Software Sizing for Cost/Schedule Estimation

Mr Philip Morrow (BA (Hons), PGDip)

Faculty of Computing, Engineering and the Built Environment

Ulster University

Thesis submitted for the degree of Doctor of Philosophy

November 2018

I confirm that the word count of this thesis is under 100,000 words

Contents

List of Figures

List of Tables

Acknowledgements

I would like to offer my gratitude to my first supervisor for this research thesis, Dr George Wilkie. He provided continued support and advice throughout the duration of this research, ensuring that the work remained focused on a clear and coherent path. The considerable time devoted to reviewing and providing feedback on the content of the research is greatly appreciated.

My gratitude is also offered to my second supervisor, Dr Ian McChesney. The advice and feedback provided throughout helped refine the structure and presentation of this research thesis.

This research could not have been completed without the assistance of Equiniti-ICS. The provision of commercial project documentation, and the meetings held with development staff, enabled this research to gain invaluable insight into the practical application of software sizing. In particular, I would like to thank Mr Charlie Tuxworth for facilitating this work, and the timely assistance and feedback provided throughout on his part.

I would also acknowledge the financial support received from the Department for Employment and Learning (DEL).

Abstract

This research thesis presents an investigation into the use of software sizing for cost/schedule estimation, with the assistance of a commercial software development organisation. The investigation was divided into three phases, and involved the use of software sizing with documentation for eleven commercial projects.

Research Phase 1 investigated the value of using different levels of sizing rigour by using the Indicative, Estimated and Full NESMA methods with both early bid stage documentation and later stage functional specifications. The Estimated NESMA method was found to provide the optimum value, in terms of estimation accuracy and effort, but it was only deemed acceptable at the bid stage.

Research Phase 2 investigated how the use of more detailed functional size data could enhance bid stage estimation. A classification of functional categories was developed and used to compare size estimates developed at both the bid and functional specifications stages. The conclusion of this phase was that the more detailed size data is needed to provide a measure of the completeness of bid stage documentation.

Research Phase 3 investigated how the NESMA sizing methods could be adapted to provide greater value. A simplified method was developed and refined using the sample projects, and evaluated against the NESMA methods. The results of this phase showed that the simplified method provided similar estimation accuracy to the Estimated NESMA method, but with a statistically significant relative reduction in the estimation effort.

The main contributions demonstrated by this research were:

- Size estimation should focus on business value rather than estimation accuracy if it is to encourage adoption within industry.
- The level of sizing detail provided by the estimate may be pivotal in determining its value.
- Simplified sizing methods should focus on approximating the level of detail provided by full sizing methods rather than on just the overall functional size.

Achievements/Contributions

The following research articles were published during the completion of this thesis:

- Wilkie, George, McChesney, Ian, Morrow, P, Tuxworth, C and Lester, NG (2011) The Value of Software Sizing. *Information and Software Technology,* 53 (11). 1236-1249.

- Morrow, Philip, Wilkie, George, McChesney, Ian and Tuxworth, C (2012) Through the Looking Glass: What does a Tender Document Actually Tell You*? In: Proceedings of the 9th Software Measurement European Forum,* 25-26 June 2012, Rome, 125-139

- Morrow, Philip, Wilkie, George and McChesney, Ian (2014) Function point analysis using NESMA: simplifying the sizing without simplifying the size. *Software Quality Journal*, 22 (4). 611-660

List of Abbreviations

BFC	Base Functional Component
CMMI	Capability Maturity Model Integration
COSMIC	Common Software Measurement International Consortium
DET	Data Element Type
EI	External Input
EIF	External Interface File
EO	External Output
EQ	External Inquiry
E&QFP	Early and Quick Function Point
FFP	Full Function Points
FISMA	Finnish Software Measurement Association
FP	Function Points
FPA	Function Point Analysis
FSM	Functional Size Measurement
FSU	Functional Sizing Units
FUR	Functional User Requirement
GSC	General System Characteristics
IFPUG	International Function Point Users Group
ILF	Internal Logical File
ISBSG	International Software Benchmarking Standards Group
MMRE	Mean Magnitude of Error

NESMA	Originally an acronym for Netherlands Software Metrics users Association, but NESMA has now restructured itself as an independent international organisation
RET	Record Element Type
SiFP	Simple Function Points
SLOC	Source Lines of Code
UCP	Use Case Points
UFP	Unadjusted Function Points
VAF	Value Adjustment Factor

1. Introduction

The research presented in this thesis focuses on investigating the use of software size estimation on commercial project documentation. The research data presented emanates from sample projects developed within a well-established software organisation across several years. The investigation considers the application of different levels of sizing detail at two significant stages of the software development lifecycle. An assessment is provided on how software size estimation can provide value to the software development process.

This chapter provides an overview of the software estimation process in order to establish the context of the research work undertaken in this thesis. The motivation for this research is outlined, both in terms of the research problem that this work seeks to address and the opportunity for working with a local software development organisation in the course of completing this research. A profile is then provided of this local commercial software development organisation, which provided access to real world project data and highly experienced professional development personnel.

An overview of the research investigation is then presented, including a statement of the aims and objectives of each phase of the research. Finally, the overall structure of the thesis is outlined, describing how the research evolves through each chapter.

1.1 The Software Estimation Process

The nature of software estimation can be viewed as an ongoing, repeatable process within the development lifecycle. The fundamental activities typically involved in the estimation process are shown in Figure 1.1. These are illustrated as consecutive stages to indicate the general order these activities may take over the course of the estimation process. In practice, the process would not necessarily follow such a linear path and not all activities would be required for each specific estimate. For example, in the event of developing estimates at multiple points within the project lifecycle it would likely require some or all of the first four stages being repeated for each distinct estimation point, with the final stage possibly only being completed once at the end of the project development.

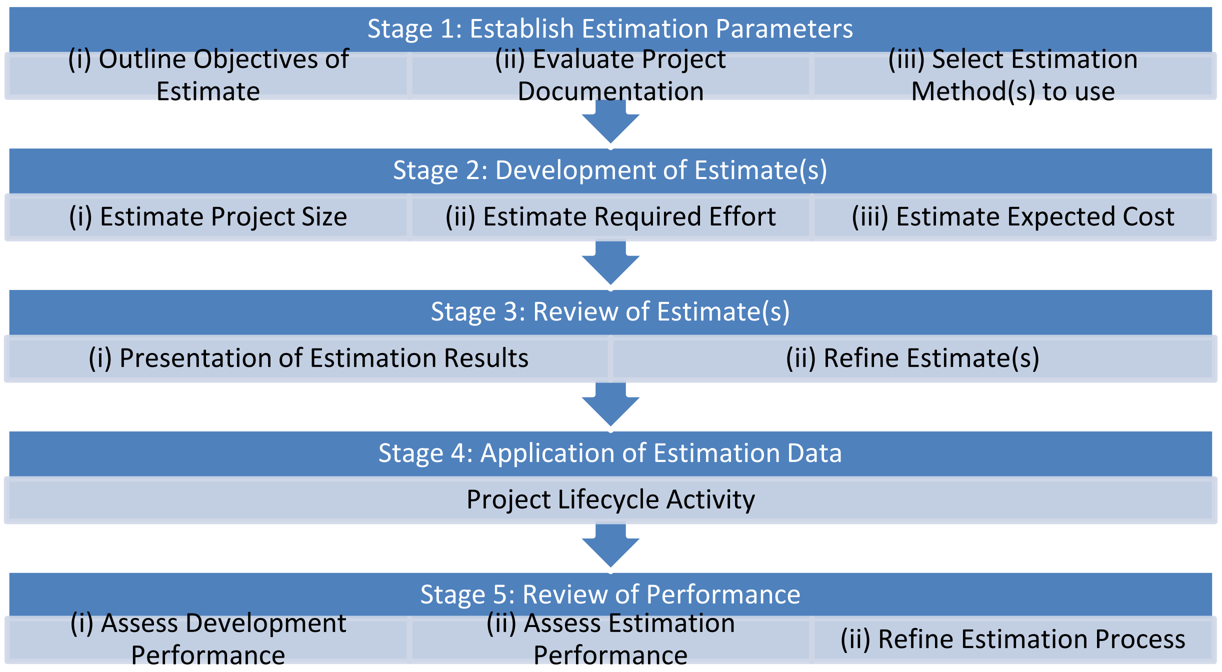
Stage 1: Establish Estimation Parameters
(i) Outline Objectives of Estimate
(ii) Evaluate Project Documentation
(iii) Select Estimation Method(s) to use
Stage 2: Development of Estimate(s)
(i) Estimate Project Size
(ii) Estimate Required Effort
(iii) Estimate Expected Cost
Stage 3: Review of Estimate(s)
(i) Presentation of Estimation Results
(ii) Refine Estimate(s)
Stage 4: Application of Estimation Data
Project Lifecycle Activity
Stage 5: Review of Performance
(i) Assess Development Performance
(ii) Assess Estimation Performance
(ii) Refine Estimation Process

Figure 1.1 - Estimation Process Activities

1.1.1 Stage 1: Establish Estimation Parameters

The first estimation stage is concerned with establishing the parameters of the estimation process. These parameters are concerned with the input and output requirements for the process. The input requirements are concerned with the project information, both available and required, to inform the use of an estimation method. The output requirements for the estimation method consider the purpose of developing the estimate. The selection of an appropriate estimation method depends upon the resources available, such as time and suitably trained and qualified members of the project team. The estimation expertise within the development organisation restricts the number of estimation methods that can be used. The availability of project information constrains the feasibility of using any particular estimation method and therefore the output that can be produced by the estimate.

These parameters are influenced by the development stage of the project lifecycle, but at the stages generally associated with estimation they will typically be drawn from the following:

(i) Outline Objectives of Estimate (Prepare project bid, produce development schedule, measure completed system etc.)

(ii) Evaluate Project Documentation (Request for Tender, Requirements Specification, Functional Specification etc.)

(iii) Select Estimation Method(s) to use (Expert Judgement, Formal Size/Cost/Effort Methods)

1.1.2 Stage 2: Development of Estimate(s)

The second estimation stage is concerned with the development of software estimates and, depending on the decisions made in the preceding steps, may comprise up to three distinct estimation measures. The estimation of these measures can form a sequence of steps where the output of one step forms the input for the next step. The first step is concerned with software size and uses a measure appropriate to the project documentation for the estimation parameters from stage one. The second step is concerned with the estimation of the effort required to develop the project, with the final step involving the derivation of the expected cost of the project. Each step can be addressed explicitly within the estimation process when a relatively formal approach is adopted. For more informal approaches to estimation one or more of the steps may be omitted, but even in such cases the development of an estimate may entail an implicit consideration of each of the steps in arriving at the final overall estimate. For this stage of the estimation process, historical project data is generally used to inform one or more steps. This may take the form of explicitly recorded project data, or the accumulated experience of the estimator.

As with the first estimation stage, the development stage of the project lifecycle influences the focus of these steps. The information arising from these steps would generally include one or more of the following examples:

(i) Estimate Project Size (lines of code, functional size, object-oriented size etc.)

(ii) Estimate Required Effort (person hours)

(iii) Estimate Expected Cost (currency)

Based on the selected method's procedures, the necessary detail is identified in the documentation and used to construct a size estimate, with previous historical data used to inform these judgements. This information may be provided broadly at an overall project level and/or more narrowly at sub-functional levels. This historical data can then be used to derive estimates of both cost and effort, which are then reviewed to determine if they are realistic and if the proposed project is indeed feasible.

1.1.3 Stage 3: Review of Estimate(s)

The next stage after the development of the estimates is to review the output they provide in order to establish an understanding of the scale of the project. Depending upon the lifecycle stage, this review may involve both development staff assigned to the project and management responsible for approval of software projects.

(i) Presentation of Estimation Results (size/effort/cost, level of risk/certainty etc.)

(ii) Refine Estimate(s) (group consensus, re-estimation etc.)

Depending on the estimation method used, the anticipated scale of the project may include measures of size, effort and cost. Such estimates should be presented with the associated level of uncertainty regarding the accuracy of the estimates. The estimates presented may subsequently be subject to refinement if they are deemed to be unsatisfactory. Such dissatisfaction may stem from wider commercial or organisational concerns rather than necessarily reflecting any perceived inaccuracy in the estimates. Refinements can range from informally reaching a consensus within the project team to repeating the previous stage of developing the estimates.

1.1.4 Stage 4: Application of Estimation Data

This stage of the estimation process is concerned with utilisation of the software estimates developed in the previous stage. The possible activities involved in this stage are reflective of the lifecycle stage at which the estimates were developed, with the examples below representing their evolution as a project develops:

- Project Lifecycle Activity (Prepare Project Bid, Develop Project Schedule, Measure Completed Project etc.)

The importance of developing software estimates can be ascertained from how the estimation output data is utilised. Inaccurate initial estimates provide an unrealistic appreciation of the cost of a project. The initial estimates are integral to the success of a project by enabling an informed decision to be made about undertaking the project e.g. bidding for the project. Subsequent estimates provide guidance for managing the development of the project and evaluating how the project evolves through the lifecycle. Typically, the formulation of a development schedule from the effort estimates is a fundamental step in the estimation process.

1.1.5 Stage 5: Review of Performance

The final stage of the estimation process is concerned with reviewing the completed project in terms of both the development and the estimation performance. This enables the estimation process to be refined on the basis of the relative performance of the completed project, assessing implications for future projects.

(i) Assess Development Performance (productivity, project success etc.)

(ii) Assess Estimation Performance (estimation accuracy, risk assessment etc.)

(ii) Refine Estimation Process (modify estimation approach, update historical data etc.)

The success of a project can be evaluated from multiple perspectives, such as customer satisfaction and whether or not it was completed on budget. The availability of actual effort and cost figures provides an explicit basis for assessing the accuracy of the respective estimates. The degree to which the actual figures match the estimated figures may also be viewed in terms of the relative productivity of the development team. The success of the project is consequently affected by both the development and the estimation performance. The results of the estimation process can be added to the knowledge base of developed projects and enable the estimation process to be refined, if necessary, for future projects. In order to maintain the integrity of historical data, suitable project development and estimation characteristics should be recorded to facilitate comparison between projects. The degree to which the estimation process can be standardised and repeated impacts the value that can be derived from maintaining historical data.

1.2 Research Problem

The research problem addressed within this thesis considers two perspectives. Firstly, the nature of the software estimation problem that persists within the software industry provides an indication of the challenges faced in incorporating estimation into software project development. Secondly, the focus of software estimation research provides an indication of the degree to which it is directed towards addressing the industry software estimation problem. These perspectives present a picture of a divide between industrial practice and academic research. The research problem is concerned with establishing a basis for delivering practical benefits to software estimation practice. The strategy adopted towards addressing this research problem is discussed within this section.

1.2.1 The Software Estimation Problem

The role of software estimation in project development is at the core of determining the success, or otherwise, of the development process. Before committing to undertake a project it should be established whether the estimated cost and effort can be justified, and estimation may play a central role in managing projects to completion. The idealised view of the software estimation process presented in Section 1.1 is not an accurate reflection of how estimation is generally conducted in industry, where less rigorous approaches to estimation are more common. The performance of software estimation within industry has failed to demonstrate any significant improvement over time according to numerous surveys (Heemstra, 1992; Lederer and Prasad, 1992; Moløkken-Østvold et al., 2004, Sauer and Cuthbertson, 2003; Verner and Evanco, 2005; Yang et al., 2008). Despite the continued challenges faced in developing accurate estimates, the trend within industry during this time has been to consistently rely upon more informal expert judgement based estimation practices, generally adopted by 70% or more of organisations (Wydenbach, 1995; Moløkken-Østvold et al., 2004; Yang et al., 2008).

1.2.2 Software Estimation Research

The software estimation problem has been acknowledged within research for over 50 years as an integral component of the wider concern of managing the development of computer systems. Farr and Zagorski (1964) focused on investigating which factors affect programming costs to facilitate the derivation of estimating relationships for such costs. The motivation behind the research was to enable the estimation process to become more *"systematic and reliable"*. In the subsequent years the dominant focus within academic research has been on the introduction and development of estimation methods, accounting for 61% of peer reviewed published literature (Jørgensen and Shepperd, 2007). This pursuit of improved approaches to software estimation has been primarily concerned with the accuracy of the estimates, without due consideration to the underlying issues affecting adoption of such approaches. The general direction of the research presents a narrow view of the value offered by software estimation, and in particular how such value may be provided to the software development process.

1.2.3 Addressing the Research Problem

The apparent gap between the focus of academic research and the estimation practices within industry highlights the failure to establish an accepted approach to the software estimation problem. The question that is posed by this research thesis is how and why progress towards addressing this problem continues to be elusive. The causes of this gap may be examined from both perspectives. The software industry operates within commercial constraints that influence the adoption of any particular approach to estimation. Software estimation methods developed within the research community must therefore demonstrate that they can provide business value to the software industry. The software estimation methods offered by academic research have thus far generally failed to attract significant interest from the software industry. This may be attributed to a number of potential characteristics of these methods:

1. The estimation accuracy of these methods is not demonstrably superior to the existing approaches used within industry.
2. The estimation data generated by these methods does not demonstrate additional utility for the software development process.
3. The estimation methods do not meet the needs of the software industry in terms of facilitating their use in the commercial environment.

The first characteristic is associated with the considerable attention devoted to improving the estimation accuracy of software estimation methods. The relative accuracy reported within both academic research and the software industry suggests that the limits of achievable accuracy may have been reached. The focus on developing software estimation practices must therefore be widened to consider the business value that may be offered by any such methods. The second characteristic highlights how this potential business value may be provided by the estimation methods. The methods must generate some form of estimation data that is not currently provided by existing estimation approaches favoured within the software industry. The last characteristic identifies that, in order to support their adoption within a commercial environment, the perceived benefits must exceed the perceived costs of using the methods. Addressing these characteristics is central to ensuring that any such benefits provided by the methods are both discernible and readily communicable to the software industry.

This research provides an appraisal of the developments within both industry and research, outlining how they fail to converge and therefore sustain the difficulties associated with software estimation. A fundamental difference is evident from Stage 2 of the software estimation process presented in Section 1.1. The different approaches to software estimation, which are dominant in research and in industry, are united in considering the effort and cost required when developing new systems, but the majority of the software industry does not typically include any formal model of software size (Heemstra, 1992; Wydenbach and Paynter, 1995; Moløkken-Østvold et al., 2004; Yang et al., 2008). In addressing the disparity, this research thesis utilises software sizing as a unifying factor whereby academic research can augment current industrial practices. An appreciation of the size of the software to be developed enables a meaningful estimate to be made of project cost and effort. Numerous methods have been developed to address the sizing of software, mostly viewing projects in terms of the functionality to be provided. This research aims to identify the practical benefits of formal software sizing and how these benefits can be obtained effectively.

1.2.4 Value of Software Sizing

The value of software sizing at each project lifecycle stage should explicitly consider the information that can be discerned at that time. Table 1 summarises the main uses of software sizing data, and the size estimation output available, according to the most significant project lifecycle stages.

At each subsequent project lifecycle stage there is an increase in the amount of sizing output that can be provided by project estimation. The contribution that software sizing can make to understanding and managing software development progress consequently increases at each stage, but the impact upon the success of an ongoing project decreases accordingly. This research thesis assesses the value of software sizing at the bid and requirements specification stages by examining the estimation output produced at each stage in order to establish the relative utility of each aspect of the size data. The efficiency of obtaining this size data is assessed by applying differing levels of rigour in the estimation method in order to determine if value is indeed provided by software sizing.

Table 1 - – Use of software sizing data by lifecycle stage

Project Lifecycle Stage	Project Estimation Activities	Size Estimation Output
Bid/Proposal Stage	Estimation of the software size Derivation of an effort/cost estimate Assessment of software reuse Preparation of commercial bid/price negotiation	Overall Software Size Basic Size Profile
Requirements Specification	Update size and effort/cost estimates Provide a measure of evolving size Identify potential scope creep Develop project plans/schedules	Overall Software Size Detailed Size Profile Comparative Analysis with initial size estimates
Project Completion	Measure final overall size of project Evaluation of estimates against actual project effort	Overall Software Size Detailed Size Profile Comparative Analysis with initial size estimates

	and cost figures Evaluating the project productivity Facilitating estimation for reuse and/or maintenance	Analysis of size estimation accuracy Historical Sizing Data

1.3 Profile of the Commercial Software Development Organisation

The research in this thesis has been completed with the support of a local commercial software development organisation with substantial experience in the industry. The following sections outline both the nature of the software development projects undertaken by this organisation and the project documentation that has been provided for this research. The existing software estimation process adopted within the organisation is discussed to highlight the contrast between industry-based and academic-based approaches to this practice.

1.3.1 Overview of Equiniti ICS

The organisation with which this research study was conducted, Equiniti-ICS, was established in May 2009, when ICS Computing Ltd was acquired by the Equiniti Group (Equiniti-ICS 2009). At the time of the acquisition the Equiniti Group was the fourth largest business services provider in the UK, with over 3,500 employees in 22 office locations supporting over 2,200 clients. Prior to the acquisition, ICS Computing had been operating since 1966, and had undergone significant growth in its last three years, supporting over 270 clients. The acquisition of ICS Computing provided Capabilities development in Case Management, Payroll services and Business Process services. Equiniti-ICS employs approximately 300 staff, with offices in Belfast, Newbury, London and a development centre in Chennai. This includes approximately 95 staff members who are directly engaged in the development of software systems.

Equiniti-ICS work with organisations across the private, public and not for profit sectors. Specific expertise has been developed through this work in regulated environments, health and education. The main solutions provided by Equiniti-ICS can be categorised as follows:

- Human Resource (HR) and payroll services
    - e.g. Auto Enrolment, Payroll Services, HR Software
- Case Management
    - Case Management, Complaints Management, Regulatory Compliance,

- Document and record solutions
    - Document Management Software, Records Management, Scanning Services, Workflow
- Enterprise Solutions
    - Accounting Software, Business Consultancy Services, Mobile Computing
- IT Services
    - Services on Demand, Managed Services, Systems Integration, Web Development, IT Outsourcing

1.3.2 Software Development Work

The focus of their software development work is on bespoke and framework-based products within these main categories. The projects undertaken within Equiniti-ICS are primarily based on fixed contract terms, necessitating the submission of a bid within a commercial tender process. The categories of solution most relevant in the context of this thesis are 'Case Management' and 'Document and record solutions', due to the nature of the sample commercial projects analysed within each research phase.

The main types of functionality associated with the sample projects include the following:

- Case Handling
- Integration with workflow modules to automate business processes
- Reporting capabilities and data analysis
- Search and retrieval mechanisms
- Document scanning and storage
- Version Tracking
- Print Solutions

As a commercial organisation, Equiniti-ICS continually evolves its development practices and techniques in order to stay aligned to current best practices. The following sections describe the nature of their software development approach during the period from which the sample projects were developed (2000 to 2012), and do not reflect how this has subsequently evolved.

1.3.2.1 Profile of Sample Software Projects

Table 2 shows the main characteristics of the eleven software projects that were supplied by Equiniti-ICS. The software projects were all developed within this software organisation during the period from 2000 to 2012. The development teams involved comprised between three and nine seasoned professional development staff in each case.

Table 2 – Software Project Characteristics

Characteristic Type	Characteristic Detail
Application Type	Data driven applications e.g. Case Management System
Requirements Methodology	Waterfall-Based
Development Team	Between 3 and 9 development staff
Development Language	.NET/C#, Visual Basic 6
Database Type	Relational
Interface Type	Graphical User Interface
System Architecture	N-tier

Each of these projects utilised a traditional waterfall-based approach during the requirements phases of the software development lifecycle. Equiniti-ICS adopted increasingly agile characteristics in the subsequent phases of the development lifecycle during the period covering these projects. There are therefore differences in how some of the projects were subsequently realised after the requirements phases. This research is limited to investigating the use of software sizing on requirements documentation. In this context, a uniform approach has been adopted in how the requirements documentation was developed for these projects. Each of the systems developed were commercial applications employing a relational database with a graphical user interface, within standard client-server architectures. These systems were data-driven applications, typically taking the form of Case Management Systems. The projects are therefore considered to be similar in nature. Each of these projects had been completed and delivered to their respective customers prior to being studied in this research.

1.3.2.2 Profile of Software Project Documentation

The general characteristics of the software project documentation are shown in Table 3. The outline of the requirements documentation represents the typical nature of the documentation for each project.

The level of detail provided in the Bid stage requirements documentation varies due to the customer oriented nature of their development. This reflects the commercial reality of software development within this sector of industry, and provides a realistic measure of the effectiveness of software sizing in these circumstances. The Functional Specifications were developed within the project team, in a typical timeframe of between six to twelve weeks after the formal contract agreement with the customer.

Table 3 – Software Project Documentation Characteristics

Bid Stage Documentation	Functional Specification Stage Documentation
Produced at 'Tender' stage	Produced at conclusion of Requirements Analysis
High Level Summation of Anticipated Requirements	Detailed description of complete set of required functionality
Primarily consists of a Tender document	Primarily consists of a Functional Specification
Generally no model of required system data	Detailed data model for the system included
Generally produced by the customer	Produced by the developer

The Functional Specification stage documentation provides a much higher level of consistency due to the more uniform development practices of the Equiniti-ICS personnel. For example, the description of the functional requirements generally took the form of use-case descriptions and report specifications. The level of detail included in the Functional Specification is always greater than in the corresponding Bid stage documentation for a project. This is in terms of both the detail included about each functional requirement, and the range of functionality covered by the documentation. The most significant distinction between these two stages is found in the provision of detail provided about the data requirements of the system. The Bid stage documentation identifies system entities, but provides limited detail about the data associated with each entity. No data model was available for any of the projects in the dataset in the Bid stage documentation. The Functional Specification stage documentation provided a complete data model for each project, including the attributes for each individual entity. The difference in this level of detail affects the feasibility of using the established software sizing methods at each stage in the lifecycle.

Additional supporting documentation, such as requests for further information at the tender stage, was present for some projects. For the purposes of this research this supporting documentation, if used in developing size estimates, is considered as part of the complete documentation for the respective lifecycle stage rather than as distinct documentation. Instances of using such supporting documentation were rare and of minimal significance in the estimation process.

1.3.2.3 Estimation Practices

In terms of the estimation practices of Equiniti-ICS, the bid stage documentation was used to develop effort and cost estimates using an expert judgement-based approach. The bottom-up nature of estimating effort for different areas of functionality, from which to calculate the overall effort required, indicates that a relatively methodical process is used within the organisation. No explicit consideration is given to software size as part of the process, however, which is consistent with the overall pattern within industry. The effort and cost estimates were used to decide if, and how, to submit a bid for the project. For successful bids, these effort estimates were then used for resource planning and scheduling activities in the early stages of each project.

The resource plans and schedules for a project were refined accordingly as the more detailed specification of the system functionality was developed. An estimate for the overall project is not, however, developed at the Functional Specification stage. This reflects the fact that there is not perceived to be any substantial benefit to be derived from devoting further time to estimation at this point. Project development follows an Agile-based approach once the requirements have been fully established and specified, with estimation taking place at the level of individual sprints. These estimates have a clearer and more specific purpose than an overall system estimate at this stage, and the benefits associated with them may therefore be readily apparent to the organisation.

1.4 Research Overview

The overriding motivation of the research is to investigate the role that software sizing can play within industry, with the assistance of Equiniti-ICS. It is concerned with addressing both the key needs of the software industry and the characteristics of existing software sizing methods that limit their practicality for commercial project development.

The scope of this research is described in Section 1.4.1. The subsequent sections describe the three main phases of this research. Each Research Phase has a specific Research Aim, which is in turn comprised of individual Research Objectives.

1.4.1 Research Scope

This research study is an empirical investigation, with a case study oriented focus, into the use of software sizing within the context of a commercial software development organisation. This incorporates the use of quantitative research, analysing the relationship between software sizing data and development project data. It also incorporates the use of qualitative research through conducting meetings with development organisation staff. The qualitative data gathered is used to assist with understanding and explaining the quantitative data obtained.

- The research is divided into three main phases, where the results of each phase inform and guide the subsequent phases. These phases address the value of software size estimation, analysis of bid stage size estimation, and development of a simplified sizing method.
- The research is comprised of a dataset of 11 commercial development projects, utilising project documentation from the bid stage and functional specification stage of the project lifecycle. The project documentation was developed in accordance with a traditional waterfall development methodology.
- The size estimation method adopted for use in this research is the Netherlands Software Metrics users Association's (NESMA) developed functional size method (NESMA, 2004). This method is based on the original Function Point Analysis (FPA) approach.
- The nature of the sample software projects may be characterised as data driven case management systems, which corresponds to the recommended types of systems for use with NESMA functional sizing.
- Size estimates are developed using both the bid stage and functional specification stage documentation. The focus of the software size estimation is primarily on point estimation, and on utilising such data for statistical analysis. The size range for the sample commercial projects used in this research is from approx. 200 FP to 2000 FP.
- The case study oriented approach enables an understanding to be developed of the existing estimation practices employed within Equiniti-ICS. It also facilitates reflection on the practical application of size estimation on authentic real world project documentation.

1.4.2 Research Phase 1

This Research Phase is concerned with investigating the value of software sizing for commercial development organisations. The specific aims and objectives for this phase are presented in Table 4.

Table 4 – Research Phase 1 Aims and Objectives

(RA1)	To assess the effectiveness of software size estimation in terms of business value at differing stages of the software lifecycle.
(RO1)	To evaluate the impact on the relative sizing of projects of a sizing process where (a) sizing methods incorporating different levels of sophistication are used and (b) sizing is performed on documentation from different stages in the software lifecycle.
(RO2)	To assess the value of software sizing to assist the associated project management activities within the local software development organisation through consideration of tangible/intangible benefits and costs.
(RO3)	To assess the potential for incorporating software sizing within the existing estimation practices of the local software development organisation.

1.4.3 Research Phase 2

The focus in this Research Phase is on investigating the additional insight that detailed functional sizing data can provide, in particular to bid stage estimation. The specific aims and objectives for this phase are presented in Table 5.

Table 5 – Research Phase 2 Aims and Objectives

(RA2)	To investigate the use of detailed software size measures to address the limitations of project documentation at the initial ‘bid’ stage of the software lifecycle.
(RO4)	To assess the potential for using a detailed profile of the required software functionality components identified using both bid stage project documentation and the functional specification in order to validate bid stage estimation.
(RO5)	To evaluate the accuracy of different levels of functional size data at the bid stage for indicating both overall functional size and actual development effort.
(RO6)	To examine the degree to which the size of required functionality not specified in bid stage project documentation can be assessed at the bid stage.

1.4.4 Research Phase 3

The final Research Phase assesses the effectiveness of simplifying the size estimation approach in terms of optimising the value of the process. The specific aims and objectives for this phase are presented in Table 6.

Table 6 – Research Phase 3 Aims and Objectives

(RA3)	To investigate the effectiveness of enhancing simplified software sizing approaches in contrast to full software size estimation.
(RO7)	To assess the extent to which software sizing can be simplified while maintaining an acceptable level of accuracy in the overall size estimate.
(RO8)	To assess the extent to which software sizing can be simplified while maintaining an acceptable level of accuracy in the estimated functional profile.
(RO9)	To assess the utility of the developed simplified software sizing method in the context of the local software development organisation.
(RO10)	To evaluate the relationship between estimation accuracy and estimation effort for different detail levels of simplified software sizing.

1.4.5 Overview of Case Study Oriented Research Focus

One of main distinguishing features of this thesis is the case study oriented focus of the research conducted. Each sample project included in the dataset corresponds to a case study. The use of real project documentation from two phases of the project lifecycle facilitates an examination of how the understanding of the required system functionality at the inception of a project compares with that at the completion of the functional specification. Chapter 4, in particular, is focused on an in depth investigation of the differences between these two stages. The provision of internal expert judgement-based effort estimates and subsequent project data, such as actual development effort, provides the context for investigating how software sizing can provide further insight into the estimation process. Conducting interviews with project development and management staff enables feedback to be obtained on the results of the software sizing exercises. This qualitative research facilitates the identification of issues concerning the feasibility of adopting the use of software size estimation within a development organisation. The feedback provided at the completion of each research phase enables the subsequent phases to be focused on addressing real world concerns about the practicality and effectiveness of estimation.

In Chapter 6, the experience of conducting this investigation in conjunction with a commercial development organisation is evaluated and reflected upon in order to establish what lessons have been learned. This review of the case study aspects of the research is comprised of two main sections. Firstly, an example of the practical application of functional sizing is presented and evaluated. In the course of completing this research a request was received to provide a size estimate for an imminent bid proposal. This provided an opportunity to apply this research in a real world context, with further detailed insight gained into how software sizing can contribute in practice. Secondly, the practical considerations for software size estimation in the context of the main research phases are discussed. In particular, this addresses the use of real world project documentation that constitutes an investigation into the appropriateness of the size measurement guidelines in practice. This includes identifying the extent to which the measurement guidelines can be implemented, as well as characteristics of project documentation that would support more efficient size estimation.

1.4.6 Overview of Statistical Analysis

This research study involves the use of statistical analysis to evaluate the use of functional size estimation on 11 commercial software projects. In particular, the relationships between functional size data and other project related data are analysed in order to identify any statistically significant correlations. This involves determining for each correlation test whether the null hypothesis can be rejected i.e. there is no association between the variables being tested. There are a number of parameters to consider when designing this research. The size of the dataset has implications on the power of any statistical analysis performed using the data. The availability of suitable projects, and the time consuming nature of functional size estimation, prohibits expanding the dataset to include further projects. The level of statistical significance commonly used, $\alpha = 0.05$, means that there is a 5% probability of making a Type I error i.e. a false positive. For the size of dataset available in this study, 11 projects, a statistically significant correlation would be obtained for a correlation $r= 0.476$ (1 tailed) or $r= 0.553$ (2 tailed). Using a statistical power of 80% ($\beta = 0.2$) would mean that there is a 20% probability of making a Type II error i.e. a false negative. The size of the association between the variables being tested represents the effect size. When there is an expected effect size, based on previous studies or experienced judgement, it is possible to calculate the necessary sample size to satisfy all of the conditions set by the chosen parameters. For example, for an effect size of $r = 0.5$ with $\alpha = 0.05$ and $\beta = 0.2$ the minimum sample size required would be 23. For the size of dataset available for this study, 11 projects, the minimum effect size that could be detected under these conditions would be r = 0.7. Obtaining a statically significant correlation for values of r below 0.7 would therefore have s statistical power of lower than 80%. For a 2 tailed test the value of r would be increased to 0.75. Any results below these thresholds should therefore be regarded with caution as there is a lower than 80% chance that such a results would be found generally outside of this dataset.

With these constraints in place, steps must be taken to minimise the possibility of drawing unsound conclusions from the results of any statistical testing i.e. false positives. This research study is exploratory in nature, in that is seeks to identify potential areas of future research. It is therefore desirable to avoid overly conservative measures in minimising false positives that would instead increase the likelihood of false negatives. Regression analysis is also used within this research e.g. to derive effort estimates from the software sizing data. The reliability of the resulting regression models must be considered, requiring appropriate steps to be taken to validate such models. The approaches adopted in addressing these, and any issues that arise from the use of other specific statistical tests, are discussed prior to the presentation of the statistical analysis.

1.5 Thesis Structure

The overall structure of the remainder of this thesis is illustrated in Figure 1.2. A Literature Review of the related research is provided in **Chapter 2,** establishing the scope of this research within the software estimation process. The main patterns within industrial practice and findings from academic research are examined in order to determine where software estimation approaches can offer more practical benefits for project development. Identifying the current research issues enables this thesis to be directed accordingly towards contributing to the advancement of software estimation research.

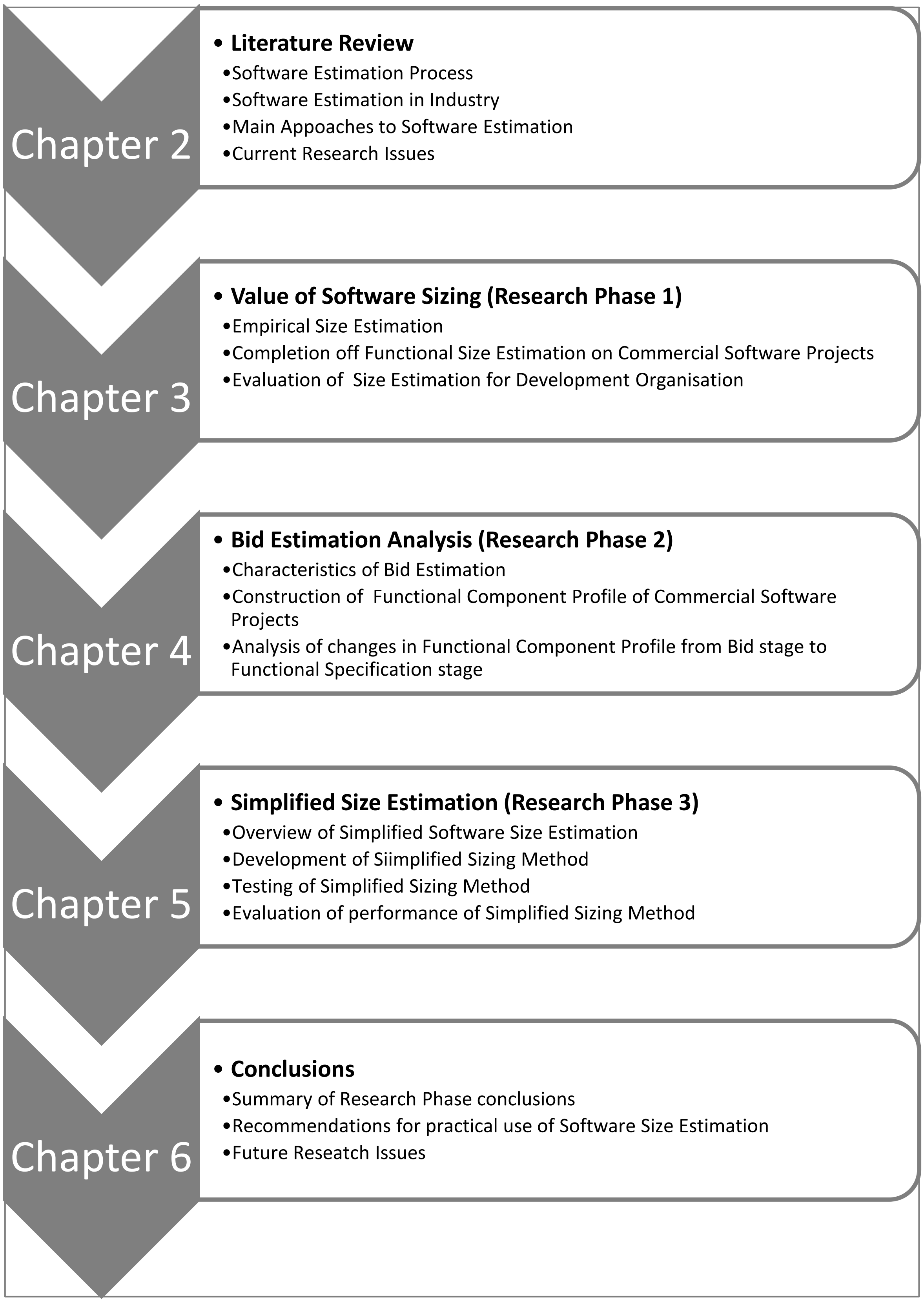
Chapter 2
• Literature Review
•Software Estimation Process
•Software Estimation in Industry
•Main Appoaches to Software Estimation
•Current Research Issues
Chapter 3
• Value of Software Sizing (Research Phase 1)
•Empirical Size Estimation
•Completion off Functional Size Estimation on Commercial Software Projects
•Evaluation of Size Estimation for Development Organisation
Chapter 4
• Bid Estimation Analysis (Research Phase 2)
•Characteristics of Bid Estimation
•Construction of Functional Component Profile of Commercial Software Projects
•Analysis of changes in Functional Component Profile from Bid stage to Functional Specification stage
Chapter 5
• Simplified Size Estimation (Research Phase 3)
•Overview of Simplified Software Size Estimation
•Development of Siimplified Sizing Method
•Testing of Simplified Sizing Method
•Evaluation of performance of Simplified Sizing Method
Chapter 6
• Conclusions
•Summary of Research Phase conclusions
•Recommendations for practical use of Software Size Estimation
•Future Reseatch Issues

Figure 1.2 – Thesis Structure

The three Research Phases are then presented in the next three chapters of the thesis. **Chapter 3** is concerned with evaluating the value of software sizing within the context of commercial project development **(RA1)**. This involves empirical software estimation research in the form of the development of size estimates from requirements documentation originating from two distinct project lifecycle stages. Firstly, bid documentation used to formulate a competitive bid for a project is used as it reflects the most significant use of software estimation. Secondly, the completed Functional Specification is used as it reflects the most complete documentation of the required functionality. The results obtained at each stage enable a comparison to be made on the estimated software size of projects at those stages **(RO1)**. The results of these sizing activities are then evaluated and contrasted with the internal estimates produced by Equiniti-ICS and the actual development effort. The potential contribution of software sizing to project management activities is assessed with feedback obtained from Equiniti-ICS **(RO2)**. Project development staff and management within Equiniti-ICS are consulted on existing estimation practices within the organisation and their feedback is provided on how software sizing would impact on these practices **(RO3)**.

Chapter 4 focuses on addressing the limitations of bid stage documentation in facilitating the development of software estimates **(RA2)**. A detailed profile of required functionality is developed both from the bid stage and from the later functional specification lifecycle stages to obtain a breakdown of specific types of required functionality for a project. Examining the differences in the Functional Component Profile at these distinct stages enables patterns to be discerned in how estimates for specific types of functionality vary across the different stages. Comparisons are made with the estimation output data produced by the existing approach utilised within Equiniti-ICS to determine if sizing data provides additional insight into assessing the thoroughness of the identified required functionality at the bid stage **(RO4)**. The functional size data produced at the bid stage is evaluated at different levels as an indicator of both the overall functional size and actual development effort in order to establish the most significant estimation data produced at this stage **(RO5)**. Relationships between functional components at both the bid stage and functional specification stage are examined to determine if appropriate ratios can be identified and used at the bid stage to indicate the likelihood of subsequent growth in size. **(RO6)**.

Chapter 5 covers the final Research Phase, assessing whether software sizing can more efficiently support the estimation process and therefore provide a more attractive proposition for implementation of such activities within industry **(RA3)**. Existing simplified sizing methods are examined to indicate both the input detail they require and the output detail they provide. In this phase a simplified sizing approach is developed, which enables different levels of requirements detail to be considered, and subsequently tested on the commercial development projects. The developed functional sizing approach is adapted at three different levels of rigour in order to assess the trade-off between simplifying the functional sizing approach and the quality of the output data produced. The sizing results obtained from using the different levels of simplified estimation are evaluated in terms of overall functional size **(RO7)** and the specific functional profile **(RO8)** against those obtained using the existing NESMA functional sizing methods. The utility of the developed simplified sizing approach is assessed through consideration of where this approach could be adopted within the development organisation and the benefits obtained from the size estimation data **(RO9)**. An evaluation of the estimation performance provided and estimation effort required at the different levels of estimation detail provides an indication of where the optimal value can be provided by software sizing **(RO10)**.

The concluding **Chapter 6** draws together the conclusions from each of the research phases, and considers how the thesis has addressed its aims and objectives. Lessons learnt regarding how the industrial application of software estimation can benefit from more formal approaches are discussed as a practical means of crossing the divide between industry and academic research. The research contributions made throughout the phases of this research are then presented. Finally, future software estimation research issues arising from this thesis are discussed.

2. Literature Review

"A good estimate is an estimate that provides a clear enough view of the project reality to allow the project leadership to make good decisions about how to control the project to hit its targets"

(McConnell, 2006, p14)

2.1 Introduction

Evaluating the merit of a software estimate requires consideration of the purpose of obtaining the estimate. How is the information provided by the estimate going to subsequently be used? Software size estimates of a project are generally considered to be a fundamental input for the subsequent estimation of effort and cost. There is a trade-off between the accuracy and level of detail provided by a size estimate, and the effort required to produce that estimate. There is also a limit to the value that may be derived from the output of an estimation method. Does the relative increase in the accuracy and level of detail provided by increasing the amount of effort required to produce an estimate provide an equivalent increase in value derived from the estimate?

Software estimation is of most value at the initial bid or feasibility stage of a project, when the information available from which to produce the estimate is at its least detailed. The estimation paradox is that when sufficiently detailed information about a project to enable the most accurate estimation is finally available, the value of such an estimate has been greatly diminished. Estimates produced at the initial stages of a project must therefore consider the impact resulting from the absence of detailed requirements information. How does the requested functionality identified at the initial stages of a project differ from the requested functionality subsequently clarified by the end of the requirements analysis?

The effort required to produce the estimate has been reported as a main barrier to the use of software estimation methods (Yang et al., 2008). It is widely considered prudent to utilise more than one estimation approach to arrive at a consensus view on a project estimate. This consideration places an additional burden on an estimation method to minimise the effort required for its use. Simplifying the estimation process to provide the optimal balance between estimation accuracy and estimation effort is one approach to enhancing the value of an estimation approach. The degree to which the simplification process reduces the accuracy and level of detail in the resulting estimate determines the value of the simplified approach. Can the estimation process be simplified without compromising the integrity of the estimate?

The main approaches to software estimation vary in the degree of formality inherent in developing the estimates. The most formalised approaches are typically Model-based estimation methods that can require substantial project data as an input to the model. The most informal approaches are typically Expert Judgement-based methods, which rely more upon the previous experience of the estimator rather than a formal process. Estimation by Analogy can vary significantly in formality, relying on either human judgement or artificial intelligence for assessing similarity between projects in order to develop appropriate estimates.

This chapter begins with a review of the main trends within the field of software estimation. This review covers the findings of surveys on the use of software estimation within industry, as well as the main estimation approaches that have been developed within research. The main issues facing this field of research going forward are then discussed, including how this research thesis seeks to investigate how software sizing can provide value to the development process. The limitations of software estimation, both in practice and in research, are then discussed. Lastly, the prospective research contribution of this thesis to software estimation is outlined.

2.2 Software Estimation Research and Practice

The use of software estimation within industry can be viewed both in terms of how successful it has been, and in the nature of the approaches to estimation that have been adopted. Separate surveys have been conducted within industry examining these issues, but differences in the research methodologies adopted hinder the direct comparison of such research. For this section of the Literature Review the inclusion of appropriate surveys is dependent upon the availability of equivalent results to facilitate comparative judgements. This section of the review looks at estimation performance within industry and the main methods used. An examination into research on software estimation methods is then provided, resulting in an assessment of the relationship between research and industrial practice.

2.2.1 Estimation Performance

The effectiveness of the estimation process within industry has been the subject of research for at least 30 years. Consideration of the performance has focused on how frequently projects exceed their estimated cost/effort, and/or the relative estimation accuracy. Comparison of research results across different industrial surveys is therefore limited by the availability of common estimation data from each survey. Figure 2.1 shows the results reported from surveys on how the actual effort required for projects compared with the estimated effort. The percentage of projects, which were underestimated in each survey, is shown as a check pattern bar with a positive value. The percentage of projects which were overestimated in each survey is shown as a plain pattern bar with a negative value. The surveys are taken from separate regions (USA, Europe and Asia) and therefore provide a broad view of estimation performance within industry.

The trend illustrated in Figure 2.1 is that the actual effort to complete software development projects has consistently been found to exceed the estimated completion efforts. The percentage of projects that have exceeded their estimated effort varies between 59% and 76% across the surveys included (Heemstra, 1992; Lederer and Prasad, 1992; Moløkken-Østvold et al., 2004; Sauer and Cuthbertson, 2003; Verner and Evanco, 2005; Yang et al., 2008). The mean magnitude of the effort underestimate reported in surveys is typically between 30% and 40% (Moløkken-Østvold et al., 2004). From those surveys which reported the percentage of projects that were overestimated, the values of the percentages ranged from 14% to 29%.

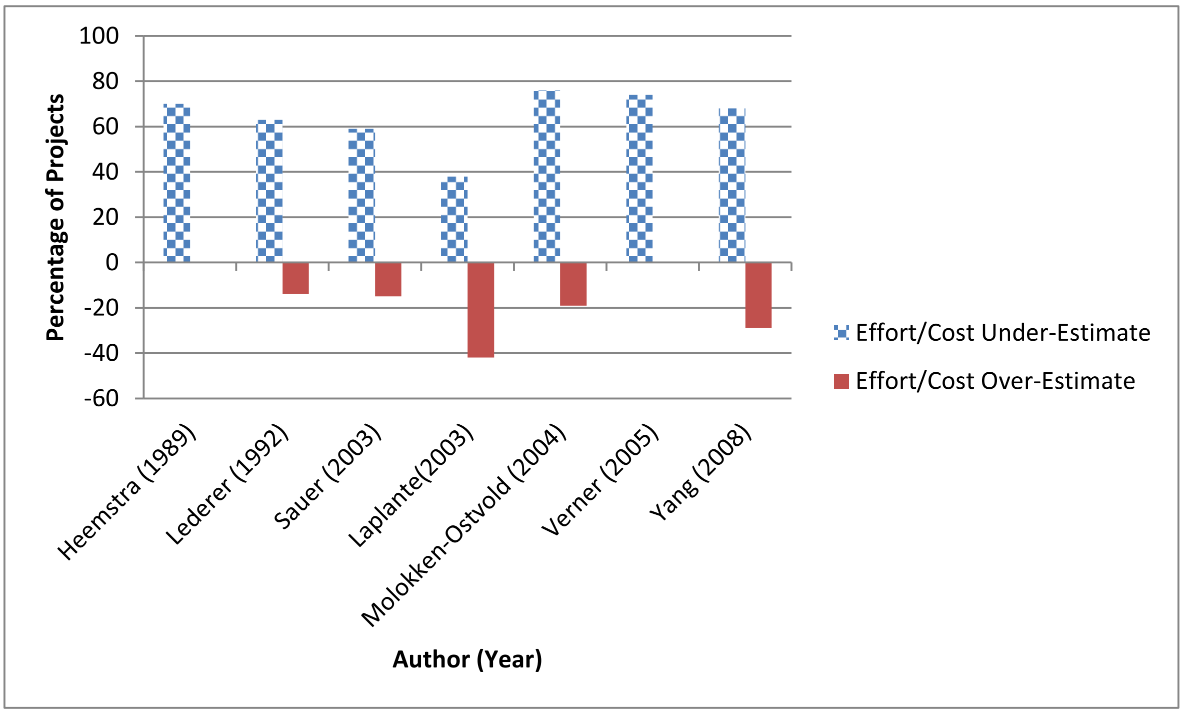

Figure 2.1 – Estimation Performance from Industry Surveys

Estimation performance can vary according to the nature of the projects undertaken. The survey of the Norwegian software industry (Moløkken-Østvold et al., 2004), reported that larger projects tended to be more likely to be underestimated. The estimation performance for "in-house" projects was found to be considerably better than that on projects completed for external clients. The reason suggested for this improved estimation accuracy was that "*in-house developers have closer proximity to the customer and more stable system properties such as* *requirements, platform and implementation language*" (Moløkken-Østvold et al., 2004 p216). These surveys show that the estimation performance within industry had not improved over this time period, remaining a challenge for current software development.

2.2.1.1 Industry Perception of Estimation

The role of software estimation in contributing to the success of projects is routinely presented in academic research as supporting its use within industry. Consideration of whether a project has been completed within the estimated time and cost has been proposed as one of the measures of the success of a project. This represents an oversimplification of the development process because projects evolve, in terms of the delivered functionality, subsequent to the estimates established earlier in the project. The pattern of unsatisfactory estimation performance reported by surveys, as highlighted in the preceding section, does not provide much insight into the sources of the relative inaccuracies in these estimates. The Verner and Evanco (2005) survey addressed whether project managers considered projects to have been successful. It was reported that while 74% of projects were underestimated, 62% of projects were considered to be a success. This suggests that estimation is not considered to be of critical importance in facilitating the successful completion of projects. There is no single accepted definition of what constitutes a 'successful project' according to a systematic literature review of the supplier perception of software development (Savolainen et al., 2012). The most prevalent success criteria were found to be related to business success for the supplier as a result of the project. There was no definition of what constitutes a 'failed project' found within this same review. The contribution of software estimation should therefore be considered beyond the overall accuracy of the estimates.

The importance of estimation was addressed in a survey of 56 organisations in the Northern Ireland software industry by the Centre for Software Process Technologies (McCaffrey et al., 2004). The results of this survey indicated that, out of the 15 areas of software engineering and engineering management addressed, software size and cost estimation was the most significant area of concern. These results are consistent with other surveys that found between 78%-87% of companies viewed estimation at the very least as an important issue (Lederer and Prasad, 1992, Moløkken-Østvold et al., 2004, Yang et al., 2008). In one of the surveys, it was indicated that the reasons for underestimates in projects were not analysed since managers were not "*concerned with analysing the performance of their methods*" (Moløkken-Østvold et al., 2004 p216). The perception of estimation within industry may therefore be that of a problem to be endured rather than one to be solved.

Yang et al. (2008) asked for the causes of inaccurate estimates to be indicated in their survey. The top ranked cause was reported to be ‘changing requirements’, which indicates that the evolution of a project through its lifecycle adversely affects the actual effort/cost incurred during development. The issue of ‘not having suitable software cost estimation methods’ was the sixth ranked cause of inaccurate estimates. This suggests that inaccurate estimates are not perceived to be an issue of ineffective estimation methods, as much as one of an unavoidable characteristic of the project lifecycle. From a research perspective this can be viewed either as an inherent constraint on software estimation, or as a limitation on the value that can be provided by estimation. Is the industry perception of software estimation an accurate reflection of the potential value of software estimation?

2.2.2 Estimation Methods Used in Industry

The software estimation process output may incorporate a range of separate measures including size, effort and cost. The estimation methods that have been developed may either encompass one or more of these measures explicitly, or consider them implicitly in more informal approaches.

Reviewing the use of estimation methods within industry is constrained by two main sources of variation. Firstly, the definition of distinct types of estimation method varies across the surveys that have addressed this topic of research. Secondly, the degree to which the respondents in a survey are able to match the estimation method(s) they use with the categories outlined in the respective survey is subject to variation across the different participants.

This appraisal of the research surveys categorises the main estimation methods as:

(1) Expert-Judgement-Based – these approaches utilise the expertise of individuals in developing an estimate for a project. These approaches can vary in the degree of formality involved, but are characterised by the reliance on the judgement of the estimator rather than specific guidelines in developing the estimate.

(2) Analogy-Based – these approaches involve analogy with historical data from previously completed projects to determine how the new project compares. These approaches can vary from more rigorous machine learning methods to more informal judgement-based methods.

(3) Model-Based – these approaches are characterised by the use of formal estimation models or adherence to algorithmic procedures. Model-based methods are driven by the specification of the method rather than the judgement of the estimator.

These three main types of estimation method do not correspond to mutually exclusive options, because development organisations may have utilised multiple estimation methods, or approaches which do not correspond to any of these methods. Other approaches have been identified, such as ‘Price-to-win’, where the final ‘estimate’ is influenced by commercial or organisational concerns. The use of the main types of estimation method may also be characterised according to whether a ‘top-down’ or ‘bottom-up’ approach was followed. Consideration of such approaches that lie outside the three main types identified is not explicitly evident across most of the surveys, and is therefore not included in this analysis of the surveys.

The relative percentage use of the three main estimation approaches within industry, as reported in surveys, is shown in Figure 2.2. Moløkken-Østvold et al. (2004) categorised Analogy-based approaches within Expert-based, so the chart in Figure 2.2 indicates the combined total for both of these types within the Expert-based column for this survey. The prevailing pattern has been for industry to refrain from the use of model-based approaches, with usage ranging from 14% to 26% (Heemstra, 1992; Wydenbach and Paynter, 1995; Moløkken-Østvold et al., 2004; Yang et al., 2008). Expert-based and Analogy-based approaches have been consistently utilised in the majority of organisations surveyed. As with estimation performance within industry, illustrated in Figure 2.1, the pattern of usage of estimation methods has remained consistent over time and across different regions of the world. In broad terms, the general approach to software estimation has not fundamentally changed despite the continued relatively poor performance obtained with this approach. Moløkken-Østvold et al. (2004) found that that from a sample of 44 analysed projects, 37 (84%) of the managers reported relying completely upon expert judgement estimation, with the other 7 projects utilising a mixture of expert judgement in combination with some model-based estimation technique. This study reported that there was no statistically significant difference in performance between solely using Expert-based methods and when additionally using Model-based methods for a combined approach. In general, however, the use of multiple estimation methods has not been explicitly reported, either within or across the three main types. It has therefore not been established whether combining different methods has any effect on the overall estimation accuracy.

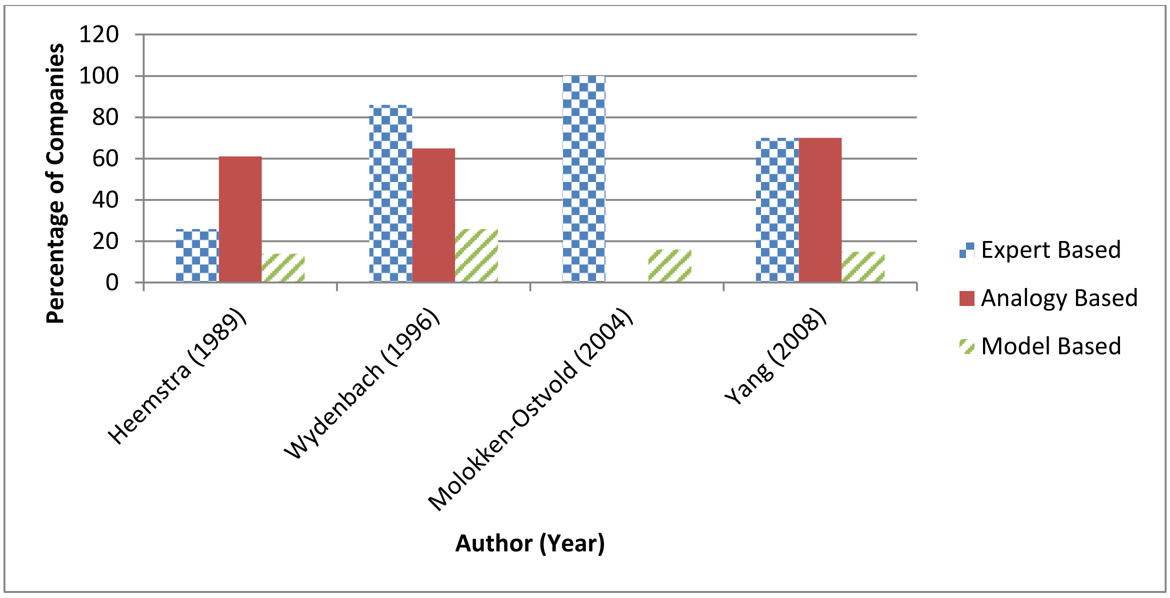

Figure 2.2 – Use of Estimation Methods from Industry Surveys

Industry surveys on the use of estimation methods have tended not to directly address the reluctance to adopt model-based estimation methods. Yang et al. (2008), however, specifically examined this issue, within relatively large software development organisations, for the use of cost estimation methods. This survey involved 112 projects in the Chinese software industry, with only 15% of projects indicating use of Model-based approaches to estimation. The reluctance to adopt Model-based approaches was attributed to the cost and effort required, with insufficient benefits to justify the additional investment in estimation practices. This supports the earlier assertion that the limitations of current estimation practices may have become accepted, and in the absence of superior alternative methods, estimation represents an unavoidable hurdle in the software development process.

The survey by Kassab et al. (2014) of software development professionals reported on the use of size and/or effort estimation based on system requirements. The use of size and effort estimation was included as part of a wider survey on requirements engineering practices and therefore not every respondent addressed this aspect. From the 60% of respondents that reported on performing estimation, it was informal approaches such as expert judgement (45%) and group estimation (30%) that were most commonly used. Formal model-based estimation approaches, such as COCOMO and Function Points, were in total used by around 30% of the respondents. It was reported that 62% of respondents included consideration of non-functional requirements in their estimates. The general pattern of estimation practices is also consistent with those reported from previous surveys discussed in this section, even if the specific approaches used are reflective of evolving approaches to software development.

2.2.3 Research into Software Estimation Methods

Jørgensen and Shepperd (2007) conducted a systematic review of peer-reviewed research in the area of software estimation. In terms of specifically reviewing the progress of research in this area, the most pertinent research questions asked during this review were concerned with three aspects.

(1) The most investigated software cost estimation topics.

(2) The most investigated estimation methods.

(3) The most frequently applied research methods (and study context).

The use of the term ‘cost estimation’ in this review encompassed the main components of software estimation e.g. size, cost, effort. Comparisons were made across three distinct time periods in order to identify patterns in how the focus of research in this area has changed over time. Figures 2.3 to 2.5 present an illustration of the patterns identified by this review. The relative percentage of research papers focused on each specific topic is shown on each chart, both within each individual timeframe and for the overall period covered by the review. The sum of the relative percentages exceeds 100% as an individual research paper may incorporate multiple topics.

As Figure 2.3 shows, the dominant software estimation topic within academic research has been that of the development and evaluation of estimation methods. This topic has appeared in around 60% of research papers, with the next highest topic only featuring in around 20% of the papers. The second largest topic has been concerned with size measures, which peaked in the 1990s, but still remained second in the first half of the subsequent decade. Organisational issues have declined slightly as a topic over time, suggesting that while it still features more than the remaining topics, its perceived importance is not strengthening. Estimation uncertainty is the sole topic demonstrating an increase in attention over time, with the other topics remaining relatively stable.

The pattern of investigated estimation methods, shown in Figure 2.4, is that most topics have increased at the expense of theory based estimation approaches, which has declined sharply from 49% to just 5%. This decline has left regression-based approaches to estimation as the dominant approach investigated, at around 50% of the research papers. This could take the form of evolving regression-based approaches, or using them as the basis of comparison for alternative approaches. The second highest approach investigated across the review has been Function Point sizing, which peaked during the 1990s but has since been in decline. The most notable increases over time are found with Expert Judgement and Analogy based approaches, which have surpassed Function Point sizing within the last time period reviewed. The growth of ‘Other’ estimation methods is an indication that there is a concerted effort to find alternative approaches in order to improve estimation accuracy.

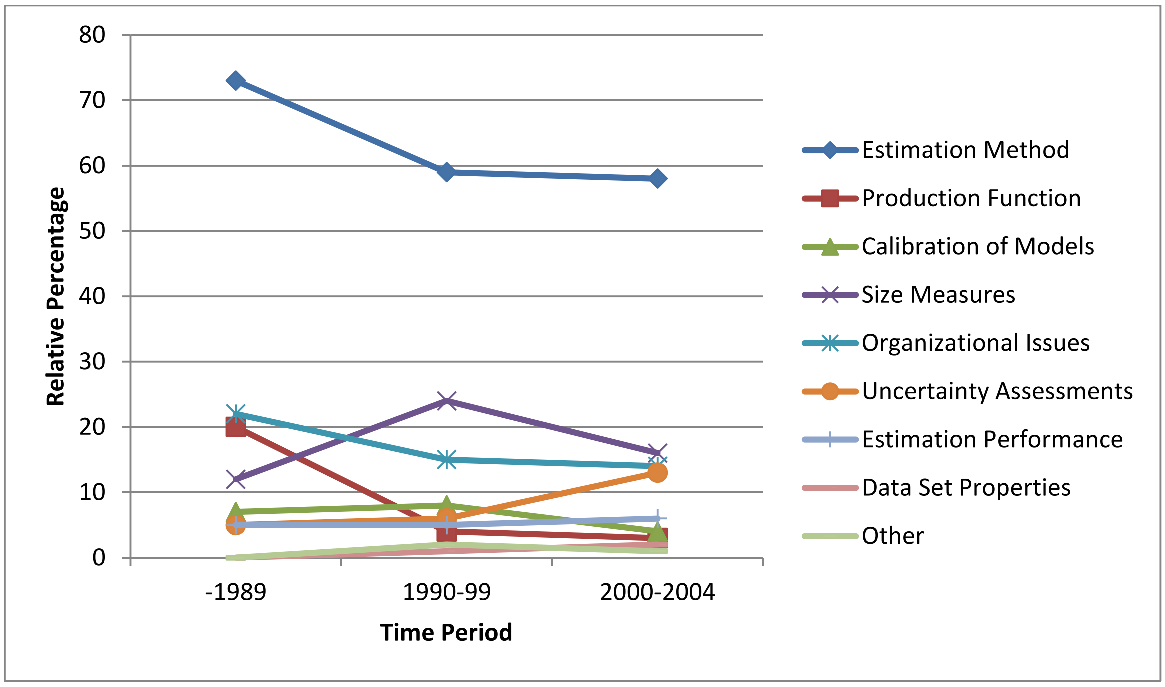

Figure 2.3 – Software Estimation Topics in Research Papers (adapted from Jørgensen and Shepperd, 2007)

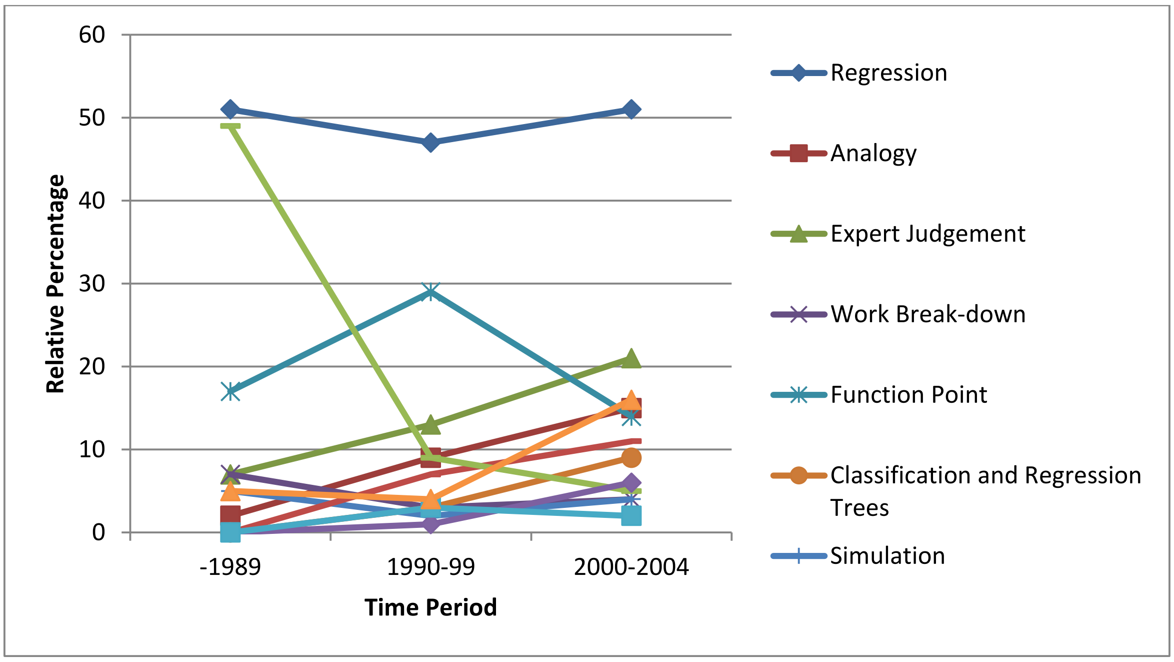

Figure 2.4 – Estimation Methods used in Software Estimation Research Papers (adapted from Jørgensen and Shepperd, 2007)

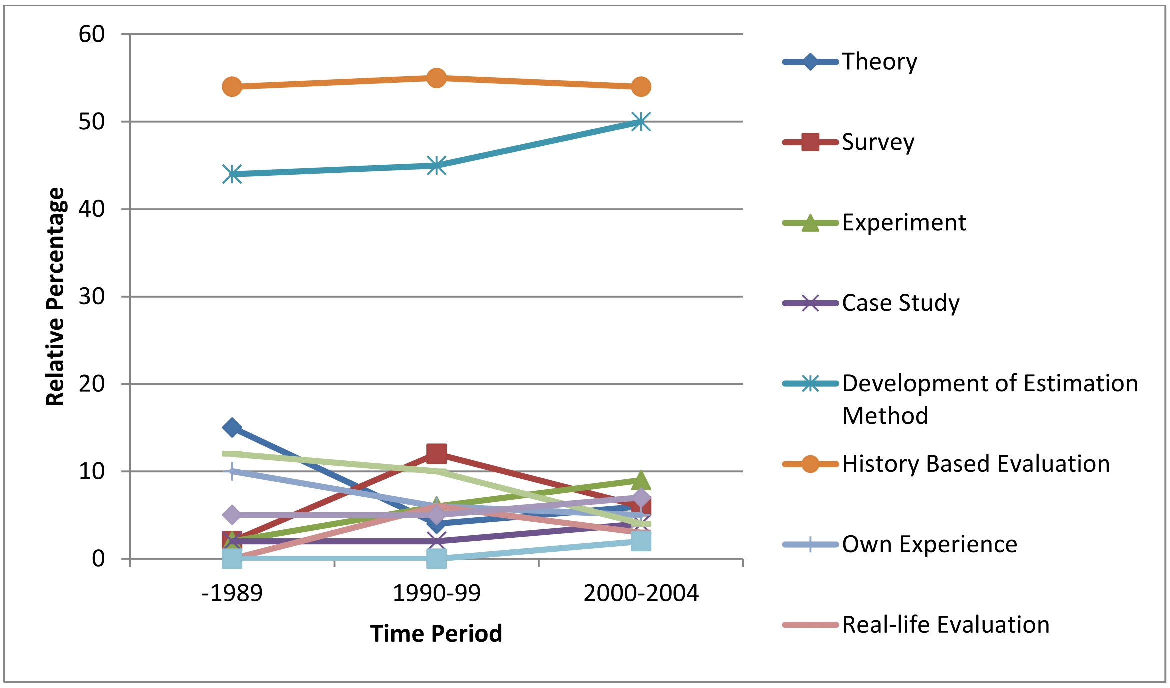

Figure 2.5 – Research Methods used in Software Estimation Research Papers (adapted from Jørgensen and Shepperd, 2007)

The research methods employed in software estimation research have been dominated by two approaches. The development of estimation methods and the use of historical data are shown in Figure 2.5 to be present in around 50% of software estimation research papers. The other approaches have demonstrated slight variations over time, but no concerted increase in the use of any of these approaches is evident. This presents problems for improving the performance of software estimation within industry. The reliance on historical data limits the degree to which the research results are applicable to modern software projects. Case studies and real-life evaluation would provide the most insight into how software estimates are developed in practice, but are among the least adopted approaches reported at 3% and 4% respectively.

Sehra et al. (2017) used natural language processing to analyse research patterns within a dataset of 1178 software effort estimation articles published from 1996 to 2016. This approach involved searching for a range of solution sizes of most significant topics e.g. what were the eight most significant topics? The effect of increasing the solution size for the most significant topics that were searched for was to dynamically produce research topics that were more specific in nature. Table 7 shows the results that were reported for the eight topic solution search of the dataset.

Table 7 Main Research Topics on Software Estimation Research (1996-2016) (adapted from Sehra et al., 2017)

Research Topic	Number of published articles
Neural networks	208
Expert Judgement	160
Size metrics	156
Estimation for re-usable components	151
Factors affecting costs	144
Soft computing techniques	128
Estimation by analogy	123
Phase wise effort distribution	108

These most significant research areas indicate a continued focus on developing and improving estimation methods, with an emphasis on using computational intelligence based approaches to analysing datasets. The number of articles focused on Expert Judgement increased in the latter part of this time period, but this was consistent with a general increase across other topics rather than a reflection of an increase of its relative importance within research. Consequently, there was no evidence of a convergence between academic research and industry practice in terms of their respective dominant estimation approaches. The research recommendations made in this review of research topics included considering how expert judgement-based estimation approaches can be integrated with computational models. The associated barriers of entry for adopting computational estimation approaches in industry may, however, limit the practicality of such advancements in research. In the subsequent chapters of this research thesis, this recommendation is instead addressed by investigating how functional sizing can complement the existing expert judgement approach used within Equiniti-ICS by providing additional insight at different stages of the project lifecycle.

2.2.4 Relationship between Research and Industry

The overall conclusion that can be drawn from the preceding sections is that there is a disconnection between the focus of academic research and the practical estimation performed within industry. The lack of improvement in estimation performance illustrates the lack of improvement in practice despite the considerable attention this topic has received in academic research. It follows therefore that research has thus far failed to address the practical problems facing industrial application of estimation practices.

The focus of software estimation research continues to be on finding a new solution to the estimation problem, focusing on more formalised approaches. Within the software industry, more informal expert judgement or analogy-based methods continue to be the prevailing choice for the estimation of projects. The literature reviews discussed in Section 2.2.3 showed that software sizing continues to be recognised as among the most common topics within academic research. The focus within industry is, however, on effort/cost estimation without any formal consideration of software size.

Bridging the ‘gap’ between academic research and estimation in practice requires consideration of why this evident trend has continued. Conventional wisdom suggests that more formalised approaches require a more substantial commitment of time and resources, without providing additional benefit to offset the cost. The limited research into this aspect prevents a clear understanding of whether this reflects a rejection of more formalised approaches on the part of industry, or merely the absence of its consideration. Conversely, the limited research into the practical application of formalised estimation approaches in a commercial environment suggests that there is an insufficient understanding within academia as to why industry has not been more open to adopting such approaches. There is therefore a need to further investigate whether there is a role for more formalised estimation approaches within the software industry.

2.3 Main Approaches to Software Estimation

Section 2.2.2 identified three main types of software estimation methods that have been utilised within industry. This section provides a review of the main estimation methods, within the three main types, which have been investigated within academic research. Size estimation is most prevalent within Model Based Estimation, whereas Expert Judgement and Analogy based methods are generally concerned with effort/cost estimation. The scope of methods reviewed in this section therefore incorporates each of these aspects of software estimation. Estimation approaches associated with specific types of software development approaches are then discussed. The use of historical data within software estimation is then considered, followed by an overview of the use of tools to support the estimation process. Lastly, evidence on the relative performance of different estimation methods is discussed.

2.3.1 Model Based Estimation

Software estimation models are ultimately concerned with estimating the effort, and hence the cost, of a project. Software size is widely recognised as a fundamental input for effort/cost models, necessitating the development of size estimates. Software sizing methods therefore represent a significant category of method within Model based estimation. There have been a multitude of sizing methods developed, with the most commonly utilised sizing approaches falling under either Functional Sizing Methods (FSM) or UML based methods. This section addresses the main FSM, while the UML based methods are discussed in Section 2.3.4.

2.3.1.1 Software Size and Estimation

The estimation role of software size requires that an appreciation of the size should be developed from the information available for the proposed system. Historically, the most common measure used to describe software size has been source lines of code (SLOC). Based on the abundance of historical data from previous projects, comparison with the size of the new project can facilitate the estimation of the effort/cost of this project. The effectiveness of this approach is limited by a number of factors.

(1) Variances in complexity across different projects.
(2) Variances in productivity across different projects.
(3) The relative level of reuse of code.
(4) Dis-economy of scale as projects increase in size.
(5) The use of automatic code generation tools.
(6) Dependence on the chosen programming language.
(7) The availability of source code for the system.

Factors (1) to (3) can be considered general to the use of software size as the means of estimating the effort and cost for a project. The use of other sizing measures would be faced with the same challenges as SLOC in accounting for these factors when deriving effort and cost estimates. Factors (4) to (7) are generally related to the particular use of SLOC as the sizing measure. The most significant factor is the availability of source code which, at a relatively late stage in the project lifecycle, is incompatible with the need to develop estimates at the outset of a project. The use of SLOC within formalised effort/cost estimation models, such as COCOMO (Boehm 1981) and SLIM (Putnam 1978), would also be prohibitive for early stage estimation. Consequently, the conversion of other size measures into SLOC has been incorporated into such models to facilitate their use in early stage estimation.

2.3.1.2 Functional Sizing

The most established alternative size measure is concerned with the functionality required by a system. FPA was proposed as a size estimation approach which addressed some of the issues present with the use of SLOC (Albrecht, 1979; Albrecht and Gaffney, 1983). Instead of viewing software size in terms of lines of code, the required functionality from the user perspective is used to establish an estimate of the size in terms of function points. This view of the system is comprised of five types of Base Functional Component (BFC), according to how the functionality is required by the user.

(1) Internal Logical File (ILF)
(2) External Interface File (EIF)
(3) External Input (EI)
(4) External Output (EO)
(5) External Inquiry (EQ)

The relative complexity of each identified component function is assessed and classified as low, average or high. This assessment is based on counting the number of Data Element Types (DET) and Record Element Types (RET) involved in the function. The sizes of the individual function values are then added to give the total number of unadjusted function points (UFP) for the proposed system. This unadjusted value can then be adjusted using a Value Adjustment Factor (VAF) calculation to determine the final functional size. This VAF comprises 14 general system characteristics with which to assess the technical and quality aspects of the system. The benefits of performing this adjustment are generally not supported by the evidence. The unadjusted total functional size has been demonstrated to be more strongly correlated to the final size of the software (Kemerer, 1987; Gaffney and Werling, 1993). Lokan (2000) found that there was no significant difference between using unadjusted and adjusted functional sizes for predicting development effort.

Table 8 – Base Functional Components of main Functional Sizing Methods

FSM	BFCs	BFC Complexity Composition	BFC Complexity Weighting	Relative Size Contribution	General Applicability
IFPUG & NESMA	Internal Logical File	Data Element Types Record Element Types	Low Average High	(7, 10, 15)	IFPUG and NESMA are stated as applicable for all Functional Domains.
	External Logical File			(5, 7, 10)	
	External Input	Data Element Types File Types Referenced		(3, 4, 6)	
	External Output			(4, 5, 7)	
	External Inquiry			(3, 4, 6)	
Mark II	Input Transaction	Data Element Types	-	0.58	Generally applicable for, but not limited to, Business Information Systems.
	Output Transaction	Data Element Types	-	0.26	
	Process Transaction	References to Data Element Types	-	1.66	
COSMIC	Entry	Entries	-	1	Business Application Software/Real Time
	Exit	Exits	-	1	
	Read	Reads	-	1	

	Write	Writes	-	1	Software
FISMA	Starting Icons; Login Windows; Menus; Selection Lists; Inquiry Windows; Browsing List Windows; Report Generation Windows	Data Items Reading/Writing References	-	0.84	Stated as for all software in any functional domain
	3-Functional Windows; 2-Functional Windows; 1-Functional Windows		-	1.57	
	Output Forms; Reports; Emails/Texts; Monitor Screen Outputs		-	1.70	
	Online Message Out; Signals to Devices; Batch Records Out;		-	1.14	
	Online Messages In; Signals from Devices; Batch Records In		-	1.57	
	Entities/Classes; Other Logical Records		-	1.75	
	Calculation Routines; Simulation Routines; Formatting Routines; Database Cleaning Routines; Security Routines; Other Algorithmic Routines		-	0.63	

2.3.1.3 Functional Sizing Methods

The subsequent development of Functional Size Measurement (FSM) methods has evolved to the extent that five of the approaches have been established as ISO standards. Table 8 provides an overview of how the BFCs of each of these methods contribute to the overall functional size of a project. The original FPA approach provided the fundamental basis for the International Function Point Users Group (IFPUG) ISO standard (International Organization for Standardization, 2009), which is the most commonly used FSM. The NESMA ISO standard (International Organization for Standardization, 2005) is also based on the FPA approach, and has evolved to the point where there are minimal differences between it and the IFPUG method.

The conventional FPA approach has been examined within research and subjected to challenges regarding the suitability for software estimation. Table 9 provides an overview of the fundamental issues that have been investigated as potential shortcomings of FPA.

Table 9 – Overview of research issues regarding FPA

Issue	Problems	Responses
Assessment of Complexity	The complexity issue lies with the fact that once a function falls within the upper boundary there is no increase in the functional size allocated regardless of how much higher the number of data elements and record elements become.	A potential solution to this issue was proposed, involving the decomposition of overly complex functions into smaller functions until every resulting function falls within the specified 'high' complexity range (Kralj et al., 2005). In evaluating this method across 20 projects, selected primarily for their recognised high complexity, it was found to produce on average a 15% increase in the functional size obtained. The subsequent use of these values to predict actual effort demonstrated, in some cases, substantial improvements in accuracy. An alternative approach is to use additional complexity categories to cater for a wider range of sizes.

Variance in Total FP across estimators	Variance in FPA points counted by individuals is inherent in the FPA approach. This variance was reported to be as high as 30% (Low and Jeffrey, 1990)	The development of expertise in the use of FPA can reduce such variance to being within 10%-12% for more experienced counters (Kemerer, 1993). This validates the approach in terms of achieving a sufficient degree of consistency once sufficient expertise has been developed.
Correlations between some types of function point components	Investigation of the sizes of the individual component types has resulted in the identification of correlations between them (Lokan, 1999). Such correlations have been presented as an invalidation of function points (FP) as they cannot be considered to be well formed metrics (Kitchenham and Kansala, 1993). The base components cannot be considered to be completely independent of one another as some elements are, in effect, being counted twice.	These correlations support the derivation of total function points from consideration of only certain types of component. For example, International Software Benchmarking Standards Group (ISBSG) Benchmark data (ISBSG 2009) on component sizes enables a size estimate to be derived from any one or more of the base component sizes. The use of industry wide data would represent an indication of 'typical' size, which may limit its utility to the provision of a secondary measure to validate primary estimates.
Use of Complexity Weightings	Kitchenham and Kansala, (1993) reported that the weightings applied to the base functional components only provided a slight improvement upon the predictive accuracy of the raw counts.	Calibrating the complexity weightings according to local development projects can improve the predictive accuracy. The relative effectiveness of forgoing the use of complexity weightings can be utilised in the form of simplified functional sizing methods.

Silva et al. (2016) reviewed published research on the reported problems from 2002 to 2014. The studies included in this review were reported to be primarily focused on technical issues with the method. These included the weightings and complexities used by the FPA method and the suitability of the associated general system characteristics (GSC) used to adjust the size estimate. A similar review by de Freitas Junior et al. (2015) reported similar findings on the focus of research during this time period, but also provided an assessment of the proposed improvements to the FPA method. Of the studies that attempted to improve the weighting and complexities, in 90% of them the approach adopted was to use one or more AI techniques such as fuzzy logic or artificial neural networks. This was the only aspect of the FPA method where improvements in estimation accuracy were reported, ranging from 8.8% to 50%. The use of AI techniques would, however, represent a barrier to widespread adoption of such an approach. The implication of these reviews is that practical issues, such as the feasibility of identifying the required functionality, have not been widely acknowledged as problems to be addressed. Chapter 4 of this research thesis investigates the issue of how functional sizing can be extended to accommodate the practical limitations of applying the method to commercial project documentation.

The remaining FSM methods have been developed to address issues that limit the applicability of the FPA approach. Enhancements to the FPA approach were proposed in the Mark II method (Symons, 1988), which has subsequently been standardised (International Organization for Standardization, 2002). This method aims to reduce the subjectivity of identifying logical files by using more straightforward entities. The logical transactions are viewed in terms ‘Inputs’, ‘Processes’ and ‘Outputs’. The FPA approach utilises a rigid scale for defining the complexity of its functional components, which has the effect of significantly limiting the degree to which projects can be estimated to differ in size. In particular, by not counting the number of DETs beyond the minimum threshold value for an upper boundary, the FPA approach imposes a maximum size on each BFC type. The Mark II method uses the specific number of DET identified, bypassing the limitation of having maximum boundaries for assigning complexity. The relative size contributions stated in Table 8 for each of the BFC types correspond to industry standard weights, with the ‘Process’ transaction type providing the most significant contribution. This method does not consider complex algorithms or real time requirements, and can therefore not be considered suitable for functional domains that demonstrate such characteristics. The applicability of this method is therefore comparable to the IFPUG and NESMA methods.

The development of the Common Software Measurement International Consortium (COSMIC) FSM method was aimed at extending the applicability of functional sizing to real time software. The current version of the method is standardised under the name COSMIC (International Organization for Standardization, 2011), and is stated as being applicable for any configuration of business and/or real time software. The standard is not defined for software incorporating complex algorithms, or specialised applications such as computer game software. It is, however, indicated that it may be feasible to adapt the COSMIC approach to provide a meaningful measure of functional size for such application domains within a local context. This method considers the specific data movements involved in a process, which distinguishes it from the conventional FPA approach. Each individual data movement corresponds to one of ‘Entry’, ‘Exit’, ‘Read’ or ‘Write’ and contributes equally to the overall functional size. It is not limited to considering functionality that is explicitly evident to the user, and does not require functions to be classified e.g. Input or Output. There is no upper limit on the size of a function which avoids obvious underestimation for functions with a substantial number of data movements. The assessment of complexity may require greater detail about the application in order to identify every specific data movement. Desharnais and Abran (2001) acknowledged the challenge of providing estimation guidelines that enable different size measurers to produce the same results when using the same requirements information. They proposed the use of a Case Based Reasoning approach to assist with common ‘problems’ encountered when applying measurement guidelines. As with the standard FPA method, the use of COSMIC BFCs for effort estimation has been investigated (Tunalılar and Demirors, 2011). A dataset from a single company was used from which COSMIC size estimates were developed. An Artificial Neural Network and multi-regression analysis was used to build an effort estimation model from the sizing data. The results reported indicated that it was necessary for the effort estimation model to treat the BFCs separately for sufficiently accurate estimates to be produced. The importance of having a more detailed profile than just the overall functional size has therefore been highlighted by this study.

The Finnish Software Measurement Association (FISMA) sizing approach (International Organization for Standardization, 2010), addresses the level of subjectivity inherent in functional sizing stemming from the identification of the BFC. FISMA identifies seven classes of component, from which 28 BFC types are derived. These main classes are:

- Interactive end-user navigation and query services
- Interactive end-user input services
- Non-interactive end-user output services
- Interface services to other applications
- Interface services from other applications
- Data storage services
- Algorithmic and manipulation services

The complete list of 28 BFC types is shown in Table 8, divided into the 7 respective classes. This approach provides a significantly more detailed classification of functionality, which is intended to have the effect of reducing subjectivity by more precisely defining each BFC. This simplifies the identification of each BFC type by reducing the level of uncertainty in ascertaining when each instance of a BFC has been identified. The complete classification of BFC types incorporates the types of functionality considered by the other main FSMs, as well as additional aspects concerned with indirect processing and algorithmic complexity. It is only necessary to measure those BFC that are identified in the system under consideration, or indeed only those relevant to another specific FSM. This enables the FISMA approach to be implemented in order to develop a size estimate compatible with a preferred FSM.

$$S = a + n/d + r/c$$

S = size of query measured in functional sizing units (fsu)
n = number of data items or fields
r = number of reading references to entities
d = BFC class specific number of data items giving 1 fsu
c = BFC class specific number of reading references giving 1 fsu
a = establishment cost of 0.2 fsu

Measuring the size for each BFC type incorporates counting such attributes as the number of data items, the number of references to entities required etc. The example equation shown above corresponds to the ‘Interactive end-user navigation and query services’ class. The specific attributes required are dependent upon the specific BFC class. Each BFC class has specific constraint values included in the calculation of size for that class. These constraint values are used to attribute the size contribution necessary, in terms of how many data items or references, to constitute 1 fsu for each respective BFC class. Specific values are suggested as part of the ISO standard, but they can be ‘corrected’ as required to preserve the balance across the set of BFC classes. The overall functional size for the system is calculated by adding the size of each of the BFC classes. The relative size contributions shown in Table 8 for each BFC class were derived from the specific class numbers stated in the ISO standard. The calculation of these relative sizes used the assumption of one data item and one reading reference where required in the equation for each BFC class. This provides the relative functional size for each BFC class when the data items and reading references are held constant. For the equation above, where $d=7$ and $c=2$, the calculation would therefore be as follows:

$$\text{Relative Size} = 0.2 + 1/7 + 1/2 = 0.84$$

2.3.1.4 Reconciling the Functional Size Measurement Methods

The development of these contrasting FSM methods presents a challenge in aligning the use of functional sizing with the practical requirements of software estimation. Table 10 outlines the main reconciliation factors surrounding the multiple FSM methods and associated developments made in accommodating these factors.

Table 10 – Reconciliation of FSM methods

Reconciliation Factor	FSM Developments
Suitability of FSMs	Each of the FSM standards provides an indication of their applicability in terms of which types of applications can be estimated. The scope of the applicability varies across the various methods, and in terms of the perception of practitioners of functional sizing.
Selection of FSM	The selection of an appropriate FSM is dependent upon both the purpose of the size estimate, and the constraints on the estimation process.
Convertibility of FSMs - Functional sizing data, both local and industry wide, provides a basis of comparison between different projects. The degree to which one FSM value can be converted into another FSM value is therefore important for enhancing the depth of comparative functional sizing data available.	The conventional IFPUG approach and the fundamentally similar NESMA approach are generally considered to be most suitable for data intensive applications, while the COSMIC approach is more suited for applications with greater algorithmic complexity. This presents a problem in establishing a conversion formula between the two approaches, as the results obtained when using either are influenced by the nature of the application being developed. This must therefore be accounted for when attempting to perform any conversion. The ability to convert is also influenced by the nature of the base components of each FSM e.g. COSMIC reads each data movement individually so a Read operation and a Write operation on the same Data Element Type within the same transaction would mean that the Data Element Type would be counted twice,

	whereas with IFPUG it would only be counted once.
Unification of FSMs	An encouraging solution to this problem was the development of a Unified Model (UM) that simultaneously produces FSM counts for three of the main methods (IFPUG, Mark II and COSMIC) from the same data (Demirors and Gencel, 2009). The UM reflects the measurement concepts and rules of each FSM by identifying the necessary base components and using set operations to define and relate the sets of base counts for each method. When tested on two case study projects the UM was able to produce the same totals as individual manual counts of each FSM. While subjected to limited evaluation thus far it suggests potential for simplifying numerous issues faced by the estimation community and countering the increasingly fractured development of these methods. FISMA KISS method allows size estimates to be developed corresponding to any other FSM (Forselius, 2006). Early and Quick Function Points are compatible with most FSM methods (Santillo et al., 2005).

2.3.2 Expert Judgement

In Section 2.2.2 it was identified from surveys that the most commonly used forms of estimation within industry have been expert judgement-based approaches. The results reported in those surveys indicated that these approaches are commonly viewed as being more transparent and requiring less effort than more formal estimation methods.

Numerous studies have been conducted to evaluate the relative performance of model-based methods against expert judgement estimation. The evidence gathered from such studies was evaluated and deemed to be inconclusive in terms of establishing overall superiority for either approach (Jørgensen, 2004). The strength of expert judgement may lie in not being restricted to considering only that information which is captured by an estimation model. In contrast, expert judgement estimation may be more open to bias, either internally from the ‘expert’ or externally from organisational issues. The comparison of the performance of expert judgement and model based estimation approaches does not imply that they are mutually exclusive options. The blurred distinction between these approaches has been highlighted, and indeed encouraged (Jørgensen and Boehm, 2009). Model-based approaches inherently require some degree of judgement-based assessments in applying the specified guidelines for a method. In contrast, expert judgement-based approaches often adopt some degree of formality in their completion, even if it is not required to be as rigorously documented as with model-based approaches.

Software Engineering has been seen to be distinct from other disciplines, such as medicine, business and project management that incorporate the use of estimation within their field. In these other disciplines, model-based estimation approaches have clearly demonstrated superior performance in comparison to expert judgement (Jørgensen, 2004). From this review of studies on expert judgement Jorgensen suggested 12 'best practice' guidelines, shown in Table 11. Each of these estimation principles is based on empirical evidence reported in expert judgement research, incorporating lessons learnt either from other disciplines or from a specific software development environment. The guidelines therefore correspond to issues that require further research into their general applicability within Software Engineering.

Group based approaches to expert judgement have been found to be effective at reducing the level of variation inherent in estimation. For example, the structured Wideband Delphi technique uses initial anonymous individual estimates, which are discussed on an iterative basis until a unanimous vote is obtained on accepting the average estimate (Boehm, 1981). This technique has been found to reduce the estimation error of the initial group average by approximately 40% (McConnell, 2006).

Expert judgement-based approaches are not based on standardised metrics and the influence of the choice of work unit for the estimate was investigated by Jorgensen (2016). In this study, software professionals from two companies were randomly allocated to either a work-hours or a work-days estimation group. Each group was required to estimate, in a randomly assigned order, two projects that differed greatly in terms of size and complexity. The work-hours group estimates were on average lower for both projects, with a difference of 33% for the larger project and 49% for the smaller project. The work-hours group took slightly less time to develop their estimates than the work-days group. The order in which the projects were estimated was also found have an effect independently of the work units used for the estimate.

Table 11 - Expert Judgement Guidelines (adapted from Jørgensen, 2004)

Overall Goal	**Estimation Principle**
Reduce situational and human biases	1 - Evaluate estimation accuracy, but avoid high evaluation pressure
	2 - Avoid conflicting estimation goals
	3 - Ask estimators to justify and criticize their estimates
	4 - Avoid irrelevant and unreliable estimation information
	5 - Use documented data from previous development tasks
	6 - Find estimation experts with relevant domain background and good estimation records
Support the estimation process	7 - Estimate top-down and bottom-up, independently of each other
	8 - Use estimation checklists
	9 - Combine estimates from different experts and estimation strategies
	10 - Assess the uncertainty of the estimate
Provide feedback and training opportunities	11 - Provide feedback on estimation accuracy and task relations
	12 - Provide estimation training opportunities

In this case estimating the smaller project first led to lower estimates for the larger project, and estimating the larger project first led to higher estimates for the smaller project. These results demonstrate that the anchor effect, widely examined in other fields of research such as psychology and behavioural science (Furnham and Boo, 2011), is evident within software estimation. When the experiment was repeated using different participants and a different estimation approach, in this case by analogy, the work-hours estimates were again lower than, in this case by 59% on average, the work-days estimates. This study suggests that even with different estimation approaches being used the choice of work unit affected the resulting estimates. The study was, however, limited to effort estimation and did not consider whether this pattern would extend to size estimation. The estimators in the study were considering the effort required to develop the project, which relates directly to their experience of working on projects. The size of a project is more indirectly related to experience of the effort involved in working on projects, so it is not clear whether the choice of size unit would have a similar effect on developing estimates. The use of a standardised sizing method would prevent such an effect being present within size estimation, thus providing greater confidence in the generalisability of the results.

The greater reliance on the estimator with expert judgement approaches raises the issue of identifying what distinguishes ‘experts’ within the field of software estimation. Boetticher and Lokhandwala (2008) conducted an experiment wherein 178 participants developed effort estimates for 28 modules of a software project using their own judgement. The relationship of various characteristics of the estimator, such as qualifications and domain experience, with the relative accuracy of their estimates was investigated using a machine learning approach. The researchers were able to produce a simple decision tree with which to, with an accuracy of 93.3%, correctly identify ‘good’ estimators. The general applicability of these results is limited by the use of a single project, but suggests that focusing on the characteristics of the estimator may contribute to the success of expert estimation.

In addition to identifying good estimators, improving the performance of experts can also be considered by reviewing how the estimation performance affects the accuracy of future estimates. A study conducted by Jorgensen and Gruschke (2009) used two groups of software developers to estimate the effort and uncertainty associated with each of a series of development tasks. The developers within one of these groups were given feedback between each task and asked to reflect upon their performance, while the control group received no feedback. The results showed that over the five tasks the intermediate feedback and reflection had no effect on the accuracy of either the effort or uncertainty estimation between the two groups. Reviewing the estimation performance of experts would therefore require greater understanding of how the necessary feedback should be determined and communicated.

Huijgens and Vogelezang (2016) investigated the effects of estimating projects that incorporate functionality from an in-house software portfolio on estimation accuracy. The difference between estimated and actual effort was found to decrease over time, with a pattern of under estimation switching to one of over estimation. Development productivity was found to have increased over time, with the effort data forming a positively skewed distribution. The improvement in estimation accuracy suggested the expert estimators were able to learn in light of improving productivity of the developers. Equiniti-ICS incorporate ‘core’ functionality into new projects as part of maintaining a base product of common functionality for reuse. The use of software sizing of the full functionality of a project in this thesis can therefore be contrasted with the expert estimates made within the organisation. The expert knowledge of which aspects of required functionality should be comprised of ‘core’ functionality would intuitively favour the existing estimation practices of Equiniti-ICS. These expert judgement-based estimates are, however, based on bid documentation that may limit the extent to which ‘core’ functionality can be accurately identified at that stage.

In Section 2.2.3 it was identified that academic research has been primarily focused on the development and evaluation of model-based estimation approaches, although there has been a trend of gradually increasing focus on expert judgement-based estimation. This more extensive research into improving model-based estimation without achieving demonstrable superiority has led to the suggestion that there remains comparatively more research potential within expert estimation (Jorgensen and Shepperd, 2007).

2.3.3 Estimation by Analogy

For estimation by analogy approaches that rely on expert judgement, the effect of bias on the resulting accuracy has been demonstrated (Jorgensen, 2013). The effectiveness of using relative estimates was reduced the further the comparison project differed from the estimation project. For example, effort estimates tend to be anchored towards the comparison project to some degree so the actual difference in size between the two projects will be underestimated. The presentation of the request for the estimate was found to influence the estimator by suggesting a desirable outcome for the estimate when none was intended.

Methods have been developed to facilitate estimation when limited project data is available, such as the Sparse Data Method (SDM) (Shepperd and Cartwright, 2001). In this approach every project is compared with every other project to give an approximate relative value e.g. Project A required three times as much effort as Project B. It is therefore only necessary for the actual value to be known for one project in order for the values to be derived for every other project. The comparisons can be across different levels of components of a project, enabling a bottom-up approach where specific components are summed together to provide overall comparisons. The conceptual support for this method is reported in the fact that at the early stage on a project relative estimates provide more accuracy than actual estimates. This study evaluated the SDM against expert estimates, finding it to be more accurate in 21 cases, compared to 12 cases where it was less accurate. Sensitivity analysis revealed that when the 'reference' case was of median relative size the results were clearly superior to when it was at either extreme end. The SDM utilises the paired comparison technique that was developed to address the scarcity of data available at the beginning of the project (Miranda, 2001). The time consuming process of making the comparisons required between each of the projects limits the feasibility of using such an approach. Development of a predefined company specific 'ruler' against which to make future comparisons was proposed to aid the estimation process (Lester et al., 2004). Successful studies have also found that equivalent estimation results can be achieved even when a reduced number of comparisons are made, requiring missing values to be compensated for (Miranda et al., 2009). It was found that good individual user story estimates could be achieved with only half of the possible comparisons. For situations where a rough estimate is considered acceptable it was found that just a quarter of the possible comparisons could provide an adequate overall project estimate.

Analogy-based approaches to estimation may incorporate the use of expert judgement and/or analytical software tools. The objective is to compare the proposed project with previously completed projects in order to derive an estimate. This involves an assessment of how analogous the required features are to previously developed features, providing the basis for determining the relative effort that will be required for the project. The intuitive nature of this approach to estimation can provide an indication of why its use within more than 60% of organisations has been reported (Heemstra, 1992; Wydenbach, 1995; Yang et al., 2008). Estimation by analogy offers a more formal and repeatable methodology than expert judgement estimation, but the suitability of this approach is more explicitly constrained by the dynamic nature of software development. The process requires the retrieval from historical datasets of the most similar projects to the proposed project, with the estimation results then fed back into the dataset for use in future project estimation. Factors such as the degree to which the development approach evolves, changes in development personnel and changes in the technology developed all impact on how analogous the historical projects in the dataset are to new projects. Maintaining a historical dataset requires not only the accumulation of project data, but also an ongoing assessment of the relevance of the historical data.

Evidence on the effectiveness of analogy-based estimation within research is inconclusive. Evidence for and against the use of analogy for estimation was found in a review of 20 research papers (Mair and Shepperd, 2005). This review suggested that variation in the research methodologies and in the estimation methods evaluated may contribute to the inconclusive evidence. The effectiveness of any one estimation method may be dependent on the quality of available historical datasets and therefore the emphasis is on determining the appropriate method in each case.

The implementation of estimation by analogy involves the selection of appropriate selection criteria for analysing the historical dataset. Identifying suitable projects, and which data about those projects, is fundamental to the effectiveness of the estimates produced. The size of the dataset and the selection criteria has implications for how the analysis should be completed in terms of human judgement versus artificial intelligence. Use of human estimators for analogy-based estimation has been shown to provide comparable performance with equivalent software tools for smaller datasets with limited selection criteria (Walkerden and Jeffrey, 1999). The human estimators in this study were inexperienced students, and while they demonstrated superior performance in selecting appropriate projects for comparison, the estimates derived from these were more prone to error than with the use of software tools. The level of computation required affected the quality of the estimates produced. Increasing the size of the dataset, and the number of selection criteria, would therefore increase the need for the use of software tools.

Wen et al. (2012) conducted a systematic literature review of machine learning based approaches to software development effort estimation. The most frequently used approaches, accounting for 80% of selected studies, were found to be Case Based Reasoning (CBR), Artificial Neural Networks (ANN) and Decision Trees (DT). In terms of the estimation accuracy of each approach CBR and ANN were found to be superior to DT, but there was no consistent pattern evident in terms of identifying an individual best approach. The characteristics of the dataset, such as its size and completeness, affect the suitability of each particular machine learning approach.

Development of Case Based Reasoning (CBR) software tools, such as ANGEL, enables estimation by analogy to be performed in these circumstances (Shepperd et al., 1996). The main weakness associated with such CBR tools is the inability to determine the appropriateness of the datasets being searched (Keung et al., 2008). The Analogy-X method, developed by Keung, provides a means of assessing the suitability of a dataset for analogy by determining whether the predictive relationship for a particular project feature is statistically significant. It has been proposed that the algorithmic approach to problem solving generally utilised within CBR tools may contribute to the limited effectiveness of the tools (Mair et al., 2009). In this view software effort estimation constitutes an ill-defined problem which requires a less restrictive approach to solving. CBR only facilitates surface-level comparisons to be made between cases when a more fundamental structural similarity is necessary to be determined.

2.3.4 Object-Oriented Estimation

The growth of Object-Oriented software development led to an increasing focus on developing software estimation approaches tailored for UML based project documentation. Object-Oriented concepts such as inheritance and communication between objects, which are not accounted for by FSM methods, may affect how the software size is reflected in the development effort. The Object-Oriented approaches to size estimation can be broadly categorised as either adapting the use of function points for object oriented systems, or developing conceptually similar methods using UML concepts.

2.3.4.1 Object-Oriented Functional Sizing

Conventional functional sizing approaches can be applied to Object-Oriented based documentation approaches such as UML modelling, where the estimator implicitly identifies the BFC from the project documentation. This process can be made explicit by specifying how the Object-Oriented concepts are to be mapped to the corresponding functional sizing BFC. Fetcke et al. (1997) investigated how FPA could be applied to an Object-Oriented methodology, focusing on identifying the required functionality from use cases. Data Functions are identified from entity (or unknown) types of object. Aggregation of objects is analogous to the logical combination of RETs into Data Functions, whereas objects within an Inheritance hierarchy can themselves be either a RET or a Data Function. Transaction Functions are identified from each use case, where the number of functions present is identified according to the FPA counting rules. The complexity of Transaction Functions, in terms of DET and file types, is also affected by Aggregation and Inheritance.

Uemura et al. (2001) focused on class and sequence diagrams as the basis for mapping Object-Oriented concepts to FPA. In their approach, messages between objects in the sequence diagrams are assessed for suitability as a Transaction Function. The project documentation required for this approach is generally not available as early in the project lifecycle as the use case diagrams. This limitation is offset, to some degree, by the refinement of a software tool developed by the authors to partially automate the estimation process.

Abrahão et al. (2006) developed the OO-Method Function Points (OOmFP) estimation method, which conforms to the IFPUG method. The OOmFP method uses the Object, Dynamic, Functional and Presentational Models to measure the size of the system. The Object Model enables the Data Functions to be estimated, and the Transaction Functions are estimated using the user interactions and process described in the other models. Measurement rules enable the identified Object-Oriented components to be mapped to the functional sizing components while retaining consistency with the IFPUG method.

The Object Oriented Function Point (OOFP**)** method adopts such an approach, based on object models, where data functions and transaction functions map to classes and methods respectively (Antoniol et al., 1999). Some degree of flexibility is provided by this method. For example, it is left to the estimator to decide which option of handling class aggregation they prefer. This enables the method to adapt to the specific circumstances of the estimate, but adversely introduces a source of inconsistency in how it is applied to different projects. Ram and Raju (2000) used generally the same mapping rules from the OOFP method, but provided additional rules for assessing the complexity of classes. Živkovič et al. (2005a) sought to unify the contrasting mapping rules found within previous approaches, resulting in their Object Oriented Function Point Method (OOFP2).

2.3.4.2 UML Component Sizing Methods

The direct use of UML concepts as the components for size estimation removes the requirement of accommodating consideration of functional sizing methods. The UCP method (Karner, 1993) focuses on identifying actors and use cases as the main components. As in FPA, the components are classified as simple, average or complex, with the complexity determined by counting the number of transactions in a use case. Technical and environmental factors are used to adjust the UCP total, providing the basis for an estimated effort to be derived. This approach has been investigated in a number of studies, including a case study where four different development companies developed a system from the same functional specification (Anda et al., 2005). The results reported that the development effort associated with use cases was strongly related to the number of transactions within those use cases. The variance in the intensity of the focus on quality associated with the development approach demonstrated by each company, however, revealed that the complexity factors fail to cater adequately for non-functional requirements. The ineffectiveness of the complexity factors utilised by the UCP method is also consistent with that aspect of FPA.

Further criticism of the UCP method includes the failure to adequately consider 'hidden' business complexity, and for being open to the same subjectivity in assessing complexity as other estimation approaches (Vinsen et al., 2004). This apparent shortcoming may, however, be more indicative of an inconsistency between practitioners in the level of detail included in the use case description than of a specific limitation of the method in this study. The presence of moderate correlations between SLOC and both External Use Cases and Classes in 12 projects has been reported (Chen et al., 2004). Correlation of SLOC with decomposed Use Cases was found to be weak in comparison. The lack of uniformity established in how UML sizing metrics should be used was suggested as a factor for the moderate performance of this approach.

Carroll (2005) reported the results of using the UCP method on over 200 software projects within a development organisation certified at Level 5 of the CMMI. For 95% of the projects the estimated cost was within 9% of the actual cost of the project. System features considered uncharacteristic in terms of the development effort required were estimated separately in order to prevent the UCP estimates from being skewed. This required the accumulation of historical data on the development effort recorded for these system features, and evaluating that data according to the technology used for their development. The analysis of the completed project data is used to identify the underlying causes of deviation within the development process, including the estimation model used within the organisation. The estimation process is subject to continuous process improvement indicative of a Level 5 Capability Maturity Model Integration (CMMI) development organisation. The degree to which the estimation performance is dependent on this level of commitment to improvement, however, would suggest that this is atypical of the commercial reality for the majority of development organisations.

UCP 2.0 presents an adapted approach, which provides specific guidelines on how the approach should be implemented (Frohnoff and Engeroff, 2008). The stated aim of this approach was to standardise the format of the specification used and assessment of complexity. An evaluation on 20 projects found that the format of the use case specification used had a significant effect on the overall UCP count (Frohnoff and Engeroff, 2008). An interesting contrast was found where state charts and business process descriptions provided the least variability in the estimation results despite 'experts' viewing these formats as the least suitable amongst those formats evaluated.

Kashyap et al. (2014) proposed addressing the perceived weakness of the complexity classification used in the UCP method. The use of three distinct complexity levels, each with a fixed weighting, simplifies the classification into ranges of transactions for each level. Fuzzy Logic was used to develop a classification scheme where the complexity weighting is adjusted gradually as the number of transactions in a Use Case increased.

The iUCP method (Nunes et al., 2011) enhances the UCP approach by considering how focusing on Human Computer Interaction provides a more thorough understanding of user requirements. For example, six complexity weightings are used for actors, in contrast to the three weightings used in the original UCP method. This allows for a greater assessment of the number of roles played by each actor, distinguishing between ‘system’ and ‘human’ actors, and enabling their complexity to be more accurately estimated. This approach views use case transaction as solely the responsibility of the system and therefore reduces the difficulty of distinguishing actors and use cases from each other. Alves et al. (2013) evaluated effort estimates, developed using both the original UCP and the enhanced iUCP methods, for seven software projects. This study reported that the iUCP method did not produce statistically significant improvements in effort estimation accuracy, indicating that further refinement is necessary with this estimation approach.

Ochodek et al. (2011) investigated the utility of using simplified UCP approaches for effort estimation of 14 software projects from multiple development organisations. The use of adjustment factors was found to not result in a significant improvement in the accuracy of effort estimates. Using a simpler measure of the number of steps used in a Use Case, rather than the number of transactions, was found to provide a similar level of accuracy. The downside of using steps rather than transactions is that they are more sensitive to variation in the writing style of documentation in the use cases. The raw counts of steps and transactions were also found to be as effective in estimating effort as using the full UCP method.

The Kassab et al. (2014) survey revealed that conventional Function Points were only used by 13% of the software professionals that indicated the use of an estimation method. In contrast the Use Case Points (UCP) method was reported to be used by just over 25% of those respondents. This may suggest that the lack of standardisation is not seen as a significant drawback in adopting UCP methods. The relative simplicity of using UCP, in comparison to functional sizing methods may also lead to a perception that it has a lower barrier to adoption within the development lifecycle.

2.3.5 Agile Estimation

Making estimates for Agile-based development project iterations typically involves comparing 'user stories' yet to be developed to establish their relative size. These comparisons are made in group sessions, commonly referred to as 'planning poker' (Cohn, 2005). These sizes are commonly expressed in the form of Story Points. For example, the expected smallest story could be assigned a value of 1 Story Point with the other stories assigned a value relative to that smallest story. The number of Story Points completed during an iteration of a project is referred to as the ‘velocity’. Estimates from previous iterations are used to inform subsequent iterations, due to the availability of historical data early in the development of a project.

Approaches to Agile estimation can be applied to conventional iterative estimation where the concept of 'velocity' is applied to sizing rather than development productivity (Hericko and Živkovič, 2008). An estimate is made for the total size of all the use cases specified as well as a subtotal for those use cases to be developed in the next iteration. The actual size of those use cases subsequently completed is then used to recalculate the overall total and subtotal for the next iteration. This refinement, assisted with the use of additional detail from class diagrams when they become available, enable the estimate of total size to converge on the actual size. While this convergence was effectively demonstrated on three relatively small business information projects the issue of handling requirements instability remains a challenge to be addressed. The applicability of conventional sizing techniques in Agile-based development may be correlated with amount of effort that they require.

A systematic literature review of effort estimation studies in the context of agile software development from 2001 to 2013 indicated that relatively subjective expert judgement-based approaches were most commonly used in research (Usman et al., 2014). This included both individual based estimates and group based approaches such as Planning Poker. A majority of the studies (72%) reported the use of a within company datasets, with only 8% involving the use of cross company datasets. This may be attributed in part to the lack of publicly available datasets in comparison to other more traditional estimation approaches. In terms of size metrics, the most commonly used were UCP and Story Points in line with the nature of the project documentation available. Size metrics were found to be used in 80% of the research articles included in the review, although traditional measures such as Function Points or Lines of Code were rarely used. The reported usage of size metrics in industry by Kassab et al. (2014) indicated that Story Points (28%) and UCP (26%) were the most commonly used.

Story Points measure size in terms of the business value of each user story. This involves implicit consideration of non-functional and environmental aspects of a project. Function Points are therefore a more specific measure that is not directly equivalent to Story Points. In certain countries, such as Brazil, Function Points are legally enshrined as the basis for measuring some software development services, and hence the payment for those services provided. The feasibility of the use of FPA as a measure of ‘velocity’ within Agile-based development was investigated by Santana et al. (2011). This study determined the correlation between Story Points and Function Points for a dataset of 2191 stories across 18 iterations from a Brazilian Government Agency. The Story Points were reported to range from 262 to 819, with a mean of 554. For the same datasets the FP values ranged from 41 to 159, with a mean of 79. A statistically significant linear correlation of 0.7137 was found between the two different size measures. In contrast, Huijgens and van Solingen (2014) found a weak negative correlation and a strong negative correlation, for two different Dutch Banks, between Story Points and Function Points. The lack of standardisation for Story Points was proposed as an explanation for the contrasting results of this study. The value of Story Point estimation data would be considered to be limited to the software organisation that developed it.

Using functional sizing in an Agile-based development context has also been proposed as a means of providing greater objectivity in the estimation process. Desharnais et al. (2011) highlighted the use of the COSMIC method to provide validation to the completeness of the documented functionality due to the provision of a five-point ordinal scale for this purpose in its guidelines. COSMIC size estimates may also be relatively more suitable for sizing individual user stories due to each discrete data movement being individually counted. The use of functional sizing for estimating individual tasks was evaluated using both the FPA and COSMIC methods (Popović and Bojić, 2012). For tasks that were sized at either the lower or upper boundary values of the FPA complexity weighting tables, the size estimates were not reflective of the range of actual efforts associated with those tasks. In practice, the user stories would be comprised of multiple tasks so such limitations of the FPA method would only be significant if most of the included tasks were either particularly simple or particularly complex.

2.3.6 Learning from Historical Data

The estimation process involves using historical data accumulated from previous projects as the basis upon which the analysis of a new project can be developed into an appropriate estimate. This historical data can be accumulated from 'within company' projects or from 'cross company' project data. Depending on the specific estimation approach, the cross company project data may be obtained from publicly available industry datasets or be incorporated directly into the estimation method e.g. represented as constants in mathematical formula.

Regardless of the specific estimation approach that is adopted, conventional wisdom suggests it is more beneficial to utilise local historical data from within the same organisation. Projects developed internally would be expected to be more similar to new development projects than those represented in industry datasets. The evidence reported in a systematic review of cross company versus within company cost estimation studies was inconclusive (Kitchenham et al., 2007). Comparisons of estimation models incorporating each type of historical data did not indicate overall superior performance for either type. Differences in how individual studies were conducted did in part prevent definitive conclusions from being drawn from the results. The suitability of either type of historical data may therefore vary according to the circumstances of each development organisation. The estimation performance from the use of local historical data is affected by how homogeneous the locally developed projects are. The use of publicly available industrial datasets must consider how closely the internally developed projects are ‘typical’ of those within the dataset.

The effect of the quality of historical data used in the estimation process was addressed by Ohlsson et al. (1998). This study examined whether the level of formality used in recording the historical data affected the effort estimation accuracy. The results indicated that even with a degree of formalism applied in the collection of historical data there was not a significant improvement in estimation performance. The fact that this study was conducted in an academic setting with inexperienced students prevents other factors being accounted for e.g. the previous experience of the estimation personnel in successfully utilising historical data.

Regardless of the source the following barriers to the use of historical data in the estimation process are present:

- Project data has a limited lifespan in terms of applicability due to the evolving nature of both the systems developed and the development methods employed.
- Developing a local historical dataset can require considerable time and effort to ensure accuracy, and must be maintained with due diligence to ensure the continued relevance of the data.
- New projects which may be considered atypical would not be represented by the historical data, necessitating adjustments to the estimation approach for those projects.

2.3.7 Software Estimation Tool Support

The manual effort required to develop more formal software estimates represents one of the most significant barriers to widespread adoption of their use. The development of software estimation tools that specifically reduce this effort has continued as an attempt to resolve this problem.

The degree to which the effort required to develop the estimate is reduced is dependent on type of functionality offered by the estimation tool. Efe (2006) categorised FSM estimation tools in three ways:

(1) Repository tools - data collection and calculation functionality
(2) Expert tools - interactive support of the development of the function count
(3) Automated tools – derivation of function count from appropriate information sources

Only the last category of these tools would represent a significant breakthrough in reducing the estimation effort required. The survey of seven tools, still actively supported, by Efe (2006) found that most of them fell under the first category of repository tools that were designed to support only the IFPUG standard. The feasibility of developing automated tools is largely dependent on the suitability of available information sources. Such automation tools can be categorised as either source code based or requirements documentation based.

2.3.7.1 Source Code Automated Estimation Tools

Tools for automating functional size estimates from application source code have been developed for use within industry. For example, an automated version of the FPA method has been proposed by the Object Management Group (2014). The Automated Function Points method is described as being generally consistent with the official IFPUG guidelines. The method has gained support within industry in the form of CAST software tools (Lavazza, 2015). In his evaluation of the method, however, Lavazza noted some fundamental differences between the two methods. For example, one of the core BFCs, the EQ component, is not included in the automated method. The size estimation data produced by the automated method cannot therefore be considered directly equivalent to the FPA method. The nature of the differences between the methods should be considered for size estimates developed for the same application using each method e.g. the FPA method early in the project lifecycle and the automated method upon completion of the coding. Limitations on its applicability also exist, such as only being suitable for transaction-oriented applications with data persistency. Despite some differences existing, this automated method was still considered by Lavazza to represent the first proposed method that is to a large extent compliant with the official IFPUG method guidelines.

Huijgens et al. (2016) conducted a survey on the use of automated functional sizing from source code within industry. A total of 336 FSM specialists from around the world, working in a variety of industry sectors, participated in the survey. Automated tools used within industry lack independent validation and are generally poorly documented. This survey therefore provided insight into the perceived value of this aspect of software size estimation. Participants were asked the extent to which they agreed with statements regarding these software tools using a scale of 1 to 5, although the number of responses to each question did vary. In terms of the specific FSM method that is automated, there was a slight preference from respondents for the COSMIC method (25%) over the standard IFPUG method (21%). While 54% of respondents agreed that automated tools that developed estimates from source code will be helpful, it was also agreed by 50% of respondents that this is difficult to achieve. Reasons given for the difficulty in automating the process included the complexity and variation in the code, as well the variety of technologies used in software projects. It was noted that it was respondents from more technical roles rather than business oriented roles that were more sceptical regarding the feasibility of automating functional sizing. This was also the case for certified versus non-certified FSM specialists. As a simpler alternative to automation, the use of backfiring, as described by Jones (1995), for conversion between lines of code and functional size was only seen as reliable by 23% of respondents. The purpose of automating functional sizing from source code was generally seen as a means of establishing an application size portfolio and historical database for benchmarking. The measurement of the final functional size of a system is necessary to fulfilling those objectives, but the technical difficulties in using source code for this process lead to the question of how the measurement should be made. As the measurement is focused on system functionality the complete specification of this functionality is more suitable than source code for this process. Automation of the functional sizing presents a challenge in either case, so the question becomes one of how size estimates can be more efficiently developed in such circumstances. Simplified sizing methods have generally focused on early lifecycle stage estimates developed from an incomplete statement of system requirements. Such methods are concerned with approximating the overall functional size rather than developed a detailed profile of the system size. In Chapter 5 the issue of simplifying the functional sizing process using detailed functional requirements documentation is investigated. The focus of this chapter is on how the accuracy of a detailed profile of the system size can be maintained, and thus it may provide an alternative to automatically deriving size estimates from source code.

2.3.7.2 Requirements Documentation Automated Estimation Tools

Tools focusing on model driven project documentation offer greater potential for automation in the estimation process. Tools developed for specific contexts have demonstrated satisfactory performance relative to that of applying the manual equivalent of the method. The prototype U-EST tool has been developed for automatically extracting Use Case Points and determining the associated complexity. The main limitation of this tool is that it only extracts information from Use Case Descriptions written in Japanese (Kusumoto et al., 2004). The μ_cROSE CASE tool automates the process of mapping Rational Rose RealTime model concepts to COSMIC Full Function Point (FFP) concepts to derive the functional size for a real time system (Diab et al., 2005). A tool was developed for automating the FPA method through extracting detail from WebML models for web oriented applications (Fraternali et al., 2006). Bagriyanik and Karahoca (2016) proposed automating the COSMIC FSM by adopting the use of a requirements engineering ontology. In this case the information gathered and formalised during the requirements engineering phase provides a suitable model for automatically deriving a COSMIC size estimate. Validation of the approach, on a small dataset of five measurements, showed that the automated estimates were within 2% of the manual estimates in each case. The requirements ontology incorporates a large variety of document types, however, and was developed within the context of a specific industry.

Automation of the FPA based on requirements text, rather than a model driven documentation, was investigated by Adem and Zarinah (2010). The requirements in this study were expressed as ‘goals’ and ‘scenarios’, from which the individual BFC can be extracted and counted. The approach requires the text to be in a specific format, e.g. subject, verb, direction and way. The estimation accuracy was within 2.35% of the manual estimate, but, the validation of the automation process consisted of estimating the size of a single application. Lent and Qu (2015) proposed automating the FPA method in the context of user interfaces expressed in a valid XML format. The approach was validated on a set of 11 projects, with the estimates of the automated estimated found to differ from the manual estimates by an average of 9.7%. The sizes of the sample applications were relatively small, however, ranging from 10 FP to 141 FP. The results were also found to be adversely affected by applications with a more complex architecture.

Simplified sizing approaches that aim to approximate the size estimates produced by the standard FSMs have also been the focus of research into automation. The premise behind the use of simplified sizing approaches is that they have lower input requirements and may therefore be more suitable for automation. Ochodek (2016) investigated approximating both COSMIC and FPA estimates based on use case names. In this approach each use-case name is assigned to one of 13 categories e.g. create, update, link etc. Each use-case is then allocated a size, measured in either COSMIC or FPA units. This size can be approximated either by using the average size of a use-case for its given category, or by using the Bayesian Network probabilistic graphical model. The approach was found to outperform random guessing i.e. it was predicting functional size, with the Bayesian Network approximation method providing the greatest estimation accuracy. The extent to which these results can be generalised is limited by the lack of standardisation in how use-cases are documented. Liu et al (2016) investigated the automation of a simplified version of FPA that identifies individual functions from use-case and class diagrams. The approach was validated on a dataset of 17 projects, although they were relatively small projects according to the manual size measurements. The automated size estimates were found to differ from the manual estimates by on average 9.95%. The degree to which estimation effort was reduced is unclear, however, as the documentation had to be modified to conform to the requirements of the tool.

The research into the automation of software sizing methods has involved relatively mature requirements documentation, characterised by a consistent structure and level of detail. Requirements documentation produced at the initial inception of a project is typically, however, lacking in such qualities that would lend itself towards automation of estimates. The sample project documentation provided by Equiniti-ICS from the bid stage was generally developed by the customer as part of the competitive bid process. There is consequently significant variety in the structure and detail of the description of required functionality across the projects. The suitability of using this documentation for investigating the automation of size estimation is therefore considered to be relatively low. The functional specification stage requirements documentation which was developed within Equiniti-ICS exhibits a more consistent structure and level of detail. The primary concern of this thesis is, however, on investigating how software sizing can add value to the estimation process. The importance of examining the use of software sizing at the bid stage precludes the issue of automation being addressed directly in this investigation. The discussion of the implications of the results of this investigation, included in Chapter 6, does however consider the issue of automation.

2.3.8 Relative Performance of Estimation Methods

In a review of research studies on expert estimation, the results were judged to have thus far been inconclusive in determining the superiority, in terms of accuracy, of either model-based or expert-based estimation (Jørgensen, 2004). The accuracy of either approach can be beneficial depending upon the nature of the project. For projects where specific domain knowledge beyond the scope of estimation models is prevalent, the use of expert estimation may be superior. In other circumstances the greater objectivity of model-based methods may be advantageous by avoiding situational factors which would adversely affect an expert estimator. The conclusions of the Jorgensen study supported the argument that it can be beneficial to utilise more than one estimation method, highlighting that different methods can consider different aspects of a project.

Research into the use of multiple methods has involved machine learning techniques to ascertain the relative effectiveness of the methods. There has not been any single estimation method established which provides the greatest level of accuracy, and it has been argued that it is fundamentally impossible to rank individual methods in terms of performance for all situations (Shepperd and Cartwright, 2001). The conditions used in the study e.g. whether a particular dataset is used for 'learning' or for 'performance', were found to affect the rankings of the different estimation methods. Kocaguneli et al. (2012) investigated using different combinations of effort estimation methods in order to determine if there were specific combinations that provided more reliable estimates. It was determined that it was necessary to separate individual estimation methods broadly into either 'superior' or 'inferior' methods, and then only select from the former set when establishing an appropriate combination of methods. This approach was evaluated using 1198 projects from 20 historical datasets, and it was reported that the best combination of 13 methods outperformed all of the individual estimation methods. The estimation methods investigated included regression-based and machine learning-based approaches, which are not widely utilised within industry. The barriers to implementing such an intensive approach, in terms of the range of methods and the historical data requirements, may be prohibitive in practical terms.

In attempting to improve the accuracy of estimation methods numerous factors have been considered that affect this accuracy. In a systematic mapping study of the research into these factors, a total of 32 factors were identified and classified into 4 categories (Basten and Sunyaev 2014). The majority of the factors investigated within research fell into the categories of the estimation process and the project characteristics. The third most common category researched, that of the estimator's characteristics, demonstrated another example of how the focus within research has not aligned with the most common approach to estimation used within industry. In evaluating the strength of evidence provided by the reviewed research studies, the authors of this study rated 69% of the evidence as being of low or very low quality. The use of small sample sizes and reliance on student participants in experiments were reported as significant reasons for the low quality of evidence. More in depth case studies were proposed as a means of examining multiple factors at once and assessing the causality of such factors. The need for collaboration between industry and research was highlighted as a mutually beneficial arrangement. Software development organisations accumulate extensive project data, but performing intensive analysis on such data would be a luxury that few can afford. This research thesis utilises a rich source of commercial project data, subjected to a detailed analysis of software size that had not previously been considered by Equiniti-ICS. The reviews of this analysis during each phase of the research provide the depth necessary to incorporate an industry perspective on the value of software sizing.

The research on expert judgement-based methods and model-based methods could progress by focusing on complementing rather than on competing with each other.

2.4 Current Research Issues

Research literature was identified in Section 2.2.3 as being relatively narrow in scope, focusing mainly on estimation methods and their performance, indicating that there is a need to focus on what can be achieved in practice. In particular, the impact of wider organisational factors and the commercial characteristics of the development process must be considered. The nature of how the stage of the lifecycle affects the feasibility of developing estimates for a project is firstly discussed. The extent to which evolving approaches to how software is developed impact the suitability of different estimation methods is then considered. A review of the perception within industry of the value of more formal estimation methods is conducted, providing an indication of where this research may investigate further opportunities for such methods to contribute to software project development. The specific issue of the value of software sizing is lastly examined, thus providing the context to the proposed research in this thesis.

2.4.1 Lifecycle Factors

The software estimation paradox highlights the effect on estimation confidence of the stage in the lifecycle when an estimate is developed. This can be explained by a phenomenon observed by Boehm (1981), in which the degree of uncertainty regarding software estimates is predictable by the stage of the project life cycle. This was later referred to as the ‘Cone of Uncertainty’

(McConnell, 1996), and is illustrated in Figure 2.6. Verner and Evanco (2005 p.91) observed that, "*most software estimates are performed at the beginning of the lifecycle before the requirements phase and thus before the problem is understood.*" The amount of uncertainty associated with an estimate is therefore greatest during the initial stages of a project, as indicated by the dark line in the graph. The narrowing of this line illustrates how the amount of uncertainty decreases exponentially as the project progresses. This reduction in uncertainty is characteristic of action taken to increase understanding of the project as more detailed information is generated. The shaded area represents the more limited reduction in the amount of uncertainty, which would be observed if action is not taken. The precise nature of this reduction in uncertainty, as illustrated in Figure 2.6, has been challenged by contrasting explanations. It has been argued that the estimation accuracy for the project as a whole improves with subsequent estimates, but the amount of uncertainty regarding the remaining tasks is not reduced even by the later stages of the project (Little, 2006). The estimation results reported in industrial surveys indicate that the inaccuracy is no longer symmetrical as underestimation is more prevalent (Laird, 2006). It should be noted that reports of estimation performance by surveys can themselves be subject to uncertainty over their reliability (Jørgensen and Moløkken-Østvold, 2006). Improvements in estimation practices have also reduced the scale of the likely potential error at each stage e.g. instead of estimates demonstrating a potential inaccuracy of a factor of 4 (400%) at the beginning of a project, a factor of 1 (100%) would be more representative of the level of estimation uncertainty.

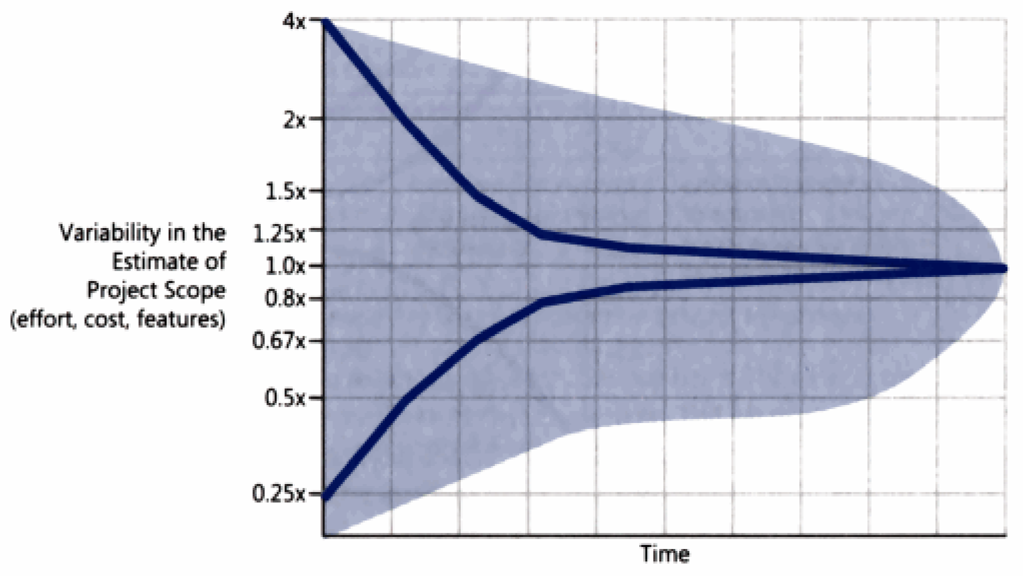

Figure 2.6 – Cone of Uncertainty (McConnell, 2006)

Recognition of this can be seen within the estimation practices of Software Development companies. In the survey of estimation activities in the Chinese software industry (Yang et al., 2008) the organisations surveyed were asked when estimates were made. The most common stage reported for conducting estimates was the requirements development stage (75% of projects). This exceeded the 57% of organisations that reported developing estimates at the initial project proposal stage, suggesting recognition of the greater accuracy possible at the later stage. Estimation became less popular further into the lifecycle, although it was still conducted in 36% of the projects by the design stage**.**

The implications of the Cone of Uncertainty stem from the fact that while the quality of project planning and estimating affect the extent to which the cone can be narrowed, there is a limit to how much uncertainty can be removed at any stage. Estimation uncertainty should be managed through recognition of the level associated with the current lifecycle stage. The degree of rigour in any adopted estimation approach should likewise consider the level of accuracy that is possible at that stage.

2.4.1.1 Managing Estimation Uncertainty

Software estimation is concerned with establishing the anticipated cost and effort required for a project, and consequently enabling effective planning and control of the development of the project. Formulating the budget and schedule for a project requires an understanding of the uncertainty associated with any estimate of effort and cost. This level of uncertainty can be expressed through the provision of a confidence level and range for each estimate. Indeed, expressing uncertainty in an estimate is recognised as an ethical responsibility of software professionals by both the IEEE Computer Society and the Association of Computing Machinery (McConnell, 2006).

Research suggests, however, that this approach is insufficiently understood in practice which undermines its reliability. In a study where the participants were required to express a minimum and maximum effort estimate that reflected a 90% confidence range, the actual value was only found to fall within this range around 60% of the time (Jørgensen et al., 2004). Providing a greater degree of confidence in an estimate should necessitate a wider range between the minimum and maximum values for the estimate. Jørgensen (2014) divided 62 software developers into two groups, which were instructed to provide an effort estimate for a relatively simple project with confidence levels of 98% and 80% respectively. The results of this estimation exercise were that there was not a statistically significant difference in the minimum to maximum range between the two groups.

The presentation of estimation uncertainty is affected by how it would be considered acceptable by management. The requirement for a level of certainty in the estimate would result in an estimate range which may be considered too wide, or a confidence level which is too low, to be of practical use for decision making. The issue of estimation uncertainty may be addressed to some degree by the provision of objective data to support the estimated value. The process of performing sizing estimation can contribute to this by providing a methodical measurement of the required functionality for a project. The process of sizing can also offset the risk associated with a project to some degree by identifying where insufficient detail has been provided in the project documentation.

Uncertainty about the required system impacts how this information can facilitate development of an accurate estimate. Consequently, the absolute application of any specific approach to software estimation may not be feasible in many circumstances. In practice, the estimator must rely on their intuition as to how to interpret how the available project details map to those required by the estimation approach. Fuzzy Logic provides a means of emulating the cognitive processes of estimators and enables imprecision to be handled by the estimation approach.

Various approaches to software estimation, including that of functional size estimation, have been enhanced by the use of Fuzzy Logic. The original FPA method was extended by Lima et al. (2003) to incorporate the use of aspects of Fuzzy Logic, resulting in the Fuzzy Function Point Analysis (FFPA) method. This approach addresses the imprecision inherent in using rigid boundaries associated with the classification of complexity for Data Functions and Transaction Functions e.g. functions which differ by only one DET can be classified with different complexities. Fuzzy Logic is used to establish a ‘fuzzy’ complexity assessment by analysing the composition of the complete set of functions and determining appropriate fuzzy number values of RET and DET for each classification of complexity. Functions which fall close to the boundary values of RET and/or DET are then assigned a FP value according to a continuous graduation rather than a rigid value according to which side of the boundary they reside. Braz and Vergilio (2004) developed the Fuzzy Use Case Size Points metric, using the same general approach of the FFPA method. These uses of Fuzzy Logic are, however, focused on dynamically refining the function complexity definitions of the methods rather than on the application of those methods. The benefits of using Fuzzy Logic within the area of functional size estimation are thus limited so far.

2.4.2 Impact of Software Development Methodology

The discussion in the preceding section of how the project lifecycle affects software estimation assumed that the project development followed the traditional Waterfall methodology. The increasing use of alternative methodologies presents different challenges for software estimation. The nature of the practices adopted for requirements engineering may influence the estimation approach used by an organisation. Kassab et al (2014) surveyed 247 software development professionals from a wide range of industries, comparing the results with those reported from previous similar surveys by Neill and Laplante (2003) and Marinelli (2008). The results of this survey indicated that Agile-based methodologies now formed the leading category of development lifecycle, utilised by 46% of the respondents. This percentage had almost doubled from that reported in 2008, replacing the Waterfall methodology as the dominant development lifecycle reported in previous surveys. The use of requirements modelling methodologies grew from 45% in 2008 to 54% in 2013 despite the rise of Agile-based methodologies that place less emphasis on documenting requirements. Of the respondents which indicated the use of a requirements modelling methodology, the most commonly used approach in 2013 was Object-Oriented analysis, reported by 70% of those respondents and 38% of all respondents. Those respondents who employed the use of requirements modelling approaches in 2013 also reported higher levels of satisfaction than those who did not use such approaches. In terms of feeling that the systems met the needs of the customer the difference was 87% compared with 69%, and in terms of feeling that they were easy to use the difference was 82% compared with 48.5%. The importance of obtaining a detailed understanding of the system requirements is therefore still evident within industry. In terms of documenting requirements, the number of survey respondents who indicated the use of natural language had actually increased from 53% in 2008 to 61% in 2013. The use of semi-formal modelling notations such as UML by 33% respondents was similar to the reported usage of Object Oriented analysis. The evolving nature of requirements engineering practices within industry has not been shown to result in fundamental changes to how requirements are documented.

Projects developed using more iterative approaches have necessitated the development of alternative estimation methods. Section 2.3.5 provided an overview of estimation methods used within an Agile-based development approach. Nguyen-Cong and Tran-Cao (2013) reviewed 32 estimation studies within a wider context of non-Waterfall methodologies. This review included estimation studies featuring Agile, Iterative and Incremental software development. These studies were conducted mostly over the previous decade, reflecting the increased use of these development approaches during this period. The Nguyen-Cong and Tran-Cao literature review found that the estimation models in the studies relied on previous ‘iterations’ of the current project to improve the accuracy of subsequent iterations, rather than using historical project data. The early availability of historical data from the current project represents a significant advantage over traditional life cycle approaches and enabled more informed subsequent estimates to be made.

The most common sizing metric reported by the literature review was Story Points, featuring in 34.38% of the estimation studies. This sizing metric is associated with the expert-based approaches to estimation such as 'planning poker'. Traditional measures such as LOC and functional sizing metrics were nonetheless still reported within these studies as model-based estimation approaches are still utilised within Agile, Iterative and Incremental software development. The range of estimation approaches reported in the review may reflect the degree to which changing development methodologies are still grounded in more traditional aspects of software development. Consequently, the more traditional approaches to software estimation can still be seen to be relevant in more recently adopted development methodologies.

A survey by Huijgens et al. (2016) included the issue of how FSM specialists viewed the impact of Agile-based methodologies on the use of functional sizing. The rise in the use of Agile-based development practices was only seen as a hindrance to functional size measurement by 22% of 245 respondents. Reasons given for the use of functional sizing included the need for an overall estimate to formulate project budgets and that it encourages more rigorous requirements management. Reasons for resistance towards the use of functional sizing included insufficient requirements documentation and the evolving nature of the project requirements.

The impact of more iterative development approaches on the suitability of functional sizing method may therefore vary according to individual circumstances. The requirements documentation provided by Equiniti-ICS was developed according to the traditional Waterfall methodology. The adoption of an Agile-based approach within the organisation to the subsequent development of the systems does therefore not directly impact the investigation into the specific use of functional size estimation in this thesis. The potential for investigating the use of functional sizing in the later Agile-based development stages is limited, as the completed requirements documentation available is not reflective of the more challenging circumstances associated with a fully Agile-based methodology. Consequently, the focus of this research does not extend beyond the three research phases stated in Chapter 1.

2.4.3 Value of Formal Software Estimation Methods

The importance of establishing estimates for software projects has been recognised as a fundamental aspect of the development process. The CMMI (Chrissis et al., 2003) addresses this within 'project planning' by identifying two specific estimation practices. Under Specific Practice 1.2, it is necessary to 'Establish Estimates of Work Product and Task Attributes'. This practice involves the use of an appropriate unit of measure, such as SLOC, classes, requirements, functions etc. The project should be measured by determining the number of instances of the appropriate units that are present. Under Specific Practice 1.4, it is necessary to 'Estimate Effort and Cost'. This practice establishes the estimated effort and costs through the outputs from the previous estimation practice. The results of these estimation practices subsequently inform the development of project schedules, and therefore provide the basis upon to control software projects.

Project scheduling issues have been identified as one of the early warning signs that projects are in danger of failing (Kappleman et al., 2006). In a survey of project failures, the most critical challenge reported in the survey was that of estimating the schedule and implementing effective project management to meet the targets of the schedule (El Emam and Koru, 2008). For projects, which were not completed, the third most stated reason by respondents for the cancellation of a project was exceeding the budget. The challenge is therefore not only developing appropriate estimates, but of managing the development of the project within the constraints of the available resources. The value derived from developing software estimates is contingent upon the purpose of the particular estimate. The purpose of an estimate, in turn, can vary according to the lifecycle stage of the project and who it is to be communicated to. Estimates developed in later stages of the lifecycle enable progress to be assessed, changes in scope to be identified, project plans to be adjusted etc. In their industrial survey, Moløkken-Østvold et al. (2004) investigated the frequency of developing estimates during the completion of a project. It was reported that the number of estimates made in projects ranged from one to six throughout the project lifecycle. This would suggest that the perceived value of estimation varies across development organisations, ranging from only providing an initial estimate for a project to assisting with its management as it progresses through the lifecycle.

The evidence from Section 2.2.1 indicates that value has not arisen from the accuracy of formalised estimation methods relative to expert-based estimation approaches. The case for adopting more formalised approaches to estimation must therefore be made on business value terms more than solely on how accurate the estimates are. The information provided by the estimation method and the practicality of developing the estimates are both integral to determining how such business value can be obtained. The information provided by the estimation method should be assessed through comparison with the output provided by existing estimation practices. The practicality of formalised estimation methods should be assessed in terms of how readily they can be incorporated into existing development practices. The focus within research on evaluating estimation methods using historical datasets, as identified in Section 2.3.6, does not assess the practicality of utilising such methods on active development projects. The use of formalised estimation methods should therefore involve engagement with development and management staff in assessing the suitability of such methods in practice. Finding a role for formalised software estimation in industry requires adapting these methods to address the needs of the development organisations.

The commitment made to the estimation process is also dependent upon the characteristics of the development organisation. A study of small to medium organisations in Malaysia and Vietnam investigated the value of specific activities within the development process (Niazi and Babar, 2009). The results indicated that the project planning activities, such as estimation, were not considered to be of high value by the majority of organisations. The companies in this study originated from developing nations, which could limit the availability of resources that may be allocated to the planning stage of a project. The assessment of the CMMI Process Areas for small to medium Northern Ireland software development organisations highlighted the difficulties facing smaller companies (Wilkie et al., 2005). This study reported that the process of estimating effort and cost was relatively well established, reflecting the importance that was associated with this practice. The estimation of the work product and task attributes was, however, often not clearly recognised or established. Indeed, of the 14 specific practices making up the project planning process area, this was the lowest scoring practice. The implication is that the relative maturity of the development process influences the recognition of the underlying basis for effort and cost estimates. This assertion is supported by a subsequent assessment of the Northern Ireland survey which investigated the level of interest in software development activities (Chen and Staples, 2007). Greater interest is apparent in 'high level' activities that need to be communicated to stakeholders, with less attention directed towards the more detailed activities that underpin them. In contrast, studies of development organisations that have reached a higher level of process maturity have been able to demonstrate a beneficial impact on estimation. Organisations that have attained Levels 4 and 5 of CMMI can be observed to have achieved quantifiable improvements in both cost and schedule estimation (Gibson et al., 2006).

2.4.4 Value of Software Sizing

This research thesis is concerned with how software sizing in particular can provide value to the development process. The perception of software sizing and how it is used within industry is examined in order to establish the context of this research. The comparative value of the main approaches to software sizing is then outlined. The usage of industry software sizing data in research is discussed in order to understand the perception of the importance of the component parts of such data. The main factors affecting the value of software sizing are discussed in terms of existing research and the approach adopted of this thesis. The importance of bid stage estimation is highlighted, with an overview of how the use of software sizing is investigated as a means of providing greater insight at the initial stage of a project. The main approaches to software sizing are then introduced to highlight their limitations in terms of the value they provide.

2.4.4.1 Software Sizing in Industry

As indicated in Section 2.2, there is limited practice in the software development industry of obtaining an underlying measure of the required system components when developing a project estimate. Within model-based approaches to effort and cost estimation it is generally accepted that software size is a primary input. Examination of the explicit use of software sizing within industry provides an indication of the value with which it is held. In a survey of the Northern Ireland software industry (McCaffrey et al., 2004) it was reported that, out of 15 areas of software engineering and engineering management addressed, the topic of estimation was the single most problematic area. The survey involved 56 organisations, mostly in the category of small-to-medium-sized enterprises (SMEs), as defined by European Commission (2003). In a further investigation of Northern Ireland software companies Wilkie et al. (2005), reported that explicit software size estimation is very much a minority activity, despite the recognition of estimation as a significant problem. In a survey of the Brazilian software industry by MCT Ministério da Ciência e Tecnologia (2001, cited by Cândido and Sanches, 2004) it was reported that only 29% of software companies develop software size estimates. The development of effort estimates requires some level of understanding of the “size” of the product, even if it is not explicitly estimated. The absence of any explicit sizing activity or documented output may reflect a lack of recognition of the significance of software size as a key factor in the estimation process.

The perception of the value of software sizing may also be viewed in the level of maturity of the development practices of organisations. The Wilkie et al. (2005) study involved six commercial organisations, which had no prior involvement with CMMI at the time of the research. An investigation of software estimation trends in one organisation targeting CMMI level 3 (Nasir and Ahmad, 2006), found that software size estimation was relatively informal. Heuristic approaches to sizing, such as expert judgement and wide band Delphi, were generally favoured over parametric models. The potential value of software sizing was demonstrated in a study of projects completed by CMMI level 5 companies (Agrawal and Chari, 2007). This study reported that at this high level of process maturity, the size of the product being developed was the only significant factor that affected the effort and development time required for projects.

Wilkie et al. (2005) evaluated six SMEs using the CMMI and found that it would be impossible to develop effort estimates without some level of understanding of the “size” of the product. However, they discovered that this initial step seems often to be almost subconsciously undertaken, without any explicit activity or documented output.

Czarnacka-Chrobot (2010) reported on a survey of the Polish Business Software Systems industry concerning the use of software estimation methods. This industry sector can be considered to be broadly similar to that of the Northern Ireland software companies included in the studies cited in this chapter. The size of the development organisations included would be equivalent to SMEs, and the software developed was typically for financial institutions, public administration institutions etc. The author of the study was concerned with the use of FSM methods in particular as the nature of the software projects were the primary intended type for those methods. The survey was carried out twice in order to measure changes in industry practice, with firstly 44 and secondly 53 responses received. The results from this survey, which are most relevant to this research, are shown in Table 12.

Table 12 – Results from survey on the use of software estimation methods in Poland (adapted from Czarnacka-Chrobot, 2010)

Survey Topic	Percentage of Respondents (2005/6)	Percentage of Respondents (2008/9)
Use of Methodology Based Estimation	45%	53%
Use of Expert Judgement	36%	43%
Use of FSM methods	20%	26%
Awareness of FSM methods	27%	34%

It was reported that only around half of the respondents indicated the use of any specific approach to estimation, with the other respondents stating that price-to-win was the preferred approach. For those respondents that used a methodology based approach, the most commonly reported method was that of expert judgement. There was a modest increase in the use of Expert Judgement from 36% to 43%, with the use of FSM methods showing a similar increase from 20% to 26%. The increases were reflective of a general modest increase in using a specific methodology for estimation. The use of FSM methods was only the fourth most common out of the five approaches surveyed, although this represented around 75% of those respondents who indicted awareness of such methods. The relatively low awareness of FSM methods provides an indication that the low uptake of such methods in industry is a reflection of how academic research in this area has been progressing independently of industrial practice. Sheetz et al. (2009) investigated differences in how SLOC and FP size measures are perceived in industry. A survey was conducted involving 28 developers and 42 managers. In terms of the sizing measure each respondent was most familiar with, 24 indicated SLOC and 46 indicated FP. No statistically significant differences were found on how each group perceived the SLOC size measure. For the FP measure, developers had a significantly stronger belief in their applicability throughout the project lifecycle as well as their greater predictive qualities. In contrast managers felt that the FP measure was more sensitive to changes in the software. Only managers believed that SLOC was more readily available than FP, although both groups agreed that SLOC was the easier measure to obtain and understand. The results of this survey indicate that developers have a greater understanding of the benefits of the FP measure. The perception gap between the two groups suggests that a focus within research on how functional sizing can contribute towards project management would be more productive than demonstrating its technical superiority.

In the Norwegian industry survey (Moløkken-Østvold et al., 2004) estimation was commonly conducted at more than one stage of the project lifecycle, with sometimes as many as six estimates conducted on a project. Khan and Beg (2013) conducted a survey on the use of size estimation with 59 project managers and practitioners within the development industry. The survey respondents were asked to indicate how often size estimation was used during the lifecycle of projects according to five options. The results for this question are shown in Table 13, indicating the percentage of respondents who selected each option.

Table 13 – Frequency of using Size Estimation during project lifecycle (Khan and Beg, 2013 p.29)

Selected Option	Percentage of Responses
Only once (at the time of proposal/bidding for the project)	53%
Twice (First at the time of proposal/ bidding and then at the time of closure of the project)	34%
At least Thrice (At inception, closure and whenever there is a scope change request)	8%
Never Required (Only Cost-Estimates and Schedule-Estimates are worked-out)	2%
Depends on customer's requirement for detailed size-estimates to be submitted	3%

The options provided in the survey indicate both the frequency and the stage of the project lifecycle at which size estimation was used. The most common responses indicate a clear pattern of developing size estimates at the beginning of a project, with only 2% of respondents indicating that projects are never explicitly sized. The focus of this survey on specifically the use of size estimation does not represent the wider use of software estimation, where in general the use of explicit sizing approaches is substantially less prevalent. The importance of estimation at the beginning of a project is universally acknowledged despite the difficulty arising from the limited detail about a project at that stage. Size estimation was indicated to be undertaken at the completion of a project by a moderate 34% of respondents, but rarely at any intermediate stage in the lifecycle. The implication that can be drawn from this survey question is that the value of software sizing is not recognised in between the inception and completion of a project. There is some recognition of the value of judging the size of the completed project against the initial size estimate, which can facilitate assessment of the sources of inaccuracy at the beginning of a project. The issue of how intermediary size estimation could enable inaccuracies in the initial estimate to be identified as a project progresses, and enable adjustments to be made to the planned schedule, is not acknowledged in practice. The research also conducted a more detailed analysis of eleven projects, incorporating interviews with staff involved in the development of the projects. Each of the eleven projects was reported to have been underestimated at the outset. For nine of these projects, the size estimates were developed using a top-down process which was neither documented nor validated through the maintenance of historical estimated and actual size values. For eight of these projects, size estimates were only developed at the initial proposal stage. The application of software sizing within these eleven projects can therefore be concluded to be relatively informal and insufficient to enable an assessment to be made on the potential value of size estimation. Guidelines were proposed by the authors to indicate the utility of size estimation at each stage of the project lifecycle. The discussion addressed how size estimates should be used, but it did not address the nature of the size data necessary to provide such utility. The value of software sizing is dependent upon both the utility of the sizing data and how efficiently such data can be obtained.

2.4.4.2 Comparison of Value of Approaches to Software Sizing

The overall current status of FSM and UML approaches to estimation are summarised in Table 14. The Object Oriented approaches to estimation have not provided any standardised methods at present. The absence of standardisation limits the establishment of industry wide estimation data derived from such approaches. The development of local historical data may also require careful consideration of how the estimation method was applied for each project. The suitability and difficulty of using publicly available datasets may limit any advantage offered by FSM in this regard.

Table 14 – Overall Comparison of FSM and UML estimation

	Functional Sizing Methods	**Object Oriented Based Estimation**
Estimation Method Maturity	ISO Standardised Methods, Publicly available datasets	No standardisation of the estimation approach
Documentation Requirements	User Functionality Requirements not restricted to specific format	Estimation methods based on specific UML documentation types
Tool Support	Limited by nature of documentation	Suitable for more formalised documentation types

The use of an Object Oriented estimation method can provide a closer fit with the format of the project documentation than alternative approaches. This reduces the degree of subjectivity on the part of the estimator in identifying the specific components required for an estimation method. For Object Oriented Functional Sizing methods, the estimation output data is more readily comparable with conventional functional sizing methods. This would offset, to some degree, the limitation arising from utilising industry wide estimation data by potentially benefitting from the substantial datasets established for functional sizing methods. The use of Use Case concepts as the basis of the sizing method is, in particular, subject to variance in how the Use Cases are specified. The value derived from UML sizing methods may be enhanced by converting the output into functional sizing measures. Cuadrado-Gallego et al. (2014) investigated the relationship between the UCP and IFPUG method in order to assess the feasibility of such conversions. Sizing using both sizing methods was completed by University students selected based on the expertise demonstrated after receiving training in the methods. This sizing exercise provided results from each method for 17 projects, and was assessed along with 15 projects in a previously published dataset (Minkiewicz, 2004). The study concluded that one IFPUG FP was approximately equal to one UCP, where the error range associated with such a conversion was 25%. These results provide a positive indication of the feasibility of converting UCP estimates into FP, although improving the confidence of such conversion was identified as a future research issue. The conversion of the overall size measure does not address the richness of the output detail providing by a sizing method. The value of the output sizing data should be assessed by considering how different aspects of a system contribute to the overall size.

The commercial nature of software development affects the availability of the necessary detail about a project. For projects subject to commercial tender processes in particular, the earliest stages in the lifecycle are limited in terms of the project details provided for informing the submission of bids. Bid documentation is typically produced by the prospective customer and will typically have been developed to a level of detail that will limit the degree to which Object Oriented estimation methods can be used. The feasibility of using such methods may therefore be limited to later stages of the project lifecycle. The value, which may be derived from Object Oriented estimation, would therefore be expected to increase as more detailed specifications of the project functionality are produced.

The availability of tool support for software estimation is limited by the difficulty in deriving automated estimates from natural language documentation. In particular, the identification of specific and unique functional sizing concepts from text based requirements specifications is subject to the same issues faced by human estimators. For example, any ambiguity and inconsistencies inherent in project documentation require subjective judgements to be made by the estimator. Object oriented approaches to estimation that utilise UML models can be considered to be a relatively simpler process to automate as the identification of the necessary components is more straightforward. The overall value of using either Functional Sizing or Object Oriented approaches is dependent upon the specific nature of the software development practices adopted by an organisation. In Chapter 3 the suitability of both of these approaches for use in this investigation of the value of software sizing is evaluated.

2.4.4.3 Use of Industry Software Sizing Data in Research

Fernandez-Diego and Gonzalez-Ladron-de-Guevara (2014) investigated the use of the ISBSG dataset in 107 published software engineering research publications from 2000 to 2012. The most common use of this dataset was in effort estimation studies, accounting for 70.5% of published papers. From the period of 2005 to 2011 there was a marked increase in the use of this dataset within research, with the topic of dataset properties emerging as the second most common research topic. In 55% of the research papers the only support used for the studies was the ISBSG dataset. The effect of the characteristics of the dataset on the accuracy of any estimation method would be understood more clearly with the use of multiple datasets. The public datasets, most commonly used, appearing in more than 10 research papers, to complement the use of ISBSG were Desharnais (Desharnais, 1988), Cocomo81 (Boehm, 1981), Bank63 (Maxwell, 2002), Kem87 (Kemerer, 1987) and Albrecht (Albrecht and Gaffney, 1983). Usage of the ISBSG dataset was further analysed to determine the specific data fields that were most commonly utilised within software effort estimation research (González-Ladrón-de-Guevara et al., 2016). Size variables accounted for 30.6% of all variables for this research. The dominant FSM was the IFPUG method, used in 76.2% of studies, with COSMIC a distant second at 7.1%. The unadjusted FP values were the preferred choice over the adjusted values, with 61.7% of studies using the former compared to 28% for the latter. This is in line with official guidelines for IFPUG that recommend using the unadjusted FP value. The use of individual BFC types was evident in around 15% of research papers, with their usage becoming more widespread after being overlooked prior to 2005. The percentage of missing values in the dataset was, however, relatively high at 67%. This suggests that recognition of their importance was limited within those industry practitioners that submitted estimation data. Focusing research on the richness of size data that can be developed using functional sizing may encourage the collection and submission of more complete estimation data from industry.

Quesada-Lópeza and Jenkins (2016) investigated the relationships between the FPA BFCs in order to determine their suitability for inclusion in an estimation model. The study used functional size data on 202 business applications from the ISBSG dataset. Statistically significant correlations were found between various BFC, most notably for EI and EQ, and also between various BFCs and the overall UFP. These results supported previous research on the correlation between BFC that suggest redundancy in measuring the functional size of a system. It also indicated that the overall functional size may be derived from a subset of the BFCs. The relationships between the BFC sizes and development effort were assessed, in comparison to using the overall unadjusted functional size. In comparison to the UFP, some of the BFCs were found to have similar correlations with development effort, as did some subsets of the BFCs. For example, using either the Transaction Functions or a combination of EI and ILF were similarly effective for predicting development effort as the UFP. These results illustrate that it may not necessary to perform the full functional sizing method steps if the goal is to obtain the overall functional size or predict the development effort.

2.4.4.4 Software Sizing Value Factors

Research into the use of software sizing has considered multiple factors to determine whether they influence how effective size estimation can be. The effect of the level of detail considered by the sizing method has been investigated (Cândido and Sanches, 2004; Zivkovic et al., 2005b; van Heeringen et al., 2009; Popović and Bojić, 2012), as has the level of requirements detail available (Demirors and Gencel, 2004; van den Berg, 2005; Zivkovic et al., 2005a; Frohnoff and Engeroff, 2008; Gencel et al., 2009). The effect of the project lifecycle stage was also investigated (Zivkovic., et al 2005b; Gencel et al., 2009; Agresti et al., 2010; Popović and Bojić 2012). Even in studies that examined more than one of these factors they were generally investigated independently of each other, and estimation accuracy was the primary concern when evaluating the estimation results. Figure 2.7 illustrates how the main factors investigated are interrelated, suggesting that a broader perspective should be adopted into how software size estimation provides value to the development process.

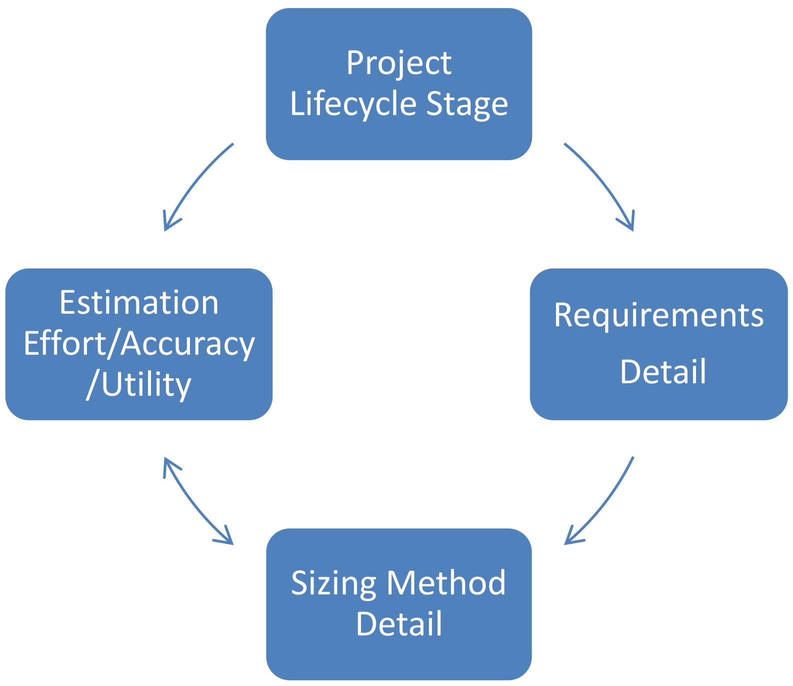

Figure 2.7 – Factors affecting value of software sizing

The project lifecycle stage affects how mature an understanding of the system requirements has been developed, and hence the level of requirements detail provided by documentation. The level of requirements detail affects the feasibility of using particular sizing methods, consequently limiting the choice of method used. The level of detail considered by the chosen sizing method affects the effort required to develop the size estimate, as well as the accuracy of the estimate and its level of utility. The acceptability of the estimation effort varies according to the project lifecycle stage, generally decreasing as the project progresses. The estimation accuracy generally increases as the project progresses and together with the level of utility determines the benefits of using software sizing. Constraints on the acceptable estimation effort, and the level of accuracy and utility desired, in turn affect the choice of sizing method detail. In the research studies listed above, only the studies by Demirors and Gencel (2004) and van Heeringen et al. (2009) explicitly reported on the estimation effort required despite it being a central factor in determining the value of software sizing. Considering which level of sizing method detail is most appropriate at different lifecycle stages has not been investigated as part of these studies. In order to examine how these factors affect the value of software sizing, the trade-off between the benefits and costs should be investigated at different lifecycle stages. This approach is adopted in the investigation into the value of software sizing presented in Chapter 3, which includes an assessment of the research studies listed in this section.

2.4.4.5 Bid Stage Software Sizing

The discussion of lifecycle factors in Section 2.4.1 highlighted how estimation uncertainty is greatest at the initial inception of a project. The understanding of the required functionality of a project is at its most limited at this stage and developing realistic estimates in support of submitting a bid is a significant challenge. Most software estimation research that addresses the development of bid stage estimates has been from the perspective of expert judgement-based approaches. The experience of the expert can compensate for the lack of requirements clarity to some degree, but lack of transparency in such intuition limits the degree of objectivity that can be associated with any such estimate. The use of experts to provide a range for bid estimates with an acceptable level of certainty has been demonstrated to be relatively unsuccessful (Faria and Miranda, 2012). The competitive bidding process itself has been reported to adversely affect the estimation process by encouraging unrealistic bids in order to win (Jørgensen and Grimstad, 2005).

The development of a size estimate using a detailed approach is generally not feasible at the bid stage due to the limited description of required functionality that is available. The estimation accuracy of size estimates has been demonstrated to improve as more detailed requirements documentation becomes available later in the lifecycle (Demirors and Gencel, 2004; Agresti et al., 2010). The question of what can be learnt from these more accurate size estimates, in order to improve bid stage estimates, has not been explored by such research. Development of a detailed profile of functional size estimates at both the initial bid stage and once the functional requirements have been fully established would facilitate an investigation into their differences. The preceding section outlined the proposed investigation into the use of different levels of sizing method at these two distinct stages of the project lifecycle. Chapter 4 builds upon this by analysing the functional size estimates at a greater level of detail than the standard sizing method components. Understanding the nature of the functionality that is and is not specified at the bid stage may provide insight into how bid stage documentation can be validated by size estimates. This would serve to provide greater transparency and hence objectivity to bid stage estimates through the use of software sizing in support of expert judgement-based approaches.

2.4.4.6 Simplified Software Sizing

The development of simplified sizing methods has been driven by two characteristics of the standard approaches to size estimation. Firstly, the level of requirements detail required to develop full size estimates is generally not available at the initial stages of the project lifecycle. Secondly, the estimation effort required to develop size estimates presents a barrier to its adoption within real world project development. Simplified sizing methods have sought to address these two issues by streamlining the process to provide an approximation of the full size estimate. The core sizing activities associated with the conventional FPA method are shown in Figure 2.8. The sizing activities that a simplified method scales back on commonly include the intermediary identification and measurement of the functional components. For example, simplified approaches to the conventional FPA method can partly reduce the effort required by eliminating the measurement, in terms of complexity, of the BFCs. An example of this approach is found within the NESMA ISO standard (International Organization for Standardization, 2005). In the Estimated NESMA method, each instance of a BFC is assigned an assumed complexity. Simplified approaches that seek to further reduce the estimation effort can achieve this by scaling back the identification of BFCs to include only a subset of these components. An example of this approach is the Indicative NESMA method, wherein only the ILF and EIF components are identified and counted. The total numbers of each component are subsequently multiplied by 35 and 15 respectively to produce the overall functional size. Numerous simplified methods based on the FPA BFCs have been developed, but only the NESMA simplified methods have been included within an ISO standard. These simplified sizing methods can either use externally established weightings to derive the overall of functional size or use local calibration to develop appropriate weightings.

The COSMIC method provides guidelines for simplified sizing methods (COSMIC, 2015), although these do not constitute part of its ISO standard. These methods generally require the average size and/or the number of functional processes to be estimated. The overall functional size is then typically derived either by using the average size as a scaling factor or by using a size classification scheme to assign each size. This process of deriving overall functional size typically requires calibration using a local dataset. The simplified sizing methods described in these guidelines have generally not been subjected to significant validation. The guidelines can therefore be considered to be primarily of an informative nature, outlining ongoing research, rather than an endorsement of their validity. Simplified sizing methods that facilitate differing levels of sizing detail to be considered have also been developed, such as the Early and Quick Function Point Method 2.0 (Santillo et al., 2005).

Chapter 3 of this thesis concerns the use of differing levels of rigour for software sizing in order to assess their relative value for commercial project development. The merits of using both the NESMA and COSMIC based approaches to simplified sizing for this investigation are consequently discussed in more detail in Chapter 3. The objective of these simplified sizing methods has been on approximating the overall functional size of more detailed estimates. The question of whether viewing functional sizing in such narrow terms of what it may contribute to the estimation process has not been addressed within research into simplified functional sizing.

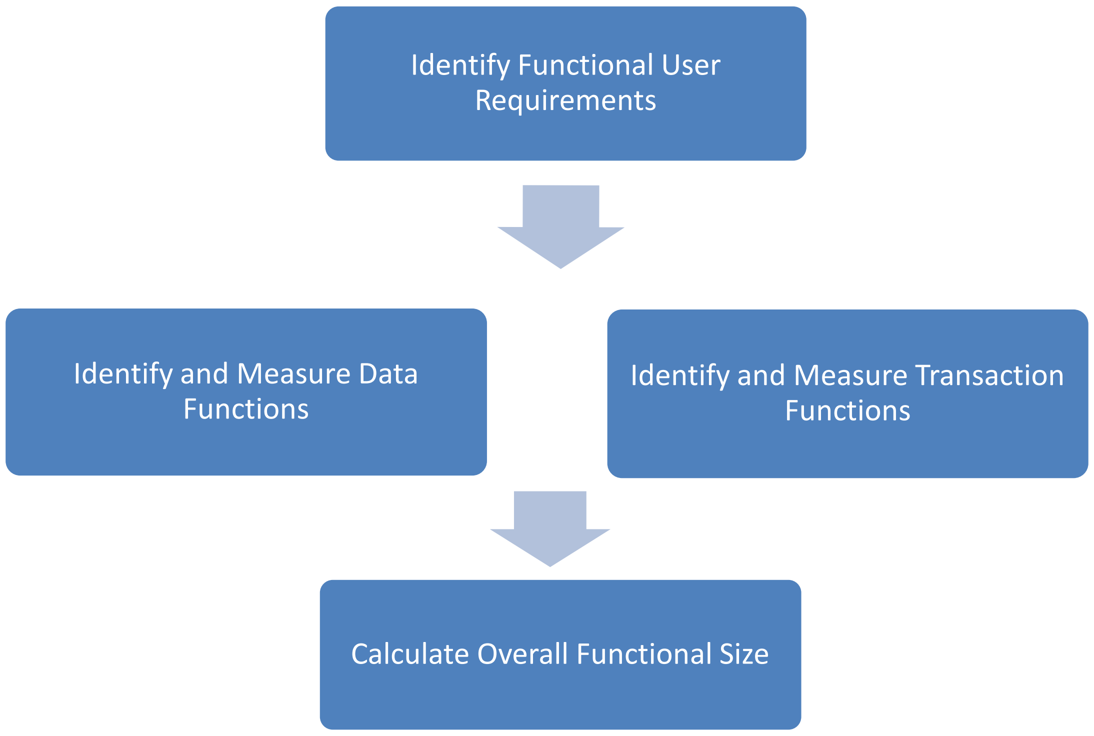
Identify Functional User Requirements
Identify and Measure Data Functions
Identify and Measure Transaction Functions
Calculate Overall Functional Size

Figure 2.8 – Core FPA sizing activities

Chapter 5 of this thesis is concerned with enhancing simplified sizing approaches to focus on providing a more detailed profile of functional size at an acceptable level of estimation accuracy. Attempting to preserve estimation accuracy at a finer level of detail than the overall functional size is necessary to provide value to the estimation process. This investigation includes an in depth examination of the existing approaches to simplified sizing that have introduced in this section, identifying where enhancements can be made to provide greater utility for the estimates.

2.5 Limitations of Software Estimation

Section 2.3 provided a discussion of the main approaches to software estimation, while Section 2.4 addressed the significant research issues that are being addressed. In considering how to improve the software estimation process it is important to recognise the limitations of what has been achieved and what can be realistically achieved. Software estimation within industry is constrained by various factors associated with the software development process. Software estimation research has often been too concerned with estimation accuracy at the expense of acknowledging wider concerns that could advance our understanding of the estimation process.

2.5.1 Limitations of Software Estimation in Industry

The extent to which the effort and cost of a project can be accurately estimated is limited by various factors.

- Estimation Practices
    - The most accurate estimation results are correlated with the quality of data utilised by an estimation method rather than the method itself (Berlin et al., 2009).
    - There is an element of subjectivity in software estimation, which prevents variance in estimates from being eliminated.
    - The barriers to use, such as the resources required, prohibit the application of more intensive methods for many organisations. Failure to implement formal, rigorous estimation practices is noticeably more common within smaller organisations.
- Organisational factors
    - Commercial development projects are subject to economic or managerial pressures which may influence the estimate. For example, in a tender situation the objective may be one of 'price-to-win' rather than to accurately estimate cost. Even for internally sanctioned projects, pressure to meet desired deadlines or budgets can lead to overly optimistic estimates being presented.
    - Changes in personnel, development methodologies, or the nature of projects within an organisation necessitate ongoing refinement of the estimation process.

The main factors responsible for inaccurate estimates reported in studies have been requirements issues, management pressure and lack of estimation resources available (Lederer and Prasad, 1995; Yang et al., 2008). Requirements specifications can hinder the accuracy of estimates at the outset of projects by lacking sufficient detail or clarity. Subsequent instability in the form of evolving requirements as the project progresses further impacts the degree to which the estimated effort corresponds to the actual effort. The impact of other business concerns and the limited availability of common resources across projects have resulted in the need to consider wider issues beyond the estimation method adopted (Magazinovic and Pernstal, 2008).

The prevalence of more informal expert or analogy-based methods within industry constitutes a reliance on more subjective approaches. For example, the presence of irrelevant or misleading information has been demonstrated to adversely affect expert estimators (Jorgensen and Grimstad, 2008). The presence of information such as cost expectations, suggestive wording etc. were found to influence the resulting estimates. The provision of explicit instructions to ignore any such information was found to be ineffective in negating its effect, and the estimators themselves were subsequently found to underestimate the effect the inclusion of such information had on their estimate. This is consistent with studies within other fields of research that demonstrated that even when participants were provided with advance warning of an anchoring effect they were unable to avoid it (Wilson et al., 1996).

There is general acceptance that there will be no universal solution to software estimation, and as such the value of improving any specific estimation method will be restricted. The question may therefore be one of how the progress of academic research in software estimation can be guided by industrial practice in order to reduce the divide between them. The approach supported by researchers such as Jorgensen is to focus on developing a greater understanding on how expert judgement is conducted in order to increase its effectiveness. Addressing the divide from the opposite side requires focusing on how model-based estimation methods can provide business value to the estimation process.

Minkiewicz (2009) acknowledged that any desire to size software in terms of its business value will be hindered by the impact of non-functional factors upon the effort required to develop the specified functional requirements. It was proposed that attention is therefore needed on the development of a business value based language with which to describe software. This would enable business driven functionality to be measured separately from information technology driven functionality.

2.5.2 Limitations of Software Estimation Research

The main approaches to software estimation adopted within industry were summarised in Section 2.2.2. The nature of academic research in the field of software cost estimation was addressed in Section 2.2.3, highlighting the systematic review of cost estimation research by Jorgensen and Shepperd (2007). This review reported the most frequently investigated estimation methods and the research methods most commonly employed. The main pattern evident from this was the ‘gap’ between how estimation is being researched and how it is being used within industry. The limitations arising from this can be summarised as follows:

(L1) The continued focus on the introduction and evaluation of estimation methods has not produced a discernible general improvement in the performance of estimation methods.

(L2) Research has generally adopted a technical focus on improving estimation models without consideration of appropriate related industrial contextual factors which affect the estimation process.

(L3) The evaluation of estimation methods is predominantly performed using historical project data rather than actual project documentation, which limits the consideration of the practicalities of applying such methods.

(L4) The evaluation of estimation methods is primarily concerned with the accuracy of such methods with limited consideration of how such methods can provide value for industry estimation practices.

The overall quality of research has also been subjected to criticism in terms of how rigorously the conclusions have been established. The systematic review of Jorgensen and Shepperd (2007) indicated that there was an overly narrow focus in terms of where software estimation research is published, with just three journals accounting for almost half of the published articles. References to other published articles were found to be similarly limited in breadth. The validity of studies which 'prove' the performance of their new estimation method has been assessed as being less than robust (Kitchenham and Mendes, 2009). The performance of estimation methods is often evaluated against linear regression, but in some cases was found to incorporate the use of an insufficient number of data sets and the incorrect application of linear regression. The selection of historical datasets which are to be included in these research studies are often not clearly justified. The research has continued to utilise outdated or inappropriate data sets. The level of statistical verification used in the research of estimation methods is often insufficient to support the associated stated conclusions. The main limitations arising from these criticisms can be summarised as:

(L5) The research studies are prone to overlooking highly relevant articles which would provide greater validity to their conclusions.

(L6) The reliance on historical datasets continues to adversely affect the reliability of the estimation results.

(L7) The applicability of developed estimation methods can be restricted by the limited evaluation of their performance.

The prevailing conflict between subjectivity and objectivity underlies the development of improvements in the estimation approaches. In essence more informal approaches, such as Expert Judgement, are viewed as being fundamentally more subjective than more formal approaches. In practice, any expert would adopt some degree of objectivity in how they approach developing their estimates. The development of estimation models is viewed as providing a more objective foundation for software estimates. In practice, there is an element of human judgement in the use of any estimation method wherein the ‘inputs’ to the method have to be established by the estimator. This should be understood in order to appraise the effectiveness of using these methods. The question is whether these contrasting estimation approaches can complement each other. A fundamental difference between formal and informal estimation approaches is the relative absence of software metrics in the latter category. Consequently, the use of software metrics within industry has been limited due to the reliance on more informal estimation approaches.

Mair and Shepperd (2011) proposed that estimation research should address the human judgement inherent in developing estimates by utilising cognitive psychology. In particular, it is necessary for estimators to understand how biases in their judgements are caused and effectively challenged. An essential component of such research was identified as how software metrics are put into practice within the estimation process. This change in focus in software estimation research would facilitate the development of a greater understanding of how more formal approaches can be adopted in practice. However, to address the limited use of formal estimation methods in industry it is necessary to establish that there is business value in adopting these methods. Investigating the business value of software metrics in the context of the estimation process is therefore a key objective to be addressed.

2.6 Prospective Contribution to Research

The primary issue to be addressed within this thesis is the enduring contrast between the focus of academic research and that of industry practice. This division can be characterised by the estimation approaches that are prevalent within the respective areas. Research into more formal estimation approaches e.g. model-based methods, has been extensive in comparison to more informal approaches used within industry. The potential advancement of software estimation can therefore be directed towards how the focus within each area can be more closely aligned. In order to work towards such an alignment, it is necessary to investigate how the contrasting approaches to estimation can inform each other. In essence, the research of more formal estimation approaches should be concerned with how they can contribute to business concerns within the software development industry, while the use of estimation within industry should incorporate some degree of formality in the process.

The research project is subject to the following specific constraints, which affect how the primary issue can be addressed:

(C1) The development practices within Equiniti-ICS preclude the investigation of estimation methods on active projects by an individual researcher. The opportunity to conduct estimation on a sufficient number of active projects is not present within the available timeframe.

(C2) Investigation into the estimation practices of Equiniti-ICS is limited to information gathering from relevant personnel from the project development teams.

(C3) The investigation of the use of formal estimation approaches in a commercial project context is restricted to the use of project documentation from previously completed software projects.

(C4) The scope of the software estimation research is limited by the industry sector, development methodology and estimation practises adopted within Equiniti-ICS.

(C5) The use of an individual researcher limits the range of formalised estimation approaches that can be incorporated into the research.

These constraints prohibit addressing all of the limitations of software estimation research outlined in Section 2.5.2. The strategy adopted for this research project is therefore to minimise the degree to which any of the limitations are present within each phase of the research. The contributions to this area of research are therefore:

1) Examine the value of formalised estimation approaches on commercial project documentation. In particular, the explicit consideration of software size should form the basis upon which to distinguish the estimation approach from expert judgement-based methods common within industry.
2) Evaluate the existing estimation practices within Equiniti-ICS in order to identify the main issues that inhibit the effectiveness of the estimation process.
3) Identify how the use of software sizing can be directed towards addressing these main issues in order to provide value to the estimation process. In particular, the use of more detailed size estimation data to provide such value should be analysed.
4) Investigate how the barriers to the use of software sizing can be diminished by focusing on how and where such activity should be directed. In particular, enhancing simplified sizing approaches in terms of the level of detail they provide should be investigated.

The overall aim of the research is directed towards investigating how software size estimation can provide value to the development process (L4). This is not focused on improving the general accuracy of size estimation methods, so limitation L1 is consequently not a fundamental characteristic of this research. The investigation into the value of software sizing, however, considers the trade-off between estimation accuracy and the effort required to develop the estimates. The potential for improving this trade-off would enable the general performance of software sizing to be reassessed in the context of real world software development. The second contribution in particular addresses consideration of the industrial factors that affect the estimation process (L2). The use of commercial project documentation avoids the reliance on historical datasets for evaluating estimation methods (L3 and L6). The constraints imposed on this research inhibit the degree to which any developments on how software estimation is performed can be validated statistically, and consequently the conclusions that can be drawn.

Limitation L7 is therefore considered in terms of primarily focusing on the process of applying software size estimation to commercial projects and the potential contribution that such a process can make. The scope of the research is not limited to the main journals in which software estimation research is published. Limitation L5 is therefore acknowledged during the completion of this research.

2.7 Conclusion

The Literature Review has outlined the progress made within software estimation research, but highlights the limitations inherent with the estimation process. The continued disparity between the focus of research and the methods adopted within industry indicates that there is an ongoing pursuit of solving the estimation problem rather than on how the estimation process can provide value to software development in practical terms. Instead of the seemingly futile objective of getting the ‘correct’ estimates, the goal should be one of using the estimates correctly.

The proposed research is therefore concerned with understanding how formal software sizing can be related to more informal estimation approaches adopted within industry. The focus is on how software sizing can meet the needs of commercial software development in order to provide value to development organisations. Understanding the practical realities of attempting to implement software sizing enables the process to be refined to provide tangible benefits in addition to the expert judgement-based approaches currently favoured in industry.

3. The Value of Software Sizing

In the Literature Review in Chapter 2, the need to examine the role of software sizing in commercial project estimation was identified. This Chapter is concerned with the first Research Phase of this investigation, namely the assessment of the value of software sizing within a commercial development environment.

3.1 Introduction

Previous industry surveys (Heemstra, 1992; Wydenbach and Paynter, 1995; Moløkken-Østvold et al., 2004; Yang et al., 2008) revealed that the use of model-based estimation approaches, such as the main software sizing methods, are limited. The prevailing pattern is one of more informal expert-based approaches to estimation, focusing on cost and effort rather than explicitly estimating software size. The CMMI (Chrissis et al., 2003) suggests that an explicit appreciation of software 'size' is expected. The size of the software facilitates an assessment of both project progress and the root causes of inaccurate estimates. This research phase investigates the use of software sizing on commercial projects, focusing on the value of this sizing, and incorporating an appropriate sizing method for the application domain of the sample projects.

In order to assess the value of software sizing in this context, the sizing results are evaluated against the existing expert-based estimation approach utilised by the research partner, Equiniti-ICS. This evaluation considers both the output produced by the contrasting estimation approaches and its subsequent utility in project management. The practical use of software sizing methods is evaluated in order to determine the feasibility of incorporating such as approach within a commercial development environment.

The value of software sizing can be dependent upon both the degree of rigour involved in the sizing method, and the stage of the project lifecycle at which the estimate is developed. For the sample commercial projects utilised in this research, requirements documentation is available from two distinct stages of the lifecycle. Firstly, the bid documentation, and secondly, the completed functional specification for each project facilitate the development of size estimates. The benefits of using these two distinct lifecycle stages derive from the different levels of requirements detail available at each stage. Firstly, the effect of having a more detailed statement of requirements can be assessed. Secondly, the level of detail can affect the degree of rigour feasible in a sizing method. This research phase therefore examines the extent to which size estimation can be utilised on commercial projects, and assesses how it can provide the most benefit within the overall project management process. This enables the value of adopting software sizing in a commercial context to be assessed, taking into account the effort and cost involved in the estimation process, together with the benefits for project management and bidding that may be gained from a deeper insight into the product size.

3.2 Related Work

Previous research on the use of software sizing has covered a range of sizing techniques. These can be broadly categorised into Functional Sizing methods and UML based sizing methods. Empirical studies on the use of these sizing techniques are discussed in the following section.

3.2.1 Empirical Size Estimation Studies

Methodical approaches to estimation, such as software sizing, require a greater commitment of time and effort than more informal estimation approaches. The justification of such a commitment should be that value is derived from the results of software sizing. The research issue is therefore one of identifying and assessing the relative value provided when software sizing is completed. Related work on this issue has examined different aspects of software sizing in assessing the value it provides. The effectiveness of software sizing has considered the following dimensions.

(1)The effect of the level of rigour of the estimation method on the accuracy or usefulness of the size estimate produced.
(2)The effect of the level of detail provided by the project documentation being assessed on the accuracy of the size estimate produced.
(3)The effect of the lifecycle stage the project documentation is produced at on the accuracy of the size estimate produced.
(4)Does the size of the documentation to be examined scale linearly with the effort required to complete the estimation?
(5)Does the effort required to develop size estimates outweigh any benefits derived from such estimates?

Table 15 provides an overview of previous work which has addressed the issues that would affect the proposed assessment. The search process for the studies to include within this related work was conducted within the context of the overall literature review for this thesis. This search process utilised multiple online databases such as IEEE Explore and ScienceDirect, as well as Google Scholar. During the overall literature research articles found under all searches were categorised according to their main research topics. Specific searches including terms related to the individual issues identified in Table 15 were performed as part of this process. This approach enabled articles outside of those returned by specific searches to also be identified for potential inclusion. In addition, the reference lists from the selected potential studies were examined in order to identify any further potential articles for inclusion. The candidate studies were then assessed to determine which were of satisfactory quality and relevance for inclusion in this phase.

Cândido and Sanches (2004) examined the effectiveness of the different levels of the NESMA approach, and found that the estimation accuracy of the simpler NESMA methods was significantly less than that of the full NESMA approach. The basic Indicative Method has a median error of 48% compared to the full NESMA method, while the more detailed Estimated Method still demonstrated a median error of 18%. A tailored version of the Estimated Method was able to produce an acceptable median error of only 4% by incorporating the typical profile of previously completed projects. The potential for improving the basic Indicative Method was not however considered.

Demirors and Gencel (2004) investigated sizing at an early stage of a large military application and found that the least detailed approaches were subject to the most relative inaccuracy. The Early Function Point method accuracy was seen to improve, so as to be comparable with the more rigorous Mark II method, as more detailed requirements became available. A weakness of this comparison was the need to compensate for the lack of detail available at this stage for the completion of the Mark II method.

(Zivkovic et al., 2005a) investigated whether the amount of detail included in a class diagram would affect the accuracy of their modified Object Oriented Function Point Method (OOFP2). It was determined that an incomplete class diagram could in theory lead to an additional 50% error in the estimate produced, while evaluation on five sample applications demonstrated a still significant additional 16% error on average. The impact of the detail provided by project documentation was demonstrated by (Frohnhoff and Engeroff, 2008) using a UCP sizing approach. A study was conducted across 20 projects and found that the format of the use case specification had a significant effect on the overall UCP count, with state charts and business process description providing the least variability in results despite 'experts' viewing these formats as the least suitable amongst those tested. van Den Berg et al. (2005) examined the use of NESMA and COSMIC on UML based project documentation and concluded that a set of distinct document types should be examined when assessing the functional user requirements. For example, certain functionality could only be identified from activity diagrams, but data movements were identified from the use case descriptions. It can be inferred from these studies that the value obtained from examining project documentation will vary depending on which type of documentation is selected.

A comparison was made of sizing performed on thirteen applications at three distinct stages in the lifecycle, ranging from a simple statistical approach used at the initial use case diagram stage to a more detailed OOFP approach on the last stage's class diagram (Zivkovic et al., 2005b). While the results at the initial stage showed a significant relative error of 40%, a modest decrease to 36% by the intermediate stage would indicate that there would be negligible justification for conducting estimates at that stage. The final stage showed a more significant improvement in accuracy, sometimes by as much as 50%, which supported the belief that accuracy improves as more detailed data becomes available.

Table 15 - Overview of Related Software Sizing Studies

Author	Methods	Estimates/Measures	Method Detail	Data Detail	Lifecycle Stage	Document Size	Estimation Effort	Combined Method/ Stage
Cândido and Sanches, 2004	Indicative, Estimated, (new) Simplified NESMA	Size	Yes	No	No	No	No	No
Demirors and Gencel, 2004	Mark II, Jones very Early, Early Function Points	Size	Yes	Yes	No	No	Yes	No
van den Berg, 2005	Full NESMA, COSMIC FFP	Size	No	Yes	No	Yes	No	No
Zivkovic et al., 2005a	IFPUG, Mark II, COSMIC FFP, OOFP, OOFP2	Size	No	Yes	No	No	No	No
Zivkovic et al., 2005b	ISBSG Statistics, OOFP	Size, Effort	Yes	No	Yes	No	No	No
Frohnoff and Engeroff, 2008	UCP	Size/Variation	No	Yes	No	No	No	No
Gencel et al., 2009	SLOC, Bytes of Code, IFPUG, COSMIC FFP	Size	No	Yes	Yes	No	No	No
van Heeringen et al., 2009	Indicative, Estimated, Full NESMA	Size	Yes	No	No	No	Yes	No
Agresti et al., 2010	SLOC	Size	No	No	Yes	No	No	No
Popović and Bojić, 2012	Indicative NESMA, Estimated NESMA, IFPUG/Full NESMA, Mark II, COSMIC FFP, UCP	Size, Effort	Yes	No	Yes	No	No	No

(Gencel et al., 2009) focused on the need to use the size measures obtained from early lifecycle products, in particular the requirements specification, as the basis for estimating the size of future products such as the code developed. The focus of this study was to investigate the relationship between Functional Size measures typically used early in the lifecycle and Code Sizing measures for the completed code. At the level of international project data, the conventional FPA approach was found to be weakly correlated with SLOC. However, COSMIC FFP and Bytes of code were found to have a significantly stronger correlation, on the limited number of components tested, possibly due to the greater independence from how the functionality is implemented. The implications of this study are that a strong correlation between Functional Sizing and the size of the completed system may not be found which could limit the effectiveness of this approach to estimation. The question of whether there is a correlation between Functional Sizing and the relative sizing of completed systems could provide an additional insight into the value that may be obtained from the sizing process.

(Agresti et al., 2010) investigated how to incorporate additional information from the product being developed into size estimation models as it becomes available e.g. Library units, context coupling, visible program units, hidden program units etc. This involved the development of a 'family of models’ that facilitate the refinement of the size estimate as development progresses. As such, this represents a similar approach to that of the COCOMO II estimation model (Boehm et al., 2000) that caters for distinct stages in the lifecycle. In their study Agresti et al. (2010) identified four distinct stages from design to implementation at which information became available and hence a new size estimate could be made. The estimates were produced from the completed system code, based on the assumption of which 'variables' would have been available at a particular stage of development. Linear regression was used to develop the models for each stage and evaluated across four projects. Results showed improvements at each stage, with the first stage explaining 47% of the variation in SLOC compared to the final stage which explained 89% of the variation. While this study supports the assertion that estimates improve as more detailed information becomes available, the use of SLOC for this approach would leave it subject to the same limitations associated with this particular size measure.

Popović and Bojić (2012) evaluated the use of a variety of sizing methods on real world project documentation from three separate lifecycle stages. A dataset of 30 projects from a single CMMI Level 2 organisation was used in the investigation, with detailed actual development effort figures available for each project. The focus of the study at each lifecycle stage was on using the most detailed sizing method feasible according to the documentation available from that stage. At the initial ‘inception’ stage, size estimates were developed using only the Indicative NESMA method and a simple count of the number of use cases. The second ‘elaboration’ lifecycle stage was subdivided into earlier and later documentation. Use case scenarios facilitated the use of Estimated NESMA, Mark II and UCP methods. The later class diagrams facilitated the use of the IFPUG/Full NESMA method. The functional sizing methods outperformed the UML based methods at both stages, with each more detailed version of the methods producing a slight improvement in the MMRE of the estimated efforts. The Indicative NESMA method at the inception stage demonstrated an MMRE of 15%, with the Estimated NESMA method at 13% and the IFPUG/Full NESMA method at 10%. For the final ‘construction’ lifecycle stage the focus was on actual effort for specific tasks rather than the overall system size, reflecting a focus on controlling projects at that stage. COSMIC FFP estimates were very strongly correlated with actual effort for specific tasks at this stage, with an MMRE of 8%. The IFPUG/Full NESMA method in contrast had issues with estimating effort for specific tasks due to the upper and lower boundaries on function size. Task with sizes at the upper and lower boundaries did not reflect the variation in actual effort recorded for those tasks. In this regard, the conventional FPA approach may be considered to be less suitable for estimating at such a low level of granularity for a project.

The issues of documentation size and effort required to complete an estimate were rarely acknowledged in these studies and even in those cases were not subject to any rigorous investigation. None of the studies investigated combining the comparison of estimation methods of varying detail with the comparison of estimating at different stages of the lifecycle. While more detailed estimation can be performed later in the lifecycle there would still need to be appreciable value obtained over continuing with a simpler approach. Given that accurately estimating actual size has remained somewhat elusive, it may be that relative size represents an achievable goal that it is still of value to an organisation.

The final column in Table 15 is concerned with combining the examination of the effects of the level of rigour in the sizing method and the stage of the project lifecycle. Thus far, related work in this area has not addressed whether the type of sizing method adopted should vary according to the development stage of the project. This research phase explicitly addresses this issue in order to assess where value can be obtained from the use of software sizing.

3.3 Research Aim

This research phase is concerned with investigating the value of software sizing to the project management process using commercial project documentation. This investigation is aimed at identifying the potential for software sizing to assume a role within the software development process. The evaluation of software sizing is therefore required to be within the context of the existing expert-based estimation approach currently employed within Equiniti-ICS. The research therefore utilises both the commercial project documentation, covering the system functional requirements for completed projects, and the associated Equiniti-ICS estimation and development data for these projects. The use of software sizing on these completed projects should incorporate the use of sizing methods that involve differing levels of rigour. These sizing methods should be used to develop size estimates on requirements documentation from different stages in the project lifecycle.

For this investigation, 'value' is therefore defined in terms of three different factors affecting the use of software sizing. Firstly, the trade-off between benefits and costs considers where the optimal balance lies for implementing the use of software sizing. Secondly, the degree of utility considers how the sizing data generated by the estimation process can be utilised to provide further technical insight within the project management activities. Thirdly, the relationship of how software sizing would relate to existing estimation practices considers whether this approach can be incorporated into the software development process.

3.3.1 Research Questions

The research questions for this research phase are concerned with the practical application of software sizing on commercial project documentation. This enables constraints arising from 'real world' project documentation to be incorporated into the research approach.

RQ1: Which combination of software sizing rigour and project lifecycle stage provides the most stable indication of project size?

This question seeks to identify where the optimal trade-off between estimation effort and estimation accuracy can be achieved. The estimation performance of software sizing is evaluated against the existing estimation approach used within Equiniti-ICS.

RQ2: How can size estimation data assist associated activities such as project planning within the development company?

This question is concerned with the benefits provided by software sizing above those obtained by the existing estimation practices. The insight provided by obtaining the size of a project should generate additional guidance on the subsequent development of a project.

RQ3: How can the software development process of the company accommodate the use of software sizing?

This question assesses the practical implications of implementing the use of software sizing within software development companies. The adoption of software sizing should not necessitate fundamental changes to existing development practices.

3.4 Research Approach

In Chapter 1 a profile was provided of the commercial development organisation, Equiniti-ICS, that is facilitating this investigation into software sizing. This profile addressed the nature of the sample projects, the requirements documentation and the estimates practices of this organisation. This section discusses the additional project data that was provided by Equiniti-ICS as part of this research. This is followed by the assessment of the potential sizing approaches that are presently used, and ultimately the selection of an appropriate approach for use in this research. An overview of the selected approach is then provided, followed by a description of each of the research stages necessary to complete this investigation.

3.4.1 Organisation Project Data

The effectiveness of software sizing can be measured against the estimation approach currently employed within Equiniti-ICS, and actual measures available for each completed project. Effort and cost estimates were developed by Equiniti-ICS at the initial Bid stage of each project, with corresponding actual development effort figures maintained for each project. The commercial value of each individual project, in terms of the amount of revenue derived, is also included as part of the project data provided for this research.

This research phase is concerned with the size of the software, which provides the basis for deriving the subsequent effort and cost for a project. Effort and cost estimates therefore provide a measure with which to evaluate the software size estimates developed as part of this research. However, it would additionally be necessary to consider the ‘actual’ size of the completed software for each project in some way as a means of validating the size estimates. No direct measurement of the completed software size is currently made by Equiniti-ICS as part of their development process. It is therefore necessary to develop an appropriate alternative measure of software size as part of this research. Equiniti-ICS were asked to provide an assessment of the relative size of the respective system developed for each project. This research stage is described in Section 3.4.4.1.

3.4.2 Software Sizing Approach

This investigation into the value of software sizing is focused on evaluating the potential benefits of adopting such an estimation approach. The scope of this investigation is on existing software sizing methods that have been established and the subject of peer reviewed research. This provides a degree of applicability to this research phase, as well as an appropriate basis for assessment of the results of the software sizing exercise. The existence of standards and guidelines provides validity to the reliability of using a specific sizing method.

The selection of an appropriate software sizing method required consideration of both the characteristics of the sample dataset, and the research questions outlined in Section 3.3.1. The sample dataset presents the following constraints on the sizing approach:

(C1) The applications developed in the sample projects are data intensive, with the primary emphasis on recording case management application data and processing case updates.

(C2) The Bid stage documentation consists primarily of textual descriptions of functional and non-functional requirements.

(C3) The Functional Specification consists primarily of use case descriptions of user functionality, with supporting report specifications and data models.

(C4) No local historical software size estimation data is available within Equiniti-ICS.

These constraints influence the general suitability and feasibility of using the available software sizing methods. The research questions presented the following constraints on the sizing approach:

(C5) Different levels of rigour in the sizing approach must be facilitated by the selected method(s).

(C6) Software sizing must able to be performed on the Bid documentation and the Functional Specification.

(C7) The sizing output data must facilitate evaluation within the project management process used within Equiniti-ICS.

(C8) The sizing approach must facilitate evaluation of the ease of integration into the software development practices of Equiniti-ICS.

These constraints have a more specific influence on the selection process for the sizing approach to be used in this research phase. A sizing method would need to incorporate a flexible approach to provide estimates at different levels of sizing rigour (C5). The Bid stage size estimate is limited by the amount of requirements detail available in the documentation, which prevents the use of some sizing methods (C6). The size estimate output detail is required to be the same for both stages, preventing fundamentally different approaches to be used at the two stages (C6). In addition, the size estimate output detail should facilitate evaluation in terms of subsequent project management activities (C7). The sizing approach should rely upon the same project documentation currently created during the development of each project, in order to minimise potential disruption to the development process (C8).

3.4.2.1 Overview of Software Sizing Methods

The scope of this sizing investigation covers the initial and final stages of the requirements documentation. Software sizing approaches such as Lines of Code are therefore not applicable at this stage. Methods for specific development approaches e.g. User Stories in Agile development are not appropriate as the sample projects reflect the commercial situation where the complete system size estimate is required at the outset of the project. The most relevant software sizing methods are therefore those which are focused on estimating the user functionality required by the system from requirements documentation.

The most established approaches are the FSM methods approved by ISO. Two of these are based upon the original FPA approach, namely IFPUG (International Organization for Standardization, 2009) and NESMA (International Organization for Standardization, 2005). Variations on this approach, Mark II (International Organization for Standardization, 2002) and COSMIC International Organization for Standardization, 2011) have also been standardised. The FISMA (International Organization for Standardization, 2010) is the final ISO standardised Functional Sizing method. An ISO published standard International Organization for Standardization, 2006) addresses the issue of selecting an appropriate Functional Sizing Method, which provides guidance in this research stage.

(1) The functional domain of the software, considered under C1 in Section 3.4.2.2
(2) The performance requirements of the estimation method, considered under C5 in Section 3.4.2.3
(3) The BFCs considered by the method, considered under C6 in Section 3.4.2.3
(4) The purpose of using Functional Sizing, considered under C7 and C8 in Section 3.4.2.3

The main alternative sizing approaches are concerned with UML representations of user functionality. These can attempt to replicate the functional sizing measures by mapping FP concepts to object oriented concepts. Alternatively, an equivalent measure such as UCP can be used which assesses actors and use cases in a similar manner to the FPA approach of assessing

BFCs. The UML based approaches have been the focus of related research, but there remains no standardised method for use in this investigation.

The evaluation in the following sections incorporates consideration of the main Functional Sizing methods as well as the use of UML based approaches.

3.4.2.2 Evaluation against project characteristics of dataset

The ISO standard (International Organization for Standardization, 2006) on FSM methods identifies the functional domain of the software to be developed as a selection factor. The sample project systems have been identified as corresponding to a data driven rather than process driven focus. Both the FSM methods and the UML approaches are appropriate for these types of applications. IFPUG is the most commonly adopted method in this functional domain, with the similar NESMA providing an equivalent choice. These methods are considered to be relatively weaker at catering for algorithmic complexity in the system functionality than alternatives such as COSMIC. One of the projects in the sample dataset can be distinguished by its greater inherent algorithmic complexity, but this is not considered to affect the selection process in this case because the individual project represents a small proportion of the overall dataset. The UML based approaches are also suitable for this functional domain due to the focus on data driven user interaction. (C1)

The Bid stage documentation provides an outline of the required functionality, but generally does not provide detail on the specific data elements each system entity is comprised of. At this stage it is therefore feasible to identify the main functions required but not the complexity of these functions. The full functional sizing approaches are therefore not feasible at the Bid stage. It is therefore necessary to use a more simplified sizing approach for the estimates at this stage. From the ISO standards, only NESMA includes simplified methods as part of its official standard. COSMIC and FISMA consider simplified methods as part of supplementary guidelines. The UML based approaches are not considered ideal for the Bid stage due to insufficient detail present in the requirements documentation. (C2)

The Functional Specification stage provides a completed description of the required functionality, including detail on the individual data elements present in a data model. Each of the ISO FSM methods is therefore considered suitable for developing size estimates at this stage. The use case description format of the Functional Specifications supports the use of a UML based approach. For the sample projects, the reliance on textual descriptions rather than other UML diagrams means that there is not considered to be any significant difference in the ease of using either a Functional Sizing or a UML based approach at this estimation stage. (C3)

The absence of historical local size estimation data does impact upon the suitability of the full recognised FSM Methods. The simplified versions of the COSMIC approach generally require size estimates to be developed on a ‘training’ dataset in order to establish appropriate ‘weightings’ to be determined for subsequent estimates. The eleven available sample projects represent an insufficient quantity to establish an appropriate ‘training’ dataset, thus preventing most simplified versions of COSMIC from being utilised in this research stage. The use of sampling of typical functionality from the project being estimated is an alternative to ‘training’ for some simplified versions of COSMIC. The bid stage documentation does not provide sufficient detail for a sample to be accurately measured, so this alternative approach cannot be adopted in this research. The NESMA and FISMA simplified methods do not require the use of local historical data and are therefore suitable options. The UML based approaches are not as established in comparison, so the suitability of utilising the suggested ‘weightings’ is subject to greater uncertainty. The use of local historical data would serve to reduce the degree of uncertainty, but is not feasible within this research. (C4)

3.4.2.3 Evaluation against research questions

The main focus of this research phase is in evaluating varying degrees of sizing rigour at different stages of the development process. This investigation therefore entails the development of a range of size estimates at each stage, differing in the sophistication of the estimation method used for each estimate. The performance requirements of prospective estimation methods are to enable such a range of estimates to be developed, up to the most rigorous size estimate feasible at each lifecycle stage. It is therefore necessary to compare performance for both detailed and less sophisticated methods. An established basis of comparison would therefore require compatibility between the sizing methods. Only the FSM approaches that include simplified methods as part of their published standard facilitate direct comparisons. The IFPUG and NESMA are fundamentally equivalent, essentially providing similar sizing results. (C5)

The feasibility of identifying the necessary BFCs varies according to the stage of the development process the estimate is required. For this research the detail required to assess each BFC is not available at the Bid stage and as such simplified sizing methods would be required for these estimates. From the evaluation in the preceding section, only the NESMA and FISMA approaches are suitable for this aspect of the software sizing. (C6)

The purpose of performing the Functional Sizing is to investigate the value of software sizing on commercial projects. The most detailed software sizing feasible at each stage is therefore required to determine the extent to which such estimates can benefit the project management process. This aspect of the research does not impact upon the suitability of the Functional Sizing method beyond the previous considerations. The UML based approaches provide no advantage over the Functional Sizing methods for this aspect of the investigation. (C7, C8)

3.4.2.4 Selection of Software Sizing Approach

From the preceding evaluation sections, only the NESMA and FISMA methods satisfy the requirements of this research. In terms of validating the results against existing research, the NESMA approach has been the most prevalent in related studies. As it is fundamentally the same as the IFPUG method it is also more applicable to FPA in general. For these reasons the NESMA sizing methods have been selected as the sizing approach to be used in this research phase.

3.4.3 Overview of NESMA sizing methods

The NESMA method is established as a full ISO standard (International Organization for Standardization, 2005) for functional size measurement. In common with all such sizing methods, NESMA adopts a unit of measure, the ‘function point’, which gives an appreciation of the amount of information processing to be undertaken in terms of “what” processing is required without reference to “how” the processing may actually be accomplished (design considerations). The method focuses upon identifying BFCs for each Functional User Requirement (FUR). FUR’s may be understood from an examination of many sources such as requirements definition documentation, data analysis artefacts, user operation manuals or physical programs and screens. In this particular work, a combination of the first two sources has been used.

There are five types of BFC, namely: (i) Internal Logical Files (ILF); (ii) External Interface Files (EIF); (iii) External Inputs (EI); (iv) External Outputs (EO) and (v) External Enquiries (EQ). Each File consists of one or more logically related system entities referred to as Record Element Types (RET). Each RET is comprised of one or more attributes referred to as Data Element Types (DET). The method specification provides details on how to ascribe a complexity of either: HIGH, AVERAGE or LOW, to each BFC. Using a look-up table, the person performing the counting exercise can then convert each BFC into a number of function points $\{(ILF)_{FP}, (EIF)_{FP}, (EI)_{FP}, (EO)_{FP}, (EQ)_{FP}\}$. If $(FUR)_{FP}$ represents the function point count for a given Functional User Requirement, then:

$(FUR)_{FP} = (ILF)_{FP} + (EIF)_{FP} + (EI)_{FP} + (EO)_{FP} + (EQ)_{FP}$

and:

$\text{TOTAL FUNCTION POINTS} = \Sigma_{FUR}(FUR)_{FP}$

For the bid-level documentation used in this study, there was insufficient information to properly assess the complexity of individual BFC's. Two simplified versions of NESMA exist, which allow for this lack of detail and still enable function point counts to be calculated. The simplified versions are known as the Estimated NESMA and the Indicative NESMA methods. Both of these simplified versions are also described in the ISO specification (International Organization for Standardization, 2005).

In the Estimated NESMA method, all five previously mentioned BFC types are used, but an assumed AVERAGE complexity is ascribed to each of the Transaction Function types (i.e. EI, EO and EQ), while a LOW complexity is assumed for the Data Function types (i.e. ILF and EIF).

In the Indicative NESMA method, only the Data Function types (ILF and EIF) are considered. A count is made of the number of entity types in each of the ILF and EIF category and weightings are applied depending on whether or not a conceptual data model or a normalised data model is available. The bid-level documentation available for this research included sufficient detail to enable either of these simplified methods to be applied**.**

3.4.4 Research Stages

Research Stage 1 is concerned with how appropriate organisation project data can be accumulated, which provides the basis for evaluating the software sizing results. The steps involved in developing the functional size estimates are then outlined in Research Stage 2, including an indication of how this process satisfies the requirements of this research. In order to maintain the integrity of the functional size estimates, these first two research stages were completed independently. The organisation project data was generated separately within Equiniti-ICS and was withheld until Research Stage 2 has been completed. Likewise, there was no informal information about the sizes of the sample projects conveyed by Equiniti-ICS at this stage of the research. This ensured that there was no pre-existing knowledge about the size of the sample projects prior to the development of the functional size estimates.

Research Stage 3 involves the formulation of alternative classification of sizing results for the purpose of addressing whether there are limits to the accuracy that can be expected of sizing methods. Metrics associated with the functional sizing process are then introduced in Research Stage 4 as a means of obtaining a high level appreciation of the sizing process. Finally, in Research Stage 5 the process of accumulating feedback from Equiniti-ICS on the completed functional size estimates is described.

3.4.4.1 Research Stage 1 - Development of Organisation Project Data

In addition to the provided effort/cost estimates, actual effort data and commercial value for each project, it was necessary to develop an understanding of the actual size of the sample projects. The Technical Director at Equiniti-ICS, with experience on each of the projects, was asked to establish an appropriate measure of the sizes of these projects. The following potential measures were considered:

Number of process / screens - the senior management team did not wish to re-create estimates using sizing techniques similar to those which would be used in this research.

Lines of Code - This approach was considered to be inappropriate for reflecting system size for a number of reasons. The use of different underlying technologies hinders direct comparison of the lines of code. The systems in newer projects are model driven, resulting in the generation of substantially more code. The inability to separate the customer specific code from the reusable (licensed) elements would complicate the counting process.

Size of Executable files - The use of different technologies in the projects precludes the use of the size of the files as a relevant measure. In addition, the applications have a large number of standard executable files that have to be included in a project, some of which are in effect compiled into the executable. The issue of separating reused elements from the customer specific aspects applies to this measure as well.

The most effective approach was thus determined to be a management survey, conducted by the Technical Director of Equiniti-ICS. Five of the Senior Managers, with experience on one or more of the projects, were asked to rank the projects in terms of size. For projects where the Senior Manager had no direct involvement, they provided a judgement based on knowledge either from similar work or through reputation. The results of this management ranking of projects are therefore relatively subjective in nature, but also reflective of the extent of knowledge held within the organisation regarding project sizes.

This analysis, combined with the previously existing project data, completed the accumulation of the internal company projects rankings. The distinct categories of organisation project rankings developed in this research are shown in Table 16.

The management size ranking is based on the judgements of how the projects compare to each other. The estimated effort ranking is based on the expert judgement estimates produced at the initial bid stage of the projects. The actual development effort ranking is based on the recorded effort values for each project, and the commercial value ranking is based on monetary value derived from each developed system.

Table 16 - Project Rankings Categories

Organisation Rankings
Management Size Ranking
Estimated Effort (Expert)
Actual Development Effort
Commercial Value

3.4.4.2 Research Stage 2 - Functional Sizing Steps

Functional size estimates were required for the bid and detailed functional specification stages for each of the eleven commercial projects used in the research. Three variants of the NESMA approach were selected in Section 3.4.2.4., with which to develop these estimates. For each Project/Method size estimate pair it is necessary to record both the total FP and the effort in hours required to complete the sizing. The use of a single person to complete this software sizing introduces potential threats to the validity of the results. The use of different sizing methods on the same project presents the possibility of acquiring knowledge about the project from the use of one method, subsequently benefitting the use of a subsequent method. In particular, the understanding gained from the first method could reduce the effort required when using the next method. This effect could also potentially arise as a result of developing size estimates at two distinct stages for each project. In this case, the understanding gained from the

Bid stage estimation could reduce the learning required in the subsequent Functional Specification estimate. To addresses these potential threats, it is necessary to seek to minimise the 'learning' effect both between the various NESMA methods, and between the two stages of estimation.

Each of the NESMA methods effectively extends the previous method, with a high level summary of the sizing activities illustrated in Figure 3.1. For each NESMA method the 'new' tasks are indicated in bold, demonstrating how these tasks extend the previous NESMA methods in the process.

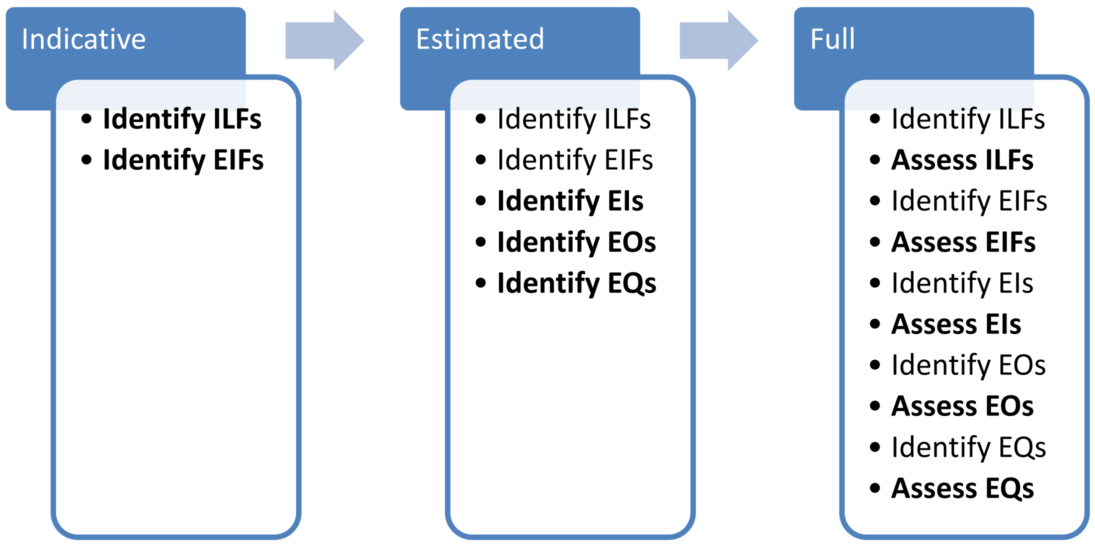

Figure 3.1 – Summary of sizing activities for each NESMA method

The Indicative NESMA method is concerned only with the identification of the Data Functions, including the grouping of RETs together into ILF or EIF components as appropriate. The Estimated NESMA method extends this by identifying the Transaction Functions, and determining the specific type of each function i.e. EI or EO or EQ. The Full NESMA method further extends the estimation by assessing the complexity of each of the Data Functions and Transaction Functions. The use of the NESMA methods in this order generally separates the estimation process into distinct components, but there are some potential overlapping activities as shown in Table 17.

The Full NESMA method is not applicable to the Bid stage due to insufficient detail in requirements documentation. The absence of a Data Model at the Bid stage necessitates the use of the textual description of the required functionality in identifying the Data Functions. Some degree of understanding of the required Transaction Functionality is inevitable in completing the use of the Indicative NESMA method at the Bid stage. The availability of the Data Model at the Functional Specification stage provided a greater degree of independence between Data Functionality estimation and Transaction Functionality estimation. However, in determining how RETs should be grouped in ILFs/EIFs it can be necessary to consider the associated Transaction Functionality. There is therefore some overlap between the Indicative and Estimated methods at both stages of estimation.

Table 17 - Relationship between NESMA methods at each lifecycle stage

Method Relationship	Bid Stage	Functional Specification
Indicative-Estimated	Identification of Data Functions from Functionality Requirements	Refinement of Data Functions from Functionality Requirements
Indicative-Full	-	Grouping of RETs into ILFs/EIFs
Estimated-Full	-	Refinement of Transaction Functions

The grouping of RETs into ILF/EIFs provides part of the complexity assessment of the Data Functionality, which considers the number of RETs and DETs for each Data Function. This assessment leads to the partial overlap between the sizing activities of the Indicative and Full methods. The complexity assessment of Transaction Functions can lead to the identification of 'duplicate' functionality e.g. the identical set of RETs and DETs are involved in separately specified functions. The identification of Transaction Functions from the use of the Estimated NESMA method can therefore be refined during the complexity assessment from the Full method.

In addition to the 'learning' effect across the different NESMA methods, the knowledge developed during the Bid stage estimate can aid in developing an understanding of the system in the Functional Specification stage estimate. Conversely, this pre-existing knowledge can also present some uncertainty in the interpretation of the Functional Specification. The impact of the earlier size estimate on the effort required for the subsequent estimate can therefore be both positive and negative. The research approach adopted should seek to minimise this impact by reducing the potential for carrying pre-existing knowledge into the latter Functional Specification estimate.

To address these potential threats, the use of these software sizing methods has been organised into specific steps designed to minimise any 'learning' effects. Table 18 shows the steps involved in completing the software sizing exercise for each individual project.

Table 18 - Functional Sizing Steps for each lifecycle stage

Sizing Method/Documentation Stage	Bid Stage	Functional Specification
Indicative NESMA	Step A1	Step B1
Estimated NESMA	Step A2	Step B2
Full NESMA	-	Step B3

The overall process for each project is to separate the Bid stage estimate from the Functional Specification stage estimate. This is achieved by completing Step A1 and Step A2 for each project in turn, enabling the Bid stage estimates to be developed across the dataset. Upon completion of the Bid stage estimation, the sizing moves onto the remaining Steps (B1 to B3) for each project in turn. This establishes a buffer between the two stages for each project, diminishing the effect of possessing pre-existing knowledge for the latter stage estimate. The design of each Step addresses the overlap between the individual NESMA methods to ensure estimation effort is appropriately recorded.

Step A1

- The RETS are identified and assessed in order to establish the ILFs/EIFs for the project.
    - Assessment is focused on identifying system entities from main indicators in the documentation e.g. section headings.
    - The estimation effort is recorded for the Indicative NESMA method.
- The understanding of Transaction Functionality is minimised in this step.

Step A2

- The Transaction Functions are identified categorised as EI, EO or EQ as appropriate.
    - This process involves consideration of the complete detail provided in the documentation.
    - The estimation effort is recorded for the Estimated NESMA method.

- The more detailed description of the functionality enables the RETs, and consequently the ILFs/EIFs to be refined.
    - This estimation effort is recorded for the Indicative NESMA method.

This procedure minimises the potential ‘learning’ effect between the Indicative and Estimated methods for the Bid stage estimation in this research. The design of Step A1 and Step A2 additionally provides a more reciprocal division of the ‘learning’ effect across the Indicative and Estimated methods, further reducing any potential threat.

Step B1

- The Data Model is assessed independently of the remainder of the Functional Specification.
    - The focus is solely on the Data Functionality in this step.
    - The estimation effort is recorded for the Indicative NESMA method.

Step B2

- The Transaction Functionality is identified and categorised as EO, EO or EQ as appropriate.
    - This process involves consideration of the complete Functional Specification and any related documentation.
    - The estimation effort is recorded for the Estimated NESMA method.
- The analysis of the Transaction Functionality enables the RET, and consequently the ILF/EIF to be refined.
    - This estimation effort is recorded for the Indicative NESMA method.

Step B3

- The complexity of the Data Functions and the Transaction Functions is assessed from the Data Model and the Functional Specification.
    - For any function with insufficient detail available the assumed complexity from the Estimated NESMA method is retained.
    - The estimation effort is recorded for the Full NESMA method.
- The analysis of the Transaction Functionality enables the identification of Transaction Functions to be refined where appropriate.
    - This estimation effort is recorded for the Estimated NESMA method.

The potential ‘learning’ effect between the Indicative and Estimated NESMA methods for the Functional Specification stage estimation is less significant due to the availability of the Data Model. The design of Step B1 and Step B2 similarly provides a more reciprocal division of the ‘learning’ effect across the Indicative and Estimated NESMA methods, further reducing any potential threat.

In Step B1 the grouping of RETs into ILFs/EIFs contributes to the complexity assessment in Step B3. The estimation effort for the Indicative NESMA method is implicitly included in the estimation effort for the Full NESMA method. Consequently, there is no requirement to separate the estimation effort for this aspect of Step B1 when allocating the effort towards each method. The overlap between the Indicative and Full methods is not considered to represent a potential threat to the accuracy of the recorded estimation effort.

The design of Step B2 and Step B3 address the overlap between the Estimated and Full methods in a similar manner as with the other steps. The understanding of the complexity of the Transaction Functions is minimised during the use of the Estimated NESMA method. The potential ‘learning’ effect is effectively divided by delaying more considered use of complexity in distinguishing Transaction Functions until Step B3.

The completion of the software sizing process outlined in this section was accomplished in two distinct halves. Each of the NESMA methods were used on the first five projects (‘A’ to ‘E’ in the results tables). During the subsequent Research Phases, the dataset was expanded to incorporate an additional six projects. The software sizing process using the NESMA methods was repeated for these projects (‘F’ to ‘K’) and the analysis of the results refined using the complete sizing results.

3.4.4.3 Research Stage 3 - Functional Sizing Limit Analysis

Functional sizing provides an estimate of the size of a project, but there is a degree of uncertainty associated with any estimate. Given this uncertainty, there may be a limit on how ‘accurate’ size estimates can reasonably be expected to be, and therefore how specifically the estimation results should be interpreted. This research stage considers two alternative classification schemes for the functional sizes of projects that correspond to a broader interpretation with lower expectations of accuracy. Such classifications may facilitate a greater level of consistency in the estimation results smoothing out inaccuracies inherent in software sizing.

The first classification approach is Rank Ordering, where projects are inserted into an evolving classification of the company projects based on the estimated functional size. Projects can therefore have their effort and cost estimated based on their position within the overall rank order. The second classification approach is Size Banding, where the estimated functional size for a project is mapped into the respective size band. These size bands could then be used as the basis for deriving an estimate of the effort/cost.

For this research, the focus is on the use of these classification schemes with the NESMA sizing results produced in Research Stage 2, and by the organisation project data. The following steps are involved in devising the Rank Ordering:

(1) For the Bid Stage Indicative NESMA sizing results arrange the projects into order of overall functional size, from largest down to smallest.
(2) Repeat step (1) for the Estimated NESMA method.
(3) For the Functional Specification Stage Indicative NESMA sizing results arrange the projects into order of overall functional size, from largest down to smallest.
(4) Repeat step (3) for the Estimated and Full NESMA methods.
(5) For the Equiniti-ICS Bid stage cost estimate figures arrange the projects into order of estimated cost, from largest down to smallest.
(6) Repeat step (5) for the Actual Effort and Commercial Value figures for each completed project.

The rank orderings produced in these steps are added to the Management Ranking of the projects provided by their personnel. This enables the rank ordering produced by the NESMA methods to be evaluated against those produced by the organisation project data.

For the Size Banding classification scheme there is no equivalent Equiniti-ICS data suitable for providing a comparison. This aspect of the research is therefore limited to comparing the Size Banding results across both stages by the NESMA methods. The definition of appropriate size bands is dependent upon the range of functional sizes demonstrated within the research dataset. The specific size bands are therefore not selected until the NESMA sizing results are produced in the previous research stage. In principle, each size band would have an upper and lower functional size to define the boundaries of the band. For example, the lowest size band could be from 0 FP to 300 FP. The following steps are involved in producing the Size Band classification results:

(1) For the Bid Stage Indicative NESMA sizing results place each project into the size band range that its overall functional size falls within.
(2) Repeat step (1) for the Estimated NESMA method.
(3) For the Functional Specification Stage Indicative NESMA method sizing results place each project into the size band range that its overall functional size falls within.
(4) Repeat step (1) for the Estimated and Full NESMA methods.

3.4.4.4 Research Stage 4 - Functional Sizing Process Metrics

The decision to develop a Functional Size estimate for any project necessitates the allocation of staff and time to complete the sizing process. This, in turn, requires an understanding of the estimation effort that will be required to develop the size estimate. This understanding can be acquired from experience of previous projects, and a judgement of how the new project compares to them. In effect, the 'size' of the project may enable judgement of the effort required to develop a formal size estimate. The assessment of the value of Functional Sizing in this Research Phase involves the use of metrics related to the Functional Sizing process.

The previous Research Stage provides the Functional Size and associated estimation effort for each project. This provides a basis for assessing whether the Functional Size of a project is related to the estimation effort required. In addition, the size of the requirements documentation used to develop the estimates provides a simple metric for providing an indication of the size of a project. This metric is also evaluated as an indicator of the estimation effort that would be required for a Functional Size Estimate. The precise Functional Sizing Process Metrics used are as follows:

Function Count: Corresponds to the Functional Size of a project, determined at both the Bid stage and the Functional Specification stage. In order to compare this metric at the two stages the same NESMA method was selected to provide the Functional Size. The Estimated NESMA method, as the most detailed method used in both stages, is the most appropriate choice.

Estimation Effort: Corresponds to the effort required to develop each of the size estimates using the Estimated NESMA method. Each Function Count metric therefore has a corresponding Estimation Effort metric.

Page Count: Corresponds to the number of pages used to describe the required functionality for each size estimate. It therefore only includes pages relevant to informing the size estimate, rather the overall page count of the requirements documentation. Each project will therefore have a Page Count metric for both the Bid stage and Functional Specification stage.

Statistical analysis was then performed to determine the significance, if any, of the relationship between these metrics at both the Bid stage and the Functional Specification stage.

- The correlation between each combination of page count, function count and estimation effort was assessed using the Pearson correlation coefficient r.

    - This measurement returns a result ranging from -1 (perfect negative correlation) to +1 (perfect positive correlation) for the variables being examined. A result of 0 would indicate an absence of any correlation between the variables.

- Visual representation of the correlation used to verify the results of the correlation calculations.
    - As the Pearson correlation measurement is concerned with linear correlation, scatter graphs were plotted to obtain a visual representation of the relationship between these variables.
    - A line of best fit is included in each graph to illustrate the results of the respective linear correlation calculation.
    - The strength of the relationship, R-squared, is included on each graph.

- The correlation coefficients obtained were tested for statistical significance in order to confirm that they did not occur by chance.
    - The *a priori* hypothesis of this correlation analysis was that positive correlations would exist between each of these variables, so the significance level assessment was performed using a One Tailed Test.
    - Any significance level of five percent ($p<0.05$), or lower, is considered to be statistically significant.

The effectiveness of the Page Count metric can be further evaluated by analysing the strength of the relationship with the subsequent development effort and the commercial value of projects. The Statistical Analysis outlined above was therefore repeated for these additional project characteristics. The results of this analysis were compared with those produced using the Function Count metric in place of the Page Count metric. This provides an indication of the effectiveness of using a simple size metric in place of a formal size estimate.

3.4.4.5 Research Stage 5 - Accumulation of Company Feedback on Software Sizing

The company feedback on the use of software sizing was completed using a structured interview session. The content to be addressed in this interview session was prepared and forwarded to Equiniti-ICS in advance of the interview. This enabled the Technical Director to select appropriate staff to attend the interview, and for feedback to be formulated prior to the meeting.

The preceding Research Stage produced functional sizes for each of the sample projects using each of the NESMA methods. In addition, the amount of estimation effort required when using each method on each project was recorded. These results were then combined with the organisation project data and provided to Equiniti-ICS for their interpretation of the results. In order to assess the value of software sizing, investigation into the existing estimation practices of Equiniti-ICs was an integral component of the feedback process. The potential for integrating software sizing into the existing development practices was discussed, providing an evaluation of the feasibility of software sizing contributing to the company.

The following questions, concerning the main sizing issues (SI), were presented to the company in preparation for the interview session.

(SI1) How agreement is reached between a project manager and the senior manager regarding effort/schedule and how could this estimation data help with that negotiation?

(SI2) The accuracy of the sizing results for each level of evaluation –

a. At which level of sizing method are acceptable results provided?
b. How do the results of our sizing activity compare with manager's perceptions of the size, scale, effort for each project?

(SI3) The amount of effort required to complete the sizing tasks –

a. How does it compare to the amount of time taken to develop using expert judgement?
b. Would the required effort represent an unacceptable delay during the planning stages of such projects?

(SI4) The usefulness of the sizing data produced by each method –

a. Could it contribute to developing project plans or providing customer quotations?
b. Could the sizing data be more effectively presented?
c. Would it provide a better basis for the subsequent effort estimation and scheduling activities?
d. What percentage of the overall project tasks could be influenced by the software sizing results?

(SI5) What further value/benefit could be derived from the use of these techniques? For example -

a. Recognition of requirements change and the impact of this change on plans and schedules.
b. Improved identification and assessment of potential project risks.

The feedback obtained from the interview session facilitates an assessment of the cost/benefit trade-off for each of the NESMA methods. An indication of where the potential value of software sizing can be found serves to direct the subsequent phases of this research.

3.5 Results

The results of the functional sizing for the commercial projects are presented alongside the organisation project data in the initial section. In the next section these results are analysed in order to compare the performance of the NESMA methods with that of the current estimation approach used by Equiniti-ICS. An evaluation is then provided of how effective functional sizing is in a commercial project context. The results section then concludes with an analysis of the value that functional sizing can provide to the software development process.

3.5.1 Project Sizing Results

Table 19 presents the organisation project data for all of the eleven projects (‘A’to‘K’), including the results of independent retrospective analysis by the Senior Management team at Equiniti-ICS (the ‘Management Ranking’ column). The ‘Expert Opinion Estimate’ column shows the rankings of the original estimates that were used in preparing each project Bid, and the subsequent project planning activities. The ‘Actual Effort’ column shows the rankings by the actual development effort expended for each project. The ‘Commercial Value’ column shows the rankings by the derived commercial value of each project to the company. The commercial value of each project is defined as the amount of revenue that the organisation has derived from the respective project.

Comparison of the rankings indicates that they are broadly similar across the different categories. However, Project H in particular indicates an example of where relationships between these categories can be uncertain. The commercial value of the project fell significantly below the initial expert estimate, while the actual effort exceeded the estimate. The commercial value column demonstrated the most variation against the other categories, suggesting that it is less dependent on project measures such as effort or size. The use of functional sizing may provide a contrasting view of these relationships across the projects.

Table 19 – Internal Company Project Rankings

Rank Order	Management Ranking Size(Survey)	Expert Opinion Estimate	Actual Effort	Commercial Value
Largest (1st)	J	J	J	J
(2nd)	C	K	K	K
(3rd)	K	G	H	C
(4th)	B	C	C	B
(5th)	E	B	B	G
(6th)	G	H	E	F
(7th)	H	E	D	I
(8th)	A	D	G	E
(9th)	D	A	A	D
(10th)	I	F	I	A
Smallest (11th)	F	I	F	H

Table 20 presents the results of the sizing of each of the eleven projects using each of the three NESMA methods at both the Bid and Functional Specification stages. For all projects, with the Indicative NESMA, it was assumed that all data was in a non-normalised form at the Bid stage and normalised form at the Functional Specification stage.

Table 21 presents the corresponding estimation effort (in person-hours) to complete each of the size estimates shown in Table 20.

Table 22 presents the rank-ordering of the eleven projects, comparing the organisation project data with the functional sizing estimates for each of the NESMA methods.

Table 23 presents the results of the size banding analysis on each of the NESMA methods. The figures in brackets represent the number of bands difference from the results predicted using the Full NESMA method. For example, B (-2) means that the project was placed by the respective sizing method two bands lower than with the Full NESMA method.

Table 24 presents the overall performance in terms of estimation effort and estimation accuracy for the distinct NESMA methods.

Table 25 presents the Functional Sizing Process metrics for each project at both the Bid stage and Functional Specification stage.

Table 26 presents the correlation coefficient *r*, and associated significance level (if any), for (i) page counts and estimation effort, (ii) function counts and estimation effort and (iii) page counts and function counts, using both Bid and Functional Specification requirements documents.

Table 27 presents the correlation coefficient *r*, and associated significance level (if any), for (i) function counts and development effort, (ii) page counts and development effort, (iii) function counts and commercial value, and (iv) page counts and commercial value, using both Bid and Functional Specification requirements documents.

Table 20 – Total Functional Size Estimates

Size Method	Total Functional Size (FP)																					
	Project A		**Project B**		**Project C**		**Project D**		**Project E**		**Project F**		**Project G**		**Project H**		**Project I**		**Project J**		**Project K**	
	Bid	FS	Bid	FS	Bid	FS	Bid	FS	Bid	FS	Bid	FS	Bid	FS	Bid	FS	Bid	FS	Bid	FS	Bid	FS
Indicative NESMA	695	430	975	765	1205	1295	770	510	890	685	225	235	560	585	900	565	490	460	1030	1360	980	1355
Estimated NESMA	605	647	1329	1355	1674	1994	612	970	635	854	149	237	500	731	848	985	309	409	751	1434	1045	1960
Full NESMA	-	629	-	1405	-	1892	-	933	-	806	-	218	-	776	-	961	-	429	-	1503	-	1931

Table 21 – Estimation Effort (person-hours) for each Project/Method

Size Method	Total Estimation Effort (Hours)																					
	Project A		**Project B**		**Project C**		**Project D**		**Project E**		**Project F**		**Project G**		**Project H**		**Project I**		**Project J**		**Project K**	
	Bid	FS	Bid	FS	Bid	FS	Bid	FS	Bid	FS	Bid	FS	Bid	FS	Bid	FS	Bid	FS	Bid	FS	Bid	FS
Indicative NESMA	4	2	7	4	5	6	5	4	3	2	0.5	1	1	1	2	1.5	0.5	1.5	2	4	3	5
Estimated NESMA	9	6	16	9	10	16	13	12	6	7	1.5	2.5	3	5.5	5	5.5	2	3.5	5	15	8	16
Full NESMA	-	14	-	25	-	40	-	21	-	17	-	4.5	-	14	-	11.5	-	7.5	-	31.5	-	39

3.5.2 Project Size Analysis

The analysis of the project size measures first considers the use of Rank Ordering, and then Size Banding, as alternatives to using the specific overall functional size for a project. The overall performance of the NESMA methods, in terms of estimation effort and estimation accuracy, is then analysed. Correlation analysis is then presented, firstly for functional sizing process metrics, and then for overall project metrics.

3.5.2.1 Rank Ordering

The level of agreement between the different approaches to ranking the sample projects was analysed using Kendall's Coefficient of Concordance, referred to as Kendall's W. This analysis returns a value for W between 0 and 1, with 0 representing perfect disagreement and 1 representing perfect agreement between the different ranking approaches. For the Equiniti-ICS rankings presented originally in Table 19, a W value of 0.805 was obtained that indicated a strong level of agreement between the different approaches. Analysing the rankings in Table 22 produced only by the NESMA sizing methods produced a W value of 0.931, indicating a very strong level of agreement between these approaches. The results for both of these Kendall's W tests were found to be statistically significant, with $p < 0.001$. In terms of analysing the rank order produced by individual approaches, each of them was evaluated against the rank order produced by the actual development effort. This analysis involved the use of Spearman's rank correlation coefficient (referred to as Spearman's rho in Table 22). This provided a measure of the level of agreement between the Actual Effort rank order and that of each individual approach, expressed as a value between -1 and 1. The results in Table 22 show that the rank order produced by most approaches demonstrated a strong, statistically significant, correlation with the Actual Effort rank order. The NESMA methods slightly outperformed the Equiniti-ICS approaches in this regard, with correlations of over 0.9 demonstrated at both the bid stage and functional specification stage. Overall no single approach, or stage of estimation, demonstrated superior rankings in comparison to the Actual Effort ranking for the projects.

This statistical analysis supports the feasibility of establishing a rank ordering of projects using NESMA sizing methods, but further consideration of individual projects is required to assess the suitability of such rankings. The Bid Stage rankings in Table 22 for both the Indicative and Estimated NESMA methods are quite similar to those produced by the Expert Opinion estimates, although Project G was ranked significantly lower by the NESMA methods. The Actual Effort ranking is more consistent with the functional size estimates for this project, as well as for Project H which had been noted previously to have been in contrast to the Equiniti-ICS assessment. For Project J the more comprehensive Estimated NESMA method produced a notably less accurate ranking than the simpler Indicative NESMA method.

Consideration of the specific functional sizes estimated for the project reveals some limitations on the use of rank ordering. Project J is ranked first within the Equiniti-ICS metrics, and third by the NESMA methods at the Functional Specification stage. On the surface this does not appear to be a significant difference, but Project J was estimated to be around 500 FP smaller than Projects C and K.

3.5.2.2 Size Banding

The Size Banding results in Table 23 indicate that the Indicative NESMA method will generally underestimate the final Functional Size by between one and two size bands (range of 200 FP). The average difference actually demonstrated an increase for the later Functional Specification estimate, although the standard deviation reduction indicates less volatility at this stage. The Estimated NESMA method demonstrated similar average performance at the Bid Stage to the Indicative method, although this was more influenced by a few significant underestimates within the dataset. At the Functional Specification stage there was only one project which was not placed in the same Size Banding by the Estimated NESMA method. In order to test the level of agreement between the Full NESMA method and the other NESMA methods, a linearly weighted version of Cohen's Kappa was used. This analysis returns a value for Kappa of 1 when there is perfect agreement between the two approaches, and 0 when there was no agreement beyond that of random chance. Using a weighted version of Cohen's Kappa enables the level of disagreement to be considered i.e. how many size bands any two estimates differ by for the same project. The results in Table 23 show that statistically significant results were obtained in each case, but this is not uncommon for even relatively low values of Kappa. Most of the values obtained for Kappa in Table 23 were close to 0.5, indicating that the level of agreement was only moderate. The Estimated NESMA method, with a Kappa value of 0.968 at the Functional Specification stage, did however demonstrate a very strong level of agreement with the Full NESMA method.

Overall, it can be seen that at the Bid Stage even identifying the correct Size Banding was not effective for some projects. More reliable results can be achieved at the Functional Specification stage, as would be expected from the greater requirements detail available. It should be noted that in no case did either the Indicative or Estimated methods place a project in a higher Size Band than the Full NESMA method.

3.5.2.3 Overall Sizing Performance

Table 24 summarises the relative estimation effort and relative accuracy of the NESMA sizing methods, using the Full method as the baseline comparison. Results are included for comparison from published research involving the use of each of the NESMA methods. The Functional Specification stage results only can be included in this comparison, as the Full NESMA method could not be performed on the Bid stage documentation.

The reduction in estimation effort was only reported previously in the study by van Heeringen et al. (2009). For the average reduction in estimation effort relative to the Full NESMA method, similar results for the Indicative method were found in this research and in the van Heeringen et al. study. Achieving a relative reduction of over 80% represents a substantial benefit of the Indicative method. The Estimated method demonstrated greater difference between these two research datasets. In van Heeringen et al. the Estimated method demonstrated an average reduction in Estimation effort of 33%, compared to 55% for the dataset in this research phase. The van Heeringen et al. study included projects from multiple development organisations, with no indication provided of the relative level of detail of the project documentation within these organisations. The greater saving in estimation effort found in the eleven projects in this research phase may be due to the nature of the documentation used to develop the estimates.

Similar average relative errors were obtained for these research results, at 26.30% and 28.44% respectively. Cândido and Sanches (2004) reported an even higher average relative error for Indicative method at 48%. The difference in that study can be explained to some extent by the relatively low complexity of the required functionality of the applications studied. The relative inaccuracy of the Indicative method is likely to limit the confidence in the use of this method. The Estimated method, in contrast, demonstrated a more acceptable performance relative to the Full method. For this research dataset, and the van Heeringen et al. dataset, the average relative error was around 5%, which is more likely to be considered sufficient for practical use. The relatively simple applications used in the Cândido and Sanches study resulted in a more substantial average relative error of 18%. This emphasises the issue of the suitability of applying simplified approaches to Functional Sizing for different application types.

Table 22 – Rank-ordering of the sampled projects by organisation and calculated size

					Bid Ranking		**Functional Specification Ranking**		
Rank Order	Management Ranking (Survey)	Expert Opinion (Estimate)	Actual Effort	Commercial Value	Indicative NESMA	Estimated NESMA	Indicative NESMA	Estimated NESMA	Full NESMA
1	J	J	J	J	C	C	J	C	K
2	C	K	K	K	J	B	K	K	C
3	K	G	H	C	K	K	C	J	J
4	B	C	C	B	B	H	B	B	B
5	E	B	B	G	H	J	E	H	H
6	G	H	E	F	E	E	G	D	D
7	H	E	D	I	D	D	H	E	E
8	A	D	G	E	A	A	D	G	G
9	D	A	A	D	G	G	I	A	A
10	I	F	I	A	I	I	A	I	I
11	F	I	F	H	F	F	F	F	F
Spearman's rho (Significance Level)	0.855 (p<0.01)	0.827 (p<0.01)	-	0.464 (Not Significant)	0.918 (p<0.001)	0.827 (p<0.01)	0.882 (p<0.001)	0.909 (p<0.001)	0.927 (p<0.001)

Table 23 presents the banding results for the 11 projects based upon using a granularity of 200 function points. Where: B1 0-200 fps; B2 201-400 fps; B3 401-600fps; B4 601-800 fps; B5 801-1000 fps; B6 1001-1200 fps; B7 1201-1400 fps; B8 1401-1600 fps; B9 1601-1800 fps and B10 1801-2000 fps.

Table 23 – Sampled projects categorised by size-banding (at 200 fps)

Project	**Indicative NESMA**		**Estimated NESMA**		**Full NESMA**	
	Bid	FS	Bid	FS	Bid	FS
A	B4 (0)	B3 (-1)	B4 (0)	B4 (0)	-	B4
B	B5 (-3)	B4 (-4)	B7 (-1)	B7 (-1)	-	B8
C	B7 (-3)	B7 (-3)	B9 (-1)	B10 (0)	-	B10
D	B4 (-1)	B3 (-2)	B4 (-1)	B5 (0)	-	B5
E	B5 (0)	B4 (-1)	B4 (-1)	B5 (0)	-	B5
F	B2 (0)	B2 (0)	B1 (-1)	B2 (0)	-	B2
G	B3 (-1)	B3 (-1)	B3 (-1)	B4 (0)	-	B4
H	B5 (0)	B3 (-2)	B5 (0)	B5 (0)	-	B5
I	B3 (0)	B3 (0)	B2 (-1)	B3 (0)	-	B3
J	B6 (-2)	B7 (-1)	B4 (-4)	B8 (0)	-	B8
K	B5 (-5)	B7 (-3)	B6 (-4)	B10 (0)	-	B10
Average Difference	-1.36	-1.64	-1.36	-0.09	-	-
Standard Deviation	1.69	1.29	1.36	0.30	-	-
Weighted Cohen's Kappa (Significance Level)	0.463 (p<0.001)	0.428 (p<0.01)	0.527 (p<0.01)	0.968 (p<0.001)	-	-

Column →	P	Q	R	S	T	U

Table 24 – Comparison of Overall NESMA Sizing Results

Estimation Method	**Average Effort Reduction (%)**	**Average Relative Error (%)**
Functional Specification Stage		
Indicative NESMA	85%	26.30%
Estimated NESMA	55%	4.50%
Full NESMA	-	-
van Heeringen et al. (2009)		
Indicative NESMA	82%	28.44%
Estimated NESMA	33%	5.73%
Full NESMA		-
Cândido and Sanches (2004)		
Indicative NESMA	-	48%
Estimated NESMA	-	18%
Full NESMA	-	-

3.5.2.4 Correlation Analysis for Functional Sizing Process Metrics

The Functional Sizing Process metrics used for this analysis are shown in Table 25. The values for the Function Count and Estimation Effort are taken from the use of the Estimated NESMA method in each case.

Table 25 – Summary information from Bid and Detailed Documentation

Project	**Bid Stage**			**Functional Specification Stage**		
	Page Count	Function Count (FP)	Estimation Effort (Hours)	Page Count	Function Count (FP)	Estimation Effort (Hours)
A	55	605	9	110	647	6
B	30	1329	16	160	1355	9
C	90	1674	10	210	1994	16
D	30	612	13	145	970	12
E	15	635	6	120	854	7
F	2	149	1.5	75	237	2.5
G	25	500	3	260	731	5.5
H	40	848	5	150	985	5.5
I	7	309	2	70	409	3.5
J	20	751	5	550	1434	15
K	25	1045	8	340	1960	16

The scatter graphs for each tested relationship are shown in Figures 3.2 to 3.7, enabling a visual inspection of the individual data points. The specific correlation, r, and the associated statistical significance of each relationship are shown in Table 26.

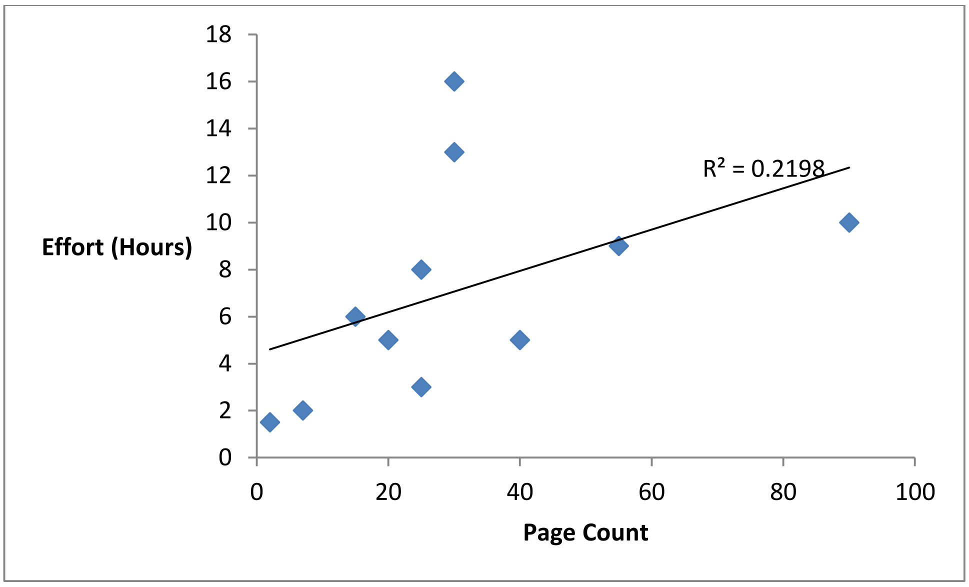

Figure 3.2 - Scatter graph for Page Count and Estimation Effort at the Bid stage

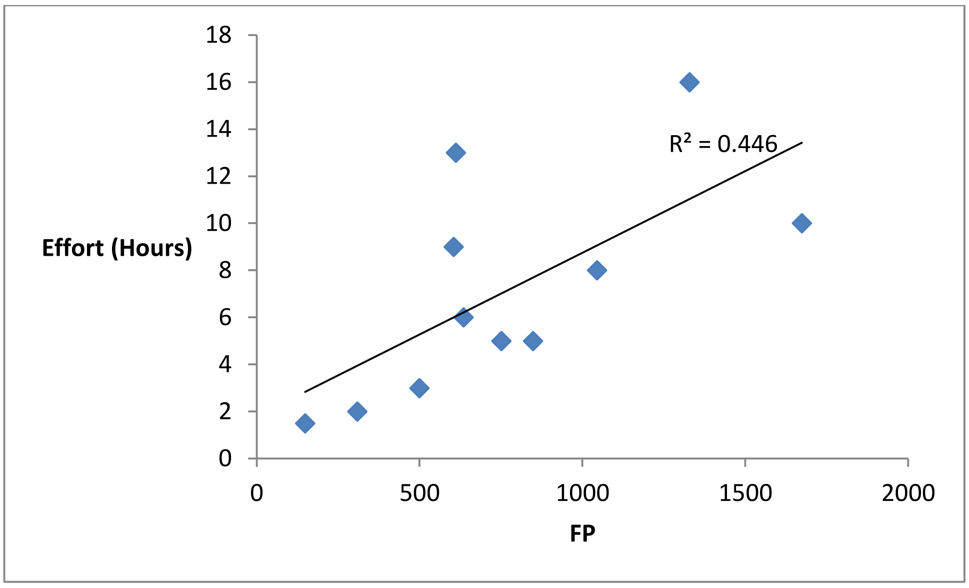

Figure 3.3 - Scatter graph for Function Count and Estimation Effort at the Bid stage

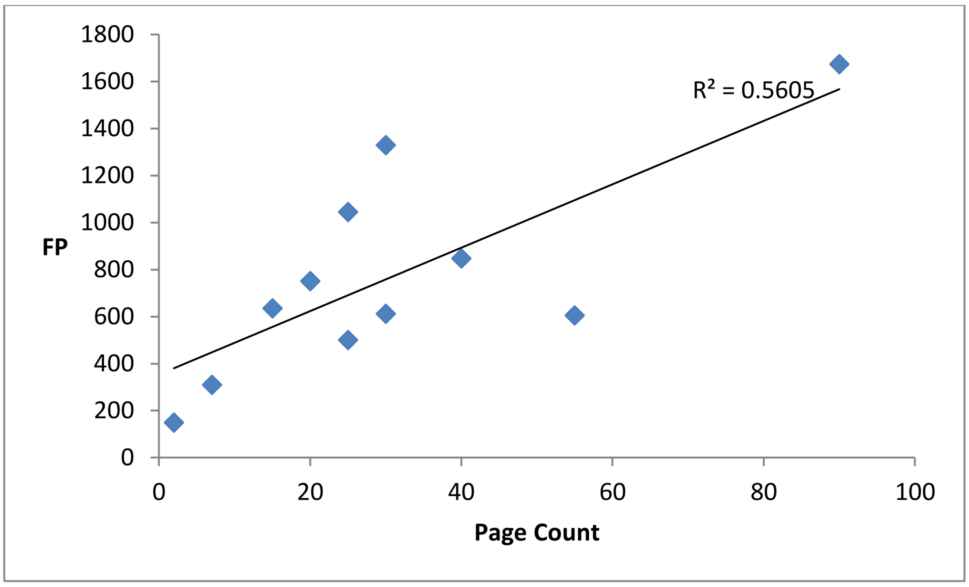

Figure 3.4 - Scatter graph for Page Count and Function Count at the Bid stage

Figure 3.2 shows a moderate correlation between Page Count and Estimation effort at the Bid stage, but this is not statistically significant. In contrast, Figure 3.3 shows a statistically significant correlation (p<0.025) between Function Count and Estimation Effort at the same stage. The correlation between Page Count and Function Count demonstrated an even higher level of statistical significance (p<0.005). In this case the scatter graph in Figure 3.4 illustrates that the project with the largest Page Count excerpts excessive influence over the result. The result of removing this larger Page Count from the dataset is a moderate correlation that is not statistically significant.

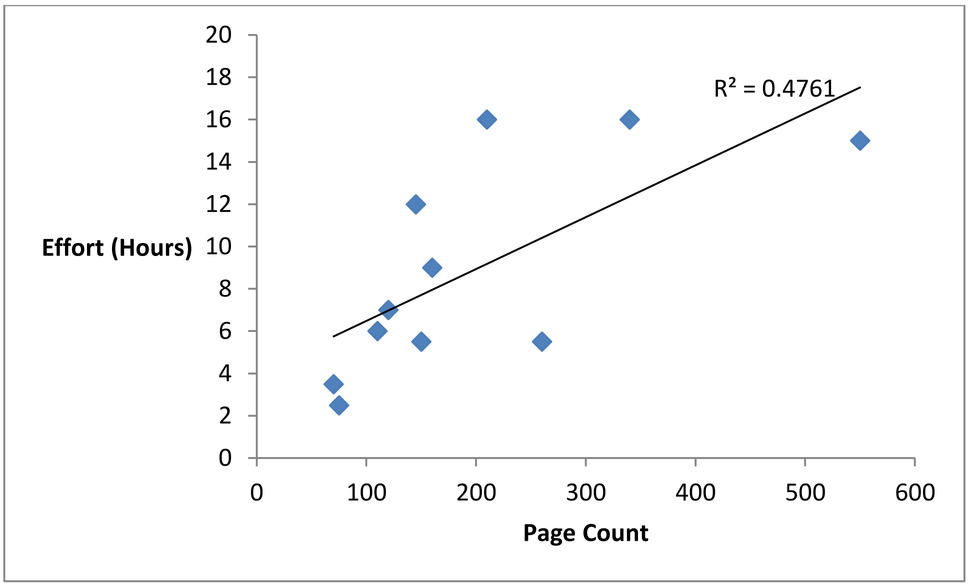

Figure 3.5 - Scatter graph for Page Count and Estimation Effort at the Functional Specification stage

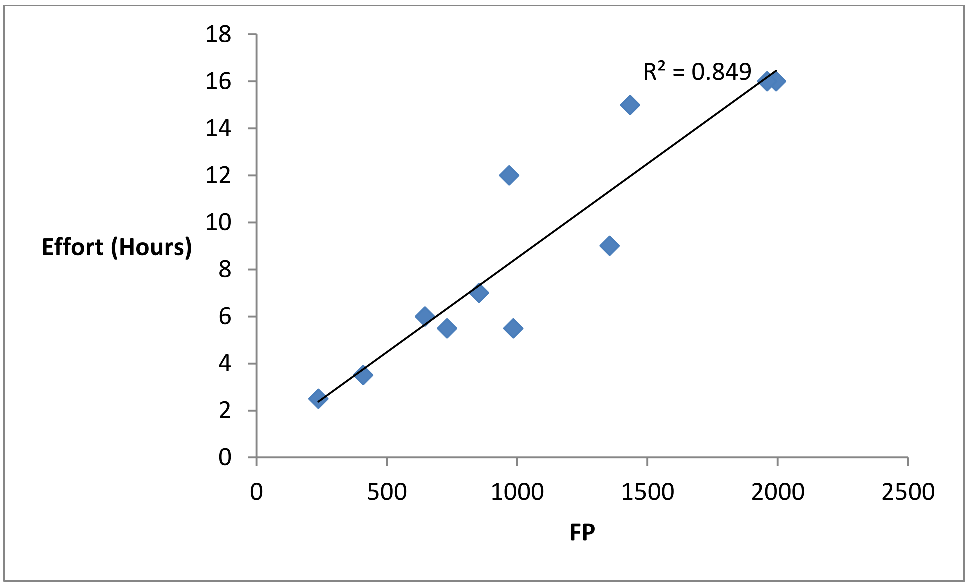

Figure 3.6 - Scatter graph for Function Count and Estimation Effort at the Functional Specification stage

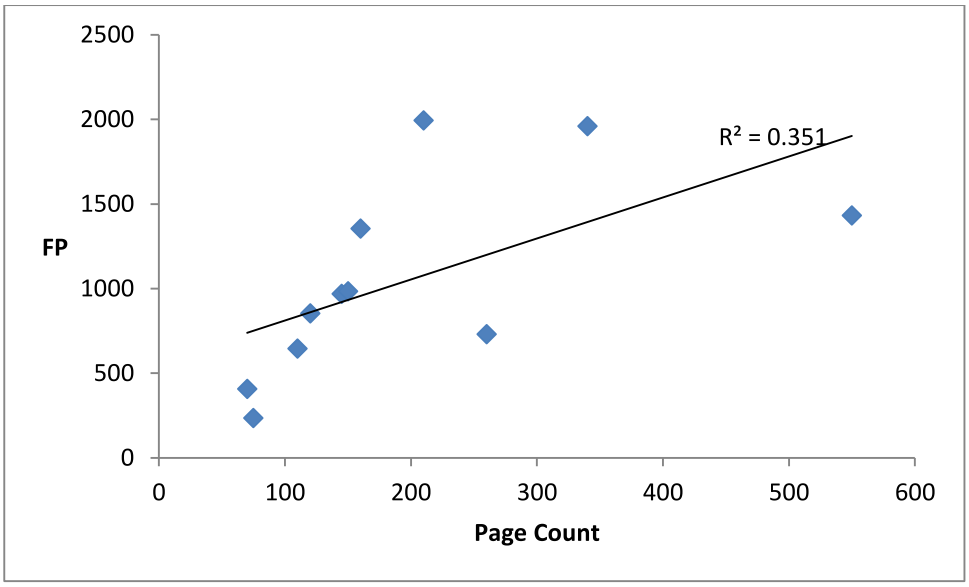

Figure 3.7 - Scatter graph for Page Count and Function Count at the Functional Specification stage

In contrast to the results at the Bid stage, the remaining Figures (3.5 to 3.7) illustrate that stronger linear correlations generally appear to exist at the Functional Specification stage. The correlation between Page Count and Estimation Effort is now statistically significant (p<0.01) at the Functional Specification stage. The correlation between Function Count and Estimation Effort is more highly statistically significant at this later stage (p<0.005). Finally, the correlation between Page Count and Function Count is lower compared to the Bid Stage.

However, the correlation is statistically significant ($p<0.05$) at the Functional Specification stage, and the scatter graph in Figure 3.7 does not indicate any outlier results. The relationship between these two metrics can therefore be considered to be stronger at the Functional Specification stage.

Table 26- Statistical Correlation Analysis of Size Estimation Process

Stage (Measure)	**Correlation (*r*)**	**Significance *p*-value**
Bid (Page Count/Estimation Effort)	0.469	*p*=0.073
Bid (Function Count/ Estimation Effort)	**0.668**	***p*=0.012**
Bid (Page Count/ Function Count)	**0.749**	***p*=0.04**
Functional Specification (Page Count/Estimation Effort)	**0.690**	***p*=0.009**
Functional Specification (Functional Count/ Estimation Effort)	**0.921**	***p*<0.001**
Functional Specification (Page Count/ Function Count)	**0.592**	***p*=0.027**

3.5.2.5 Correlation Analysis with Overall Project Metrics

Table 27 shows the results of the correlation analysis concerning the Development Effort and the Commercial Value of the projects. Consideration of both the Function Count and the Page Count at the Bid stage did not produce a statistically significant correlation in any case. Only the Function Count and Commercial Value of projects demonstrated a moderate correlation. At the Functional Specification stage there were statistically significant correlations found in each case. Scatter graphs (Figures 3.8 to 3.11) are therefore provided to enable visual inspection of these relationships.

The most intriguing results found were that the simple Page Count provided a more statistically significant result with the Development Effort than the more sophisticated Function Count. Inspection of the scatter graph in Figure 3.9, however, indicates that the presence of the data points for the two largest Page Counts could potentially exaggerate the strength of this correlation. Cook’s Distance can be used to estimate the influence of an individual data point in a regression model. It measures the effect of removing a data point from the regression model, where a high Cook's Distance indicates a highly influential observation. There are different cut-off points suggested to signify highly influential data points, such as 1 or 4/n. With a dataset of 11 projects, the cut-off value for the latter suggestion would be equal to 0.36. Calculating Cook's Distance for each data point in this model indicates that only the data point for the second largest Page Count exceeds this, with a value of 0.48. Removing that data point and recalculating Cook's Distance for the remaining observations resulted in the data point for the largest Page Count returning a much larger value of 11.42. A similar effect was produced by instead just initially removing the data point for the largest Page Count rather than for the second largest, and to a lesser extent by instead removing both data points. Retaining both data points, one of which is considered to be marginally over influential, would therefore be deemed preferable in this model. For the relationship with Commercial Value, the statistical significance demonstrated was more even for the Function Count and Page Count. For both metrics the correlation was found to be highly significant ($p<0.005$), with the scatter graphs not revealing any outliers affecting the result. Some limitations should be noted regarding the models shown in these scatter graphs. In each case a negative Development Effort or a negative Commercial Value would result from function counts or page counts below certain values. The magnitude of the residuals for the observations in some graphs exhibited a pattern of increasing for larger data values, indicating some degree of heteroscedasticity. These limitations were not considered severe in terms of impacting the results, but suggest that some caution should be exercised in interpreting these results.

Table 27 - Statistical Correlation Analysis of Functional Sizing and Completed Project Data

Stage (Measure)	**Correlation (*r*)**	**Significance *p*-value**
Bid (Function Count/Development Effort)	0.294	*p*=0.190
Bid (Page Count/Development Effort)	-0.053	*p*=0.438
Bid (Function Count/Commercial Value)	0.490	*p*=0.063
Bid (Page Count/Commercial Value)	0.160	*p*=0.319
Functional Specification (Function Count/Development Effort)	**0.655**	***p*=0.014**
Functional Specification (Page Count/Development Effort)	**0.921**	***p*<0.001**

Stage (Measure)	Correlation (*r*)	Significance *p*-value
Functional Specification (Function Count/Commercial Value)	**0.794**	***p*=0.002**
Functional Specification (Page Count/Commercial Value)	**0.858**	***p*<0.001**

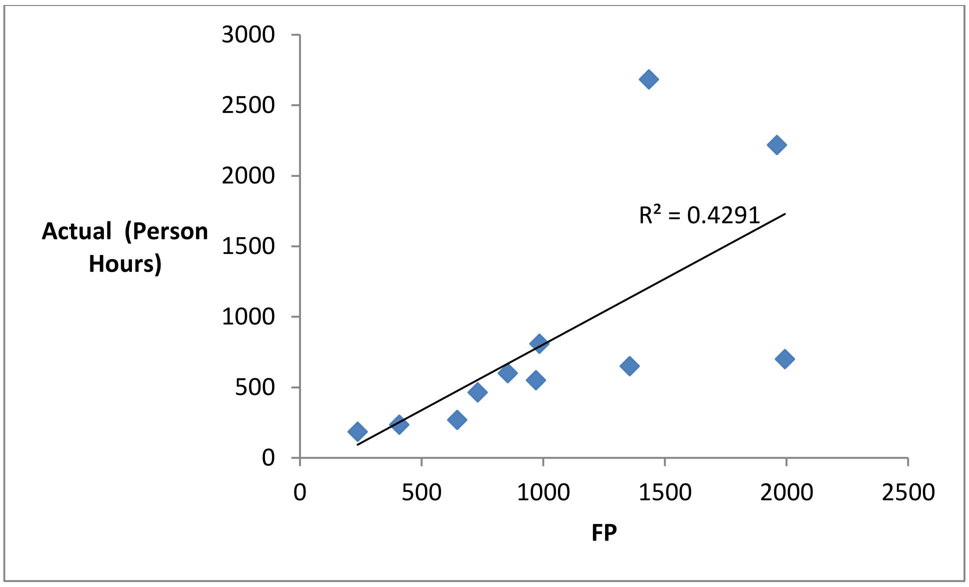

Figure 3.8 - Scatter graph for Actual Development Effort and Function Count at the Functional Specification stage

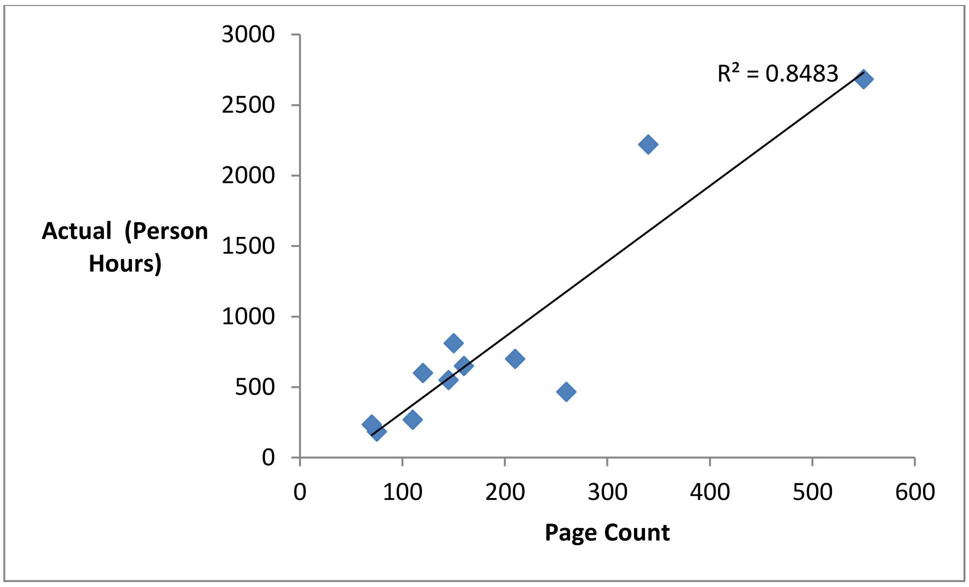

Figure 3.9 - Scatter graph for Actual Development Effort and Page Count at the Functional Specification stage

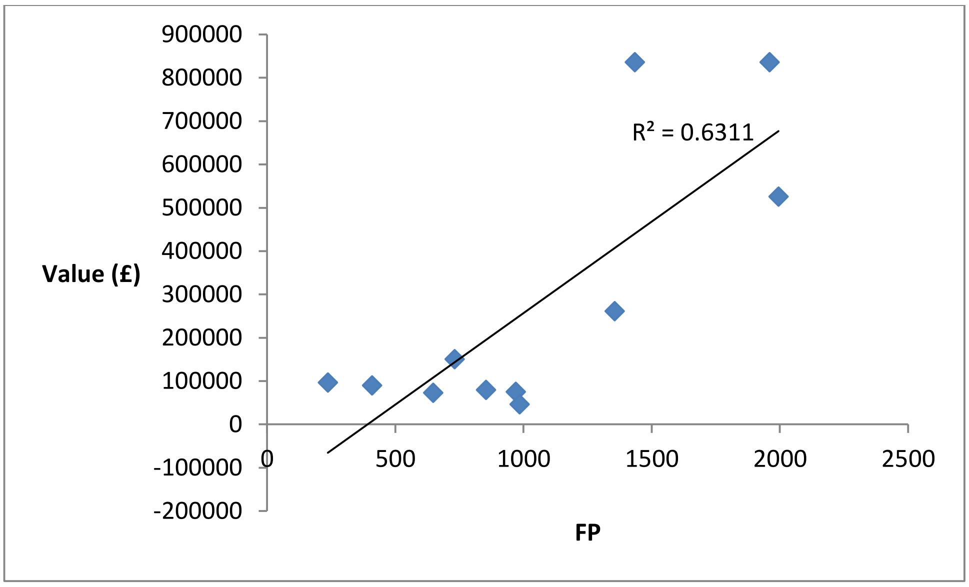

Figure 3.10 - Scatter graph for Commercial Value and Function Count at the Functional Specification stage

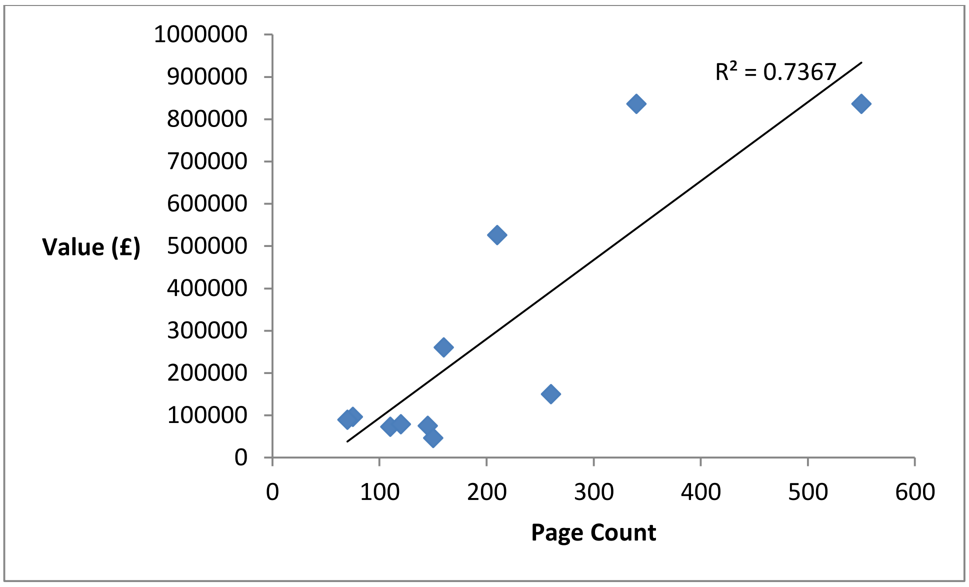

Figure 3.11 - Scatter graph for Commercial Value and Page Count at the Functional Specification stage

3.5.3 Evaluation of Results

The overall results from the use of the NESMA methods in this research are evaluated in the context of related research concerning these methods. The limits of the accuracy of the sizing results are then evaluated by considering the effectiveness of using Rank Ordering and Size Banding to classify the size estimates. The effect of the project lifecycle on the development of size estimates is then evaluated, followed by the effect of the project documentation.

3.5.3.1 Overall NESMA Method Performance

The use of different levels of rigour in software sizing can be evaluated from the results of the three NESMA methods. In terms of relative accuracy, the results of this research for the Functional Specification stage are generally consistent with previous research. The accuracy of the Indicative NESMA method, relative to the Full method, would be considered insufficient for use as the fundamental basis of detailed project decisions. The Estimated NESMA method, however, provided an acceptably small relative inaccuracy to be considered as a legitimate alternative to using the Full method. The Cândido and Sanches study highlighted the fact that applications which do not conform to typically average complexity would result in a greater level of inaccuracy for the Estimated NESMA method. The simplified versions of NESMA were found to offer clear savings in the estimation effort required in developing the size estimates. The Indicative NESMA method has been shown to provide a substantial reduction in the estimation effort required, at the cost of accuracy in the result of the estimate. The reduction achieved by the Estimated NESMA method was greater in the dataset used in this research phase, compared with the larger van Heeringen et al. dataset. In both datasets the same conclusion can be drawn, however, regarding the use of the NESMA methods. The Estimated NESMA method provides the optimal trade-off between relative accuracy and the estimation effort required. The increase in accuracy from using the Full NESMA method is insufficient to justify the additional estimation effort required in assessing the complexity of the functionality. This conclusion is, however, dependent upon the degree to which the assumption of Average complexity is appropriate for the type of applications developed by an organisation.

3.5.3.2 Limits of Software Sizing Accuracy

The potential limitations of the accuracy of software sizing were considered using alternative assessments of the size results, using Rank Ordering and Size Banding approaches.

The use of Ranking Ordering within the dataset for this research demonstrated that the NESMA sizing methods produced rankings that were generally similar, if slightly superior, to those produced using organisation data. Potential weaknesses with this approach were identified from some volatility in the rankings produced by different NESMA methods and at the different stages of estimation used. The sources of such variation include the distribution of project sizes within the dataset. Projects that are similar in size can lead to variations in the rankings produced without necessarily indicating substantial differences in actual size. Large ‘gaps’ between the sizes of adjacent projects in the rankings are not transparent simply from the ranking list, potentially obscuring substantial differences in actual size. Analysis of the larger van Heeringen et al. dataset provides an indication of whether these weaknesses remain as the size of the dataset is increased. The projects in the van Heeringen et al. dataset range from 32 FP to 5632 FP, representing a greater overall range of size within the dataset. In the absence of ‘actual’ organisation project data for the van Heeringen et al. dataset, the rankings produced by the Full NESMA method are assumed to be the ‘correct’ rankings. From the 42 projects included in the dataset, the Estimated NESMA method ranked 23 in the ‘correct’ position, with a further 11 projects at one ranking out of position. The Indicative NESMA demonstrated less similarity, with only 7 ‘correct’ projects and 6 out of order by one position. The uneven distribution in the size of the projects would be obscured by relying solely on the rank ordering. Adjacent pairs of projects could differ greatly in size range, from being identical in size to being over 2000 FP apart. The Indicative NESMA method, in particular, demonstrated some relatively large differences in rankings for individual projects, with the maximum difference being 12 positions out of place.

The use of Size Banding within the dataset of eleven projects in this research produced mixed performance from the NESMA methods. The Indicative NESMA method was inconsistent at both the Bid Stage and the Functional Specification stage, when compared with the Full NESMA method at the latter stage. The Estimated NESMA method was effective at placing the projects within the correct Size Banding at the Functional Specification. The Size Banding analysis was repeated for the van Heeringen et al. dataset, using the same granularity of 200 FP for each band and the Full NESMA method for the ‘correct’ bandings. The Indicative NESMA method placed 23 projects correctly, with 13 projects only one banding apart from the correct level. Exception projects were evident though, with the highest difference corresponding to 15 levels of banding. The Estimated NESMA method provided promising results, with 31 projects placed correctly and the remaining projects only one banding apart from the correct level.

An obvious potential weakness of the Size Banding approach is in the definition of the bands the projects are placed into. The use of a different size for each band rather than 200 FP, as used in this research, may provide different levels of consistency across the sizing methods and lifecycle stages. The most appropriate size to use may also be affected by the range of sizes present in the dataset of projects. The use of the Size Banding approach for subsequent effort estimation would allocate a single effort value for every project placed within the same size band. The level of accuracy for estimating effort in this approach would favour smaller size bands, but this undermines the concept of using the Size Banding approach. Projects that are estimated at sizes close to the boundary values for a size band are potentially subject to differences in subsequent effort estimates exceeding the relative difference in the size estimate. Smaller band sizes would also increase the volatility of a project’s banding at different stages of the lifecycle. It would therefore be necessary to investigate the use of a more flexible definition of the size bands, in contrast to using the same size range for each band. The other main limitation of the Size Banding approach is the loss of detail in terms of the exact functional size estimate produced by the estimate. Providing confidence interval limits in addition to the exact functional size would provide an equivalent indication of the range of likely sizes for a project, without sacrificing any detail in the estimate. Size bands may, however, represent a simpler, more intuitive concept for communicating to management within an organisation. The value of Size Banding would likely be as no more than as a complimentary size estimation approach given its limitations.

Overall, the Size Banding approach provides a more promising approach than Rank Ordering, particularly at the Functional Specification stage. In the Size Banding approach, the project size estimates are positioned within a fixed scale, regardless of the method and stage involved, rather than against the relatively volatile rank ordering of the set of projects.

3.5.3.3 Effect of Project Lifecycle

The development of size estimates at two distinct stages provides an indication of how the level of understanding about a project affects the assessment of the size of the project. The main indication of such an effect is found in the size estimated at each stage. The Indicative NESMA method provided no clear pattern, with increases and decreases in size obtained at the later Functional Specification stage. For the Estimated NESMA method, however, every project produced a larger size at the later Functional Specification stage. In the most extreme cases the size had almost doubled from the initial Bid stage estimate. The reasons for these patterns could be attributed to differences in how well the Data Functions and Transaction Functions can be identified at the Bid stage. Both the Indicative and Estimated NESMA methods specifically identify the Data Functions, but it is only the latter method which identifies the Transaction Functions. The difference between the two methods therefore corresponds to the difference between the assumed Transaction Functionality and the actual identified Transaction Functionality. A more detailed consideration of the nature of the Transaction Functionality would provide insight into which types of functionality are not generally identified at the Bid stage, and how they differ from the assumed Transaction Functionality.

The use of Rank Ordering demonstrated variations across the two stages, with the Functional Specification stage providing the rankings most similar to the organisation rankings. The Estimated and Full methods differed only in the ranking of the largest two projects at the later stage. For the Size Banding approach, the Estimated NESMA method at the Functional Specification provided notably reliable results across the dataset. The relationship between Functional Size and Development Effort was only found to be statistically significant at the Functional Specification stage.

The results of this analysis are consistent with related work, indicating that the more detailed view of the required functionality at a later stage in the project lifecycle enables more reliable size estimates to be developed.

3.5.3.4 Effect of Project Documentation

The effect of variance in the ‘quality’ of Project Documentation was not limited to the level of detail associated with the lifecycle stage. The Bid stage documentation demonstrated notable variation in structure and detail due to being primarily produced by the customer. The format of the documentation produced for the Functional Specification stage within Equiniti-ICS evolved from the earlier to the later projects in the dataset. There were two mains effects observed on the development of size estimates during this research.

Firstly, the estimation effort required for the projects could vary independently of the functional size of the projects. For example, the effort to analyse the ‘bid’ level documentation on project C can be seen in Table 25 to be less than the corresponding figures for projects B and D, but the functional size of project C and the page count of its project documentation are greater than those for projects B and D. The reason for this apparent anomaly is the availability of an electronic copy of the requirements documentation for project C, which facilitates tasks such as searching for specific requirement details, but only paper copies of the relevant documentation for both projects B and D. Secondly, the degree to which the Full NESMA method could be followed at the Functional Specification stage varied according to the specification of DETs involved in the Transaction Functionality. In the earlier projects this detail was less readily discernible, preventing the exact assessment of complexity at times. In such circumstances it was necessary for those Transaction Functions to be assigned the assumed complexity from the Estimated NESMA method. For the subsequent projects, the format of the Functional Specifications became more uniform, with DETs more explicitly indicated in the Transaction Functionality. The need to default to the Estimated NESMA assumed complexity was consequently less frequent for these projects. In addition, the estimation effort required to assess the complexity could be affected as it was not possible to assess a greater proportion of functions, although the assessment for individual functions was more straightforward.

3.5.4 Analysis of the Value of Software Sizing

This section presents an assessment of the value that may be derived from the use of software sizing. This analysis incorporates the feedback received from company personnel during the review of the sizing data. The current expert judgement-based estimation approach utilised by Equiniti-ICS is discussed, to provide the context for the subsequent analysis. The value of software sizing was then considered according to the three different factors affecting the use of software sizing identified in Section 3.3. These were the trade-off between benefits and costs, the degree of utility, and the compatibility of software sizing with the existing estimation practices.

3.5.4.1 Existing Expert Judgement-Based Estimation

The software development methodology employed by Equiniti-ICS has influenced the approach they adopt for effort/cost estimation. From the initial inception of a project until the completion of the Functional Specification a “waterfall”-type software development lifecycle (Royce, 1970) is utilised. The adoption of ‘Agile’ development processes after the Functional Specification stage involves the use of ‘Sprints’, where individual effort estimates for each takes the form of User Stories. The software sizing completed within this research phase using the Functional

Specification is not directly comparable to the current company practices, but may be considered as an input to later stages in the project lifecycle.

The most significant estimation undertaken within Equiniti-ICS occurs at the initial Bid stage of a project. The effort/cost estimates are produced independently by two business analysts, typically using the Request for Tender documentation provided by the customer. These estimates incorporate an assessment of the proportion of the core company product that could be reused as part of the new required system. The senior manager then arbitrates between the results to arrive at a balanced final cost for the project. The cost estimate is used to inform the bid process, but other commercial concerns also influence any chosen bid amount. The result of this process must balance the need to submit a competitive bid price with the need for that price to be economically feasible.

3.5.4.2 Benefit/Cost Trade -off

A summary of the benefits and costs of using the different NESMA methods is provided in Table 28. The benefits are concerned with the accuracy of the estimates and the output detail provided by the sizing method. The costs are concerned with the effort required to develop the estimate, and the necessary requirements detail to facilitate the use of the sizing method.

The assumption of this aspect of the research is that the Full NESMA result represents the ‘correct’ Functional Size for a project. The Estimated NESMA method can therefore be considered to provide an acceptable level of accuracy, with an average relative accuracy of around 95%. The relative accuracy of the Indicative NESMA method would be unacceptable, in comparison, if the objective of the estimate was to provide the most accurate functional size. The post sizing interview session with Equiniti-ICS addressed the issue of the performance of the NESMA methods (SI2).

In terms of the sizing data output provided, the Estimated and Full NESMA methods provide an identical level of detail. The Indicative NESMA method only provides the overall functional size, which as discussed in Section 3.5.4.2 restricts the degree of utility of this method.

The main constraint to adopting the use of software sizing is the estimation effort that is required with these methods. The issue of how acceptable the estimation effort of the NESMA methods was to Equiniti-ICS was addressed in the interview session (SI3). The Full NESMA method requires more effort than the existing Expert Judgement-based approach utilised by Equiniti-ICS. The Estimated NESMA method reduces the effort of the sizing process by around 55%, and was considered by the management to be broadly equivalent to the Expert Judgement approach currently used. The Indicative NESMA method required the least effort of the sizing methods, and represented a lower effort than the Expert Judgement approach. The conclusion of the company management was that performing the Estimated NESMA would be quite acceptable within their organisation. It was emphasised, however, that software sizing would not replace their existing approach to estimation. The use of software sizing could be used to confirm their estimation analysis, with the additional time required considered to be an acceptable overhead. An important distinction could be observed in the effect of project size on the estimation effort required.

Table 28 – Benefits and Costs of each NESMA method

	Indicative NESMA	Estimated NESMA	Full NESMA
Benefits			
Estimation Performance (Accuracy Relative to Full NESMA)	73.70%	95.50%	100%
Estimation Output Detail	Overall Functional Size	Overall Functional Size, Individual BFC Sizes	Overall Functional Size, Individual BFC Sizes
Costs			
Estimation Effort	85% less than Full NESMA, Less effort than Expert Judgement	55% less than NESMA, Broadly Equivalent to Expert Judgement	More substantial than Expert Judgement
Requirements Detail	Data Functions	Data Functions, Transaction Functions	Data Functions, Transaction Functions, Individual Data Elements

For the Expert Judgement approach, the amount of time allocated to developing the estimate was relatively inflexible. The effort required to use the NESMA methods could, in contrast, be observed to differ significantly in line with the size of the projects. This may be a reflection of the circumstances in which the estimates were developed.

The Expert Judgement estimates were developed in a commercial environment where time is a finite resource to be allocated to the task. The use of the NESMA sizing methods in this research was not constrained in this manner, enabling the sizing guidelines to be applied completely on the requirements documentation. In practice, the use of software sizing in a commercial environment may have to factor in a 'rough' assessment of project size before committing to the use of a specific sizing method.

The requirements detail considered increases across the different NESMA methods, as one method essentially extends the preceding method. The Indicative NESMA method is only concerned with the Data Functions, while the Estimated NESMA method additionally considers the Transaction Functions. The Full NESMA requires a more detailed description of these functions in terms of the DETs involved. The Full NESMA method incurs the greatest cost, therefore, in terms of requirements detail that must be documented prior to the development of a size estimate.

Weighing the costs against the benefits suggests that the optimal balance is found with the Estimated NESMA method. The additional benefits provided by the Full NESMA method do not offset the additional costs that are incurred. The minimum costs incurred with the Indicative NESMA method are unlikely to be considered sufficient to offset the lower benefits that are accrued in comparison.

3.5.4.3 Degree of Utility

The potential utility of the NESMA sizing methods is summarised in Table 29, with an indication provided of how the sizing output required determines which NESMA methods supports each respective use of software sizing.

Table 29 – Utility of NESMA methods

	Output Required	Supporting Methods
Effort/Cost Estimation	Overall Size	Indicative, Estimated, Full
Profile of Reuse Functionality	BFC Sizes, Functional Categories/Areas	Estimated, Full
Project Planning	Functional Categories/Areas	Estimated, Full
Estimate Validation	BFC Sizes, Functional Categories	Estimated, Full

The importance of the Bid stage estimates was emphasised by the company managers. These estimates form the basis for each project's fixed price contract, and therefore they are critical to the business. The main use of software sizing is as the basis for deriving an estimate of the development effort and cost. As each of the NESMA methods provide the overall Functional Size, all of these methods support Effort/Cost Estimation. The managers within the company noted the uncertainty associated with each estimate, which would caution them against using the size value as the fundamental basis for contract negotiation in the bid process. The value of the sizing estimates was considered to result from using them as a more objective basis for determining the relative scale of a proposed project. The suitability of each method, however, may extend beyond consideration of the overall Functional Size. The relationship between a more detailed profile of functionality and effort/cost may prove a more suitable basis for such estimation. It is therefore necessary to consider the relative accuracy of the functional profile produced by the sizing method, as well as the overall Functional Size accuracy.

Equiniti-ICS evaluate the degree of reuse from their core product for each new project, with the percentage of reuse factored into the effort/cost estimate. NESMA guidelines for sizing maintenance projects (which can be considered equivalent to re-used functionality) require examination of the proportion of DETs that are modified for each function. This would provide a measure of the level of reuse associated with a new project, and therefore a measure of the Functional Size of the 'new' functionality. The adoption of this in practice is problematical due to the characteristics of commercial project development. Insufficient requirements detail is available at the initial Bid stage of a project, preventing analysis of the DETs associated with each function. The cost in terms of estimation effort is also prohibitive to the adoption of this detailed assessment of re-use within the estimation process. Consideration of the issue of reuse from the perspective of functional sizing is therefore more appropriately focused on identification of specific types of functionality. Within each functional area of a project, the

NESMA approach identifies each specific Data Function and Transaction Function. This enables functionality that is generic to the application domain, and incorporated into the core product, to be identified from the size estimate. It is therefore necessary for the profile of each BFC type to be provided, and further decomposed according to specific Functional Categories. This level of detail precludes the use of the Indicative NESMA method for catering for this application of functional sizing.

The development process of Equiniti-ICS incorporates characteristics of ‘Agile’ practices after the Functional Specification has been finalised. In particular, the implementation is divided into ‘Sprints’ focusing on specific features of the application within each ‘Sprint’. These features correspond to specific Functional Categories/Areas as identified from the size estimates, therefore requiring that the output detail provided by the estimation method should facilitate decomposition into this level of detail. All managers within Equiniti-ICS agreed that all tasks within their plans and schedules could be developed either directly, or via “uplift”, from the size estimates. This application of Functional Sizing is supported by the Estimated and Full NESMA methods.

During the results feedback session, there was significant interest from the managers on the differences in functional size produced at the Bid and Functional Specification stages. The completed Functional Specification provides the basis for assessing the degree to which the initial estimate of the required functionality is reflected in the final view of the required functionality. This could take the form of simply comparing the overall Functional Size at the Bid stage and Functional Specification stage. In all cases, for the Estimated NESMA method, the functional size increased at the Functional Specification stage. The presence of ‘requirements creep’ could be indicated when the Functional Size at the latter stage is larger, but it cannot be confirmed without more detailed assessment. The Management at Equiniti-ICS indicated that consideration would be given to adapting their internal processes to accommodate management action on the basis of any differences between projects at these two stages. In particular, an examination of the reasons for the differences on future projects could enable material discrepancies to be identified and acted upon. Comparison of the BFC sizes at both stages, and of specific Functional Categories if necessary, enables the nature of variation in the Functional Size across the different stages to be analysed. For example, increases in Functional Size could arise from increases in the scope of functionality requested at the Bid stage. Conversely, ‘new’ functionality could have been identified during the Requirements Analysis and Specification process that was not indicated at the Bid stage. Functional Sizing therefore enables validation of the progress of a project by identifying the nature of any differences in the required functionality at the completion of the finalised Functional Specification. The level of output detail required from the estimation approach again limits this application to the Estimated and Full NESMA methods.

3.5.4.4 Compatibility with existing estimation practices

The software size estimates developed at the Bid stage must be considered in parallel with the current effort/cost estimation process used by the company. The software size estimates developed at the Functional Specification stage do not have a corresponding equivalent within the company. These estimates must therefore be considered as an additional estimation component for the overall software development process.

Table 30 shows the main issues of integrating the use of the NESMA methods into the existing estimation practices of Equiniti-ICS. The first issue is whether the NESMA method can be utilised at the required stage(s) of the project lifecycle. Currently, Equiniti-ICS develop estimates at the Bid stage of each project, focusing on effort/cost rather than size. The Indicative and Estimated NESMA methods only are feasible using the level of detail available at the Bid stage. Further estimates of the entire project are not currently developed on the completed Functional Specification documentation. Individual estimates are produced in accordance with an ‘Agile’ approach by adapting the Functional Specification into User Stories for each development ‘Sprint’. Use of the Full NESMA method is feasible at the Functional Specification stage, but would not be considered an appropriate use of resources at this point in the lifecycle.

In terms of the necessary requirements detail in the project documentation, the NESMA guidelines specify that the Indicative method requires a Data Model to be available. The nature of the Tender process, commonly utilised within this sector of the software development industry, generally prevents the availability of such a model at the Bid stage. In practical use of the method, the absence of a Data Model can be overcome providing sufficient detail is available from which the estimator can develop such a model. The current expert judgement-based estimation approach used by Equiniti-ICS incorporates the identification of the main system entities.

Table 30 – Compatibility of NESMA methods with existing estimation practices within the company

	Indicative NESMA	Estimated NESMA	Full NESMA	Overall Compatibility
Lifecycle Requirements	Bid, Functional Specification	Bid, Functional Specification	Functional Specification	Indicative, Estimated
Documentation Requirements	Data Model	Transaction Functionality	DET	Indicative, Estimated
Estimation Effort Requirements	Acceptable	Acceptable at Bid Stage	Not Acceptable	Indicative
Project Planning Requirements	Not Compatible	Compatible at both Stages	Compatible at Functional Specification Stage	Estimated

The Documentation Requirements for the Indicative NESMA method are therefore not considered incompatible with the existing estimation practices. The description of the required Transaction Functionality is a standard component of project documentation, even at the initial Bid stage. The availability of specific DET detail is generally insufficient at the Bid stage, and often partially incomplete at the Functional Specification stage. Consequently, while the Estimated NESMA method is considered compatible in this regard, the Full NESMA method is only partially compatible in terms of the documentation requirements.

The estimation effort required by the Indicative NESMA method is acceptable for adoption within Equiniti-ICS. The relative inaccuracy of this method, however, offsets this compatibility with the existing estimation practices. The Estimated NESMA method is considered to be acceptable, in terms of estimation effort, at the Bid stage, and as the Full NESMA method is not feasible at this stage, it represents the most detailed approach that is both feasible and acceptable. The use of either the Estimated or Full NESMA methods at the Functional Specification stage is not acceptable, particularly as the estimation effort is more substantial at this stage with more detailed documentation being available. The acceptability of the estimation effort is limited to projects that fall within the range demonstrated by this dataset. The possibility of the size of a project prohibiting the use of the Estimated NESMA method at the Bid stage cannot be ruled out, due to the time constraints associated with commercial tendering processes.

The Indicative NESMA method does not facilitate project planning requirements as it only provides the overall functional size. Both the Estimated and Full NESMA methods provide the necessary functional sizing data to support project planning. The relative accuracy of the Bid stage estimates is limited due to the incomplete specification of the required functionality. Consequently, even though the Estimated NESMA method can be used at this stage, the value it could potentially provide for project planning is diminished compared to the Functional Specification stage.

The overall compatibility of the NESMA sizing approach with the existing estimation practices within Equiniti-ICS remains limited despite the availability of the three NESMA methods. The management team within Equiniti-ICS was keen on the use of a consistent sizing approach at both the bid and detailed functional specification stages, as a means of evaluating the evolving product. The Estimated NESMA method provides such an approach, but the cost at the Functional Specification stage is not acceptable. The prevailing problem of the software estimation paradox is not alleviated by the existing NESMA methods. At the initial Bid stage, the incomplete specification of the required functionality cannot be overcome through the use of the NESMA methods. The Functional Specification stage does not attach the same level of value to the software size estimates produced using the NESMA methods. The subsequent research phases will therefore address this software estimation paradox.

3.6 Review of Research Phase 1

The functional sizing results produced by the NESMA methods have been analysed and evaluated in order to determine how software sizing can benefit the software development process. The following section therefore provides responses to each of the research questions posed at the outset of this research phase. This enables the focus of the subsequent research phases to then be outlined.

3.6.1 Response to Research Questions

RQ1: Which combination of software sizing rigour and project lifecycle stage provides the most stable indication of project size?

This Research Question was concerned with the trade-off between estimation effort and estimation accuracy when completing software sizing. The general pattern found from the sizing results was that the estimation effort was less substantial at the earlier Bid Stage. The most accurate estimates were obtained, however, at the Functional Specification stage where the completed description of the required functionality was available. The use of alternative classifications for the functional sizes was considered in the form of Rank Ordering and Size Banding. These approaches demonstrated some interesting results across the dataset, but no advantage over using the specific functional size was determined.

A straightforward consideration of the trade-off between estimation effort and estimation accuracy would suggest that the use of the Estimated NESMA method at the Functional Specification stage is optimal. The overall functional size and its relationship with the actual development effort has stabilised at this stage, with no additional benefits accrued from using the Full NESMA method. This optimal estimation performance is not, however, necessarily considered the optimal value of software sizing since it is at the Bid stage where estimation is of the most importance. The use of the NESMA sizing methods at these two distinct lifecycle stages cannot be considered 'optimal' in terms of meeting estimation needs within industry.

RQ2: How can size estimation data assist associated activities such as project planning within the development company?

In terms of general performance, for cost/effort estimation, software sizing does not provide an improvement over the existing expert judgement- based estimation approach. The primary benefit of employing the use of software sizing at the Bid stage is to provide an alternative project estimate, acting as a 'reality check' on the current estimation approach. Additional benefits are potentially available at the later Functional Specification stage, where presently no overall project estimate is obtained by Equiniti-ICS. The development of software size estimates at this later stage facilitates direct comparison with those from the earlier Bid stage. This provides a validation of the Bid stage estimates, enabling the identification of changes in the overall functional size of the project by the Functional Specification stage. Project plans may therefore be adjusted, if necessary, prior to the subsequent lifecycle stages. The software size estimates could also be readily decomposed to assist the 'Sprint' planning sessions during the development of the system. The decomposed size estimates may also enable the generic functionality within the 'core' company product to be quantified in terms of functional size.

RQ3: How can the software development process of the company accommodate the use of software sizing?

The accommodation of software sizing into the current software development process was considered at both the Bid stage and the Functional Specification stage. The sizing results for this dataset support existing research in finding that the additional estimation effort required for the Full NESMA method is not justified sufficiently by any increase in estimation accuracy. The assessment of the use of the Full NESMA method in this research further negates its use in terms of compatibility within the existing software development process. At the Bid stage the Full NESMA method is not feasible, while at the Functional Specification stage the estimation effort is prohibitive. The Indicative NESMA method is acceptable in terms of estimation effort at both the Bid and Functional Specification stages. The main drawback of this method, other than its relative inaccuracy, is the limited degree of utility due to its provision of only the overall Functional size. The issue with the Indicative NESMA method is therefore concerned with its inability to fulfil a role within the software development process. The Estimated NESMA method is the most compatible, providing the superior combination of estimation effort, estimation accuracy and output detail. The only drawback with this method is with the estimation effort required at the Functional Specification stage. The reduced value associated with estimation at this later stage in the lifecycle prevents the Estimated NESMA method from being considered necessary to the project development at this stage.

3.6.2 Researching the Role of Software Sizing

The results and analysis of this Research Phase support the assertion that software sizing has a role to play in the estimation practices in industry. The limited uptake of software sizing within industry indicates that challenges remain in defining the role. The use of the NESMA methods in this research has identified the preliminary findings in attempting to identify how software sizing can provide value to the existing estimation practices of a development organisation.

- The Estimated NESMA method provides an acceptable balance between benefits and costs of implementing this approach to software sizing.
- Software sizing can be used to support existing expert judgement-based approaches to estimation by providing an alternative analysis to validate the primary estimate.
- The sizing data provided by the Estimated NESMA method can support the formulation of project schedules for the subsequent project lifecycle activities.

These benefits enable value to be added by providing a greater level of assurance in the reliability of the effort/cost estimates. The use of the NESMA sizing methods did not, however, provide superior performance to the existing Expert judgements-based approach used within Equiniti-ICS. The demonstrable benefits of software sizing at the conclusion of this initial Research Phase do not extend beyond a supporting role to the established estimation approach in typical development organisations. The results provide evidence of limitations in how effective software sizing methods are in providing value to the development process.

- The estimation performance of the NESMA sizing methods in terms of use in Effort/Cost estimation is superior at the later Functional Specification stage.
- The level of requirements detail available at the initial Bid stage is generally insufficient to enable the Full NESMA method to be utilised.
- The clarity of the required functionality described in this documentation is the main limitation in identifying each individual function when using the Estimated NESMA method.
- The cost of using software sizing methods is only considered acceptable at the Bid stage, which is the only lifecycle stage where estimation is currently utilised within Equiniti-ICS.

These results are consistent with the software estimation paradox, wherein the size estimates are most reliable at a stage in the lifecycle when they are no longer considered as important. Subsequent Research Phases in this thesis are therefore focused on examining how the sizing data produced by the NESMA sizing methods can be further exploited to overcome the practical limitations on the effectiveness of software sizing.

- How can the size estimate data developed at the initial Bid Stage be used to provide an assessment of the risk of relative inaccuracy in comparison to the final size of the project?
- How can software sizing 'learn' to adapt to the lack of sufficient requirements detail at the Bid stage?
- How can the software sizing adapt to deliver the current level of performance in a more efficient manner?

3.7 Conclusions

This research phase has highlighted the value that software sizing can provide to the software development process. The potential of software sizing is not, however, fully realised by the existing FPA approach, as represented by the NESMA methods. Further research is therefore necessary to enhance the role that software sizing can play within the software industry.

The functional size of a project changes from the Bid stage to the Functional Specification stage. This highlights the need to investigate the nature of these changes, and how they can be anticipated at the earlier Bid stage. The functional sizing data should be analysed beyond the current focus of the NESMA methods in order to discern whether the Bid stage software sizing data can provide additional insights into required functionality that may emerge beyond this stage.

The increased accuracy available at the Functional Specification stage provides a source of potential value to the project management process. The value of estimation has, however, diminished by this stage of the project lifecycle as the initial Bid stage estimate effectively determines the realistic chance of success for the project in terms of meeting its budget. The challenge lies therefore with how the potential value of the Functional Specification stage size estimate can be realised. The use of a more simplified sizing method addresses the issue of the effort required to develop such estimates. However, the Indicative NESMA method does not demonstrate an equivalent level of performance compared to the more detailed versions of NESMA. This highlights the need to investigate how simplified approaches to software sizing can be adapted to better meet the needed performance requirements.

4. A Functional Sizing Oriented approach to Bid Estimation

In Chapter 3, the most important estimation priority was acknowledged within Equiniti-ICS to be that of informing the bid proposal. This Chapter is concerned with investigating how software sizing can contribute to bid stage estimation.

4.1 Introduction

The value of Software Estimation will vary according to when it is utilised in the project lifecycle. In the previous chapter, the value of software sizing for Software Estimation was examined at two distinct stages of the project lifecycle – the initial bid stage and the later completion of the detailed functional specification. This chapter is concerned with how the value of software sizing can be increased at the initial bid stage of the project lifecycle.

The most important decision for a project, particularly for fixed-price contracts, is the initial decision to proceed with the project. In a commercial tender situation this takes the form of submitting a bid to the customer, based upon the estimated development cost for the project. Consequently, in a project tender situation the tender (or bid) estimate is crucial in informing this decision. The ability to successfully manage a project is dependent on developing a reliable appreciation of the required project functionality. The value of the estimate is therefore greatest at the bid stage; paradoxically it is at this stage when the least requirements detail is available to inform the estimate. The accuracy of any estimate may be associated with the uncertainty associated with the project at that stage of the lifecycle. Boehm (1981) illustrated that the estimation uncertainty decreases as the project progresses through its lifecycle. This phenomenon has been referred to as the 'Cone of Uncertainty' (McConnell, 1996), and underlines how the most critical software estimate must be made under the greatest uncertainty. How can this estimation uncertainty be managed when it is at its greatest? This chapter addresses this question, by investigating what may be learned from more accurate size estimates at the functional specification stage to identify the degree of uncertainty associated with the initial bid stage estimates. In the previous chapter, software sizing results were presented for the bid stage and the later functional specification. The focus of this chapter is on determining what can be discerned from what the bid document does tell you, in order to identify what it does not tell you. For this research the sizing results from these two distinct stages are compared and analysed in order to discern where required functionality is 'missing' at the bid stage. A detailed assessment is made of how well specific types of functionality are represented in the bid documentation. The next stage involves the development and assessment of possible size metrics associated with future growth from bid sizing data. Finally, the potential for discerning that growth in functional size had occurred at the Functional Specification is assessed through consideration of potential size metrics at that stage. The value added through software sizing would consequently be found through the additional insight not provided by existing expert-based approaches to software estimation.

4.2 Related work

The reliability of using unstructured expert judgement-based estimation at the bid stage was investigated by Faria and Miranda (2012). In their study, 14 employees with an average experience of 7.5 years were randomly assigned into one of two groups. Each participant was given documentation equivalent to a request for proposal document, from which to develop an estimate of the effort for the project allocated to their group. For project 1, the effort estimates differed significantly from 120 to 2016 staff days. For project 2, the range of effort estimates was from 55 to 940 staff days. The use of a more structured estimation method is investigated in this thesis, but the characteristics of the requirements documentation may also impact upon the reliability of estimates made at this early stage. Estimates made at the initial bid stage of a project are commonly developed from Request-For-Tender (Bid) documentation written by the customer. The structure and relative detail of these documents will be subject to more variation from one customer to another than is found within the internal project documentation of a development organisation, and consequently the estimation uncertainty will be greater at this early stage. More detailed specifications of required functionality have been shown to lead to larger estimated values being produced. In a study of a real life bidding process Jorgensen and Carelius (2004) instructed one group of 17 development companies to produce a bid based firstly on a brief one-page description of the proposed system, and secondly based on a more detailed 11-page requirements specification. On average, the bids developed from the detailed specification were 73% higher. A second group of development companies in the same study, who only received the detailed specification, produced lower bids, on average, than the first group. The authors proposed the explanation that the greater uncertainty present with the brief description of required functionality leads to an overcompensation effect of building the associated risk into the bid. The subsequent reduction in estimation uncertainty was then considered to be insufficient to overcome this initial overcompensation. For the sample projects used in this thesis, the description of system functionality in the bid stage documentation range in size from 2 pages to 90 pages, with an average of 30.8 pages. The relationship between documentation size and the estimated functional size for the sample projects will be examined as part of the analysis of bid stage estimation.

The bidding process itself leads to the phenomenon of the 'Winner's curse' (Jørgensen and Grimstad, 2005) wherein realistic bids are less likely to be successful when there are numerous competitors. For fixed price contract projects, in particular, the selection of an overly optimistic bid is unlikely to result in either a profit for the winning development company or an entirely satisfactory system for the customer. Size estimates must therefore communicate some measure of the associated risk so that informed bidding decisions can be made. Assessing the accuracy of estimates simply by comparing the overall actual magnitude against the overall estimated magnitude may provide an unrealistic indication of the relative success of a project estimate. The nature of the underestimated functionality may provide an indication of the likely customer satisfaction with the completed system. This chapter assesses the differences between the initial identification of the requested functionality, as outlined in the bid documentation, and the later detailed specification of the functionality.

Agile development practices have become more prevalent in industry, but the bidding process typically requires an estimate based on the overall project. Jamiesen et al. (2005) propose that the tender procurement process must also adopt agile principles in order to minimise the extent of estimation uncertainty at the bidding stage. Applying a bidding process to each iteration of the project effectively ensures small projects, and may enable the estimation accuracy of each phase to improve as actual data from previously completed phases becomes available. This approach would, however, require a fundamental change in the customer-supplier tender relationship from what commonly exists in industry.

Estimates are often used at different stages in the project lifecycle, and may differ in magnitude in part due to the required functionality evolving as the project progresses. In some development companies as many as six estimates, made at different lifecycle stages, have been observed for a software project (Moløkken-Østvold et al., 2004).

Agresti et al. (2010) demonstrated how system detail developed during the various stages of design could be utilised to facilitate more accurate size estimates as the project progresses through the lifecycle. Zivkovic et al. (2005b) found that early size estimates had, on average, a relative error of 40% compared to those estimates developed from subsequent more detailed documentation. Demirors and Gencel (2004) developed Function Point estimates at five separate stages of requirements analysis for a project module. The size obtained increased at each stage, rising from an initial 940 FP to 2089 FP. This highlights the inadequacy of the size estimates developed from the initial high level documentation, and introduces the prospect of learning from subsequent estimates later in the lifecycle. Can the reasons for these variations be identified and used to inform bid estimation in future projects?

In the previous chapter software size estimates were developed at both the bid stage and the detailed Functional Specification stage of the project lifecycle for eleven commercial projects. In each case the functional size was found to be larger when determined at the Functional Specification stage, with the relative increase in size ranging from 1.96% to 90.95%. The phase of research in this chapter aims to understand the reasons for this pattern and determine if the lessons learnt can be used to inform future size estimates made at the bid stage of the project.

The reasons for less accurate initial estimates may include imprecise descriptions of the required functionality effectively hiding the complete requirements; an incomplete consideration or understanding of the required functionality; the customer changing the requirements as the project develops; the inability of the estimation approach to adequately assess the required functionality based on the available detail. Compared to expert-based estimation approaches, model-based estimation approaches are more reliant upon detailed input data and may therefore be more affected by imprecision in bid documentation. Expert-based approaches may therefore be able to more effectively 'fill in the gaps' and compensate for the lack of detail. However, as these approaches are more subjective they may be more prone to be influenced by other factors. Expert judgement has been shown to be affected by the presence of irrelevant information, such as the expectations of the customer, even when the estimators were explicitly instructed to disregard this information (Jørgensen and Grimstad, 2008). Managerial pressures may also affect the estimate, such as the desire to win the contract by providing a lower bid than the competitors. Model-based approaches provide a more objective basis for estimation, and therefore are less likely to be affected by other factors. The degree of objectivity may be affected by the lack of detail limiting the ability of the estimator to identify the required input detail, and there may also be a greater reliance on making assumptions on what functionality is required. During the initial stages of a project it may only be feasible to utilise a simplified version of model-based estimation that requires less input detail, such as the Indicative NESMA method **(**NESMA, 2004**)**. Regardless of this, the greater rigour involved even in simplified model-based approaches may enable functionality to be more accurately identified, or it may indicate where functionality would be expected but has not been specified.

In the event that the size of the system is found to differ significantly from the size determined at the initial bid stage, the damage has effectively already been done. It is therefore of interest to determine if reasons for this variance in size may be identified at the initial bid stage of a project. After the full functional specification is complete the uncertainty surrounding the required system functionality has been minimised. There is limited scope for improving the accuracy of size estimates, and consequently for examining the extent to which software size is one of the factors affecting estimation uncertainty, beyond this stage. This chapter, therefore, is focused on comparing the reasons for the differences in the software size estimate based on the initial bid documentation and the later functional specification.

4.3 Research Aim

The main focus of this research phase is to assess the relative completeness of the bid documentation available for each project. Consideration of the potential inaccuracy of the developed estimates at that stage can provide an indication of the risk associated with those estimates. The current estimation process employed within Equiniti-ICS incorporates a consideration of the risk inherent in each project. In developing estimates, Equiniti-ICS analyses the content of the bid document for a project, where the prospective customer is commonly the author of the documentation. An assessment is therefore made by the estimators of their confidence in the level of description provided in the bid documentation. This assessment would focus on the perceived expertise of the customer's personnel involved in the production of the bid documentation, the number and nature of queries made on the bid document and responses received. In essence, the customer's understanding of the proposed system is evaluated, and in turn, of the completeness of the required functionality. This reliance on the expertise of the estimator in making subjective assessments presents some limitations of this approach. The nature of subjective judgements can be difficult both to transparently justify their assessment, and to document for future projects. The experience gained from making these assessments is generally limited to the ‘expert’, limiting the extent to which it may benefit the expertise of the project team. This research phase seeks to develop a more objective basis to assist with such deliberations, by investigating how functional size estimation can be utilised to inform bid estimates.

4.3.1 Research Questions

The role of functional sizing within bid estimation is to provide an objective measurement of the size of the system to be developed as ascertained from available requirements detail. The value derived from functional sizing is dependent upon how the estimated size related to both the size of the final system and the effort required in developing the system. This chapter is concerned with assessing the value that can be added by having a more detailed specification of required functionality for informing bid stage estimation. This will be facilitated by the comparison of the functional size estimates developed from the bid and full functional specification documents. This comparison will focus on specific types of functionality identified within the existing BFC types associated with FPA. The profile of the specific types of functionality obtained at both the bid stage and the functional specification stage may enable us to identify the degree of functionality missing from bid-level documentation. This, in turn, facilitates an assessment of the risk that a project may change in size as a more detailed specification of the requirements is developed. Value can be added to the functional size estimation process from the identification of specific sizing data associated with relative change in functional size. In addition, the detection of such changes in functional size could be indicated without requiring further detailed size estimates. Consideration of a more detailed classification of required functionality can provide value by identifying which types of functionality are related most strongly to the development effort. The research questions of this chapter are therefore:

RQ1 - Can a detailed assessment of the profile of required functionality at the bid and the functional specification stages in the project lifecycle provide a validation of the adequacy of requirements documentation at the bid stage of a project?

RQ2 – What level of detail of functional sizing data at the Bid stage of a project is most strongly associated with the overall Bid functional size and required development effort?

RQ3 – Can relative change in overall functional size be associated with sizing data, and if so, what level of sizing detail is required?

4.4 Research Approach

This research phase involves the eleven commercial projects provided by Equiniti-ICS, and incorporates a consideration of their current bid estimation approach. These projects were completed on a fixed-price contract basis, where the estimated cost was used as the basis for the submission of a bid. In this context, the importance of developing accurate estimates at the initial bid stage is paramount to the success of a project.

In the previous chapter software sizing using each of the NESMA functional sizing methods (NESMA 2004) was completed on the eleven commercial projects using documentation taken from two distinct stages of the project lifecycle. The requirements documentation at the initial bid stage generally took the form of Request for Tender documents. These were produced by the customer and therefore provided differing levels of detail and presentation formats. The later Functional Specifications (referred to in the results as the Detailed Stage) were produced by the development teams, and therefore provided a more uniform approach to documenting the requirements. Each of the projects involved data driven applications, generally in the form of Case Management Systems. The general similarity of the projects supports the development and use of a common classification of functionality types. The main stages of this research phase are as follows:

(1) – Develop a Detailed Profile classification of functionality types to facilitate the production of the detailed functional profile.

(2) – Analyse the size estimation output data in order to classify each instance of BFC under the developed Detailed Profile classification. This will produce the more detailed functional profile for each project.

(3) – Evaluate detailed functional profiles at bid stage and functional specification stage for validation of bid size estimation. **(RQ1)**

(4) – Perform Linear Regression Analysis on the results of stage (2) for evaluating the relationship with overall functional size and development effort. Compare these results with those produced by the conventional NESMA sizing output data. **(RQ2)**

(5) – Develop and assess potential size metrics associated with functional growth at the bid estimation stage. **(RQ3)**

(6) – Develop and assess potential size metrics associated with functional growth at the functional specification stage. **(RQ3)**

The conventional NESMA sizing output data consists of the sizes of the ILF, EIF, EI, EO and EQ components. For this investigation, this is referred to as the NESMA Functional Profile. The more detailed classification developed in Research Stage 1 is referred to as the Detailed Functional Profile.

4.4.1 Research Stage (1) – Detailed Functional Profile classification of Functionality Types

The NESMA sizing completed on the sample projects produced the detailed listing of each instance of each BFC. The IFPUG functional sizing method (IFPUG, 2010) uses these same components, so in order to remain compatible with this IFPUG method the additional types of functionality are drawn directly from the main BFC. Each new type of functionality in the developed Detailed Functional Profile classification is therefore a subtype of an existing BFC, rather than a new distinct component type. Developing the classification requires ‘common’ types of functionality to be identified from the sample commercial projects. The potential limitation of this approach is to categorise specific types of functionality that are not prevalent outside of the domain of the project dataset. The new types of functionality should therefore remain sufficiently broad to preserve the generalizability of the results of this research phase.

Occurrences of each type of BFC in the NESMA sizing output detail for each project were analysed in order to determine if suitable subtypes of functionality could be identified. Data Functions were generally represented more clearly in the bid documentation than Transaction Functions, and changes in the later Functional Specification were more likely to reflect a change of scope rather than insufficient detail in the description of requirements at the bid stage. In contrast, the description of Transaction Functions is often broad (or missing), and are only explicitly stated in the later Functional Specification. The Transaction Functions therefore provide the most potential for beneficial decomposition into further subtypes of functionality.

4.4.1.1 Internal Logical Files

This BFC provided limited potential for decomposition into useful subtypes of functionality. The NESMA sizing approach requires the individual RETs that an ILF is comprised of to be identified. The comparison of the total number of RETs to the total number ILF provides an indication of the extent of 'associated data' for each main logical entity in a system. RETs would be of interest in assessing the extent of associated transaction functionality at the bid stage, but they do not constitute a subtype of ILF. Consequently, the only type of functionality to be added to the classification for this aspect of the phase is the existing BFC.

(C1) Internal Data Functionality: Refers to any file accessed and maintained by the system to be developed.

4.4.1.2 External Logical Files

This BFC was not found to occur to any significant degree in the sample projects, and therefore was not suitable for decomposition. The existing BFC is therefore the only type added in this case to the classification.

(C2) External Data Functionality: Refers to any file accessed by this system, but which is maintained by an external application.

4.4.1.3 External Inputs

The Indicative NESMA method assumes that there will be, on average, three External Inputs associated with each ILF identified in the system. These correspond to 'Add', 'Amend' and 'Delete' functions and represent obvious candidate subtypes for this BFC. These types of functionality are consistent with types of functionality required within the sample projects, and were therefore added to the Detailed Profile classification. The case management nature of the sample projects was reflected in the significant maintenance of 'Links' between related data records. For example, a 'Case' record could be 'linked' to multiple other records such as 'Document', 'Action', 'Party' etc. 'Link' functionality specifically required to maintain these links was therefore determined to be an appropriate type of functionality to be included in the classification. The maintenance of data in the sample projects frequently centred on recording the completion of specific business processes. The fact that the bid documentation was generally written by the customer may suggest that the description of the system requirements would be relatively focused on the perspective of the business functionality. This introduces the candidate type of ‘Process’ functionality, which is focused on specific business processes rather than generic data management processes. The following types of functionality are therefore added to the classification.

(C3) Add Functionality: Refers to any function concerned with adding a new record to a file within the system e.g. a Case record added using an ‘Add Case’ function.

(C4) Link Functionality: Refers to any function concerned with linking records from different files together, by creating a new record in a 'link' file e.g. a ‘Link Document to Case’ function that creates a Case Document record for linking a Document record to a Case record.

(C5) Amend Functionality: Refers to any function concerned with the general modification of a record within a file e.g. modifying a Party Address record using a ‘Modify Party Address’ function.

(C6) Process Functionality: Refers to any function concerned with amending existing details of a record through the completion of a specific business process e.g. an ‘Approve Recommendation’ function which updates the status of a Case Recommendation record.

(C7) Delete Functionality: Refers to any function concerned with removing a record from the system, or marking the record as deleted in the system e.g. removing a Party from the system using a ‘Delete Party’ function.

4.4.1.4 External Outputs

The External Output functionality described in the NESMA guidelines provides an indication of the main types of functionality typically required within Information Systems. The functionality associated with searching for data can be identified through the use of selection criteria and the varying size of the output. In the absence of selection criteria, the output of details held in multiple records is commonly provided in the form of ‘List’ Functionality. The output of statistical reports is often provided explicitly through ‘Report’ functionality. These three types of functionality were each evident in the sample projects and were therefore appropriate candidate types of functionality. Outside of these types, the remaining required External Outputs in the sample projects did not provide further opportunity for decomposition. The most common remaining functionality was concerned with creating detailed output regarding individual cases for specific business processes. This type of functionality can be classified as ‘Output Functionality’. Any remaining miscellaneous output functionality can be placed in this type as it is generally tied to business processes, and does not conform to the other types. The following types of External Output are added to the Detailed Profile classification.

(C8) Search Functionality: Refers to any function concerned with searching for records within the system, in particular the provision of the search results where more than one record matches the search criteria e.g. a ‘Search Assigned Cases’ function which retrieves each of the cases a Team Member is assigned to work on.

(C9) Output Functionality: Refers to any function concerned with providing output detail, typically regarding an individual case, and for a specific business process e.g. a ‘Create Document Bundle’ function which generates the relevant data associated with an individual case.

(C10) Report Functionality: Refers to any function concerned with producing reports, typically involving many records and generated summary statistics, e.g. a ‘Total Number of Open Cases Report’.

(C11) List Functionality: Refers to any function concerned with listing numerous records within a particular file without the need for selection criteria e.g. a ‘List Case Events’ function which generates a list of all the Case Event records associated with a Case.

4.4.1.5 External Queries

This BFC provided insufficient potential for decomposition into useful subtypes of functionality due to limited deviation within the nature of EQ components in the sample projects. The relatively small contribution of this BFC to the overall functional size of the projects would lead to insignificant contributions for any decomposed subtypes of functionality. This suggests that there would be greater insight obtained in maintaining the existing EQ component, as defined under NESMA, for inclusion in the Detailed Profile classification.

(C12) Query Functionality: Refers to any function concerned with displaying the details from a specific record, and where no additional processing was required in producing the output e.g. displaying a detailed Case record selected from a list of assigned cases.

4.4.1.6 Limitations of Detailed Functional Profile classification of Functionality Types

The development of the NESMA size estimates facilitated a detailed understanding of the main types of functionality required in the sample projects. The selection of the types of functionality to include in the Detailed Functional Profile classification is a reflection of the nature of Case Management Systems. The suitability of this classification can be assessed in terms of:

(i) Degree to which the identified functionality fits the classification – as each functionality type is derived from a NESMA BFC, it is envisaged that each individual function should map to a functionality type in the Detailed Functional Profile classification. The functionality types are designed to be sufficiently distinct to provide a detailed functional profile of a project, and sufficiently broad to be fully inclusive of the required functionality. The evident exception lies with the inclusion of the NESMA defined FPA functionality, covering areas such as lookup functionality. This type of functionality accounts for an insignificant proportion of the functional size of a project, and is essentially of 'fixed' size when the Estimated NESMA method is used. There is therefore no value in including this aspect in the classification. The assessment of the degree of inclusion in the Detailed Functional Profile classification will take the presence of this 'missing' functionality into account.

(ii) General applicability of the Detailed Functional Profile classification – the required functionality of the Case Management Systems included in the sample projects is broadly typical of the data intensive systems considered suitable for functional sizing. The ILF, EIF and EQ components of the NESMA method have direct equivalents in the classification. The 'Add', 'Amend' and 'Delete' functionality types correspond to NESMA assumed functionality. The 'Search', 'Report' and 'List' functionality types are common examples of EO components. The majority of the types included in the Detailed Functional Profile classification can therefore be considered to be generally applicable to the use of functional sizing.

4.4.2 Research Stage (2) – Allocation of Functionality Types

The objective of this Research Stage is to develop a more detailed profile of each of the sample projects. The NESMA sizing data from the previous Research Phase provides a listing of each individual instance of a BFC identified within each project, at both the bid stage and the Functional Specification stage. Each instance of a BFC is categorised using the Detailed Functional Profile classification of functionality types developed in Research Stage (1). The following steps are completed for each project, firstly from the bid size estimate data and then the functional specification size estimate data.

(i) For each instance of ILF, record an instance of C1.

(ii) For each instance of EIF, record an instance of C2.

(iii) For each instance of EI, determine which functionality type (C3, C4, C5, C6, C7) the component corresponds to. Record the appropriate instance or, in the event that no type is suitable, leave the component unclassified.

(iv) For each instance of EO, determine which functionality type (C8, C9, C10, C11) the component corresponds to. Record the appropriate instance or, in the event that no type is suitable, leave the component unclassified.

(v) For each instance of EQ, record an instance of C12.

Upon completion of each project size estimate allocation, calculate the percentage of BFC instances that were classified. This will indicate the degree to which the Detailed Functional Profile classification accounts for the required functionality for each project.

4.4.3 Research Stage (3) – Comparative Analysis of Bid Stage and Functional Specification Sizing Data

The previous Chapter provided a comparison of the overall functional size at two distinct stages of the project lifecycle. In Research Stage (3), the objective is to provide a more detailed analysis of the functional size at both the bid stage and the functional specification stage. The functional size for each functional type in the Detailed Functional Profile classification equals the Estimated NESMA assigned function point value for the BFC type the component corresponds to. The following steps are completed for each project, firstly from the bid size estimate data and then the functional specification size estimate data.

(i) For each instance of C1, record a functional size of 10 FP.

(ii) For each instance of C2, record a functional size of 7 FP.

(iii) For each instance of (C3, C4, C5, C6, C7), record a functional size of 4 FP.

(iv) For each instance of (C8, C9, C10, C11), record a functional size of 5 FP.

(v) For each instance of C12, record a functional size of 4 FP.

The overall functional size for each functionality type is then calculated by adding the function point values for each instance of that type. The relative change in functional size, from the original bid estimate to the subsequent detailed functional specification estimate, is then calculated for each type of functionality.

In order to assess the significance of any changes in functional size within the Detailed Functional Profile classification of functional types, a comparison is also made of the percentage contribution that each type made to the overall functional size of the project at both the bid and detailed stages. This Research Stage addresses RQ1 by providing a more detailed functional profile for each project at both stages of the lifecycle.

4.4.4 Research Stage (4) – Statistical Analysis of Functional Sizing Detailed Functional Profile classification Data

The primary use of functional size estimation is often to facilitate the subsequent estimate of the development effort required to complete a project. Can the provision of a more detailed functional profile at the bid stage facilitate more accurate development effort estimation? This can be investigated in two main ways. Firstly, the relationship with development effort can be examined directly by using the bid estimation data. Secondly, the relationship can be investigated indirectly by examining how the bid estimation data is related to the functional specification estimation data. Analysing the correlations for these relationships must consider how the results are affected by the volume of testing. The use of multiple significance tests within correlation analysis impacts the number of false positive results that may be obtained. For an individual significance test with $\alpha = 0.05$, there is only a 5% probability of achieving a statistically significant result by chance. For 100 tests considered together, commonly referred to as a 'family' of tests, at the same significance level the number of expected false positives would be 5 due to the volume of testing. Techniques to control the number of false positives generally do so in terms of either the familywise error rate or the false discovery rate. Familywise error rate focused techniques are particularly effective at controlling false positives, but the conservative nature of such techniques may lead to an excessive number of false negatives. The research conducted in this thesis is exploratory in nature and would therefore be limited if the number of false negative results were inflated. The less conservative false discovery rate approach to controlling false positive is therefore considered more appropriate, providing greater statistical power for discovering statistically significant results. The Benjamini-Hochberg procedure enables the false discovery rate considered acceptable to be set e.g. a rate of 10% would correspond to a probability of 10% of obtaining a false positive result. The Benjamini-Hochberg procedure adjusts the α significance level to a critical value for each result, such that any individual significance result within its level would not lead to the false discovery rate being exceeded. The definition of what constitutes a 'family' of results is concerned with identifying a set of inferences that should be considered collectively in reaching an appropriate overall decision. The Benjamini-Hochberg procedure is less sensitive to decisions regarding which tests are considered to be part of a 'family' than familywise error rate focused techniques. The proportion of results determined to be statistically significant under this procedure would generally remain similar, regardless of 'family' size, when the distribution of *p* values within the 'family' remains similar. The selection of a 'family' for each use of the Benjamini-Hochberg procedure is indicated where appropriate in the discussion of these statistical analysis results. The false discovery rate adopted for use in this research is 10%, and statistical significance results where the Benjamini-Hochberg procedure has been applied are labelled as such in the results table.

In the preceding chapter, it was determined that functional size estimates based on the more detailed functional specification were more strongly correlated with development effort than those based on the initial bid documentation. Using the bid estimation data to predict the size at the functional specification stage would enable more accurate size estimates to be obtained at the bid stage. This, in turn, would enable more accurate development effort estimates to be made at the same stage. Research Stage (4) therefore uses Linear Regression to investigate how the detailed bid estimation data is related with both the development effort and the functional specification size. The use of Multiple Linear Regression Analysis enables multiple non-dependent variables to be investigated together in order to consider the effects of interactions between these variables. This can be used to determine which predictor variables are necessary to form the most accurate model of the relationship with the criterion variable. The limitations of this approach impact upon its suitability for Research Stage (4). In particular, overfitting of the regression model occurs when the number of non-dependent variables used provides greater complexity in the model than is permissible by the available test data. Green (1991) investigated previously reported rules-of-thumb for the recommended ratio of test cases to non-dependent variables for linear regression models, with ratios that ranged from 5 to as much as 20. Austin and Steyerberg (2015) reported that a ratio as low as two could be used for adequate estimation of regression coefficients in a model with a fixed number of non-dependent variables. In the dataset available for this research there are eleven projects (test cases) with which a regression model can be investigated. This is essentially the same as the number of functionality types included in the Detailed Functional Profile classification from Research Stage (1). The suitability of investigating the complete Detailed Profile classification of functionality types in the same Multiple Linear Regression model is therefore not recommended to be reliable. However, the use of Multiple Linear Regression can provide insight into the potential nature of how the various functionality types interact to determine each dependent variable considered. The use of the complete Detailed Functional Profile classification of functionality types is not expected to produce meaningful results for the dataset when all of the types are included as non-dependent variables. Consequently, where necessary, steps should be taken to identify which specific non-dependent variables are necessary to best explain variance in the criterion variable.

For the results from the Multiple Linear Regression Analysis, completed using the SPSS software application, the R Square value provides a measure of the proportion of variance in the criterion variable that is explained by the model. The limited ratio between the number of test cases and the number of non-dependent variables is considered through the reporting of the Adjusted R Square value. This provides a more realistic measure of the 'success' of the model, by accounting for the size of the dataset and the number of non-dependent variables.

The safest approach to Multiple Linear Regression Analysis, when there are a limited number of test cases, is to use the 'Enter' method where all of the non-dependent variables are simultaneously considered. Each non-dependent variable is assessed as though it has entered after all the other non-dependent variables have been entered. An assessment is thus made on what each non-dependent variable adds to the prediction of the dependent variable. For situations where the numbers of non-dependent variables are closer to the number of test cases, the likelihood of meaningful results being obtained diminishes. The use of a 'statistical' based approach, such as the 'Stepwise' method, would provide a possible insight into the Regression model by selecting only those non-dependent variables which significantly add to the model. In accordance with the desire to establish the minimum number of predictor variables required, the 'Stepwise' method will be used alongside the 'Enter' method.

4.4.5 Research Stage (5) – Development and Assessment of Bid Stage Size Metrics

The size estimates developed from the Functional Specification were found to be more strongly correlated with the actual development effort figures for the projects. This Research Stage is concerned with assessing the extent to which the size estimates at the Functional Specification stage can be anticipated at the Bid stage. Two main approaches are adopted in completing the assessment of the Bid stage sizing data. Firstly, the relationship with the overall Functional Specification size is examined. Section 4.4.5.1 outlines the different levels at which this analysis is undertaken. Secondly, the relationship with the relative change in functional size between the two stages is examined. This relative change is referred to as 'Functional Growth' in this research as each project increased in functional size between the two stages. The development of Bid stage size metrics associated with functional growth is focused on assessing the extent to which the required functionality of the completed system has been identified in the bid documentation. Functional sizing views a system in terms of Data Functions and Transactions Functions. In essence, the Data Functions will have associated Transaction Functions that maintain the data in a system. The ratio of Data Functions to Transaction Functions therefore forms the foundation for the development of potential size metrics. The nature of both Data Functionality and Transaction Functionality is first considered in order to establish suitable ratio components (Sections 4.4.5.2 and 4.4.5.3). The outline of the developed size metrics is provided in Section 4.4.5.4. Characteristics of the Bid documentation data are then used to develop further metrics to evaluate the relationship with Functional Growth (Section 4.4.5.5).

4.4.5.1 Relationship with Overall Functional Specification Size

The analysis of the relationship with overall Functional Specification size incorporates the following different levels of Bid stage sizing detail are:

(1) – Overall Bid Functional Size
(2) – NESMA Functional Profile Level (BFC)
(3) – Detailed Functional Profile Level (Classification of Functionality Types)

This enables the level of sizing detail required at the bid stage to most accurately indicate the overall size at the Functional Specification stage. In order to provide a directly equivalent comparison between the stages, the Estimated NESMA method provides the size at both stages. Firstly, the correlation with the overall Functional Specification size is assessed for the individual components at each level of detail. This enables the individual components that are most strongly related to the overall Functional Specification size to be identified.

Secondly, Multiple Linear Regression Analysis is then completed within each level to determine if the combination of multiple components is more strongly associated with the overall Functional Specification size. This regression analysis is completed subject to the same general approach described in Section 4.4.4.

4.4.5.2 Data Functionality

The main type of Data Functionality found within the sample projects was, as expected, the ILF component. This component is therefore considered to be a suitable candidate for developing a ratio to use as a size metric.

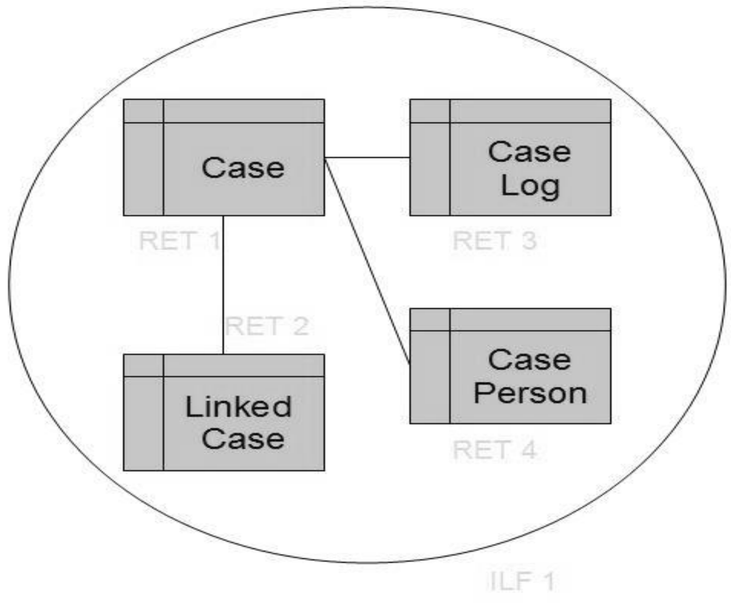

Figure 4.1 - Sample Data Model

The NESMA functional sizing approach identifies each ILF, including the individual Record Element Types (RET) each of the ILFs may be comprised of. The completion of NESMA functional sizing in the previous chapter enabled a listing of ILFs, and the respective RETs, to be developed for each project. Figure 4.1 shows a typical example of part of a data model from a Case Management System. The four individual RETs shown represent one ILF in the system. In this example the ‘Case’ RET is the core RET within the ILF. The 'Case Log', 'Linked Case', and 'Case Person' RETs would only be recorded in the system in conjunction with the 'Case' RET.

In Functional Sizing, common functionality such as 'Add' functionality is expected to be required for each complete ILF. The functional sizing performed on the sample projects, at the Functional Specification stage, indicated that such common functionality was instead frequently requested for each individual RET i.e. there would be four 'Add' functions rather than the one function in this example.

This pattern was less evident in the bid documentation, in part due to the less detailed description of the system requirements provided at this stage. The individual RETs may be more identifiable at the bid stage than the specific associated Transaction Functionality. The RET component is therefore also considered to be a suitable candidate for developing a ratio.

The EIF component is an obvious potential candidate for inclusion as it is a BFC of the NESMA functional sizing approach. The instances of this component within the sample projects were, however, negligible and it is therefore not considered to be a suitable ratio candidate.

4.4.5.3 Transaction Functionality

The main NESMA components of Transaction Functionality are EI, EO and EQ. In Research Stage (1) these components were further decomposed into a Detailed Profile classification of functionality types. This provides ten potential ratio components for use in the development of size metrics. The completion of Research Stage (2) resulted in a listing of the occurrences of each functionality type within each project. The overall pattern found was that each functionality type was commonly found across the dataset, but in some select cases a specific functionality type was not present within every project at both the bid stage and the functional specification stage. This is not considered to affect the suitability of these components, but the results should exclude any instance where a component type was not present within a project at either stage. Each of the ten Transaction Functionality types from the Detailed Profile classification is therefore included as ratio components in this stage of research. The analysis will accommodate the size of the dataset for each specific ratio component into determining the statistical significance of the results.

4.4.5.4 Functional Based Size Metrics

These size metrics are concerned with comparing the amount of Data Functionality with the amount of Transaction Functionality. The degree to which the specified amount of ‘associated’ Transaction Functionality in the Bid documentation conforms to the expected amount is investigated as a potential size metric associated with Functional Growth. Table 31 shows the functional size metrics investigated in Research Stage (5).

The groupings of the ratios shown in Table 31 enable each combination of level of detail for Data Functionality and Transaction Functionality to be compared. These ratios are firstly evaluated individually by obtaining their correlation with overall Functional Growth. The effects of combining multiple ratios is then analysed through the use of Multiple Linear Regression. This analysis is conducted within the different groupings described in Table 31, both to compare different levels of detail and to restrict the number of variables included in each regression model.

Table 31 - Overall Growth Functional Based Size Metrics

RET/ILF	The ratio of the number of RETs to ILFs represents the number of 'associated' entities for each main logical entity. The extent of the 'associated' entities in a project may implicitly reflect the amount of associated Transaction Functionality.
ILF/EI, ILF/EO, ILF/EQ	These ratios directly compare the Data Functionality and the 'associated' Transaction Functionality. In these cases, the Transaction Functionality is represented by the standard NESMA BFC.
RET/EI, RET/EO, RET/EQ	These ratios also directly compare the Data Functionality with the 'associated' Transaction Functionality. In these cases the Data Functionality is represented at a more detailed level by the RET component.
ILF/Add, ILF/Link, ILF/Amend, ILF/Process, ILF/Delete, ILF/Search, ILF/Output, ILF/Report, ILF/List	These ratios incorporate the more detailed level of Transaction Functionality from the Detailed Profile Classification Functionality Types. In these cases the Data Functionality is represented at the level of the ILF component.
RET/Add, RET/Link, RET/Amend, RET/Process, RET/Delete, RET/Search, RET/Output, RET/Report, RET/List	These ratios utilise the more detailed levels for both Data Functionality and Transaction Functionality.

4.4.5.5 Documentation Based Size Metrics

Using size metrics based on characteristics of the Bid documentation is based upon developing a measure of how ‘complete’ the description of the requirements is. The most basic measure would be a simple consideration of the size of the documentation in terms of a page count. Bid documentation includes varying levels of discussion on non-functional requirements, the bidding process etc. For the purpose of measuring the size of the documentation, the page count is focused solely on the functional requirements content.

The more considered approach to assessing the documentation would be to measure how detailed the contents is in terms of identifying functionality. The average amount of functionality identified on each page of the Bid documentation (i.e. the FP/Page Ratio) can be obtained by dividing the overall functional size by the page count. This will provide an indication of how ‘dense’ the description of the functional requirements is in the Bid documentation. The ratio will seek to provide insight into whether denser description reflects greater potential to be expanded into increased functionality, or that it represents a clearer specification of the project functionality less likely to change.

The size metric ratios for this stage are:

(1) – Bid Page Size: where each page that includes relevant description of the required functionality is counted to provide the documentation size.

(2) – FP/Page: where the Estimated NESMA method overall functional size at the Bid stage is divided by the above metric.

Each individual metric is assessed by determining the correlation with overall growth. The effect of combining the metrics is then assessed through the use of Multiple Linear Regression.

4.4.6 Research Stage (6) – Development and Assessment of Functional Specification Stage Size Metrics

The diminishing value of estimating the software size further into the project lifecycle has been identified in the related work. The case for undertaking size estimation at the Functional Specification stage may therefore be perceived to be less justified. This Research Stage examines whether it is possible to identify at the Functional Specification stage any size metrics associated with change in functional size without incurring the effort of detailed software sizing. There are two main approaches adopted within this stage of analysis. Firstly, an assessment is made of whether changes in the overall size can be associated with changes in specific types of functionality. This would reduce the extent of the functional sizing necessary at the Functional Specification stage. Secondly, an assessment is made of whether simple documentation based factors are associated with change in overall size. This removes the need for any form of functional sizing in indicating functional growth.

4.4.6.1 Functionality Type Size Metrics

Each of the functionality types from the Detailed Functional Profile classification in Section 4.4.1 is evaluated as a potential size metric. Section 4.4.3 produces sizes for each type of functionality at both the Bid stage and the Functional Specification Stage. This enables the relative change in size to be calculated, and for the correlation with functional growth for each individual type to be determined.

Multiple Linear Regression analysis is then undertaken to assess whether combining types of functionality provides an improved indication of functional growth.

4.4.6.2 Documentation Based Size Metrics

This analysis involves the development of simple metrics using the page sizes of the documentation for each project. The size metrics for this stage are:

(1) – Functional Specification Page Size: where each page that includes relevant description of the required functionality is counted to provide the documentation size.

(2) – Percentage Page Size Increases: calculation of the relative difference between the Bid Page Size (calculated in Section 4.4.5.5) and the Functional Specification Page Size.

Each individual metric is assessed by determining the correlation with overall growth. The effect of combining the metrics is then assessed through the use of Multiple Linear Regression.

4.5 Results

The results obtained using the Detailed Functional Profile classification of functionality types facilitate the assessment of the overall functional profile of the sample projects at both the bid and the functional specification stage of the lifecycle. The relationships between differing levels of functional profile, NESMA and Detailed, and general project metrics are then presented in order to establish the value provided at each level. These differing levels of functional profile, as well as related functional ratios, are then tested as size metrics associated with the relative change in project functional size from the bid stage to the functional specification stage. Finally, the results are presented of potential size metrics, at the functional specification stage, associated with the relative change in functional size at this later stage of the project lifecycle.

4.5.1 Overall functional profile

The first column of Table 32 shows the overall average percent change for each type of functionality in the Detailed Functional Profile classification in terms of the change in size (FP), from bid to functional specification. The overall average percent contribution to the overall functional size for each functionality type at each stage is also shown in the next two columns. This provides an indication of the significance of the contribution of each functionality type at both stages. The final two columns show the correlation between the sizes at both stages, along with its statistical significance, as an indication of the strength of the pattern demonstrated by the sample projects.

The functional profile exhibited is discussed according to the broad nature of functionality each individual type represents.

Data Functionality: Internal Data Functionality was the largest type of functionality at both the bid stage and the functional specification stage. The overall pattern for Internal Data Functionality was for a relatively modest increase in size at the later stage, but a slight decrease in the percentage contribution towards the overall functional size of the projects. This pattern suggests that Internal Data Functionality is generally specified quite clearly at the bid stage, compared with the Transaction Functionality, as it becomes a slightly less significant type of functionality despite the increase in size. The highest correlation between the two stages was demonstrated with Internal Data Functionality (significant at $p<0.01$), indicating that the largest type of functionality is also the most predictable. External Data Functionality demonstrated a similarly modest increase at the Functional Specification stage. The contribution of this type of functionality to the overall size was similar at both stages, with a statistically significant correlation evident between the bid stage and functional specification stage. The contribution, at around 1%, is the lowest of all the functionality types and would not be considered to have a high impact.

Input Functionality: Statistically significant correlations were demonstrated by three of the Input Functionality types, namely ‘Add’, ‘Link’ and ‘Process’. These Functionality types were also found to provide the smallest relative changes in size between the two stages. In each of these types the percentage contribution was slightly decreased at the functional specification stage. The largest reduction was evident in the ‘Process Functionality’ which was the only type to decrease in functional size at the later stage. This can be explained by the ‘business processes’ oriented nature of the bid documentation, reflecting the focus of the customer on specifying their requirements in this documentation. In contrast to these three types, the 'Amend' and 'Delete' types demonstrated large average increases in size, 105% and 344% respectively, at the functional specification stage. The percentage contribution increased slightly for 'Amend Functionality' at this later stage. The percentage contribution of 'Delete Functionality' almost tripled in size, although it remains one of the lowest relative contributors of functionality. Modest correlations were evident for 'Amend' and 'Delete' types, but these were not statistically significant. The under specification of these two types at the bid stage suggests that more 'generic' functionality is not explicitly addressed by the customer at the bid stage.

Output Functionality: None of the Output related functionality types demonstrated a statistically significant correlation between the bid stage and functional specification stage. 'Search Functionality' and 'List Functionality' demonstrated large overall average increases at the functional specification stage. The contribution of the 'Search' type to the overall functional size approximately doubled at this later stage, accounting for an average of 9.10% of the functional size. The extent of this increase may be slightly exaggerated due to the nature of some of the functional specifications. For some earlier projects, the less precise specification of the search requirements suggested the existence of more substantial distinct search functions. The more precise specifications found in later projects indicated that search functionality was not divided into distinct functions to the same degree. The search functionality may therefore have been overestimated for some of the earlier projects. The contribution of the 'List' type to the overall functional size had almost tripled at the functional specification stage, accounting for an average of 18.75% of the overall size. This type of functionality represents the second highest component for the dataset, and the moderately high correlation, while not statistically significant, suggests a moderately strong pattern was found. The nature of Case Management Systems provides an explanation for the significance of 'List' functionality within the sample projects. The distinction between 'Search' and 'List' functionality was not as clear in one case. For one project, it became apparent in the functional specification that, unlike the other projects, viewing lists of related records was to be predominantly catered for by firstly eliciting search criteria from the user. This type was allocated as 'Search' according to how it was specified, rather than by the broader purpose of the functionality. For this project, the functional size growth occurred in the 'Search' rather than the 'List' category. The functional sizes at each stage for 'Output Functionality' were weakly correlated, with a modest average reduction in size. There was a reduction in the percentage contribution to the overall size at the functional specification stage to approximately a third of the bid stage contribution. The significant increases in other types would explain the relative decrease in contribution of 'Output Functionality'. It could also partly be down to the less specific specification of some output functionality at the bid stage leading to this more general classification at the bid stage. 'Report

Functionality' demonstrated a modest average increase in size, but the percentage contribution to the overall size had nearly halved at the functional specification stage.

Table 32 - Overall Average Size Pattern

Factor	Average % Change in FP	Average % of Overall Bid Size	Average % of Overall Functional Specification Size	Correlation between Bid Size/Functional Specification Size	Statistical Significance *p* value (2-Tailed Test)
Internal Data Functionality	29.62%	22.82%	20.58%	**0.821**	***p*=0.002**
External Data Functionality	35.71%	0.90%	1.14%	**0.712**	***p*=0.014**
Add Functionality	18.02%	13.08%	10.22%	**0.733**	***p*=0.01**
Link Functionality	6.05%	3.48%	2.13%	**0.783**	***p*=0.004**
Amend Functionality	105.04%	6.32%	8.43%	0.548	*p*=0.081
Process Functionality	-16.04%	11.80%	7.33%	**0.664**	***p*=0.026**
Delete Functionality	344.17%	1.26%	3.43%	0.568	*p*=0.068
Search Functionality	257.96%	4.16%	9.10%	0.378	*p*=0.251
Output Functionality	-41.01%	12.05%	3.64%	0.232	*p*=0.493
Report	31.18%	11.13%	6.63%	0.089	*p*=0.796

Functionality					
List Functionality	490.17%	6.77%	18.75%	0.590	*p*=0.056
Query Functionality	235.82%	4.23%	7.24%	0.234	*p*=0.489

The extremely weak correlation for this type between the two stages indicates the volatile nature of this type of functionality in the sample projects. For some of the projects, the bid stage documentation included substantial lists of 'Report Functionality' with limited detail into the contents of each report. In contrast, the Functional Specifications frequently treated this type of functionality in a more generic way, with less distinct functions outlined. There could therefore be a disparity in how the customer anticipates the requirements of this type of functionality in the bid documentation, and in how it is subsequently documented in the functional specification by Equiniti-ICS.

Query Functionality: This type of functionality demonstrated a large average increase in size from the bid stage to the functional specification stage. The percentage contribution to the overall size had almost doubled by the later stage, now accounting for 7.24% of the overall functionality. This type of functionality was frequently not explicitly specified at the bid stage, reflecting how more generic functionality is not as thoroughly acknowledged at this stage as some other types. The distinction between 'Query Functionality' and other types of functionality was not always clear at the functional specification stage. The low correlation between the two stages for this type of functionality indicates that no consistent overall pattern is evident, with significant increases and decreases found across the projects. The percentage contribution of the 'Query' functionality to the overall functional size could change by a factor of up to 5 in this dataset.

4.5.2 Relationship of functional profile components with Overall Functional Size and Actual Development Effort

This section presents an assessment of the relationship between the functional profile components and other project attributes. Firstly, the correlation with the overall size provides a measure of how accurately the functional size can be determined from the size of an individual component. The correlations between individual components and overall size are included for the bid stage and the functional specification respectively. Table 33 presents the results from the

NESMA Functional Profile obtained using the conventional NESMA approach. The RET count is included in the results as, while not one of the BFC, is produced by using the NESMA approach. Table 34 presents the functional profile obtained using the more detailed classification developed in this research. In both tables the correlations are presented for each individual component within their respective profile. The correlations found at both the Bid and the Functional Specification stages are included for comparison. The correlation between overall Bid size and actual effort is relatively low (0.294), and provides a base figure for comparing the individual components against. The expectation is that an increase in any individual component will be reflected in an increase in the overall size for a project. The statistical significance of each correlation is therefore assessed using a 1-tail test. The use of multiple correlation significance tests on the same dependent variables in this section is controlled through the use of the Benjamini-Hochberg procedure. In both Table 33 and Table 34 each column corresponds to a separate dependent variable and is therefore regarded as a separate 'family' of results. The Factors in Table 33 and Table 34 were also considered together as part of each 'family', excluding results common to both tables. This analysis across both tables therefore involved 4 separate families each comprised of 15 unique Factors. The p values shown in these tables are for each individual correlation significance, where p<0.05 would be considered statistically significant. Results that have their statistical significance changed after applying the Benjamini-Hochberg procedure are affixed with (S) for significant or (NS) for not significant.

Table 33 – NESMA Functional Profile Correlations with Overall Size and Actual Effort

Factor	**Correlation with Overall Size (Bid)**	**Correlation with Overall Size (Functional Specification)**	**Correlation with Actual Effort (Bid)**	**Correlation with Actual Effort (Functional Specification)**
Internal Logical Files	**0.873**	**0.911**	0.567	**0.827**
Statistical Significance p value (1-Tailed Test) (Benjamini-Hochberg)	***p<0.001***	***p<0.001***	*p=0.035 (NS)*	***p=0.001***
Record Element Types	**0.727**	**0.881**	**0.731**	**0.799**
Statistical Significance p value (1-Tailed Test)	***p=0.006***	***p<0.001***	***p=0.005***	***p=0.002***

(Benjamini-Hochberg)				
External Interface Files	0.230	0.363	-0.097	0.066
Statistical Significance p value (1-Tailed Test) (Benjamini-Hochberg)	*p=0.248*	*p=0.137*	*p=0.388*	*p=0.423*
External Inputs	**0.945**	**0.952**	0.330	**0.680**
Statistical Significance p value (1-Tailed Test) (Benjamini-Hochberg)	***p<0.001***	***p<0.001***	*p=0.161*	***p=0.011***
External Outputs	**0.950**	**0.920**	0.196	0.505
Statistical Significance p value (1-Tailed Test) (Benjamini-Hochberg)	***p<0.001***	***p<0.001***	*p=0.282*	*p=0.057*
External Queries	**0.786**	**0.785**	0.090	0.327
Statistical Significance p value (1-Tailed Test) (Benjamini-Hochberg)	***p=0.002***	***p=0.002***	*p=0.397*	*p=0.163*

Table 34 – Detailed Functional Profile Correlations with Overall Size and Actual Effort

Factor	**Correlation with Overall Size (Bid)**	**Correlation with Overall Size (Functional Specification)**	**Correlation with Actual Effort (Bid)**	**Correlation with Actual Effort (Functional Specification)**
Internal Data	**0.873**	**0.911**	0.567	**0.827**

Functionality *Statistical Significance p value (1-Tailed Test)* (Benjamini-Hochberg)	***p<0.001***	***p<0.001***	*p=0.035 (NS)*	***p=0.001***
External Data Functionality	0.230	0.363	-0.097	0.066
Statistical Significance p value (1-Tailed Test) (Benjamini-Hochberg)	*p=0.248*	*p=0.137*	*p=0.388*	*p=0.423*
Add Functionality	**0.670**	**0.830**	0.192	**0.694**
Statistical Significance p value (1-Tailed Test) (Benjamini-Hochberg)	***p=0.012***	***p=0.001***	*p=0.286*	***p=0.009***
Link Functionality	0.352	**0.914**	**0.693**	**0.755**
Statistical Significance p value (1-Tailed Test) (Benjamini-Hochberg)	*p=0.144*	***p<0.001***	***p=0.009***	***p=0.004***
Amend Functionality	**0.787**	**0.828**	0.398	**0.669**
Statistical Significance p value (1-Tailed Test) (Benjamini-Hochberg)	***p=0.002***	***p=0.001***	*p=0.113*	***p=0.012***
Process Functionality	**0.813**	**0.553**	0.092	0.237
Statistical Significance p value (1-Tailed Test) (Benjamini-Hochberg)	***p=0.001***	***p=0.039***	*p=0.394*	*p=0.241*
Delete Functionality	**0.775**	**0.714**	-0.001	0.301

Statistical Significance p value (1-Tailed Test) (Benjamini-Hochberg)	***p=0.003***	***p=0.007***	*p=0.499*	*p=0.184*
Search Functionality	0.398	**0.584**	0.148	0.072
Statistical Significance p value (1-Tailed Test) (Benjamini-Hochberg)	*p=0.113*	***p=0.030***	*p=0.332*	*p=0.417*
Output Functionality	**0.533**	**0.813**	0.310	0.446
Statistical Significance p value (1-Tailed Test) (Benjamini-Hochberg)	***p=0.046***	***p=0.001***	*p=0.177*	*p=0.085*
Report Functionality	**0.735**	0.299	-0.042	0.003
Statistical Significance p value (1-Tailed Test) (Benjamini-Hochberg)	***p=0.005***	*p=0.186*	*p=0.451*	*p=-0.496*
List Functionality	**0.744**	**0.843**	0.621	**0.768**
Statistical Significance p value (1-Tailed Test) (Benjamini-Hochberg)	***p=0.004***	***p=0.001***	*p=0.021 (NS)*	***p=0.003***
Query Functionality	**0.786**	**0.785**	0.090	0.327
Statistical Significance p value (1-Tailed Test) (Benjamini-Hochberg)	***p=0.002***	***p=0.002***	*p=0.397*	*p=0.163*

Table 33 shows that the ILF component initially demonstrated a statistically significant correlation with both the overall size and the actual effort at both stages. The bid stage correlation with actual effort was, however, no longer considered statistically significant after applying the Benjamini-Hochberg procedure. The correlations for this BFC were very strong for overall size, and the Functional Specification stage showed improvements for both size and actual effort. The RET count effectively provides a more detailed measure of the internal data functionality than the ILF size (from the Estimated NESMA method), by identifying exactly how many RETs the ILFs are comprised of. The correlation of RETs with the overall size was statistically significant at both stages, although slightly lower than the ILF values. The correlation of RETs with actual effort demonstrated a higher level of statistical significance than the ILF component at the bid stage. The more detailed consideration of internal data functionality would therefore seem beneficial at the Bid Stage of a project. The correlation was similar to the ILF component at the functional specification stage, suggesting that the more developed data model at this later stage has evened out the significance of RETs and ILFs in determining actual effort. The question remains whether the ILF component provides the more accurate indication of the overall size of a project (as measured from the Functional Specification), or merely a superior indication of the less accurate overall size estimated at the Bid Stage. The EIF component demonstrated weak correlations throughout, and would therefore be of no value as in individual measurement. The insignificance of this component in contributing to the overall size of a project expectedly proved to offer no insight into the sample projects.

For the EI, EO, and EQ components, the correlations with overall size were statistically significant at both stages. In each case there was no significant difference in the results obtained at the Bid and Functional Specification stages. The EI and EO components demonstrated the highest correlation with overall size of all of the components. The relationship with actual effort was notably weaker, with no statistically significant correlations found at the Bid stage. The correlations improved by the Functional Specification stage, although only the EI components demonstrated a statistically significant level. The Transaction Functions have not been demonstrated to provide an accurate indication of actual effort, but do provide a good indication of overall size (as determined as the same stage).

For the more detailed functional profile in Table 34, the ILF, EIF and EQ are directly represented in the Detailed Functional Profile classification and therefore the results are identical to those in Table 33. The interest therefore lies in how the more detailed breakdown of Inputs and Outputs compare with the results in Table 33. The results for the EI related components were weaker than for the main EI component, in terms of the relationship with overall size. However, in most cases the correlations were statistically significant. Unlike with the main EI component, there were changes in the statistical significance of the correlations between the two stages for most components. The significance for the ‘Add’ and ‘Link’ components increased at the Functional Specification stage, and decreased for the ‘Process’ and ‘Delete’ components. This supports the observation that some types of functionality are more difficult to identify at the Bid stage, and therefore the degree to which they are reflected in the overall size would be expected to vary. These increases and decreases would effectively cancel each other out when combined into the overall EI component. The more detailed functional profile also revealed a statistically significant correlation with actual effort at the Bid stage. The ‘Link’ component was found to be strongly correlated at this first stage, and the correlation increased at the Functional Specification stage ($p<0.005$) and exceeded that obtained with the EI component. The ‘Amend’ component also had a statistically significant correlation at the Functional Specification stage.

The EO related components demonstrated a similar pattern to the EI related components. The correlations with overall size were weaker than found with the overall EO component, but in most cases they were statistically significant. The level of significance also tended to either increase or decrease between the Bid stage and the Functional Specification stage for the components. These changes effectively cancelled themselves out for the overall EO component. The strongest correlations were found with the ‘List’ component, which would be expected to be strongly related to the number of Data Functions in a project. This component was also quite strongly correlated with the actual effort. While the correlation was not significant at the Bid stage after applying the Benjamini-Hochberg procedure, a significance level of $p<0.005$ was found at the Functional Specification stage. The Detailed Functional profile therefore provided insight into the actual effort not available with the NESMA Functional Profile. In particular, the correlations at the Bid stage were stronger for the ‘Link’ and ‘List’ components than for the ILF component.

4.5.2.1 Multiple Linear Regression Analysis – Multicollinearity Testing

The use of the different levels of functional profile to determine overall size and actual effort was evaluated using Multiple Linear Regression Analysis. This enables the components of the functional profiles to be evaluated together to determine if multiple components can more effectively model the relationship with overall size and actual effort than any individual component. The Literature Review in Chapter 2 highlighted research demonstrating the existence of significant correlations between the individual BFC used by the NESMA method. The existence of such multicollinearity has implications for the use of Multiple Linear Regression. When some of the independent variables are strongly correlated with each other, the standard error associated with individual independent variables and their statistical significance can change erratically in response to small changes to the model or the data. The use of diagnostic tests for multicollinearity was therefore conducted for regression models with both the NESMA Functional Profile and Detailed Functional Profile. Table 35 shows the Tolerance value and Variance Inflation Factors for the variables for regression models with both Profiles. Tolerance values of less than 0.1 indicate that the variance for a variable can be accounted for by other variables to such an extent that it the variable may be considered redundant. As a rule of thumb a Variance Inflation Factor greater than 10 is considered to problematic, with values between 5 and 10 considered to be potentially problematic. With the danger of overfitting regression models that are based on a dataset of limited size, the preferred remedy is to remove problematic variables. The procedure for completing this testing was to identify model variables with a Variance Inflation Factor in the problematic range, consider the cross correlations with other variables, and if necessary remove the most problematic variable from the model. This stage is then repeated for the resulting model until a model is obtained that contains no further problematic variables. For the regression model using NESMA Functional Profile the EI variable exceeds this value, while the ILF variable is relatively close to 10. Examination of the cross correlations indicated that strong correlations with other variables were most commonly found with these two variables. These two variables were removed from the regression model in separate stages and the resulting diagnostic tests were examined again each time.

Table 35 – Regression Model Initial Multicollinearity Statistics

Profile Type	Model Variable	Tolerance	Variance Inflation Factor
NESMA Functional Profile	ILF	0.115	8.691
	RET	0.231	4.329
	EIF	0.813	1.230
	EI	0.057	17.512
	EO	0.256	3.914
	EQ	0.166	6.012
Detailed Functional Profile	External	0.014	72.397
	Add	0.032	31.655
	Link	0.005	190.844
	Amend	0.037	26.798
	Process	0.007	136.971
	Search	0.068	14.734
	Output	0.043	23.313
	Report	0.196	5.102
	List	0.003	317.275
	Query	0.046	21.855

The results, shown in Table 36, indicated that none of the remaining model variables exhibited a problematic Variance Inflation Factor. These four variables therefore formed the NESMA Functional Profile for the use of regression models using the Enter method. For the use of the Stepwise method the original six variables are retained for initial inclusion, with the regression method selecting the most suitable variables.

For the regression model using the Detailed Functional Profile, Table 35 shows that all but one of the model variables exceeded a value of 10 for the Variance Inflation Factor. In this case the Internal and Delete variables had already been removed from the model at this initial stage due to having a Tolerance value of 0. Testing then proceeded by removing the variable with the highest value for Variance Inflation Factor and examining the resulting values for the remaining variables. This process was repeated until none of the remaining variables produced a problematic value for its Variance Inflation Factor.

Table 36 – Regression Model Final Multicollinearity Statistics

Profile Type	Model Variable	Tolerance	Variance Inflation Factor
NESMA Functional Profile	RET	0.594	1.683
	EIF	0.918	1.090
	EO	0.514	1.946
	EQ	0.507	1.974
Detailed Functional Profile	External	0.662	1.511
	Add	0.477	2.097
	Link	0.512	1.953
	Search	0.585	1.709
	Output	0.321	3.115
	Report	0.582	1.718
	Query	0.279	3.589

The final Detailed Functional Profile, shown in Table 36, contains 7 variables from the original set of 12 variables. As with the NESMA Functional Profile, these 7 variables are included in regression models for the use of the 'Enter' method only, with all 12 variables retained for use with the Stepwise method.

4.5.2.2 Multiple Linear Regression Analysis – Overall Bid Size and Actual Effort

Regression Analysis was performed using both the 'Enter' and 'Stepwise' methods, for the standard NESMA Functional Profile and the Detailed Functional Profile. Table 37 shows the results, where applicable, from performing this analysis. The correlation, 'R', is shown along with the strength of this correlation, 'R Square'. The 'Adjusted R Square' provided a measure of how strongly this correlation would be replicated beyond the dataset used in this analysis. The 'Statistical Significance' accounts for the number of test cases and the number of model variables in reporting the significance of the results. The Beta coefficients are provided for each model variable, which can provide an indication of the relative strength of these predictors within the regression model. The statistical significance of the effect of each model variable is also shown in the table. The standard error of the estimate is provided for both the overall model and for each individual model variable, providing an indication of how close on average the predicted values are to the observed values. The use of the 'Enter' method for Overall Bid Size is of negligible interest, as estimating the complete profile would provide that overall value. The interest is therefore with the use of the 'Stepwise' method and whether a more restricted subset of each profile could determine overall size

The results for the NESMA Functional Profile show very highly statistically significant results, $p<0.001$, in explaining the variation in the Overall Bid Size. The use of the EO and EI components were selected by the 'Stepwise' method, with each variable demonstrating a similar statistically high significant effect in the model. The Detailed Functional Profile analysis also provided a very highly statistically significant result at $p<0.001$. The most effective model included the 'Internal File', 'Query' and 'Report' components, with a similar level of statistical significance found for each variable. The overall performance for this Detailed Functional Profile was similar to that found with the NEMSA Functional Profile, although there was a higher standard error of the estimate for this more detailed sizing level. The results of the Multiple Linear Regression suggest that some improvement can be found by considering more than a single component in predicting the Overall Bid Size. The significance of the improvements is limited due to the high correlation found between Overall Bid Size and some of the individual components.

The results from performing the Multiple Linear Regression Analysis with the Overall Development Effort are presented in Table 38. The use of the 'Enter' method with the NESMA Functional Profile produced a relatively high correlation with the Overall Actual Development Effort.

Table 37 - Overall Bid Size Multiple Linear Regression Analysis

Model Parameters	Overall Size (Bid) NESMA Functional Profile			**Regression Method**	Stepwise
Model Variables	**R**	**R Square**	**Adjusted R Square**	**Standard Error of Estimate**	**Statistical Significance**
EO, EI	0.998	0.997	0.996	27.991	*p*<0.001
Coefficients	**Unstandardized Coefficients**		**Standardized Coefficients**	**t**	**Statistical Significance**
	B	**Std Error**	**Beta**		
(Constant)	51.977	18.187		2.858	*p*=0.021
EO	5.301	0.330	0.537	16.057	*p*<0.001
EI	5.847	0.380	0.515	15.406	*p*<0.001
Model Parameters	Overall Size (Bid) Detailed Functional Profile			**Regression Method**	Stepwise
Model Variables	**R**	**R Square**	**Adjusted R Square**	**Standard Error of Estimate**	**Statistical Significance**
Internal, Query, Report	0.996	0.992	0.988	47.536	*p*<0.001
Coefficients	**Unstandardized Coefficients**		**Standardized Coefficients**	**t**	**Statistical Significance**
	B	**Std Error**	**Beta**		
(Constant)	-32.503	48.855		-.665	*p*=0.527
Internal	3.322	0.373	0.418	8.896	*p*<0.001
Query	4.863	0.440	0.450	11.062	*p*<0.001
Report	0.902	0.100	0.372	9.048	*p*<0.001

The significantly lower ‘Adjusted R Square’ value indicates that this correlation would not be found outside of this dataset. The results were not found to be statistically significant in this case. The use of the 'Stepwise' method with the NESMA Functional Profile did produce highly statistically significant results. The combination of the RET and EI components demonstrated the most accurate explanation of the variance in the actual effort, significant at *p*=0.006. This model provided a significant improvement over the use of just the RET component, although it was the RET component that had the most significant effect in the model. The standard error of the estimate was the lowest of the regression models used in this section for actual development effort. The use of the 'Enter' method for the Detailed Functional Profile demonstrated a high correlation, but as with the NESMA Functional Profile the 'Adjusted R Square' was much lower. The results for this regression model were therefore not found to be statistically significant. The use of the 'Stepwise' method for the Detailed Functional Profile did produce a statistically significant result (*p*=0.018) for the inclusion of the 'Link' component in the model. This model did, however, demonstrate the highest standard error of the estimate for actual development effort.

The use of Multiple Linear Regression Analysis with Overall Actual Development Effort suggests that statistically significant improvements can be obtained compared to using individual components. The use of the NESMA Functional Profile provided the best performance in explaining variance in the actual development effort.

Table 38 - Overall Actual Development Effort Multiple Linear Regression Analysis

Model Parameters	Overall Actual Effort NESMA Functional Profile			**Regression Method**	Enter
Model Variables	**R**	**R Square**	**Adjusted R Square**	**Standard Error of Estimate**	**Statistical Significance**
RET, EIF, EO, EQ	0.845	0.714	0.523	568.000	*p*=0.074
Coefficients	**Unstandardized Coefficients**		**Standardized Coefficients**	**t**	**Statistical Significance**
	B	**Std Error**	**Beta**		
(Constant)	-257.906	405.258		-0.636	*p*=0.548
RET	37.410	10.135	1.046	3.691	*p*=0.010
EIF	-40.383	153.561	-0.060	-0.263	*p*=0.801
EO	-1.441	5.597	-0.078	-0.257	*p*=0.805
EQ	-36.794	24.691	-0.457	-1.490	*p*=0.187

Model Parameters	Overall Actual Effort Detailed Functional Profile			**Regression Method**	Enter
Model Variables	**R**	**R Square**	**Adjusted** **R Square**	**Standard Error of Estimate**	**Statistical Significance**
External, Add, Link, Search, Output, Report, Query	0.946	0.895	0.650	486.692	*p*=0.158
Coefficients	**Unstandardized** **Coefficients**		**Standardized Coefficients**	**t**	**Statistical Significance**
	B	**Std Error**	**Beta**		
(Constant)	-69.171	428.828		-0.161	*p*=0.882
External	16.763	30.983	0.124	0.541	*p*=0.626
Add	0.385	4.502	.0023	0.086	*p*=0.937
Link	22.720	5.606	1.060	4.053	*p*=0.027
Search	-13.195	7.737	-0.417	-1.705	*p*=0.187
Output	13.613	4.386	1.026	3.104	*p*=0.053
Report	0.982	1.108	0.218	0.887	*p*=0.441
Query	-17.921	7.132	-0.891	-2.513	*p*=0.087
Model Parameters	Overall Actual Effort NESMA Functional Profile			**Regression Method**	Stepwise
Model Variables	**R**	**R Square**	**Adjusted** **R Square**	**Standard Error of Estimate**	**Statistical Significance**
RET, EI	0.852	0.727	0.658	480.599	*p*=0.006
Coefficients	**Unstandardized** **Coefficients**		**Standardized Coefficients**	**t**	**Statistical Significance**
	B	**Std Error**	**Beta**		
(Constant)	-62.518	325.598		-0.192	*p*=0.853
RET	47.596	11.196	1.331	4.251	*p*=0.003

EI	-15.703	6.609	-0.744	-2.376	*p*=0.045
Model Parameters	Overall Actual Effort Detailed Functional Profile			**Regression Method**	Stepwise
Model Variables	**R**	**R Square**	**Adjusted R Square**	**Standard Error of Estimate**	**Statistical Significance**
Link	0.693	0.480	0.423	624.686	*p*=0.018
Coefficients	**Unstandardized Coefficients**		**Standardized Coefficients**	**t**	**Statistical Significance**
	B	**Std Error**	**Beta**		
(Constant)	419.688	240.638		1.744	*p*=0.115
Link	14.851	5.148	0.693	2.885	*p*=0.018

4.5.3 Overall Bid Stage Size Metrics

This section presents the results of assessing potential size metrics associated with how the functional size of projects may change between the Bid Stage and the Functional Specification. Section 4.5.3.1 is concerned with relating Bid Stage sizing data to the overall Functional Specification size. This analysis involves relationships between different levels of Bid Stage sizing data and the overall Functional Specification size. Section 4.5.3.2 is concerned with how functional sizing metrics are associated with the relative growth in functional size between the Bid Stage and the Functional Specification.

4.5.3.1 Analysis for Overall Functional Specification Size

The use of Bid stage functional sizing data with the overall Functional Specification size was evaluated at three different levels of detail. First, the relationship between the highest level, overall Bid Size, and the overall Functional Specification size is shown in Figure 4.2.

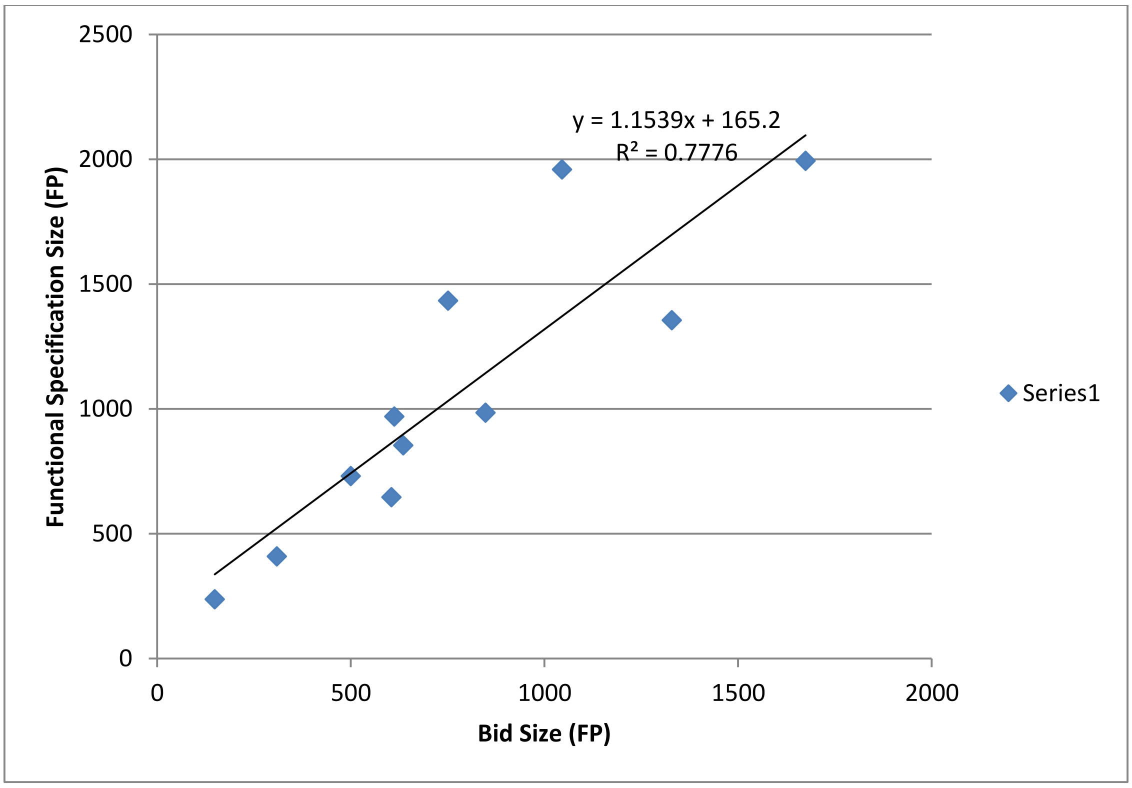

Figure 4.2 - Correlation between Bid Size and Functional Specification Size

The graph indicates that a linear relationship exists between the Bid Size and the Functional Specification size. The correlation between the overall functional sizes, r=0.882, is highly statistically significant (*p*<0.01). The subsequent levels of detail involved the use of the Functional Profile and the Detailed Profile, including both individual components and Multiple Linear Regression Analysis.

The correlations between bid level components and the overall functional specification size are presented in Tables 39 and 40. There is no a priori expectation that the Bid size should be positively correlated with the Functional Specification size. The determination of statistical significance therefore used a 2-tailed test in this analysis. For the analysis of multiple correlation significance tests using the Benjamini-Hochberg procedure, the results from both tables were considered together as part of the same 'family'. The size of this 'family' was therefore 15, after accounting for results common to both tables. For the NESMA Functional Profile components, Table 39 shows that initially four of the components demonstrated correlations with the overall Functional Specification size that have a high level of statistical significance (*p*<0.01). In particular, the ILF and RET components provide a slightly higher correlation than found with the overall Bid size. The use of the Benjamini-Hochberg procedure in this instance resulted in the EQ component additionally being considered statistically significant.

Table 39 - Relationship between Bid NESMA Functional Profile and Overall Functional Specification Size

Functionality Type	**Correlation with Overall Functional Specification Size**	**Statistical Significance (2-tailed test) (Benjamini-Hochberg)**
Internal Logical Files	**0.904**	***p*<0.001**
Record Element Types	**0.899**	***p*<0.001**
External Interface Files	-0.023	*p*=0.946
External Inputs	**0.870**	***p*<0.001**
External Outputs	**0.805**	***p*=0.003**
External Queries	**0.584**	***p*=0.059 (S)**

In Table 40, it can be seen that half of the Detailed Functional Profile components initially demonstrated a statistically significant correlation with the overall Functional Specification size. The number of significant results increased further after applying the Benjamini-Hochberg procedure. In most cases, however, the correlations are less significant than those found with the NESMA Functional Profile components. This suggests that the effect of combining Detailed Functional Profile components into NESMA Functional Profile components can provide a stronger relationship with the overall Functional Specification size.

Table 40 - Relationship between Bid Detailed Functional Profile and Overall Functional Specification Size

Functionality Type	**Correlation with Overall Functional Specification Size**	**Statistical Significance *p* value (2-tailed test) (Benjamini-Hochberg)**
Internal Data Functionality	**0.904**	***p*<0.001**
External Data Functionality	-0.023	*p*=0.946
Add Functionality	**0.597**	***p*=0.052 (S)**
Link Functionality	**0.674**	***p*=0.023**
Amend Functionality	**0.753**	***p*=0.007**
Process Functionality	**0.631**	***p*=0.037**
Delete Functionality	**0.637**	***p*=0.035**
Search Functionality	0.420	*p*=0.198
Output Functionality	0.443	*p*=0.173
Report Functionality	**0.574**	***p*=0.065 (S)**
List Functionality	**0.792**	***p*=0.004**
Query Functionality	**0.584**	***p*=0.058 (S)**

4.5.3.2 Multiple Linear Regression Analysis – Functional Profiles and Overall Functional Specification Size

The results of performing Multiple Linear Regression Analysis with the Overall Functional Specification Size are presented in Table 41. The use of the 'Enter' method with the NESMA Functional Profile demonstrated a very highly statistically significant explanation, at *p*<0.001, of the variance of the Overall Functional Specification size. The standardized coefficients for the RET and EO variables indicated that these two components had the most effect on the predicted Functional Specification size. The results obtained with this method slightly exceeded that obtained with the use of the 'Stepwise' method, with the latter method selecting all but one of the same model variables used with the former method. The use of the 'Enter' method with the Detailed Functional Profile produced a statistically significant result for the regression model, although with a higher *p* value of 0.022 and approximately twice the standard error of the estimate than was found with the NESMA Functional Profile. The use of the 'Stepwise' method with the Detailed Functional Profile improved the level of statistical significance to *p*<0.001, in part due to including only two variables in the regression model. The standard error of the estimate was, however, further increased in this case suggesting that the model may be less reliable.

The results of this analysis suggest that the combination of the results of multiple Bid level components is more strongly associated with the overall Functional Specification size than any individual sizing component. It would appear that there was no advantage, however, in adopting a more detailed sizing estimate than provided by the NESMA Functional Profile.

Table 41 - Overall Size (Functional Specification) Multiple Linear Regression Analysis

Model Parameters	Overall Size (Functional Specification) NESMA Functional Profile			**Regression Method**	Enter
Model Variables	**R**	**R Square**	**Adjusted R Square**	**Standard Error of Estimate**	**Statistical Significance**
RET, EIF, EO, EQ	0.994	0.988	0.979	83.444	*p*<0.001
Coefficients	**Unstandardized Coefficients**		**Standardized Coefficients**	**t**	**Statistical Significance**

	B	Std Error	Beta		
(Constant)	82.040	59.535		1.378	*p*=0.217
RET	17.019	1.489	0.676	11.430	*p*<0.001
EIF	-91.167	22.559	-0.192	-4.041	*p*=0.007
EO	7.264	0.822	0.562	8.834	*p*<0.001
EQ	-7.529	3.627	-0.133	-2.076	*p*=0.083
Model Parameters	Overall Size (Functional Specification) Detailed Functional Profile			**Regression Method**	Enter
Model Variables	**R**	**R Square**	**Adjusted R Square**	**Standard Error of Estimate**	**Statistical Significance**
External, Add, Link, Search, Output, Report, Query	0.987	0.974	0.912	171.430	*p*=0.022
Coefficients	**Unstandardized Coefficients**		**Standardized Coefficients**	**t**	**Statistical Significance**
	B	**Std Error**	**Beta**		
(Constant)	134.510	151.048		0.891	*p*=0.439
External	-14.350	10.913	-0.151	-1.315	*p*=0.280
Add	2.771	1.586	0.237	1.748	*p*=0.179
Link	9.166	1.975	0.608	4.642	*p*=0.019
Search	-1.943	2.725	-0.087	-0.713	*p*=0.527
Output	5.319	1.545	0.570	3.443	*p*=0.041
Report	1.649	0.390	0.519	4.226	*p*=0.024
Query	-2.303	2.512	-0.163	-0.917	*p*=0.427
Model Parameters	Overall Size (Functional Specification) NESMA Functional Profile			**Regression Method**	Stepwise
Model Variables	**R**	**R Square**	**Adjusted R Square**	**Standard Error of Estimate**	**Statistical Significance**

RET, EO, EIF	0.989	0.979	0.969	101.260	*p*<0.001
Coefficients	**Unstandardized Coefficients**		**Standardized Coefficients**	**t**	**Statistical Significance**
	B	**Std Error**	**Beta**		
(Constant)	106.689	70.795		1.507	*p*=0.176
RET	15.911	1.687	0.632	9.433	*p*<0.001
EO	6.505	0.894	0.504	7.278	*p*<0.001
EIF	-95.783	27.243	-0.202	-3.516	*p*=0.010
Model Parameters	Overall Size (Functional Specification) Detailed Functional Profile			**Regression Method**	Stepwise
Model Variables	**R**	**R Square**	**Adjusted R Square**	**Standard Error of Estimate**	**Statistical Significance**
Internal, Link	0.948	0.899	0.874	205.411	*p*<0.001
Coefficients	**Unstandardized Coefficients**		**Standardized Coefficients**	**t**	**Statistical Significance**
	B	**Std Error**	**Beta**		
(Constant)	-305.182	195.629		-1.560	*p*=0.157
Internal	7.827	1.319	0.753	5.935	*p*<0.001
Link	4.887	1.912	0.324	2.556	*p*=0.034

4.5.3.3 Analysis for Growth in Overall Functional Size – Bid Functional Sizing Data

The use of Bid stage functional sizing data involved the sizing metrics developed in Section 4.4.5.4. Table 42 shows the correlations with functional growth produced by the Functional Ratios. These ratios were designed to measure the degree of functionality associated with the internal data files. The a priori expectation is that the less associated functionality identified at the Bid stage, the greater potential for growth in functional size to occur later in the project. The statistical significance of the correlations was therefore evaluated using a 1-tailed test. The number of projects included for each ratio is shown, due to the absence of values for some ratios in some of the projects. The Benjamini-Hochberg procedure was again applied to account for the use of multiple correlation significance tests. The ‘family’ of results for this analysis included all 25 sizing metrics shown in Table 42.

The RET/ILF ratio provides a measure of the associated data functionality for each main ‘logical’ entity in a system. Table 42 indicates a moderate correlation with Functional Growth for this ratio, suggesting that while the relationship is not statistically significant, there may be some influence evident from the associated data functionality. The ratios developed using the main Transaction Function components (EI, EO, EQ) demonstrate that higher correlations are obtained when they are used in conjunction with the RET component rather than the ILF component. The RET/EI and RET/EQ ratios provided the best correlations, initially considered statistically significant at *p*=0.023 and *p*=0.007 respectively. The use of the Benjamini-Hochberg procedure negated the significance for both of these ratios in this instance. These results nonetheless indicate that the relationship between associated data functionality and transaction functionality provides the best indication of subsequent functional growth. The ratios developed using the Detailed Functional Profile components were less effective in indicating a relationship with functional growth. The only ratio that initially provided a statistically significant correlation from these ratios was RET/Delete. The relatively small contribution of ‘Delete’ functionality to the projects, further indicated by the fact that three of the projects had no value for the ratio, suggests that the RET/Delete ratio would be less likely to be reliable outside of this dataset.

Table 42 - Functional Ratios (Bid Stage) and Overall Functional Growth

Ratio	**Correlation with Functional Growth**	**Number of Projects**	**Statistical Significance *p* value (1-tailed test) (Benjamini-Hochberg)**
RET/ILF	0.412	11	*p*=0.104
ILF/EI	0.109	11	*p*=0.375
ILF/EO	0.191	11	*p*=0.287
ILF/EQ	0.524	11	*p*=0.049 (NS)
RET/EI	0.610	11	*p*=0.023 (NS)

Ratio	**Correlation with Functional Growth**	**Number of Projects**	**Statistical Significance *p* value (1-tailed test) (Benjamini-Hochberg)**
RET/EO	0.421	11	*p*=0.099
RET/EQ	0.711	11	*p*=0.007 (NS)
ILF/Add	-0.040	11	*p*=0.453
ILF/Link	-0.462	10	*p*=0.089
ILF/Amend	0.178	11	*p*=0.300
ILF/Process	-0.097	11	*p*=0.389
ILF/Delete	0.583	8	*p*=0.065
ILF/Search	0.303	10	*p*=0.197
ILF/Output	-0.167	11	*p*=0.312
ILF/Report	0.142	10	*p*=0.348
ILF/List	0.166	11	*p*=0.313
RET/Add	0.370	11	*p*=0.132
RET/Link	-0.426	10	*p*=0.110
RET/Amend	0.410	11	*p*=0.105
RET/Process	0.123	11	*p*=0.360
RET/Delete	0.658	8	*p*=0.038 (NS)
RET/Search	0.422	10	*p*=0.112
RET/Output	0.013	11	*p*=0.484

Ratio	**Correlation with Functional Growth**	**Number of Projects**	**Statistical Significance *p* value (1-tailed test) (Benjamini-Hochberg)**
RET/Report	0.206	10	*p*=0.284
RET/List	0.210	11	*p*=0.268

4.5.3.4 Multiple Linear Regression Analysis – Functional Metrics and Overall Functional Growth

The result of using Bid functional sizing data to explain overall functional growth using Multiple Linear Regression Analysis is presented in Table 43. The use of the 'Enter' method demonstrated a high correlation between the NESMA Functional Profile and functional growth. Although the inclusion of four model variables produced a reduction in the 'Adjusted R Square' value, the results of using this model were deemed to be statistically significant at *p*=0.047. The RET component was the only variable found to have a statistically significant effect in the model. The use of the 'Stepwise' method did not produce any results, due to none of the individual components demonstrating a correlation sufficient to be entered into the model. The use of the Detailed Functional Profile did produce a higher 'Adjusted R Square' value with the 'Enter' method than the NESMA Functional Profile. The use of seven model variables meant that the results were only slightly more statistically significant at *p*=0.041, but the standard error of the estimate was reduced from 19.317 to 11.067. The use of the 'Stepwise' method did not produce any results, with no variables entered into the model.

Due to the number of potential model variables, the use of the Functional Ratios was separated into those derived from the NESMA Functional Profile components and those derived from the Detailed Functional Profile components. Multicollinearity testing was applied to both of these ratio profiles using the same approach described in Section 4.5.2.1. The model variables listed for the results of 'Enter' method with these ratio profiles in Table 43 are therefore the final sets after variables with problematic Variance Inflation Factors had been removed. The use of the 'Enter' method for the Functional Ratios derived from the NESMA Functional Profile demonstrated a relatively high correlation, but the lower 'Adjusted R Square' value resulted from the inclusion of four model variables. The result of this model was correspondingly found to not be statistically significant. The use of the 'Stepwise' method did provide a statistically significant result for the model produced. In this case, however, only the 'RETEQ' ratio was included in the model, suggesting that no advantage could be found compared to the use of the individual ratios to explain functional growth. For the Functional Ratios derived from the Detailed Functional Profile, the use of the ‘Enter’ method produced a high ‘Adjusted R Square’ value despite the inclusion of six model variables. This result was the most statistically significant at *p*=0.01, with five of the model variables demonstrating individual statistically significant effects. The lowest standard error of estimate, at 7.112, would indicate that this was the most reliable regression model for explaining overall growth in functional size. The use of the ‘Stepwise’ method did not produce any results for these functional ratios.

Table 43 - Bid Stage Functional Sizing Metrics Multiple Linear Regression Analysis

Model Parameters	Overall Growth in Size NESMA Functional Profile			**Regression Method**	Enter
Model Variables	**R**	**R Square**	**Adjusted R Square**	**Standard Error of Estimate**	**Statistical Significance**
RET, EIF, EO, EQ	0.870	0.757	0.595	19.31687	*p*=0.047
Coefficients	**Unstandardized Coefficients**		**Standardized Coefficients**	**t**	**Statistical Significance**
	B	**Std Error**	**Beta**		
(Constant)	36.169	13.782		2.624	*p*=0.039
RET	1.026	0.345	0.777	2.976	*p*=0.025
EIF	-10.673	5.222	-0.429	-2.044	*p*=0.087
EO	-0.173	0.190	-0.255	-0.908	*p*=0.399
EQ	-1.776	0.840	-0.598	-2.115	*p*=0.079
Model Parameters	Overall Growth in Size Detailed Functional Profile			**Regression Method**	Enter
Model Variables	**R**	**R Square**	**Adjusted R Square**	**Standard Error of Estimate**	**Statistical Significance**
External, Add, Link, Search, Output,	0.980	0.960	0.867	11.06685	*p*=0.041

Report, Query					
Coefficients	**Unstandardized Coefficients**		**Standardized Coefficients**	**t**	**Statistical Significance**
	B	**Std Error**	**Beta**		
(Constant)	42.558	9.751		4.364	*p*=0.022
External	-1.350	0.705	0-.272	-1.917	*p*=0.151
Add	0.000	0.102	0-.001	-0.003	*p*=0.998
Link	0.687	0.127	0.868	5.389	*p*=0.013
Search	-0.571	0.176	-0.489	-3.247	*p*=0.048
Output	0.371	0.100	0.757	3.723	*p*=0.034
Report	0.010	0.025	0.062	0.408	*p*=0.710
Query	-0.726	0.162	-0.978	-4.479	*p*=0.021
Model Parameters	Overall Growth in Size NESMA Functional Ratios			**Regression Method**	Enter
Model Variables	**R**	**R Square**	**Adjusted R Square**	**Standard Error of Estimate**	**Statistical Significance**
RET/ILF, ILF/EI, RET/EO, RET/EQ	0.847	0.717	0.528	20.86780	*p*=0.072
Coefficients	**Unstandardized Coefficients**		**Standardized Coefficients**	**t**	**Statistical Significance**
	B	**Std Error**	**Beta**		
(Constant)	-46.035	46.383		0-.992	*p*=0.359
RET/ILF	22.151	15.001	0.398	1.477	*p*=0.190
ILF/EI	-3.262	80.273	-0.012	-0.041	*p*=0.969
RET/EO	9.815	15.997	0.144	0.614	*p*=0.562
RET/EQ	5.644	2.219	0.694	2.543	*p*=0.044
Model Parameters	Overall Growth in Size Detailed Functional Ratios			**Regression Method**	Enter

Model Variables	**R**	**R Square**	**Adjusted R Square**	**Standard Error of Estimate**	**Statistical Significance**
ILF/Amend, ILF/Search, ILF/Output, RET/Amend, RET/Process, RET/List	0.991	0.983	0.949	7.11203	*p*=0.010
Coefficients	**Unstandardized Coefficients**		**Standardized Coefficients**	**t**	**Statistical Significance**
	B	**Std Error**	**Beta**		
(Constant)	-189.442	23.625		-8.019	*p*=0.004
ILF/Amend	11.391	23.090	0.045	0.493	*p*=0.656
ILF/Search	52.406	5.342	0.903	9.811	*p*=0.002
ILF/Output	85.174	13.468	0.731	6.324	*p*=0.008
RET/Amend	168.811	14.398	1.453	11.724	*p*=0.001
RET/Process	-42.826	7.758	-0.538	-5.520	*p*=0.012
RET/List	8.370	1.221	0.563	6.856	*p*=0.006
Model Parameters	Overall Growth in Size NESMA Functional Ratios			**Regression Method**	Stepwise
Model Variables	**R**	**R Square**	**Adjusted R Square**	**Standard Error of Estimate**	**Statistical Significance**
RET/EQ	0.711	0.506	0.451	22.49009	*p*=0.014
Coefficients	**Unstandardized Coefficients**		**Standardized Coefficients**	**t**	**Statistical Significance**
	B	**Std Error**	**Beta**		
(Constant)	1.396	14.758		0.095	*p*=0.927
RET/EQ	5.789	1.906	0.711	3.037	*p*=0.014

4.5.3.5 Analysis for Growth in Overall Functional Size – Bid Documentation Data

Table 44 shows the results of using simple documentation based metrics to explain potential functional growth. The a priori expectation was that there was no basis for anticipating a positive (or negative) correlation with functional growth to be found. The statistical significance of the correlations was therefore determined using a 2-tailed test.

The use of the Bid Page Size was found to demonstrate a moderate negative correlation with functional growth. The result is not considered to be statistically significant, although it suggests that smaller bid documentation was to some degree associated with functional growth. The 'FP/Page' ratio in comparison demonstrated a weaker, positive correlation with functional growth. The degree of how 'dense' the bid documentation is in terms of functionality was therefore more loosely associated with functional growth. The use of Multiple Linear Regression did not produce any improvement in correlation with Functional Growth. The analysis of documentation based factors indicates that they cannot independently explain the occurrence of functional growth in this dataset. The role, if any, played by the characteristics of the bid documentation is therefore unclear.

Table 44 - Documentation Metrics (Bid Stage) and Overall Functional Growth

Variable	**Correlation with Functional Growth**	**Statistical Significance *p* value (2-tailed test) (Benjamini-Hochberg)**
Bid Page Size	-0.455	*p*=0.160
FP/Page	0.336	*p*=0.312

4.5.4 Overall Functional Specification Stage Size Metrics

This section presents the results of two approaches to assessing the association with changes in the functional size of projects from the Bid Stage to the Functional Specification. Firstly, the analysis focuses on the use of changes in the size of individual functionality types to explain the overall functional size change. Secondly, the analysis addresses the use of documentation based metrics to explain change in functional size.

4.5.4.1 Analysis for Growth in Overall Functional Size – Functional Sizing Data

The relationship, in terms of change in functional size, between the individual functionality types from the Detailed Functional Profile and the overall functional size are presented in Table 45. The a priori expectation was that changes in individual functionality types should not necessarily be positively correlated with changes in overall functional size. The statistical significance of the results was therefore determined using a 2-tailed test. The Benjamini-Hochberg procedure analysis for these tests involved a ‘family’ of 12 results. This analysis did not, on this occasion, lead to the statistical significance for any of the results being changed. The results show that ‘Internal Data Functionality’ demonstrated the highest correlation, statistically significant at p=0.008, with the overall functional growth. This is consistent with the overall pattern of results in this research, wherein this type of functionality is representative of the overall functional size. The statistically significant correlation found for the ‘External Data Functionality’ is not considered important due to the established insignificance of this type of functionality within the sample projects. In terms of the remaining types of functionality, only ‘Add’ and ‘Amend’ demonstrated a statistically significant correlation with overall functional growth. These two types correspond to ‘assumed’ functionality in the Indicative NESMA method, where each ILF is expected to have this associated functionality. This suggests that their correlations with overall functional growth may be explained implicitly through their association with ‘Internal Data Functionality’. There were some moderate correlations found among the other functionality types, such as ‘Delete’ and ‘Query’, which are commonly not explicitly specified at the Bid stage. Negative correlations were also found, for ‘Search’ and ‘Report’, indicating contrasting patterns between the types of functionality.

4.5.4.2 Multiple Linear Regression Analysis – Functional Metrics and Overall Functional Growth

The use of Multiple Linear Regression Analysis on these functionality types was affected by the absence of some types at either stage of a project. From the dataset of eleven projects, there were only five projects that contained values for each type of functionality type at both the Bid stage and Functional Specification. The first stage of this analysis was therefore to use the regression methods on these five projects separately. The second stage involved reducing the number of functionality types included in the regression model in order that all eleven projects could be included in the analysis. It was determined that the selection of seven functionality types would enable this stage of the analysis to be completed.

Table 45 - Correlation between Functionality Type Growth and Overall Functional Growth

Functionality Type	**Correlation with Overall Functional Growth**	**Number of Projects**	**Statistical Significance *p* value(2-tailed test) (Benjamini-Hochberg)**
Internal Data Functionality	**0.749**	11	***p*=0.008**
External Data Functionality	-0.625	7	*p*=0.134
Add Functionality	**0.666**	11	***p*=0.025**
Link Functionality	-0.180	10	*p*=0.620
Amend Functionality	**0.686**	11	***p*=0.020**
Process Functionality	0.475	11	*p*=0.139
Delete Functionality	0.477	8	*p*=0.232
Search Functionality	-0.075	10	*p*=0.837
Output Functionality	0.208	11	*p*=0.540
Report Functionality	-0.302	10	*p*=0.396
List Functionality	0.244	11	*p*=0.469
Query Functionality	0.462	11	*p*=0.152

The use of the 'Enter' method did not produce any meaningful results in the first stage including all of the functionality types. The reduction of the dataset to five projects in this case had expectedly resulted in this outcome. Table 46 shows that the use of the 'Stepwise' method did produce a statistically significant result, with the inclusion of only the 'Add' functionality type. The level of statistical significance was lower than that obtained from the analysis of individual functionality types. The use of the 'Enter' method for the second stage, incorporating all eleven projects, demonstrated a relatively high correlation. The inclusion of seven model variables negated the significance of this correlation, with a sharp reduction obtained in the 'Adjusted R

Square' value. The results were therefore not found to be statistically significant. The use of the 'Stepwise' method resulted in the selection of only the 'Internal Data' type of functionality, indicating that the inclusion of other types of functionality did not improve the performance of the regression model.

Table 46 – Growth/Functional Metrics Multiple Linear Regression Analysis

Model Parameters	Overall Growth in Size Functional Metrics (All Variables, 5 Cases)			**Regression Method**	Stepwise
Model Variables	**R**	**R Square**	**Adjusted R Square**	**Standard Error of Estimate**	**Statistical Significance**
Add	0.888	0.789	0.718	19.31922	*p*=0.044
Coefficients	**Unstandardized Coefficients**		**Standardized Coefficients**	**t**	**Statistical Significance**
	B	**Std Error**	**Beta**		
(Constant)	12.085	9.725		1.243	*p*=0.302
Add	0.531	0.159	0.888	3.346	*p*=0.044
Model Parameters	Overall Growth in Size Functional Metrics (7 Variables, All Cases)			**Regression Method**	Enter
Model Variables	**R**	**R Square**	**Adjusted R Square**	**Standard Error of Estimate**	**Statistical Significance**
Internal, Add, Amend, Process, Output, List, Query	0.892	0.796	0.320	25.04493	*p*=0.364
Coefficients	**Unstandardized Coefficients**		**Standardized Coefficients**	**t**	**Statistical Significance**
	B	**Std Error**	**Beta**		
(Constant)	13.368	19.384		0.690	*p*=0.540
Internal	0.423	0.338	0.522	1.249	*p*=0.300
Add	0.085	0.260	0.133	0.327	*p*=0.765
Amend	0.071	0.174	0.234	0.410	*p*=0.710

Process	0.162	0.316	0.232	0.513	*p*=0.644
Output	-0.018	0.229	-0.030	-0.079	*p*=0.942
List	0.019	0.025	0.300	0.771	*p*=0.497
Query	-0.005	0.063	-0.046	-0.076	*p*=0.944
Model Parameters	Overall Growth in Size Functional Metrics (7 Variables, All Cases)			**Regression Method**	Stepwise
Model Variables	**R**	**R Square**	**Adjusted R Square**	**Standard Error of Estimate**	**Statistical Significance**
Internal	0.749	0.561	0.512	21.21623	*p*=0.008
Coefficients	**Unstandardized Coefficients**		**Standardized Coefficients**	**t**	**Statistical Significance**
	B	**Std Error**	**Beta**		
(Constant)	23.269	8.304		2.802	*p*=0.021
Internal	0.606	0.179	0.749	3.388	*p*=0.008

4.5.4.3 Analysis for Growth in Overall Functional Size – Documentation Data

Table 47 shows the correlations with overall functional growth for measures of documentation size. The percentage page size increase, from Bid documentation to Functional Specification, demonstrated a high correlation, statistically significant at *p*=0.029. The simple page count of the Functional Specification provided an almost identical correlation with overall function growth. Initially this would appear to be a surprising result, but taking the previous results of using the Bid page size to explain growth into consideration, the Functional Specification page size does appear to be the more significant factor. The Benjamini-Hochberg procedure had no impact on the significance of these results.

Table 47 - Documentation Metrics and Overall Functional Growth

Variable	Correlation with Overall Functional Growth	Statistical Significance *p* value (2-tailed test) (Benjamini-Hochberg)
Functional Specification Page Size	**0.661**	***p*=0.027**
Percentage Page Size Increase	**0.654**	***p*=0.029**

4.5.4.4 Multiple Linear Regression Analysis – Documentation based Metrics and Overall Functional Growth

The combined use of the documentation based metrics to explain overall functional growth is presented in Table 48. In this case the use of the 'Enter' and 'Stepwise' produced an identical result, with a high correlation demonstrated when both variables are included in the regression model. The standardized coefficients for each variable indicate that they had a similar effect in the regression model. The level of statistical significance for the overall model, at *p*=0.013, has improved from the use of the individual metrics. The performance of these documentation based metrics is similar to that obtained from the functional sizing data. The negligible effort required to obtain these metrics enables the extent of overall function growth to be readily ascertained.

Table 48 – Growth/Documentation Metrics Multiple Linear Regression Analysis

Model Parameters	Overall Growth in Size Documentation Metrics			**Regression Method**	Enter/Stepwise
Model Variables	**R**	**R Square**	**Adjusted R Square**	**Standard Error of Estimate**	**Statistical Significance**
Functional Specification Page Size, Page Size Increase	0.815	0.664	0.581	19.66518	*p*=0.013
Coefficients	**Unstandardized**		**Standardized**	**t**	**Statistical**

	Coefficients		Coefficients		Significance
	B	Std Error	Beta		
(Constant)	5.405	11.059		0.489	*p*=0.638
Functional Specification Page Size	0.110	0.046	0.510	2.374	*p*=0.045
Page Size Increase	0.013	0.006	0.500	2.327	*p*=0.048

4.5.5 Regression Model Predictive Accuracy Validation

The derivation of further estimates from size data, such as actual effort, is commonly achieved through the use of a regression model. Such models, as developed in the previous sections, enable predictions to be made from subsequent size estimates not included in the regression model. The regression models developed in the previous section utilised the complete dataset and are thus likely to over-estimate their predictive performance outside of this dataset. The limited confidence in the stability of these models should be investigated by choosing an appropriate form of validation. Using one dataset for developing the regression model, and another separate dataset for evaluating the model, referred to as the holdout method, would be desirable but a sufficient number of data points needs be available to provide stable results. The availability of only 11 projects for use in this research study prohibited simple division into a training and a validation subset, as the results would be expected to vary significantly according to which projects were allocated to each subset. The Leave-one-out cross validation approach evaluates the regression model by holding each data point in turn out of the model and predicting the outcome for that data point. This method reduces the variance in the prediction results by calculating the average error across the predictions for each data point. This average error is often expressed in terms of the Mean Magnitude of Error (MMRE), which is one of the most widely used measures for evaluating the accuracy of estimation models. MMRE has, however, been subject to criticism e.g. it is biased in favour of estimation models that generally produce under estimates (Kitchenham et al., 2001). MMRE is reported in the results of this section to preserve generalisability within this field of research. The average of the absolute residuals produced by the predicted values provides an alternative measure that is not subject to this bias (Shepperd and MacDonell, 2012). The mean, median and standard deviation of the absolute residuals are therefore also reported as a more reliable indication of the overall accuracy of the estimates.

The validation results for the prediction of actual development effort are shown in Table 49. The emphasis is primarily on the use of bid stage data for effort estimation, but the uses of functional specification data as well as the expert estimates provided by Equiniti-ICS are also reported for comparison. Size estimation data from the bid stage is denoted by (B), with (FS) used for the Functional Specification stage.

Table 49 – Actual Development Effort Prediction Validation

Predictor	R	R Square	Adjusted R Square	MMRE	Mean Abs Residual	Standard Dev	Median Abs Residual
Indicative NESMA (B)	0.553	0.306	0.229	95.46%	621.70	530.15	473.69
Estimated NESMA (B)	0.294	0.086	-0.015	111.47%	699.74	615.23	460.03
Indicative NESMA(FS)	0.827	0.684	0.648	59.85%	417.17	473.18	250.74
Estimated NESMA(FS)	0.655	0.429	0.366	60.15%	497.67	617.31	238.68
Full NESMA(FS)	0.690	0.477	0.418	61.17%	480.45	576.80	209.79
Expert (B)	0.868	0.753	0.726	64.34%	616.39	1155.51	238.08
ILF (B)	0.567	0.321	0.246	98.73%	623.38	530.60	463.98
RET (B)	0.731	0.534	0.482	57.27%	418.19	474.39	251.64
ILF (FS)	0.827	0.685	0.650	59.13%	417.44	474.04	213.40
RET (FS)	0.799	0.639	0.599	47.46%	421.25	586.60	152.88
RET, EIF, EO, EQ (B)	0.845	0.714	0.523	117.14%	860.85	1010.63	616.80
RET, EI (B)	0.852	0.727	0.658	55.51%	448.67	434.55	342.50
EIF, Add, Link, Search, Output, Report, Query (B)	0.946	0.895	0.650	480.54%	2456.89	2319.36	1585.93
Link (B)	0.693	0.480	0.423	76.43%	684.84	1162.04	258.00

In terms of the overall sizes produced by the NESMA methods, the Indicative NESMA provided the most accurate predictions at both stages. This may reflect the stronger relationship with actual effort that data functionality demonstrates in comparison with transaction functionality. The Expert estimates developed within Equiniti-ICS produced a stronger correlation with actual effort than produced by the overall NESMA sizing methods. When the Expert estimates are validated, however, the mean absolute residual was similar to that produced by the Indicative NESMA method at the bid stage. The standard deviation for the Expert estimates was approximately double of that produced by the NESMA sizing methods, indicating greater instability in its predictions. The NESMA sizing methods at the Functional Specification stage were outperforming the bid stage NESMA estimates as well as the Expert estimates. Focusing specifically on Data Functionality at the bid stage provided greater levels of predictive accuracy. The most detailed RET component at the bid stage produced the lowest mean absolute residual at that stage, with a much lower standard deviation than the Expert estimate predictions. These results for the RET component were also similar to those produced at the Functional Specification stage, suggesting that using this more detailed component is preferable to the standard ILF component.

The models produced using Multiple Linear Regression Analysis at the bid stage had produced stronger correlations with actual effort than those using the overall NESMA methods. The validation results in this section indicate, however, that the predictive accuracy of such models was generally inferior. The number of variables in the regression model impacted the predictive accuracy, with the seven variables in the Detailed Functional Profile in particular producing a substantially larger mean absolute residual and standard deviation than any other approach. The smaller number of variables, namely the RET and EI components, produced using the ‘Stepwise’ method on the NESMA Functional Profile provided the most accurate Multiple Linear Regression predictions. This level of predictive accuracy was similar to that produced using just the RET component, and therefore superior to using overall NESMA size results for making predictions.

Table 50 shows the validation results for the regression models developed for predicting the overall Functional Specification size from bid sizing data. The predictive accuracy from using the overall Estimated NESMA bid size was found to be outperformed by some of the other regression models. The most accurate predictions were obtained from the multiple regression models using the NESMA Functional Profile, as well as the simple regression model using the RET component. The NESMA Functional Profile based models generally produced lower mean and median residual values than were found using the RET component model. The greater number of model variables may explain why the NESMA Functional Profile had a relatively higher standard deviation value. The worst performing model, using the Detailed

Functional Profile, included the highest number of model variables further highlighting the limitations imposed by the size of the dataset.

Table 50 – Overall Functional Specification Size Prediction Validation

Predictor	R	R Square	Adjusted R Square	MMRE	Mean Abs Residual	Standard Dev	Median Abs Residual
Estimated (B)	0.882	0.778	0.753	107.01%	696.90	540.95	350.23
RET, EIF, EO, EQ (B)	0.994	0.988	0.979	70.26%	440.91	466.00	274.37
RET, EO, EIF (B)	0.989	0.979	0.969	74.36%	467.60	463.40	281.11
EIF, Add, Link, Search, Output, Report, Query (B)	0.987	0.974	0.912	177.57%	975.95	1106.92	629.53
ILF, Link (B)	0.948	0.899	0.874	104.12%	678.87	549.01	553.22
ILF (B)	0.904	0.817	0.796	111.51%	627.07	312.37	597.61
RET (B)	0.899	0.808	0.786	77.08%	464.29	361.99	393.04

The validation results for predicting growth in overall functional size from bid size data are shown in Table 51. The results show that although the correlations produced were quite varied across the regression models, the predictive accuracy was quite similar for each model. Regardless of the prediction accuracy measure considered, there was no single regression model that could be considered to be the best performing. The use of either Functional Profiles or Functional Ratios would provide similar results for predicting growth in overall functional size. The general level of predictive accuracy was inferior to that found with predicting overall Functional Specification size for this dataset.

Table 51 – Growth in Overall Functional Size Prediction Validation

Predictor	R	R Square	Adjusted R Square	MMRE	Mean Abs Residual	Standard Dev	Median Abs Residual
RET, EIF, EO, EQ (B)	0.870	0.757	0.595	93.36%	817.21	807.08	540.55
EIF, Add, Link, Search, Output, Report, Query (B)	0.980	0.960	0.867	90.24%	806.63	806.92	583.82
RET/ILF, ILF/EI, RET/EO, RET/EQ	0.847	0.717	0.528	91.55%	812.12	820.52	512.75
ILF/Amend, ILF/Search, ILF/Output, RET/Amend, RET/Process, RET/List	0.991	0.983	0.949	94.96%	875.96	812.89	616.01
RET/EQ	0.711	0.506	0.451	90.88%	810.45	816.97	561.97

4.6 Evaluation of Research

This section is concerned with evaluating the implications of the results on both the development of bid stage estimates, and on research into this topic. Responses are then provided to the research questions that were raised in Section 4.3.

4.6.1 Bid Size Estimation Issues

(1) The initial bid stage process facilitates the submission of queries regarding the proposed system, but it would not be considered to be conducive to detailed probing of specific aspects of required functionality. The use of assumptions in the development of estimates is therefore an issue for bid estimation in particular. Generic functionality, for example ‘Amend’ and ‘Delete’ could be assumed to be required, from the inclusion of general ‘maintenance’ requirements

statements in the bid documentation. The use of an expert-based approach by Equiniti-ICS generally interprets these statements to include these different aspects of functionality. Functional sizing requires such issues to be clarified with the customer when the ‘expected’ functionality is not clearly specified, which would be unlikely to be feasible at the bid stage. The sample projects demonstrated that such generic functionality at the distinct stages of the lifecycle was not correlated to a statistically significant degree. The pattern was even more uncertain for ‘Query Functionality’. The option of assuming that these types of functionality are required would therefore be unreliable if generally applied across all projects.

(2) 'Process Functionality’ demonstrated a moderate overall decrease at the functional specification stage, and a reduction in the percentage contribution to the projects. As indicated previously, the bid documentation, written by the customer, tends to describe functionality in business process terms when, in practice, it can be more generic 'Add' and 'Amend' functionality that is ultimately specified. The Director at Equiniti-ICS indicated that business processes were prevalent in the background of more generic functionality and were significant factors affecting the effort required to develop each system. This may reflect the difference in how this type of functionality is viewed by FPA and by the development personnel in Equiniti-ICS. FPA adopts a user oriented view of functionality, so for this research ‘Process Functionality’ represented the provision of functionality related to a specific business process by the user. Business processes are therefore limited to this type of functionality in this research. The development view of customer business processes by Equiniti-ICS is focused on the implementation of the required functionality, regardless of the user view of that functionality. The extent to which business processes affect the implementation of a wider range of functionality is not related to FPA, and is therefore beyond the scope of this research.

(3) For each of the projects examined, the bid documentation did not provide a data model for the proposed system. The current estimation method employed by Equiniti-ICS involves the identification of the main system entities, analogous to the ILFs. In the project that experienced the most significant growth in functional size, the company's estimate identified 29 entities, while the functional sizing identified 27 ILF. However, the size estimate identified 90 RETs, resulting in the highest RET to ILF ratio of 3.21. In the absence of a data model in the bid documentation, this indicates the value of developing a more detailed derivation of the required data model for the system.

4.6.2 Bid Size Estimation Research Issues

(1) The use of Multiple Linear Regression analysis was limited in this research phase due to the size of the dataset. Investigating the combinative effects of model variables subject to this limitation restricted the number of model variables that could be included. The results section revealed that on occasion e.g. some uses of the Detailed Functional Profile, no meaningful results could be obtained. The potential interaction of the model variables could therefore not be assessed for every aspect of this research phase. In addition, the question of interactions across different aspects was beyond the scope of this research. For example, the Multiple Linear Regression analysis of using documentation based metrics to explain functional growth failed to discern any significant relationship for the combined use of the metrics. The question remains whether these documentation-based metrics would contribute towards explaining functional growth if they were investigated in the same model as the functional sizing metrics. Overall, an improved understanding can be obtained in some cases from the use of Multiple Linear Regression, compared to analysing variables individually. Analysing how project metrics such as functional size and effort vary should consider multiple functional components at once rather than any individual component. The extent of this understanding can only be increased by increasing the number of projects included in the dataset. The constraints of completing this research, in terms of availability of time and sample projects, prevent this phase of the research from being extended further.

(2) The pattern of functionality presented in Section 4.5.2 suggests that significant variations across different types of functionality are present within the dataset. The implication of these results extends beyond the scope of bid estimation. Assumptions of specific types of functionality being present are subject to questionable degrees of accuracy. The simplified versions of the NESMA method are predicated on fundamental assumptions proving to be broadly correct on average when applied across a complete project. How accurate are these assumptions in practice? The insight obtained from analysing the functional profile of projects requires functional sizing data to be developed at a later stage of the project lifecycle e.g. the Functional Specification stage. The economic viability of undertaking functional sizing at a later stage provides a significant barrier to the use of detailed methods. Can this functional sizing data be obtained in a more efficient manner?

4.6.3 Response to Research Questions

RQ1 - Can a detailed assessment of the profile of required functionality at two distinct stages in the project lifecycle provide a validation of the adequacy of requirements documentation at the bid stage of a project?

The detailed functional profile assessment indicated some common patterns within some types of functionality. The Data Functionality demonstrated a consistent pattern of a moderate increase in relative size between the Bid and the Functional Specification stages. The pattern for ‘Add’ and ‘Link’ functionality also provided a consistent pattern of more modest increases between the two stages. These types of functionality would therefore be less likely to result in unexpected levels of change in size. The general pattern for more generic functionality such as ‘Amend’ and ‘Delete’ was of more significant increases in between the two stages, although this was less consistent within the dataset. These types of functionality would require more careful consideration at the Bid Stage as they may reflect areas of missing functionality.

The pattern for EO types of functionality was more erratic for the most part. The most significant observation was that ‘List’ functionality generally demonstrated substantial increases between the two stages, although depending on the nature of implementation it could also be classified as ‘Query’ functionality. The nature of the sample projects led to ‘List’ functionality becoming the most significant type of Transaction Functionality. Careful consideration should therefore be given to assessing the degree to which Data Functionality has associated ‘List’ functionality at the Bid stage. The absence of a proportional amount of ‘List’ functionality presents a significant risk of missing functionality. The pattern for ‘Query’ functionality was inconsistent and would therefore be difficult to determine how substantial any relative change in size would be.

RQ2 – What level of detail of functional sizing data at the Bid stage of a project provides the most accurate indication of the overall Bid functional size and required development effort?

The degree to which individual components were correlated with the overall functional size indicated that the NESMA Functional Profile level of detail provided a stronger indication than the Detailed Functional Profile level. The greater variation in correlations for the Detailed Functional Profile components was effectively ‘smoothed’ out at the NESMA Functional Profile level.

The overall functional size at the bid stage provides a poor indication of the actual required development effort. In contrast, the internal data functionality available at the NESMA Functional Profile level provided a statistically significant correlation with the actual effort. The more detailed RET size demonstrated a stronger correlation than the ILF size in this case. For transaction functionality it was necessary to use the Detailed Functional Profile to find components, in this case ‘Link’ and ‘List’, which provided an equivalent correlation with actual development effort. The validation of predictive accuracy indicated that the RET size produced the most reliable estimates of actual development effort.

The use of Multiple Linear Regression analysis did not indicate any benefit from going beyond the NESMA Functional Profile level of detail for indicating either the overall functional size or the required development effort.

RQ3 – Can relative change in overall functional size be associated with sizing data, and if so, what level of sizing detail is required?

The two main approaches for analysing the change in overall functional size were to investigate the relationship for bid stage size data with the Functional Specification size and with the Functional Growth. The correlation between the overall functional sizes at the Bid stage and the Functional Specification stage was found to be highly statistically significant. Using the NESMA Functional Profile level of components, the ILF component demonstrated an even higher level of correlation. The use of Multiple Linear Regression analysis suggested that considering multiple components at once could provide the highest level of correlation. The Detailed Functional Profile level of components did not provide any further level of improvement in this regard. The validation of predictive accuracy indicated that both the RET component individually and the NESMA Functional Profile produced the most reliable estimates of overall functional specification size.

Focusing on the relationship with Functional Growth was less effective, although some of the functional ratios did demonstrate relatively strong correlations. The functional ratios utilised the same data generated during the functional sizing so negligible additional effort is required to perform this analysis. The use of these alternative approaches provides a measure of security by identifying contrasting warning signs.

The use of Documentation based metrics was found to be effective at explaining the occurrence of functional growth at the Functional Specification stage. The use of functional sizing data, however, at this Functional Specification stage provided a higher level of correlation with functional growth. The importance of discerning that functional growth has occurred at the Functional Specification stage would dictate which approach was preferable.

4.7 Conclusions

This phase has considered the extent to which the profile of functionality required for a system may be utilised to determine project metrics such as overall functional size and actual development effort. The overriding lesson from this research is that the use of the profile of functionality provides greater benefits than simply using the overall functional size. What level of detail is necessary for this profile of functionality? The answer is dependent upon how the profile is required to be used. Assessing whether missing functionality is present in the Bid documentation requires the Detailed Functional Profile of functionality developed in this phase of research. The conventional Functional Profile (BFCs) is most suitable for determining overall functional size, as well as explaining relative change in functional size. For actual development effort, either the RET size or some of the Detailed Functional Profile components provide a stronger correlation at the bid stage.

The type of the Case Management Systems investigated shapes the nature of the assessment of these projects. The specific types of functionality selected for the Detailed Functional Profile classification reflect typical aspects of data intensive systems. The general applicability of this research phase is therefore limited to projects that conform to these characteristics.

The primary issue arising from this research is how the level of sizing detail necessary can be obtained without incurring an unacceptable amount of estimation effort. Can existing approaches to reducing the effort involved in size estimation provide the answer to this issue?

5. Simplified Size Estimation

In Chapter 4 it was established that in order to utilise software sizing to provide greater insight into bid stage estimation, a detailed comparison of size estimates from both that stage and the later functional specification stage was necessary. The challenge for meeting this requirement stems from the unacceptable commitment of resources, at the functional specification stage, associated with using functional sizing methods that provide this necessary level of detail. This chapter is concerned with investigating how simplified sizing methods can be enhanced to provide such detail without incurring prohibitive estimation effort.

5.1 Introduction

Chapter 3 examined the value that may be obtained when different levels of rigour are applied to functional software sizing. Three standard variants of NESMA (Indicative, Estimated and Full) were used to evaluate the cost-benefit trade-off of increasing the level of size estimation rigour. The conclusion of Chapter 3 was that the Estimated NESMA method offered the optimal trade-off between sizing accuracy and estimation effort. The most significant savings in estimation effort provided by the simplest Indicative NESMA were negated by the insufficient accuracy of the sizing results. In the context of this commercial environment the primary constraint on the use of an estimation approach is the effort required to develop the estimate. The use of the Estimated NESMA method would be unlikely to be considered beyond the initial bid stage estimation. The value provided by software sizing beyond this stage has been highlighted in existing research, but the commercial reality prohibits the feasibility of this level of estimation at later stages of the project lifecycle. The estimation effort incurred is dependent on the size of the project being undertaken, but the available time to develop the estimate does not necessarily scale accordingly.

Chapter 4 examined the size estimation process at the bid stage, and how these bid estimates compared with those developed from the later functional specifications. It was found that while some general patterns within specific sub-types of functionality were observable, there was not an ‘expected’ profile that could be extrapolated to inform future project estimation. The main value obtained from this bid estimation analysis was in the stronger explanation of the differences between estimates developed at the two lifecycle stages that could be provided from a more detailed consideration of functional size than the overall size. This analysis required the estimation of both Data Functions and Transaction Functions.

This Chapter is concerned with the development of a simplified functional sizing method, derived from the fundamentals of the NESMA approach, which considers the results of the previous chapters. The needs that this simplified method should meet may be summarised as

(i) - Flexibility to adapt to the estimation process for each specific project e.g. stage of the project lifecycle, availability of requirements documentation, availability of time and staff to develop the estimate.

(ii) - Incorporate the assessment of Transaction Functionality, in order to obtain a detailed profile of such functionality, which exceeds the scope of the current Indicative NESMA method.

(ii) - Reduce estimation effort (relative to the Estimated NESMA method) to extend the feasibility of software sizing beyond the initial bid project estimate.

(iii) - Support adoption within existing software development practices.

The characteristics of the proposed simplified sizing method should therefore be independent of the level of detail available in the requirements documentation. In particular, increasing levels of input detail should be accommodated to match the differing requirements of each estimate, and to benefit from greater levels of detail when it is available. Increasing the level of detail considered by the estimator should be achieved more efficiently than with the use of the Estimated NESMA method in order to maintain its feasibility. The simplified approach should not require additional documentation requirements beyond the existing project documentation routinely developed within projects.

This chapter provides an overview of the related work into the development of simplified functional sizing methods. The existing simplified NESMA methods are described in order to establish the basis for adapting them to meets the needs of the proposed simplified sizing method. The Research Aim is then outlined, including the specific research questions that are addressed in this phase. The Research Approach is described, detailing the three main stages involved in developing and evaluating the simplified sizing method. This simplified sizing method was developed and refined using the experience of applying the existing NESMA method to the five commercial projects initially used in this research project. The initial software sizing activities enabled the consideration of the practical application of functional sizing on commercial projects. This chapter reports on the results of applying this simplified sizing method on these projects, as well as on six additional commercial projects. The performance of the simplified sizing method is primarily evaluated, in terms of its accuracy and required effort relative to the Full NESMA method, against the results obtained using the Indicative and Estimated NESMA methods. The research is lastly evaluated, considering the research issues of simplified sizing identified from this phase.

5.2 Rationale for Simplified Functional Sizing

The measurement phase of FPA is characterised by two core activities:

(1) Identification of each occurrence of a BFC

(2) Assessment of the type and associated complexity of each occurrence of a BFC.

The feasibility of undertaking these activities is determined both by the level of requirements detail available, and by the availability of time and personnel to develop the size estimate. Early in the project lifecycle there is often limited information available about the proposed requirements. The feasibility of completing the software sizing using FPA may consequently be limited to either identifying a subset of the BFCs and/or omitting the assessment of their complexity. Simplified software sizing methods may therefore be considered to be a necessity in addressing the limitations on size estimation imposed by this lower level of requirements detail. In these cases, the level of detail and the format of the information available are fundamental to selecting an appropriate simplified method.

The availability of time and personnel to allocate for size estimation tasks may vary according to the importance of the estimate. This importance may be determined from factors such as the lifecycle stage of the project e.g. at the initial inception of the project where estimation is of most value before the commitment to undertake the project. The other primary motivation for simplifying the size estimation process therefore stems from the trade-off between the estimation effort incurred and the accuracy and detail of the sizing output produced. The limited availability of resources needed to develop the estimate in turn requires demonstrable benefits to be derived from the size estimates. The benefits derived from obtaining size data diminish as a project progresses through its lifecycle. In order to provide value at later stages in the lifecycle the issue of the required estimation effort becomes more important. In essence, the feasibility of using the complete FPA approach increases inversely with the justification for doing so. The question is therefore one of how can the increased detail available later in the project lifecycle be utilised effectively with an acceptable associated estimation effort. There are a number of issues that impact on how effectively the software sizing process can be simplified, and these are discussed in the following section.

5.3 Related Work

This section considers the main issues with simplified functional sizing that have been investigated within research. The main simplified functional sizing methods are then examined, by classifying them according to their general approach and comparing their relative performance. This is then followed by the identification of the main limitations associated with these methods.

5.3.1 Simplified Functional Size Estimation Issues

The degree to which the estimation effort required for the development of functional size estimates may be reduced is dependent upon those activities that are simplified. Surveys on the estimation effort required for each phase of the FPA method were conducted by Lavazza (2017).

In the initial survey, first reported in Lavazza (2016), the respondents were asked to indicate a range of percentage effort for each individual phase. The results shown in Table 52 for Survey 1 are therefore normalised mean values. For Survey 2 the respondents were instead asked to specify a specific percentage for each phase, removing the need to normalise the results shown in Table 52. The first survey involved professionals with experience in the use of either IFPUG or NESMA, most of whom were certified in this field. The second survey involved only certified Function Point Specialists or Practitioners with an average of over ten years of experience in this field.

Table 52 - Percentage effort associated with FPA Phases (adapted from Lavazza, 2017)

FPA Phase	Survey 1	Survey 2
1 - Gather requirements documentation	14.0%	16.9%
2 - Identify application boundaries and determine measurement goal and scope	12.5%	10.2%
3 - Identify Logical Data Files (Data Functions) and Elementary Processes (Transaction Functions)	35.8%	23.3%
4 - Classify Data Functions (ILF, EIF) and Transaction Functions (EI, EO, EQ); identify RET, DET etc.; determine complexity of each function	23.3%	31.5%
5 - Calculate the functional size	6.2%	8.1%
6 - Document and present measurement	8.2%	10.0%

The main difference in the results between surveys is found in phases 3 and 4. Relatively large variances in the responses for these phases were attributed to some respondents attributing more time for the analysis of the individual functions to phase 3 and consequently less to phase 4. In survey 2 large variations were also observed for other phases e.g. a more detailed analysis of the requirements documentation could be involved in phase 1, so the responses for that phase ranged from 5% to 50%. The results from these surveys indicate that phases 3 and 4 are the main two contributors to the estimation effort associated with the FPA method. The focus of developing simplified sizing methods has primarily been on the activities associated with these two phases. The variance in the responses provided outside of these phases indicates that the potential saving made in estimation effort may depend on the measurer as well as the method. The variance in the responses may also indicate a difference in the perception of the respondents as well as any difference in how they approach the measurement process. In order the improve the understanding of the estimation effort required, greater attention is needed on explicitly recording this effort as part of studies into the use of functional sizing methods.

One of the perceived weaknesses of the FPA approach is in how it considers the complexity of the required functionality. In particular, it is focused on the number of data movements but the allocation of complexity boundaries (low, average and high) essentially introduces an upper limit that does not consider the degree to which this limit is exceeded. Algorithmic complexity is not itself explicitly assessed, which can adversely affect how the functional size will influence the development effort for a project. In FPA the functional size is ‘adjusted’ through consideration of General System Characteristics (GSC) in order to account for factors other than size, such as system complexity, that may affect the development effort. These adjustments enable a Value Adjustment Factor (VAF) to be calculated, which is then applied to the unadjusted size to provide the adjusted functional point total. The use of these adjustment factors has, however, not been generally considered to be beneficial. There has been widespread criticism regarding their lack of a sound theoretical basis, and because they do not provide an improved indication of how Function Points are related to development effort (Lokan, 2000). In addition, the ISO standard for Functional Size Measurement (International Organization for Standardization, 2007) does not include consideration of the VAF because it is related to technical characteristics rather than functional requirements. The inclusion of this aspect of FPA provides a potential source of variance between functional size counters. Simplified sizing methods may omit this aspect of FPA without sacrificing demonstrable benefits, and also avoid this potential source of subjectivity.

The absence of sufficient requirements detail about a project, particularly during the early stages of its lifecycle, presents a challenge to the development of necessary estimates. Expert-based estimation approaches can utilise the expertise of the estimator to compensate for the lack of requirements details, but formal approaches such as FPA require functionality to be specified in order to be measured. In addition, new projects may need to be calibrated using local (within-company) historical data in order to derive appropriate estimates from the identified functionality. In the absence of such local historical data, or the unsuitability of a new project for such calibration due to its different characteristics, alternative calibration data must be utilised. In both of these cases, the use of industry (cross-company) project data provides a means of compensation for absent local data. External data for functional sizing methods is available from industry datasets e.g. the International Software Benchmarking Standards Group (ISBSG) maintains a public repository of software project data (ISBSG, 2009). As part of a wider review on cross-company data studies, MacDonell and Shepperd (2007) investigated the use of the ISBSG dataset for effort estimation. For this dataset analysis, the use of within-company data was found to be perform better than the use of cross-company data. Kitchenham et al. (2007) reviewed cost estimation studies, comparing the performance of cross-company data and within-company data. This study concluded that the suitability of each type of dataset was affected by different project characteristics and the size of the development company. For example, the adoption of cross-company data was less suitable for small development companies, specialised projects, relatively small projects and when homogeneous within-company datasets were available. The evidence from existing studies was not considered to be definitive, in part due to the absence of any consensus on the approach to researching this issue. Some existing simplified sizing methods require the use of either internal or external historical data in developing the estimate. The suitability of these simplified sizing methods may therefore vary according to the characteristics of the project under consideration. Removing the dependency on either form of historical data is a necessary step for addressing the problem of their relative suitability.

One of the primary uses of software sizing is to derive an effort estimate for a project. Evaluation of the effectiveness of a software sizing approach for such use requires consideration of the level of output detail required for the optimal derivation of an effort estimate. In relation to simplified software sizing methods, does simplifying the sizing output down to an overall functional size limit their value for effort estimation? The use of a reduced number of IFPUG BFCs for predicting development effort was evaluated by Lavazza et al. (2013). In this study using the ISBSG dataset, statistically significant correlations were found between subsets of BFCs and development effort, suggesting that full consideration of all five main BFCs was unnecessary for this purpose. The authors proposed that the actual size measure itself should be simplified rather than just the measurement process. The benefits were stated as removing some of the technical issues associated with FPA, such as the correlations between BFCs that suggested they did not constitute an additive estimation model. The subjectivity and effort involved in identifying the type of each instance of a BFC would be reduced by replacing them with a simpler set of components. The issue that remains is one of whether the fact that you can reduce the size measure means that you should do so. The relationship between the different components of software sizing output (e.g. the BFCs), and the subsequent development effort for a project, provides an indication of the value that may be obtained from them. Abran et al. (2004) used regression models to investigate the relationship between the FPA functional profile of projects i.e. the individual BFC sizes, and the development effort of projects. The correlation with development effort was found to differ for individual BFC types within a dataset, suggesting that the relative size of each BFC within a functional profile would affect the development effort. Their study found that, within each dataset used, when the functional profile fell outside of the average functional profile (incorporating 80% of projects in a dataset) the correlation with development effort differs considerably from the correlation obtained using that entire dataset. In contrast, the projects that fell within the average functional profile for a given dataset produced a similar correlation with development effort to that obtained by the entire dataset. Buglione and Gencel (2008) found that using the individual BFC sizes from the COSMIC sizing method improved the modelling of the size to effort relationship in comparison to using only the overall functional size. Increases in the strength of the correlation with development effort were found to be 16.7% for development projects and 23.6 % for enhancement projects. These studies suggest that obtaining the functional profile for a project, rather than just the overall functional size, provides additional value for subsequent effort estimation. Consequently, simplified software sizing methods that produce the complete functional profile provide the ability to utilise this fact. The relative accuracy of the functional profile obtained by a simplified sizing method is therefore of interest.

The amount of detail contained in requirements documentation will vary from project to project, but it is generally dependent upon the stage of the lifecycle. The typical pattern for a project is for an increase in the amount of detail the further into the development lifecycle it progresses. The amount of uncertainty associated with an estimate is dependent upon the amount of detail provided in the requirements documentation, and would therefore be expected to diminish the further into the lifecycle the estimate is developed. The reduction in estimation uncertainty should facilitate the development of more accurate sizing estimates, at the expense of incurring greater estimation effort. This relative increase in the relative accuracy of size estimates derived from more detailed information has been demonstrated (Živkovič et al., 2005a). The use of Functional Sizing methods to derive effort estimates at multiple stages in the project lifecycle was investigated by Popović and Bojić (2012). Four distinct stages were selected for this study, ranging from the initial inception of the project up until the construction of the system. The results at each successive stage demonstrated a narrowing of the relative size range of the most accurate estimates. The pattern of these results fell within the range indicated by Boehm (1981), although a less pronounced degree of reduction was evident in this case. The impact of having greater requirements detail at each stage was reflected in the ability to use increasingly detailed sizing methods accordingly. For example, Indicative NESMA was used at the initial inception stage, but at the next stage Estimated NESMA was then able to be used. By the last two stages in the study, there was sufficient detail available to utilise the Full FPA approach. The use of increased requirements detail and the corresponding use of more detailed Functional Sizing methods where possible, have therefore been demonstrated to provide improved accuracy in size estimation. As more detailed requirements are only available at later stages in the project lifecycle, the prevailing question is whether or not the greater effort incurred in using more detailed Functional Sizing methods can still be justified. To obtain value at later stages it is therefore necessary to benefit from more detailed requirements specifications with a minimal estimation effort. Can a simplified sizing approach be used to accommodate greater input requirements detail without sacrificing estimation output detail? The Early Function Point Analysis method, developed by Meli (1997), allows for this by adapting to the level of detail provided in the documentation (the updated version of this method, the Early and Quick Function Point Method 2.0, is described in Section 5.3.2.3). Gencel and Demirors (2008) used this approach to develop functional size estimates at five distinct stages of increasingly detailed requirements analysis for a single project. Improvements in relative accuracy were provided by this method at each stage, when compared with the final stage estimate developed using the IFPUG FPA sizing method. The focus of the Gencel and Demirors study was limited to the overall functional size of the project, with no indication provided of the estimation effort required and the level of detail considered by the size estimation method at each stage. The focus of the research described in this chapter therefore incorporates an evaluation of the accuracy of sizing results obtained from different defined levels of requirements input detail, and the estimation effort required at each level. In addition, the estimation accuracy at each level will be assessed in terms of the complete functional profile produced by the estimate.

5.3.2 Simplified Functional Sizing Methods

Functional sizing methods may be simplified broadly according to two main approaches. Firstly, the assessment of the complexity of each identified BFC may be substituted by deriving the complexity from historical data (Derived Complexity Sizing Methods). Secondly, the process of identifying each BFC may be simplified by only requiring a subset of the BFC types to be identified, with the overall functional size extrapolated from this subset (Extrapolative Sizing Methods). Each of these approaches may incorporate the use of historical data (external or internal) in order to determine the functional size according to previously established average size contributions for each BFC. The use of these main simplification approaches may be flexible in their application in order to incorporate greater requirements detail in the size estimate where possible (Multi-Level Sizing Methods).

The existing simplified functional sizing methods can therefore be compared using four main features.

(1) BFC Requirements – which functional components must be identified by the sizing method?

(2) External Weightings – is the use of external data required for determining the size contribution of each BFC?

(3) Local Calibration – is the use of internal data required for determining the size contribution of each BFC?

(4) Multi-Level – is the method adaptable by accommodating different levels of requirements detail?

Table 53 shows the features associated with each of the simplified methods identified in the literature. The BFC Requirements feature identifies those simplified methods that require all of the functional components to be identified i.e. the Derived Complexity Sizing Methods. The Extrapolative Sizing methods will only require the identification of a subset of the BFCs associated with the full sizing method they are based upon. The External Weightings feature refers to either explicit weighting derived from cross-company data, or to implicit weightings from the assumption of complexity specified by the developer of the respective method. The Local Calibration feature refers to the requirement for specific weightings for the BFCs to be calculated from locally accumulated historical data. The Multi-Level feature refers to whether the size estimate can be adapted by the simplified sizing method according to the level of requirements detail available.

Table 53 - Overview of Simplified Functional Sizing Methods

Sizing Approach	BFC Requirements	External Weightings	Local Calibration	Method Type
Indicative NESMA (NESMA, 2004)	ILF,EIF	Yes	No	Extrapolative
ISBSG Extrapolative Weightings (Meli and Santillo, 1999)	One or more	Yes	No	Extrapolative
Internal Logical File Model (Tichenor, 2008)	ILF	No	Yes	Extrapolative

ILF Transaction Template (Poul Stall Vinje (cited in Meli and Santillo, 1999))	ILF	No	No	Extrapolative
Early Function Point Prognosis (Bundschuh, 2006)	EI, EO	No	Yes	Extrapolative
Simplified Indicative NESMA (Wang et al., 2008)	ILF,EIF	No	No	Extrapolative
ISBSG Average Complexity (Meli and Santillo, 1999)	All	Yes	No	Derived Complexity
Estimated NESMA (NESMA, 2004)	All	Yes	No	Derived Complexity
Early 'E' (Horgan et al., 1998)	All	No	Yes	Derived Complexity
Function Points Simplified (Bock and Klepper, 1992)	All	No	Yes	Derived Complexity
Simple Function Points (Meli, 2011)	Unspecified Generic Data Group, Unspecified Generic Elementary Process	No	Yes	Derived Complexity
Simplified COSMIC Methods (COSMIC, 2015)	All	No	Yes	Mixed
KISS (Forselius, 2006)	All (sub divided into 28 components)	No	No	Yes
Early and Quick Function Point Method	Individual and/or Aggregated	No	Yes	Yes

(Santillo et al., 2005)				
Rapid Function Point Analysis (Wu and Cai, 2008)	Individual and/or Aggregated	No	Yes	Yes
Simplified Sizing Method (This Research Phase)	ILF,EIF	No	No	Yes

5.3.2.1 Extrapolative Sizing Methods

Extrapolative Sizing Methods address the estimation effort and information requirements of the sizing process by only requiring the identification of a subset of the BFC types. The overall functional size is then extrapolated from the counts of the required BFC types, using either external or internal historical data.

The feasibility of deriving an overall functional size from a subset of BFC types is supported by the presence of correlations between BFCs. The nature of these correlations was investigated by Lokan (1999), who had previously identified factors affecting the relative contribution of each BFC: programming language, type of project developed, type of organisation and the use of prototyping during development. In this study, subsets of projects from an overall set of 269 projects were used to assess which factors had the most effect on correlations between BFCs. The correlations varied across subsets, ranging from always being present between EI and ILF, to rarely being present between EIF and any other components. The strongest correlations were found for 'new' development projects; particularly those developed using 4GL languages and application generators. It is the inherent weakness of the existence of correlations between BFCs which, conversely, enables this derivation of a total functional size from a subset of these components.

The applicability of utilising these correlations may therefore be limited to certain types of project, and require tailoring the approach adopted. The use of industry data, such as the ISBSG repository of software project data, enables the derivation of the percentage contribution, on average, that each BFC makes toward the total functional size. The overall estimated size can therefore be derived from the identification of one (or more) specific types of BFC. When using this approach early in a project lifecycle, it is recommended that an additional specified contingency size be added for functionality not apparent at an early stage.

The Indicative NESMA method only requires the identification of internal logical files (ILF) and external interface files (EIF), applying pre-established weightings to the number of identified ILF and EIF. The official NESMA counting manual specifies that errors in functional size with this method can be up to 50% (NESMA, 2004). Cândido and Sanches (2004) also found that the estimation accuracy of the Indicative NESMA method was typically at the upper end of this range, with an average error of 48% compared to the Full NESMA method reported. The van Heeringen et al. (2009) study reported that the Indicative NESMA method demonstrated on average an inaccuracy of 16.5% relative to the Full NESMA method. However, this study incorporated the direction of the relative inaccuracy for each project into the calculation, which leads to the positive and negative values cancelling each other out to some degree. In terms of the magnitude of each relative inaccuracy, the average inaccuracy compared to the Full NESMA method is 28.44%. These results are similar to those from Chapter 3 of this thesis, where the Indicative NESMA method demonstrated an average relative inaccuracy of 26.30 %.

The suitability of using externally established correlations will therefore vary according to the nature of the projects. Developing correlations internally from previously completed projects addressed the suitability of this approach for a specific development organisation. Tichenor (2008) describes an ‘Internal Logical File Model’, developed in 1994, that obtains a statistical correlation between unadjusted function points and the number of ILF from previous projects. This study reported that this model produced estimates within 10% of the actual total function point count. The author recommended that about 30 applications as a minimum must be counted before a statistically significant correlation is obtained, which may limit the feasibility of using this approach.

The need for historical data can be addressed by deriving a functional count from applying a template of transactions to identified ILFs, as suggested by Poul Staal Vinje (cited in Meli and Santillo, 1999). An application 'type' template of data inputs, data outputs and data inquiries can be defined and applied to each ILF (EIF are counted separately) in the form of an associated functional size for the appropriate template. This approach can reportedly reduce the effort to perform the function count to about 20% of the full count, although the accuracy will be limited to how closely the project matches the selected application ‘type’. The estimator has the flexibility to assign Function Point values to each function type in the template, but the effectiveness of this is dependent upon how closely the template fits each ILF on average.

Wang et al. (2008) proposed a similar template approach based on the Indicative NESMA method. The template of expected transactions is assigned to each ILF and EIF according to specific rules, which can be adjusted to suit each individual project. This approach produced results that were significantly closer to the full function count for the studied projects. The largest project included in the Wang study was only 414 IFPUG Function Points, and the projects had generally been overestimated by the Indicative NESMA method, which was consistent with an equivalent study of small web based applications (Cândido and Sanches 2004). However, for studies which included systems with a greater range of types and sizes (van Heeringen et al., 2009; Wilkie et al., 2011) this was not the typical pattern. The validation of the results in this study was limited to only the overall functional size, preventing a more detailed explanation of the reduced sizes produced by the method. The specified rules in the Wang study do not provide an indication of why their method would have led to lower estimates than the conventional method, so the general applicability of this method is unclear. These rules indicate a consideration of requirements detail beyond that of the simplified NESMA methods. The absence of reported estimation effort values for the methods used in the study leads to uncertainty into how effective this simplified approach is in reducing the amount of effort required to develop an estimate.

In contrast to methods that use one or more of the Data Functions as the input for the size estimate, the Early Function Point Prognosis methods instead uses the EI and EO components (Bundschuh, 2006). This method used the BFC counts from 78 projects to establish the average percentage of each BFC type present in the dataset. The counts for just the EI and EO components demonstrated a high correlation (greater than 0.97) with the overall IFPUG functional size. The nature of the projects in the dataset, where there was an availability of system interfaces from which to identify those BFC types, was suggested as a reason for the suitability of this method for the organisation that developed the projects. This early phase estimation method is therefore representative of how an organisation may develop a simple method according to the characteristics of the projects that they develop.

5.3.2.2 Derived Complexity Sizing Methods

Derived Complexity Sizing Methods require the identification of all of the BFC types, with the simplification of the sizing arising solely from the removal of the complexity assessment of each occurrence of a BFC. The complexity is derived either externally from established industry/standards data, or internally from the accumulation of local historical data. The pre-established complexities are then directly applied to the BFC counts, or indirectly through weighting constants, in order to produce the overall functional size for a project.

External data may be sourced from industry datasets, or it may be specifically incorporated into the simplified sizing method. For functional sizing methods, the ISBSG repository includes data for each BFC where they were provided for a project. This enables the general average complexity weighting of each BFC type to be derived. These weightings may be applied to the counts for each BFC in order to produce the overall functional size through a method referred to as Average Complexity Estimation (Meli and Santillo, 1999). The issue of the suitability of using cross-company data, identified in the previous section, suggests that this approach may not be preferable in certain cases and is more suited to substituting for the absence of appropriate internal data.

The Estimated NESMA method adopts assumed complexities for each BFC that were established by NESMA, who found that they produced functional sizes strongly correlated with those obtained using the Full NESMA method. The performance of the Estimated NESMA method was evaluated by van Heeringen et al. (2009), across 42 projects, who reported an average inaccuracy of only 1.5% relative to the Full NESMA method. The amended calculation, incorporating the magnitude of the relative inaccuracy for each project produces an average inaccuracy of 5.73%. This was consistent with the findings in Chapter 3, suggesting that there was no additional value in performing the Full NESMA method. The suitability of the Estimated NESMA method may, however, vary according to the nature of the developed systems. Cândido and Sanches (2004) examined the effectiveness of the different levels of the NESMA approach for web based applications, reporting that the Estimated NESMA method demonstrated an average error of 18% relative to the full method. The performance of this general approach was improved by the authors to an acceptable average error of only 4%, by adapting the assumed complexities to the 'typical' profile of previously completed projects. This approach still applies the associated complexity of each BFC type across every instance of that BFC, which may not be as effective outside of the relatively uniform low complexity profile of their tested applications. The use of the Estimated NESMA method was also evaluated using 36 projects developed between 2008 and 2012 (Ohiwa et al., 2014). The results of the Estimated NESMA method were found to have a high correlation of 0.97 with those of the IFPUG method. The correlation for the Estimated NESMA method results with software development effort values was also found to be high at 0.823. A distinguishing characteristic of the dataset used for the study was that the largest project had a functional size of 30000 FP, and represented a significantly larger size than the upper limit of 3000 FP reported in the official NESMA evaluation. The 36 projects included in the study were from a larger software repository of 512 software projects. The projects included in the study were the only ones to have complete functional size estimates and effort data submitted to the repository. The reasons for this were not considered in the study, leading to the question of why only a relatively small proportion of projects had submitted such data to the repository.

The 'Early E' approach (Horgan et al., 1998) provides a variation on the Estimated NESMA approach by identifying each BFC in the same way, and from this counting raw function points (RFP) i.e. how many BFCs are present. Instead of applying assumed complexities, a single weighting constant is derived from local historical projects to provide the conversion from the total RFP to a total Function Point (FP) value. The performance of this method was that it produced sizes within 25% of the full unadjusted count on 94% of the estimates. The performance in predicting effort was reportedly poor, so its use was only recommended for predicting the functional size of a system.

The Function Points Simplified (FPS) method (Bock and Klepper, 1992) also only requires each individual BFC to be identified, with regression analysis performed on historical project data to derive a separate weighting for each BFC. The FPS method omitted the EIF component, but a study by Desharnais and Abran (2003) included this component in their application of the method, and therefore provided a clearer comparison with the full FPA approach. A dataset of 47 projects was used to derive the specific weightings, which were then applied to another dataset of 42 projects. The results produced by this method demonstrated a statistically significant correlation with the detailed function counts of the FPA approach. The simplified method was found to be accurate to within 5% on average of the full function count, providing a comparable performance to the Estimated NESMA method. The method requires a significant historical dataset to produce the weightings, however, which may prove prohibitive for some organisations.

Simple Function Points (SiFP) further simplify the derived complexity approach by replacing the five main BFCs with two BFCs (Meli, 2011). The Unspecified Generic Elementary Process component represents each of the three Transaction Functions, while the Unspecified Generic Data Group represents the two Data Functions. The average respective sizes for each BFC were initially determined and evaluated using 800 projects from the ISBSG dataset, with an extremely high correlation of 0.9988 reported between the IFPUG size and SiFP size. The method was reported to be two to five times faster to perform than the full IFPUG method, but no indication was provided of how this was established. The method was subsequently validated using 140 software applications ranging in size from 103 FP to 4202 FP (Lavazza and Meli, 2014). The mean absolute difference between the SiFP size and the IFPUG size was reported to be 10.6%. Neither of the two methods evaluated were determined to be very accurate in predicting development effort, however, with MMREs of 117% and 116% respectively. The ease of learning the SiFP method was investigated using 180 analysts with no functional sizing experience. The analysts received three hours of training and then given 40 minutes to size a software application that had been previously measured at 267.8 SiFP. It was found that 75% of the analysts developed estimates that were between -10% and +7% of the ‘correct’ size, while 90% of analysts were within -18% and +16%. It was reported that 60% of the estimation effort was concerned with identifying and assessing the complexity of the five BFCs associated with the IFPUG method. The basis for this assertion was not clearly established other than from the mention of ‘field tests’. In addition, it is not clear what proportion of the 60% is concerned with distinguishing between the individual BFC types. It can therefore not be determined how much of the 60% of estimation effort that the SiFP method could potentially save. The results of the 40-minute sizing experiment were used to speculate that up to 2600 SiFP could be estimated in one working day. This would represent a significant increase on previously reported figures reported for the IFPUG method, ranging from 200-300 FP (Total Metrics, 2007) to 400-600 FP (Jones, 2008). The question of whether the imposition of a time limit in which to develop estimates for an application may influence the performance of the estimator was not considered in the study. A more accurate investigation of the estimation effort required for this method would therefore be to ask participants to develop a size estimate for a software application without a time limit. The use of the SiFP method was further evaluated using 25 Web Applications developed within a single company (Ferruci et al., 2016). Statistically significant correlations were found between the SiFP size and both the IFPUG size and development effort. This study simply derived the SiFP estimates from the IFPUG BFC counts, however, and therefore did not evaluate the use of the method or the estimation effort required. The SiFP method seeks to emphasise that it is not necessary to consider the functional size using as much detail as represented by the five IFPUG BFCs. As a means of approximating the overall functional size it may provide a superior trade-off between estimation accuracy and estimation effort than the other derived complexity methods. In narrowing the focus of the functional size to only two components, however, it weakens the level of utility of the resulting estimate further than those other methods.

The common requirement of these methods is that the data maintained by the system must be identified, with the NESMA approach specifically stating that a detailed data model should be available. COSMIC functional sizing is, however, fundamentally process focused and simplified versions of this approach only require the functional processes to be identified. Different simplified versions of COSMIC exist (COSMIC, 2015), providing alternative solutions to omitting the identification of the number of data movements involved in each functional process. The COSMIC Equal Size Bands method assigns each functional process to a particular size band, with an average size allocated to each distinct size band. These size bands are established using local calibration in order to ensure that each size band contributes equally to the overall system size. The COSMIC Fixed Size Classification method assigns each functional process to one of the locally established fixed size categories. The size categories must be sufficiently wide apart to enable functional processes to be clearly selected for each process. The use of this method has not been reported outside of the organisation in which it originated thus its generalisability is limited. The COSMIC Average Functional Process method simplifies the process further by using an average functional process size established using a sample of typical functional processes where detailed data movements are documented. This average size is then applied to a count of the number of functional processes identified in order to derive the overall system size. The reliability of this approach is dependent on how representative the sample functional processes are of the system as a whole. The unavailability of a sufficiently detailed sample from a project would prohibit the use of this method unless previous local historical data was available. The COSMIC Average Use Case method adopts a similar approach, only by using a higher level of granularity than functional processes. This approach may be subject to greater uncertainty as it depends both on how many functional processes on average are in a use case, and on the average size of a functional process. There has been no reported use of this method within research thus far.

The van Heeringen et al. (2009) study reported relatively high levels of accuracy for the COSMIC Equal Size Bands and the COSMIC Average Functional Process methods. These average inaccuracies stated, within 2% of the detailed COSMIC method, were subject to the same issue present with his NESMA data. Adjusting the calculation to only consider the magnitude of the relative inaccuracy for each project, the average inaccuracies that result are 28.25% for the Average Functional Process method and 7.71% for the Equal Size Bands Method. The accuracy of these two simplified COSMIC approaches may therefore be more in line with the simplified NESMA methods, but with the added requirement of establishing scaling factors. De Marco et al. (2013) evaluated the use of the COSMIC Average Functional Process method, as well as a simple count of the number of functional processes, for effort estimation. A dataset of 25 web applications was used to develop the size estimates used to build the effort estimation model. The simplified methods were found to produce estimates that were on average within 20% of the actual values, which was considered satisfactory for early effort estimation.

The main burden of adopting a simplified version of COSMIC is that the necessary scaling factors must be calibrated locally before the methods may be applied. The EASY (Early and Speedy) Function Points method has been proposed that eliminates the need for local calibration (Santillo, 2012). This approach, applicable to both the IFPUG and COSMIC methods, uses Fuzzy Logic to establish the size classifications used to develop the estimate. The ongoing validation of this approach that was indicated in the proposal paper has not resulted in any published research. Valdés-Souto and Abran (2012) also proposed the use of a Fuzzy Logic model to remove the need for local historical data. The use of this Estimation of Projects in a Context of Uncertainty (ECPU) model has been subsequently refined (Valdés-Souto and Abran, 2015), and evaluation has been ongoing (Valdés-Souto, 2017). Results from each of these studies suggest that use of the ECPU model for simplified COSMIC estimation can provide a slight improvement on the COSMIC Equal Size Bands method in terms of approximating the actual COSMIC size.

While the Derived Complexity Sizing methods have been demonstrated to be relatively effective, the necessary level of requirements detail to use them may not always be available for a project. In addition, the burden of identifying each individual BFC may either not be feasible or not justifiable, in terms of the expenditure in estimation effort required**.**

5.3.2.3 Multi-Level Sizing Methods

The Early and Quick Function Point (E&QFP) Method combines aspects of some of the previous approaches (Santillo et al., 2005), and refinements since 1997 enable it to be adapted to fit any of the ISO specified FSM methods. It supports the identification of existing BFCs, referred to as Base Functional Processes (BFP) and applies weightings derived from ISBSG benchmark data and/or local calibration. It also defines additional component types that represent aggregations of existing BFCs e.g. the Typical Functional Process (TFP) consists of the typical operations: Create, Retrieve, Update, Delete, (List) i.e. CRUD(L). The General Functional Process (GFP) allows a sub system of two or more BFPs to be more generally identified, while the Macro Functional Process (MFP) facilitates the aggregation of two or more GFPs. The E&QFP (Version 3.1) manual for the IFPUG method (DPO, 2012) defines four aggregation levels in total: -

(1) IFPUG BFC – including type and complexity of each instance
(2) Unclassified IFPUG BFC – no type or complexity determined
(3) Group of Unclassified BFC – e.g. Typical Functional Process
(4) Group of General or Typical Processes – e.g. Macro Process

The basis for aggregation may come from identifying requirements in the new system that correspond to a known aggregation in an existing system. The complexity of each component is assessed as being low, average or high depending on the level of requirements detail available. In addition, each component is assigned a minimum, most likely, and maximum size using statistical tables rather than a single size. These values vary according to the aggregation level used by the estimator reflecting the uncertainty associated with the estimate. The developers of this method reported that in most cases the size estimated is within 10% of the 'real' size. Three distinct sizing levels are suggested in the E&QFP manual, with each consisting of identifying components at either of two adjacent aggregation levels. These are the Summary sizing level (Aggregation Levels 3 and 4), the Intermediate level (Aggregation Levels 2 and 3) and the Detailed level (Aggregation Levels 1 and 2). An investigation using these three sizing levels of the E&QFP method was conducted using a dataset of 65 full IFPUG measurements (Meli, 2015). The sizes of the measurements in the dataset ranged from 113 FP to 1601FP. Experienced function point specialists were used to develop the estimates using the E&QFP method. The mean absolute errors for each sizing level, relative to the full IFPUG size, were reported to be very similar (either 6% or 7%) in each case. Only the overall functional size is compared, therefore the utility of the sizing data beyond this is not considered. The reduction in the estimation effort varies according to the degree to which the components have been aggregated into MFPs, but the savings reported previous to the Meli (2015) study were generally between 50% and 90%. In the Meli (2015) study estimation effort was considered in terms of sizing productivity i.e. hours/FP. The Detailed level of sizing was reported to be on average 1.9 times faster than the full IFPUG method. For the other two sizing levels the productivity increased further to 3.9 and 6.3 times that of the full IFPUG method. The study did not outline the procedure for recording estimation effort for each of the sizing levels used, only presenting a summary of the average results without further discussion. The reliability of these results is therefore unclear and would depend on the aggregation levels most commonly used by the estimator. Theoretically, the method at the lowest aggregation level is equivalent to a full functional sizing method where functional components are identified and their complexity assessed. Indeed, Gencel and Demirors (2008) reported that their use of this method at the most detailed level of their study required the same amount of estimation effort as the IFPUG FPA method. The E&QFP method provides for considerable flexibility as each part of the system can be estimated at whichever level of detail is feasible (or desirable). This enables the method to be applied when there is limited requirements detail available, but also take advantage of greater detail as and when it is available. As with any Function Point approach, the reliability of the results is limited to the ability of the estimator to identify when such component types are present in the documentation. This issue may be more significant in this case due to the additional layered classifications of components that may be identified, although (unspecified) encouraging results are reported even with novice users of this method. Suitability for the use of non IFPUG-based versions of the method remains unclear. An experiment on the use of the COSMIC version of the E&QFP method did not produce encouraging results (Almakadmeh and Abran, 2013). In this experiment, estimates developed using the E&QFP COSMIC method demonstrated an average magnitude of relative error of 1502% when compared with estimates developed using the standard COSMIC method. The average percentage difference from the mean estimate when using the E&QFP COSMIC method was 87.4%., suggesting a low level of reproducibility for the method.

Wu and Cai (2008) reported the use of their Rapid Function Point Analysis method on 20 large software applications from the telecommunications industry. From the description of the method reported in their study, however, this method is indistinguishable from the E&QFP method. The study can therefore be considered primarily as an evaluation of that method on a dataset ranging from 154 FP to 1228 FP. The method was reported to produce estimates that were between -14% and +10% of the IFPUG estimates, with an average difference of 5.65%.

The KISS method (Forselius, 2006) aims to simplify the software sizing procedure while maintaining general compatibility with any ISO standard FSM method. This method involves the counting of 28 different functional components, based on the FISMA method. The rationale behind this approach is that by providing more specific components to be counted, less complicated guidelines are required to identify those components. This, in turn, reduces the amount of training required and the subjectivity involved in performing the method. For example, External Outputs are divided into output forms, reports, text messages/emails, and monitor screen outputs. As with FISMA, algorithmic complexity is directly considered through counting components such as Calculation routines and Simulation routines. The KISS Quick level of this method only requires these components to be counted, with an 'average' size allocated by the counter enabling a functional size to be derived by applying multipliers to the 'average' sizes of the counts of each respective component. Multipliers from any of the main standard methods may be used, with a multiplier of 0 applied to any component not required by the chosen method. While the KISS Quick level has produced promising results in student trials, relative to the more detailed KISS Perfect level of this method, the case studies used were small, at approximately 200 FPs or less in size. The reliability of this approach for larger systems has therefore not been established. The feasibility of determining a reliable 'average size' for each component for larger projects may limit the suitability of this approach to those projects with a ‘typical’ profile. The estimation effort savings achieved by KISS Quick were not reported, but as each component must be counted it would not suggest that savings equivalent to the extrapolative sizing methods could be achievable. The KISS Perfect level of this method addresses some of the shortcomings of the KISS Quick level, as it allows for a more detailed consideration of the functional size by listing and assessing each individual function. However, at this level the effort required is approximately the same as the detailed standard methods, and may therefore not be suitable for classification as a simplified approach.

5.3.2.4 Relative performance of simplified functional sizing methods

Evaluations of simplified functional sizing methods have generally taken the form of comparing their estimation accuracy against the size estimates of the related full method. While the Indicative and Estimated NESMA methods have commonly been evaluated together, as discussed in the previous sections, more comprehensive studies directly comparing various simplified methods on the same projects have been largely absent. One study was conducted that evaluated a range of simplified methods against the full IFPUG size estimates on the same projects (Lavazza and Liu, 2013). The dataset consisted of nine Real-Time projects and nine non Real-Time projects. The projects included in the dataset ranged in size from 15 FP to 289 FP, representing a limited range of sizes for evaluating the performance of the methods. The results for each of these simplified methods were derived from the BFC counts developed using the full IFPUG method. No consideration was therefore given to estimation effort in this study, or to whether using the full IFPUG method leads to more accurate BFC counts than using only a simplified method. The results of this study are summarised in Table 54, showing the mean absolute error for each simplified method relative to the full IFPUG method.

E&QFP (generic) in this study used a BFC aggregation level, so while this method is a Multi-Level method in general its use in this context is equivalent to a Derived Complexity method. E&QFP (unspecified generic) is essentially the Simple Function Point method, as the method has been incorporated into the E&QFP guidelines. The Simplified FP method is essentially the same as the Estimated NESMA method but with different weightings applied to the BFC types.

Table 54 - Mean Absolute Relative Error Values for Simplified Functional Sizing Methods (adapted from Lavazza and Liu, 2013)

Sizing Method	Method Type	Mean Absolute Error (Real-time Projects)	Mean Absolute Error (Non Real-time Projects)	Mean Absolute Error (All Projects)
NESMA Indicative (non-normalised)	Extrapolative	127%	79%	103%
NESMA Indicative (normalised)	Extrapolative	63%	34%	49%
Tichenor ILF	Extrapolative	20%	32%	26%
ISBSG Distribution	Extrapolative	112%	58%	85%
NESMA Estimated	Derived Complexity	10%	8%	9%
E&QFP (unspecified generic)	Derived Complexity	9%	17%	13%
E&QFP (generic)	Derived Complexity	10%	14%	12%
Simplified FP (sFP)	Derived Complexity	17%	23%	20%
ISBSG Average Weights	Derived Complexity	10%	14%	12%

The Extrapolative methods were more affected by the distinction between Real-Time and non-Real-Time projects than the Derived Complexity methods were. The Extrapolative methods demonstrated unsatisfactory estimation accuracy in general. The Tichenor ILF method performed better overall than the other methods in this category, suggesting that the weightings used for such methods should be calibrated locally to better suit the projects developed by an organisation.

The Derived Complexity methods provided a more satisfactory level of estimation accuracy overall, as would be expected from other individual evaluations of such methods. The Real-Time projects were generally found to be more complex than the non-Real-Time projects in this dataset. Consequently, those Derived Complexity methods that use larger weightings tended to produce larger overestimates for the non-Real-Time projects. The Estimated NESMA method was the most suitable for the non-Real-Time projects, and those specific results were closer to those reported both in other studies and in Chapter 3 of this thesis for this method. The suitability of any simplified method will vary according to the nature of the projects being sized. Flexibility to adapt to the characteristics of an individual project is therefore an important consideration, rather than simply trying to establish more suitable weightings from local data to apply generally on future projects. The approach developed in this chapter considers complexity at a finer grained level in order to provide such flexibility.

5.3.3 Limitations in Simplified Software Sizing

The main issues prevalent in these existing simplified sizing methods are as follows:

(1) The reliance on historical data, or assumptions of 'typical' projects, restricts the accuracy of these methods on new projects that do not conform to the necessary precedents.

(2) The extrapolative methods, which provide the greatest savings in estimation effort/cost, generally do not provide the same level of output as the more detailed methods, therefore limiting the utility of these methods to indicating the overall functional size rather than the provision of the full functional profile.

(3) The weightings, or assumptions, upon which the methods are based, are applied uniformly across the respective BFC types, limiting the flexibility to cater for significant variations within an individual project.

(4) With the exception of the Multi-Level sizing methods, the approaches are not adaptable in terms of the level of input detail from requirements that they can accommodate. This limits their ability to benefit from a greater understanding of the required functionality of a project as a project progresses through development lifecycle steps.

The study reported in this chapter involves the size estimation of 11 commercial projects, ranging in size from 218 FP to 1931 FP, using a simplified approach adapted from the NESMA Indicative method. The adaptations seek to address each of the limitations identified above. The E&QFP Method is the only one that overcomes each of these issues, as it can essentially replicate the full function counting process where it is feasible. However, the aggregated nature of its base components, while providing the most flexibility of the identified simplified methods, introduces additional components to be identified and therefore represents the most complicated approach. The focus within this research, which adapts the simplified NESMA approaches, limits the identification of base components to the existing NESMA BFCs. The simplified sizing method focuses on identifying the 'profile' of their occurrences within a project. The relatively wide variance in the reduction in estimation effort achieved by the E&QFP method suggests that the incorporation of additional input detail into the estimate significantly impacts upon this effort. This research aims to provide a more efficient approach by facilitating refinement of the 'profile' at a broader level than the individual BFCs.

5.4 Simplified NESMA Methods

The NESMA Function Point Method measures the functional size of software in terms of five BFCs:

- Internal Logical Files (ILF) and External Interface Files (EIF)
- External Inputs (EI), External Outputs (EO) and External Inquiries (EQ)

These components are referred to as either Data Functions (ILF, EIF) or Transaction Functions (EI, EO, EQ). There are three levels of size estimation provided for within the NESMA approach, as illustrated in Fig. 5.1. The Full NESMA estimation method requires the identification and complexity classification of each of these components. The complexity of each component is determined from counting the number of Data Element Types (DET) and Record Element Types (RET) either present, in the case of files, or transported across the system boundary in the case of transactions; with a classification of Low, Average or High complexity assigned according to specified boundaries of DET and RET totals. In this context DET refers to the attributes present in a file, while each file is itself comprised of one or more RETs that exist as a 'logical' file within the system. Each individual component is assigned a Function Point value according to its classification, as shown in Table 55.

The total functional size for a system is then calculated by summing the individual Function Point values of these components:

Total Functional Size = (Sum of ILF) + (Sum of EIF) + (Sum of EI) + (Sum of EO) + (Sum of EQ)

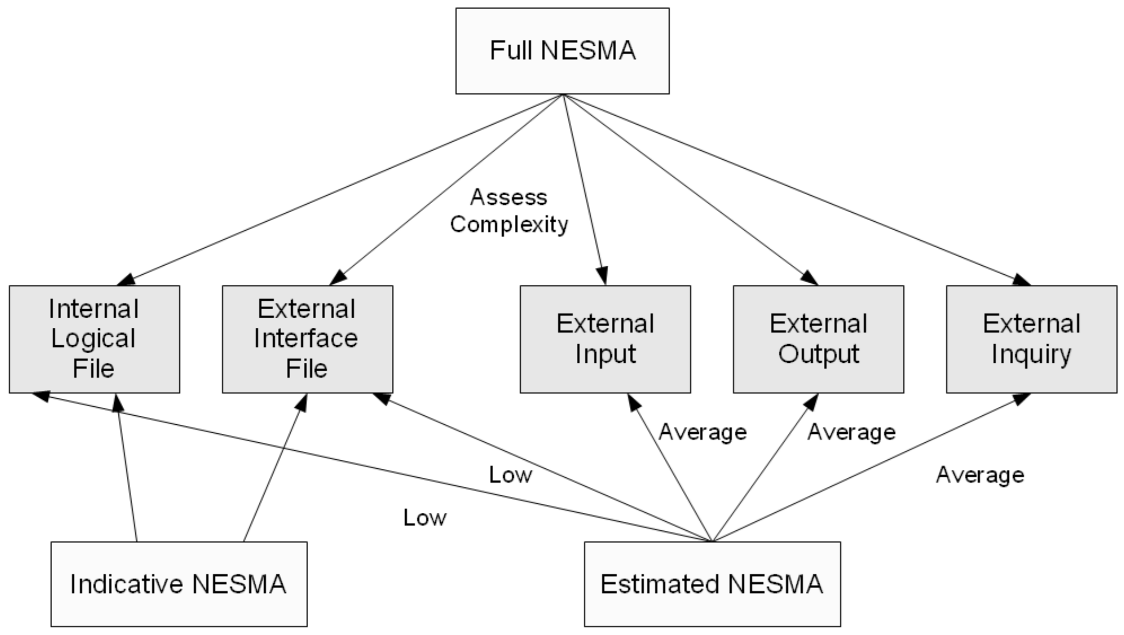

Figure 5.1 - Related Levels of NESMA Functional Sizing

The Estimated NESMA method removes the requirement of assessing the complexity of each individual component. Instead, as shown in Figure 5.1, the Data Functions are assumed to be of low complexity, while the Transactions Functions are assumed to be of average complexity. The total functional size is calculated using the same formula as for the Full NESMA method, and as such the same subtotals for each BFC are also provided by this Estimated NESMA method.

Table 55 - NESMA Complexity Weighting Scheme (adapted from NESMA, 2004)

BFC Component	Low (FP)	Average (FP)	High (FP)
ILF	7	10	15
EIF	5	7	10
EI	3	4	6
EO	4	5	7
EQ	3	4	6

The Indicative NESMA method further simplifies the process by only requiring the identification of the Data Functions, from a data model, and applying predefined weightings to the number of components identified. The weightings to apply depend on whether or not the data model being used is normalised (i.e. 3rd normal form):

*Total Functional Size = Num of ILF *35 + Num of EIF *15 (for non-normalised)*

*Total Functional Size = Num of ILF *25 + Num of EIF *10 (for normalised)*

This method therefore does not provide subtotals for each BFC, limiting the sizing output in comparison to the other NESMA methods. The rationale behind these weightings is the assumption that each Data Function will, on average, have a specific number of Transaction Functions associated with it, as shown for non-normalised Data Functions in Table 56. The three EIs associated with each ILF represent Add, Amend and Delete functionality; the two EO reflect 'report' functionality typically required for each ILF.

Table 56 - Indicative NESMA Assumed Transaction Functions

Data Function Type	Number of EI	Number of EO	Number of EQ
ILF	3	2	1
EIF	-	1	1

5.5 Research Aim

The decision on which estimation approach should be adopted requires consideration of the following main factors that affect the estimation process. The selection of a software sizing method must reflect the level of requirements detail available in order to address the feasibility of developing the estimate (Section 5.5.1). The purpose of developing the size estimate determines the nature of the output detail required to be provided (Section 5.5.2). The degree to which a size estimation method can be effective is dependent on how the specified guidelines reflect the commercial reality of practical application (Section 5.5.3).

The aim of this research phase is focused on investigating how the approach to simplifying the software sizing process can be adapted to provide utility equivalent to the full sizing method. In particular, the simplified approach should accommodate the various factors affecting the estimation process. This research therefore incorporates the development of a software sizing method that can adjust the balance between estimation performance and estimation effort according to the circumstances of the size estimate.

5.5.1 Feasibility Factors

The main factors for determining the feasibility of a functional sizing method are as follows:

(1) Which types of BFC can be identified from the requirements documentation?

(2) Is sufficient detail present to determine the complexity of specific Data and Transaction Functions?

(3) Are the necessary resources (time, staff etc.) available to develop the estimate using the method?

The majority of simplified sizing methods have specific levels of detail that must be present in the requirements documentation. The level of detail available for a project may consequently fall ‘between’ different methods, or only partially conform to a specific method.

5.5.2 Estimation Output Requirements

The main factors for determining the output detail required from a functional sizing method are as follows:

(1) Is the method required to provide the functional profile of the system in terms of the size of each BFC type (for subsequent project management activities e.g. effort/cost estimation)?
(2) The value of the size estimate in terms of significance of informing decisions.

In the preceding chapter the value obtained from different levels of estimation output detail was examined. The optimum level of detail was found to vary according to the purpose of the estimate e.g. predicting development effort, predicting future change in functional size etc. The required performance of the estimation method can vary according to the commercial significance of the project, stage of the project lifecycle etc.

5.5.3 Practical Application of Estimation Method

The main factors affecting the practical application of a functional sizing method are as follows:

(1) Is there a requirement for local historical data or calibration for the method?
(2) How well do the method guidelines relate to commercial projects in practice?

Software sizing methods generally have inherent expectations of the nature of requirements documentation. Simplified sizing methods may involve applying specific assumptions or calibrations to compensate for the absence of sufficient requirements detail. The degree to which commercial projects suit such approaches impacts upon the effectiveness of existing software sizing methods.

5.5.4 Research Questions

The simplified sizing methods previously identified are generally confined to specific levels of input and output detail. The rigidity of these methods necessitates the selection of a particular approach according to the specific circumstance of the estimate. The common theme across the factors discussed in the preceding sections is that the circumstances of each size estimate can necessitate a different sizing method to be adopted. The flexibility provided by a Multi-Level Sizing approach enables a more consistent approach to be adopted across each estimate according to individual circumstances. This research phase investigates the effectiveness of adapting the current simplified NESMA approach to enable the sizing process to be scaled according to specific circumstances in a simplified manner.

This research is concerned with assessing whether adapting the NEMSA simplified approach to functional sizing provides value to the software development process. To examine the business value of simplified functional sizing methods, consideration should be given to (i) the size estimation accuracy; (ii) the degree of utility, in terms of project management activities, provided from the size estimate output detail; and (iii) the estimation effort overhead. The performance of the developed simplified sizing method should therefore be evaluated at different levels of input detail to determine whether value is added as more detailed estimates are developed. The Research Questions, therefore, for this phase are:

RQ1: To what level of requirements detail can the FPA approach be simplified while achieving an acceptable level of accuracy (i.e. within 5% of Full NESMA method)?

RQ2: To what level of requirements detail can the FPA approach be simplified while providing the functional profile to an acceptable level of accuracy (i.e. within 5% of Full NESMA method)?

RQ3: What level of utility can a simplified sizing method provide?

RQ4: Where is the optimal trade-off between estimation accuracy and estimation effort?

For RQ1, based on the previous chapters the Estimated NESMA method is considered to be the baseline (of within 5%) for relative estimation accuracy against which the developed simplified method should be compared. For RQ4, consideration of the desirable estimation accuracy and estimation effort incorporated the feedback from Equiniti-ICS on what would be considered acceptable in a real world context. In addition, the change in estimation accuracy and estimation effort between each level of the simplified sizing method is tested for statistical significance.

5.6 Research Approach

This phase of research incorporates the use of the same eleven sample commercial projects from the previous chapters. The process of developing NESMA sizing estimates for these projects informed the adaptation of the existing simplified methods (i.e. Indicative and Estimated) adopted by the NESMA approach. The research methodology is presented in the following section, outlining the iterative nature of developing and refining a simplified functional sizing method within the overall research project.

5.6.1 Research Methodology

The aim of this research phase is concerned with adapting the current NESMA approach to simplifying functional sizing, to develop a simplified method that optimises the trade-off between estimation accuracy and estimation effort. This optimisation is to be achieved by simplifying the functional sizing process, while achieving estimation accuracy and degree of utility comparable with that from the Full NESMA method. This research phase is broadly divided into the following three stages:

(1) Appraisal of NESMA Functional Sizing Methods (Indicative, Estimated, Full)

(2) Development of simplified NESMA Adaptations

(3) Evaluation of simplified NESMA Adaptations

The first research stage is focused on learning from the application of each of the NESMA methods on the sample projects. This stage is therefore completed concurrently with the previous phases of research as an appreciation of the practical application of the NESMA methods (Indicative, Estimated, Full) is developed. The goal of this stage is to provide the identification of potential areas of improvement for the NESMA approach to simplifying functional sizing.

The second research stage is focused on developing the simplified method, incorporating adaptations of the NESMA approach. This involves adapting the existing simplest approach, the Indicative NESMA method, to facilitate aspects of the Estimated and Full NESMA methods to be utilised in a scalable fashion. The development of a size estimate can be affected by factors such as the level of requirements detail specified, the purpose of the estimate and the availability of time and personnel. The goal of this stage is to develop a simplified functional sizing method that allows the optimal level of estimation effort to be incurred according to the circumstances affecting the estimate. The development of the simplified sizing method is iterative, wherein the approach is designed and refined on the first five projects. In order to assess the trade-off between estimation performance and estimation effort, it is necessary for the simplified sizing method to accommodate different levels of rigour in assessing the required functionality of a project. The refinement process is therefore also concerned with the design of different 'levels' of software sizing for the developed method. The different levels of the completed simplified sizing method are then performed on both these initial five projects, and subsequently on the remaining six projects.

The third research stage is concerned with the evaluation of the developed simplified sizing method. The sizing results obtained through applying different levels of the simplified sizing method on the sample projects are evaluated against the results obtained from using the NESMA methods. The NESMA sizing results, and associated estimation effort figures, from the initial phase of this research project are therefore utilised in this research phase.

5.6.1.1 Composition of Research Dataset

The dataset for this phase of the research took the form of eleven commercial projects supplied by Equiniti-ICS. The development of the simplified sizing method, in Research Stage 2 above, incorporates the use of five initial projects (referred to as 'A' to 'E'). In Research Stage 3 these initial projects are combined with an additional six projects ('F' to 'K') to form the overall dataset used to evaluate the developed sizing method.

The use of a subset of the dataset for Research Stage 2 is necessary to fulfil the objectives of this research phase. The assessment of the trade-off between estimation performance and estimation effort requires that accurate measurements are made for the different levels of the simplified sizing method. This is offset by the need to develop the simplified sizing method by iteratively applying elements of the approach to projects and contrasting this with the use of the NEMSA methods in these same projects. In particular, the estimation effort incurred by the completed simplified sizing method would only be 'judged', due to the application of intermediate iterations of the method, rather than accurately recorded. In order to fully satisfy the requirement for accurate measurements, the development and refinement of the simplified sizing method should be completed prior to the application of the method. Restricting the development of the method to the initial subset of five projects enables the remaining six projects to be used to obtain accurate measurements of the estimation effort incurred. The issue of maintaining the integrity of the measurements of different sizing methods on the same sample projects is discussed in Section 5.5.4.

The initial subset of projects, (‘A’ to ‘E’), represent those which were generally completed by Equiniti-ICS earlier in the overall timeframe (2000 to 2005) than the remaining projects (‘F’ to ‘K’). In Chapter 3 it was noted that the development approach adopted by Equiniti-ICS differed in the later projects, but that the format of the documentation remained the same for each project. The contrast in these characteristics of the projects was not considered to impact upon the consistency of the sizing measurements made using the NESMA methods. The developed simplified sizing method should conform to the general principle of functional sizing, and should therefore also be independent of any contrast in project characteristics. The composition of the initial subset of projects, and those of the remaining projects, is consequently not considered to impact upon the validity of the size results obtained in this phase of research.

5.6.2 Research Stage 1: Appraisal of NESMA Functional Sizing Methods

The applicability of each NESMA method can be assessed by considering the estimation process factors listed in Section 5.5.

5.6.2.1 Assessment of Feasibility Factors

(1) Which types of BFC can be identified from the requirements documentation?

The Indicative NESMA method is appropriate when only the Data Functionality can be identified. The Estimated and Full NESMA methods require the Transaction Functionality to also be identified.

(2) Is sufficient detail present to determine the complexity of specific Data and Transaction Functions?

The Full NESMA method can only be used when it is possible to assess the complexity of the individual functions. In the absence of this level of requirements detail a simpler version of NESMA must be selected.

(3) Are the necessary resources (time, staff etc.) available to develop the estimation using the method?

The Full NESMA method requires the greatest investment of resources to develop size estimates. The larger the project is, the more pronounced the difference is in the associated estimation effort compared with the simpler Indicative and Estimated NESMA methods. The resources available to develop an estimate are greatest at the initial stage of a project where it is a primary activity, and decrease later as subsequent development activities take precedence.

5.6.2.2 Assessment of Estimation Output Requirements

(1) Is the method required to provide the functional profile of the size of the system (for subsequent project management activities e.g. effort/cost estimation)?

The Indicative NESMA method only provides the overall functional size, with the Estimated and Full NESMA methods providing the complete functional profile. The Indicative NESMA method is therefore unsuitable for project management activities that require more detail than simply the overall functional size.

(2) The value of the size estimate in terms of significance of informing decisions.

The importance of the size estimate is at its greatest at the initial bid stage of a project, and decreases as the project progresses through the lifecycle. The Indicative NESMA method has been demonstrated to provide an unsatisfactory level of accuracy to be used as more than a rough indication of overall size. The value of this method is therefore limited to the relatively simple indication of project size. The Estimated NESMA method has been demonstrated to provide the optimal trade-off between estimation accuracy and estimation effort, but the value is limited at later stages in the project lifecycle. The Full NESMA method generally does not provide value to the development process of projects. In practice, the effort incurred by this method can only be justified early in the lifecycle of a project, but at this point there is typically insufficient requirements detail available to facilitate the development of a full size estimate. This pattern is consistent with the software estimation paradox, indicating that there is a disparity between when estimation is critical and when it is accurate.

5.6.2.3 Assessment of Practical Application of Estimation Method

(1) Is there a requirement for local historical data or calibration for the method to be used?

Each of the NESMA methods does not require adjustments to be made to the official guidelines in order to be utilised in practice. The Indicative NESMA method uses predetermined weightings to be applied to the raw count of Data Functions. The Estimated NESMA method assumes the complexity of the Data Functions and Transaction Functions to be applied equally across each respective function. In practice, the performance of these simpler versions of NESMA can be improved by adjusting the weightings, or assumed complexities, based on previously estimated projects. This would introduce a potential barrier to adapting these methods to reflect local development projects due to the need to establish historical data.

(2) How well do the method guidelines relate to commercial projects in practice?

The completion of Software Sizing using the NESMA methods in the previous chapters facilitated the assessment of the practical application of this approach. The following issues were identified when using the NESMA methods:

(i) Requirement for a Data Model to be available

(ii) Selection of weightings for Indicative NESMA method

The official NESMA guidelines specify that a data model must be present in order for the Indicative NESMA method to be completed. Two different weightings are prescribed for the formula according to whether or not the data model being examined has been normalised. The stage in the project lifecycle at which the size estimate is developed affects the degree to which these guidelines can be followed. At the initial tender stage of a project it is common for no data model to be present in the requirements documentation, likely due to the customer being responsible for this documentation. It is therefore necessary that such a model is developed by the estimator from the available system description. The inclusion on the official NESMA website of a worked example of deriving the necessary detail from a user requirement statement represents a tacit endorsement of such an approach. The issue, in practical use, is therefore the objectivity inherent in deriving the necessary detail from the tender document rather than the feasibility of doing so. The extent to which the logical files identified from the data model have been normalised is subject to greater uncertainty at this early stage. Consequently, the suitability of either of the predefined weightings is also less certain at the earliest stage of a project. This second issue of selecting the weightings, in particular, is generally not acknowledged within current published research utilising this method. The first issue is not present at the functional specification stage for the sample projects in this research. The second issue is less significant for data models developed at the functional specification stage for these same projects. The degree to which projects outside of Equiniti-ICS conform to this pattern cannot be discerned from related research, but the common nature of the tender process supports the wider applicability of these observations. Removing the dependency on the Indicative NESMA method weightings would allow for increased consistency by avoiding this potential source of variation, as well as providing greater transparency in how the estimate is developed.

5.6.2.4 Accuracy of Indicative NESMA Assumptions

The accuracy of the number of Transaction Functions predicted by the Indicative NESMA method, when compared with the actual number of Transaction Functions specifically identified by the Full NESMA, provides an indication of the effectiveness of this method. In the sizing estimation research in Chapter 3, the raw counts of each component type were identified during the completion of the function counting. The sizing data developed from the functional specification stage is used as it represents the most complete statement of the required functionality. In order to extrapolate the number of predicted Transaction Functions, the assumptions underlying the Indicative NESMA method are applied to the raw counts of the Data Functions as follows:

- For each identified ILF function, count three EI functions.
- For each identified ILF function, count one EO function.
- For each identified ILF function, count one EQ function.
- For each identified EIF function, count one EO function.
- For each identified EIF function, count one EQ function.
- For any FPA tables ILF function, count one EI, one EO and one EQ function.

The EI functions for each ILF correspond to 'Add', 'Amend' and 'Delete' functions from the NESMA assumptions. For this analysis, the normalised weightings (25 and 10) have been used for the Indicative NESMA method, so each ILF will expect only one EO function. The NESMA sizing approach treats entities used for constants, decoding etc. as a special case, referring to these as FPA tables. Internally maintained entities of this type are grouped together as one ILF, and externally maintained entities grouped together as one EIF. The NESMA assumptions specify that one EI, one EO and one EQ is required for the FPA tables ILF, and that no Transaction Functions are required for the FPA tables EIF. This procedure was applied to the complete dataset of eleven projects used in this research. Table 57 presents these results, along with the percentage difference in each case. In a study of 42 projects (van Heeringen et al., 2009), the raw counts of each component type identified according to NESMA guidelines were also provided for each project. As the same normalised weightings were used in the van Heeringen et al. study, the same approach outlined above was applied to the larger dataset. As the presence of FPA table Data Functions was not indicated in the van Heeringen et al. data, all of the data functions were treated as normal, but the effect of any such functions being present would at most lead to a reduction of two EI, one EO and one EQ from the total number of expected transaction functions for any project. The effect on the overall results would therefore be insignificant in this case. Due to the presence of no transaction functions in some components of eight of the projects in the dataset, these projects were excluded from this assessment as no relative inaccuracy could be calculated for these components. Table 57 includes the overall averages obtained from this analysis of the van Heeringen dataset.

The main observations to be made from these results are:

(1) The greatest inaccuracy is found with the EQs, where, on average, the number of expected EQ was 264.76% more than actually identified during the function count. The counted number exceeded the expected number of functions in only one of the projects. This pattern was significantly more extreme in the van Heeringen et al. dataset where the average difference exceeded 1100% and every project contained less EQ than expected.

(2) The inaccuracy found with the EOs was significant, but with a different pattern than found with the EQs. Each of the eleven projects contained more EOs than expected by the Indicative NESMA method, with the average difference being 56.57%. A similar average difference was observed in the van Heeringen et al. dataset, with 31 out of the 34 projects (91%) included, conforming to the pattern of exceeding the expected number of EOs.

(3) The pattern with the EIs was more varied, particularly within the smaller dataset of eleven projects that had a modest average difference of 25.66%. In terms of the overall direction of this difference, there was a modest difference of 9.09% between the expected and the counted number of EIs. The van Heeringen et al. dataset exhibited a clearer pattern with 71% of the projects containing less EIs than expected and a more significant average difference of 87.35%.

The results from the dataset of eleven projects show that the Indicative NESMA method underestimated nine of these projects, when compared with the Full NESMA method, and the average difference across all eleven projects was 26.30%. The results in Table 57 indicate that the significant over assumption of the number of EQs was partially compensated for by the under assumption of the number of EOs. A more reliable pattern may be observed from the larger van Heeringen et al. dataset. As indicated previously, with the van Heeringen et al. dataset the Indicative NESMA method was found to have an average difference of 28.44% compared to the Full NESMA method, which resulted in an average overall over estimation of 16.30%. The analysis in this chapter suggests that to a large extent these results were achieved by the significant inaccuracies found with over assuming the number of EIs and EQs effectively being partially compensated for by under assuming the number of EOs. As a rough indicator of system size the Indicative NESMA method seemingly produces satisfactory results, but it would be incapable of providing anything of value beyond this overall size. The van Heeringen et al.study provided no indication of the types of systems analysed, or of the nature of the requirements documentation used to inform the estimates, so the general applicability of these results is uncertain. The general reliability of the original results can be assured as the projects were drawn from 'many' different organisations and the measurements were reviewed by a peer NESMA certified analyst.

5.6.2.5 Analysis of Estimated NESMA Assumptions

The Estimated NESMA method removes the need to assess the complexity of the identified functions by assuming that, on average, each instance of a particular BFC type will have the same complexity level. The suitability of this approach is dependent on how closely the functional profile of a project adheres to the anticipated pattern of complexity. The degree to which this sizing approach can be adapted is limited by requiring that each instance of a particular BFC type is allocated the same complexity level. This analysis investigates the accuracy of the functional sizes produced for each BFC type, for the dataset of eleven projects, in order to assess the degree to which the sample projects are suitable for the Estimated NESMA method. The effects of altering the complexity assumptions are then investigated in order to assess the extent to which the sizing results are affected by using a broad level of adjustment. In each case, the functional sizes are compared with those sizes obtained using the Full NESMA method.

The existing Estimated NESMA complexity assumptions are that the Data Functions are of low complexity, and the Transaction Functions are of average complexity. In the functional sizing completed previously there was insufficient detail to assess the complexity of the EIF components when using the Full NESMA method. The Estimated and Full NESMA methods therefore produced the same functional size for this component and there is no insight provided from altering the complexity assumption for the EIF component. The ILF component was rarely found to include functions with high complexity, so it is only considered necessary to investigate the use of average complexity for this component. The following alterations were therefore investigated for each project:

(i) ILF complexity level of Average
(ii) EI, EO and EQ complexity level of Low
(iii) EI, EO and EQ complexity level of High

Table 58 shows the relative average differences between the Full NESMA functional sizes and each set of complexity assumptions. The overall direction of the differences is included to show whether the assumptions result in overestimates or underestimates on average. The functional sizes for the existing Estimated NESMA method indicate that the average differences across the individual component types are greater than the overall size difference for the projects. The ILF and EQ components in particular demonstrate the more substantial relative differences at over 10% on average. The ILF component was generally underestimated, but the EQ component provided no clear pattern in terms of the direction of the relative differences. Altering the ILF complexity to average resulted in a substantial increase in the average difference to 31.21%. This indicates that for this dataset, the estimation performance cannot be improved as an insufficient proportion of ILF functions exceed the current Estimated NESMA assumption of low complexity. Altering the EQ component to either low or high complexity resulted in substantial increases in the relative average differences for functional size. The overall direction demonstrated uniform patterns of under and over estimation respectively. The EI and EO components demonstrated a similar pattern using the existing Estimated NESMA assumptions. In both cases there was an overestimation of around 5%, suggesting more limited potential for improvement. This was reflected in the results obtained from altering the complexity for these components. Using low complexity resulted in a more substantial average difference, with underestimation occurring in each project. Using high complexity simply increased the extent of the pattern of overestimation, resulting in average differences of around 50%.

In terms of the accuracy of the functional profile, this analysis indicates that there is scope for improvement across the BFC types when using the Estimated NESMA method. For this dataset, altering the complexity assumptions resulted in more substantial average differences for every change investigated. This validates the specific assumptions specified in the NEMSA guidelines to some degree, but illustrates the limitations for broadly applying the same complexity across each instance of a BFC type. Improving the accuracy of the functional profile may therefore require consideration of complexity at a finer level.

5.6.2.6 Degree of Utility of Simplified NESMA Methods

The results from Chapter 3 demonstrate that the Indicative NESMA method provides insufficient accuracy to be used beyond as a rough indication of size. The limited transparency of the method and the reduced information produced by the method has been highlighted in Section 5.6.2.4 as hiding inaccuracies in the assumed transaction functionality for a project. The Indicative NESMA method can therefore be considered unsuitable for supporting project management activities requiring detailed sizing data. The extent of the relevance of the method is primarily limited to the earliest stages of the software lifecycle where requirements documentation is least detailed.

The Estimated NESMA method has been demonstrated in Chapter 3 to provide sufficient overall accuracy to negate the use of the Full NESMA method. The estimation effort required may not be considered acceptable in practise, for larger projects in particular, in later stages of the project lifecycle.

The value of software sizing is dependent upon obtaining an acceptable trade-off between the utility provided by the size estimate and the effort required to develop that estimate. The trade-off can be considered acceptable at the earliest bid stage of a project, in providing a supporting role to existing expert -based estimation approaches. The existing simplified NESMA methods do not provide an appropriate trade-off for use at the later Functional Specification stage.

5.6.2.7 Simplified Sizing Method Requirements

The Indicative NESMA method only produces an overall functional size rather than individual sizes for each of the component types. This provides a significant reduction in transparency in terms of how the overall functional size is composed, which limits the extent to which the results can subsequently be validated. The potential effect of the functional profile on the subsequent development effort required was highlighted previously in this chapter. In terms of project planning, having a more detailed breakdown of software size across functional areas of a project may therefore provide an indication of how development effort may be distributed across a project.

The extent to which the weightings are suitable will depend on the nature of the project being developed, which may vary significantly from one application type to another. The weightings are based on the typical profile of a software application, thus requiring modification for atypical projects. In particular, NESMA identifies variation in the number of associated EO as a reason for changing the multipliers used. While such calibration for local development projects may achieve the desired applicability, this may introduce the burden of maintaining local historical sizing data in order to establish reliable weightings. The degree of flexibility in adapting to different application types is still limited by projecting the same profile across each Data Function.

This research phase is therefore concerned with adapting the NESMA approach to achieve a level of accuracy and a degree of utility beyond that of the Indicative NESMA method. The achievement of this aim involves the incorporation of aspects of the Estimated and Full NESMA methods without incurring the same degree of required estimation effort. The Estimated NESMA method requires the identification of each instance of a BFC. The estimation effort incurred is therefore greater than with the Indicative NESMA method. The Estimated method assigns the same complexity classification for each instance of a specific BFC type. The assumption made is that the complexity of each BFC type will, on average, be the same as the assigned complexity classification. This approach does not provide flexibility for refining the estimate in situations where a project's BFC complexity profile differs significantly from the assumption. The Full NESMA method requires the complexity of each instance of a BFC to be assessed, which is often the most time consuming aspect of the functional sizing. The Estimated and Full NESMA methods have a greater input detail requirement, since each individual instance of a BFC is required to be considered. The requirement of identifying the 'typical' profiles of occurrences of BFC and their associated complexity would reduce the input detail burden of the functional sizing approach. The development of an adapted approach of functional sizing should maintain a degree of consistency with the application of the standardised functional sizing methods to ensure compatibility. This enables the resulting size estimates to be compared with and, if necessary, converted into other equivalent functional sizing measures. The sizing method requirements (SMR) for the proposed simplified method are therefore as follows:

(SMR1)	- Preserve consistency with the general components and principles of the FPA approach.
(SMR2)	- Remove the dependency on ‘weightings’ (NESMA or local) to be applied to the identified Data Functions.
(SMR3)	- Provide the full functional profile i.e. FP sizes for each BFC type.
(SMR4)	- Identify the ‘typical’ functional profile, in terms of Transaction Functions to be associated with the Data Functions.
(SMR5)	- Allow the complexity of both the Data Functions, and the identified profile of Transaction Functions, to be assigned at a finer level.
(SMR6)	- Provide a level of estimation effort below that of the Derived Complexity methods (such as the Estimated NESMA method).

The provision of the complete functional profile (SMR 3) enables an assessment of the adapted method to be made against the Full NESMA method. SMR 4 and SMR 5 facilitate the accommodation of increasing levels of input requirements detail into the size estimate. An assessment of the estimation performance can be made at distinct levels. SMR 6 can be evaluated against the estimation effort figures for each NESMA method from Chapter 3.

Table 57 - Comparison of Expected and Full Counted NESMA Profiles

Project	Counted ILF	Counted EIF	Expected EI	Counted EI	Percentage Difference	Expected EO	Counted EO	Percentage Difference	Expected EQ	Counted EQ	Percentage Difference
A	16	3	46	41	12.20%	18	60	-70.00%	18	14	28.57%
B	29	4	85	69	23.19%	32	164	-80.49%	32	9	255.56%
C	51	2	151	167	-9.58%	52	151	-65.56%	52	51	1.96%
D	20	1	58	85	-31.76%	20	77	-74.03%	20	25	-20.00%
E	27	1	79	75	5.33%	28	68	-58.82%	28	5	460.00%
F	9	1	25	17	47.06%	10	13	-23.08%	10	9	11.11%
G	23	1	67	75	-10.67%	24	37	-35.14%	24	20	20.00%
H	21	4	61	87	-29.89%	25	82	-69.51%	25	15	66.67%
I	18	1	52	31	67.74%	19	30	-36.67%	19	1	1800.00%
J	54	1	160	118	35.59%	55	103	-46.60%	55	16	243.75%

Project	Counted ILF	Counted EIF	Expected EI	Counted EI	Percentage Difference	Expected EO	Counted EO	Percentage Difference	Expected EQ	Counted EQ	Percentage Difference
K	53	3	157	173	-9.25%	55	146	-62.33%	55	38	44.74%
		Average Difference (Overall Direction)			9.09%			-56.57%			264.76%
		Average Difference			25.66%			56.57%			268.40%
		Average Difference (Overall Direction) (calculated from van Heeringen dataset)			77.50%			-28.59%			1114.85%
		Average Difference (calculated from van Heeringen dataset)			87.35%			54.56%			1114.85%

Table 58 - Effect of Adjusting Estimated NESMA Complexity Assumptions

Complexity Assumptions	**ILF Average % Difference (Overall Direction)**	**ILF Average % Difference**	**EI Average % Difference (Overall Direction)**	**EI Average % Difference**	**EO Average % Difference (Overall Direction)**	**EO Average % Difference**	**EQ Average % Difference (Overall Direction)**	**EQ Average % Difference**
Estimated NESMA (Standard Assumptions)	-8.16%	10.43%	4.53%	6.78%	4.75%	5.43%	-1.07%	11.67%
Estimated NESMA (Average ILF)	31.21%	31.21%	-	-	-	-	-	-
Estimated NESMA (Low EI,EO,EQ)	-	-	-21.60%	21.60%	-16.20%	16.20%	-25.80%	25.80%
Estimated NESMA (High EI,EO,EQ)	-	-	56.80%	56.80%	46.65%	46.65%	48.39%	48.39%

5.6.3 Research Stage 2: Development of Simplified Sizing Method

The development of the simplified sizing method incorporates learning from the application of each NESMA method on an initial five projects from the dataset. The sizing method is refined using these same projects until each of the sizing method requirements has been met. These requirements are addressed within this research stage as follows:

(SMR1) - Preserve consistency with the general components and principles of the FPA approach.

The minimum fundamental requirement of the sizing method to be developed is the identification of the ILF and EIF components. These are therefore required to be identified according to the existing NESMA guidelines. Consistency with the assessment of the related Transaction Functionality is addressed concurrently alongside the remaining requirements.

(SMR2) - Remove the dependency on ‘weightings’ (NESMA or local) to be applied to the identified Data Functions.

This requirement is to be addressed in the initial development of the sizing method.

(SMR3) - Provide the full functional profile i.e. FP sizes for each BFC type.

This requirement is to be addressed in the initial development of the sizing method.

(SMR4) - Identify the ‘typical’ functional profile, in terms of Transaction Functions to be associated with the Data Functions.

This requirement is to be partially addressed during the initial development, and then this aspect is to be completed during the refinement of the sizing method.

(SMR5) - Allow the complexity of both the Data Functions, and the identified profile of Transaction Functions, to be assigned at a finer level.

This requirement is dependent upon completion of the initial development of the sizing method. It is therefore to be addressed during the refinement of the method, and may necessitate amendments to how SMR 4 is addressed.

(SMR6) - Provide a level of estimation effort below that of the Derived Complexity methods (such as the Estimated NESMA method).

This requirement will be an on-going concern during the development and refinement of the sizing method. The refined method is to be evaluated in terms of the estimation effort incurred before the method is applied to the remaining six commercial projects in the next research stage.

5.6.3.1 Simplified Sizing Levels

The simplified sizing method to be developed is required to facilitate the refinement of estimates according to the amount of requirements detail in the project documentation. From RQ4, the optimal amount of requirements detail to be considered by the method is investigated in this research phase by evaluating the effect on simplified size estimation performance by incorporating an increasing level of detail. This evaluation therefore requires the definition of distinct software sizing levels from which sizing results can be compared. Figure 5.2 provides an overview of the three different simplified sizing levels that are to be developed.

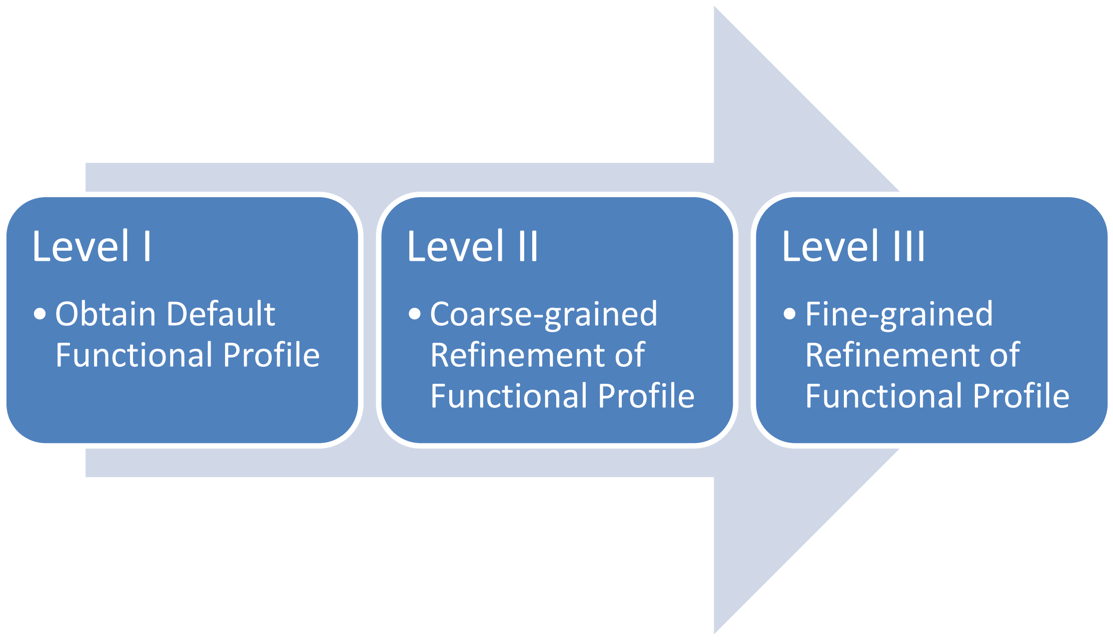
Level I
• Obtain Default Functional Profile
Level II
• Coarse-grained Refinement of Functional Profile
Level III
• Fine-grained Refinement of Functional Profile

Figure 5.2 - Overview of Simplified Sizing Levels

The Level I sizing is concerned with providing the ‘default’ functional profile of a project in terms of the sizes of each BFC type, therefore satisfying SMR2 and SMR3. This default functional profile should be obtained by identifying the required Data Functions and deriving the required Transaction Functions accordingly. Level II sizing is concerned with refining this default functional profile by considering the Transaction Functions described by the project documentation (SMR4). These refinements should be considered to be a coarse-grained refinement for a specific project, as the profile of Transaction Functions should be related to all Data Functions. Level III sizing involves assessing the extent to which types of Transaction Functions are required across the Data Functions (SMR5). These refinements should be considered to be a fine-grained refinement of the functional profile, as the specific types of Transaction Function are more explicitly estimated.

The simplified sizing levels are focused on answering the Research Questions presented in Section 5.5.4. The sizing method should therefore facilitate size estimates to be developed at distinct levels, but in practice should not be restricted to specific levels. The definition of the simplified sizing levels will be refined in conjunction with the refinement of the sizing method in order to ensure compatibility between both. The status of each SMR is indicated upon completion of an iteration of the development of the sizing method, along with the proposed actions for the next iteration. Prior to the first iteration of development, as shown in Table 59, each of the SMRs remains to be addressed. The initial development of the simplified sizing method is concerned with the transparent provision of the Functional Profile, so SMR2 and SMR3 are to be addressed in the first iteration. SMR1 and SMR6 are focused on maintaining their respective characteristics of the simplified sizing method.

Table 59 - Status of Sizing Method Requirements (Initial)

Sizing Method Requirement	Status	Next Action
SMR1 (NESMA Consistency)	Remaining	Maintain
SMR2 (Weighting Free)	Remaining	Address
SMR3 (Provide Functional Profile)	Remaining	Address
SMR4 (Determine Specific Profile)	Remaining	No Action
SMR5 (Refine Specific Profile)	Remaining	No Action
SMR6 (Low Estimation Effort)	Remaining	Maintain

5.6.3.2 Sizing Method Development (First Iteration)

The simplified sizing method uses the existing NESMA approach for identifying Data Functions, with the Estimated NESMA assumption of Low complexity applied to each Data Function. The functional size for the ILF and EIF components can therefore be calculated according to NESMA guidelines. The adaptations to the NESMA approach in this First Iteration are therefore focused on the estimation of Transaction Functions.

The first adaptation of the sizing method is concerned with addressing SMR2 by removing the use of weightings from the Indicative NESMA method. This requires the fundamental assumptions from which these weightings are formulated to be utilised explicitly rather than being incorporated into a weighted calculation. Figure 5.3 illustrates how the Indicative NESMA weightings can be adapted in this manner. The weightings currently specified by NESMA are dependent upon whether or not the data model for the project is normalised. The size estimates for this research phase are based on the functional specification stage, where the data models were assessed as being normalised, so the lower weightings (25, 10) are applicable in this case. The NESMA assumptions for this weighting are shown in Figure 5.3 as the Initial Adaptation (Level 1). Each ILF is assumed to have three associated EI functions (corresponding to ‘Add’, ‘Amend’ and ‘Delete’), as well as one associated EO and one associated EQ function. Each EIF is assumed to have one associated EO function and one associated EQ function. These Transaction Functions should therefore be considered to be an explicit part of the size estimate by recording the number and type of the functions.

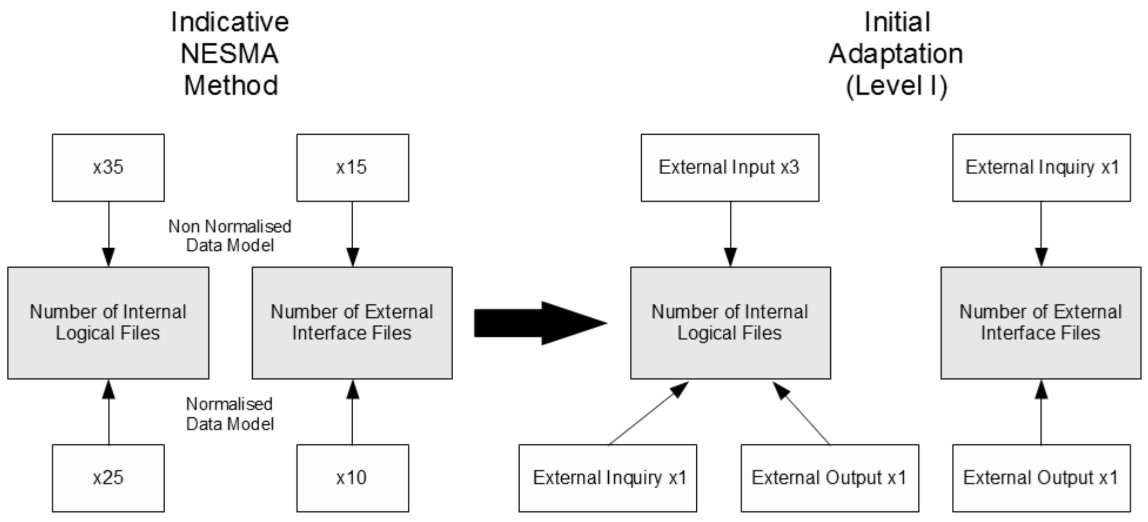
Indicative NESMA Method
x35
x15
Non Normalised Data Model
Number of Internal Logical Files
Number of External Interface Files
Normalised Data Model
x25
x10
Initial Adaptation (Level I)
External Input x3
External Inquiry x1
Number of Internal Logical Files
Number of External Interface Files
External Inquiry x1
External Output x1
External Output x1

Figure 5.3 - Initial Adaptation (Sizing Level I) for Simplified Sizing Method

SMR3 requires that the functional profile of a project is to be provided by the estimate i.e. functional sizes must be provided for each BFC type. The profile of the Transaction Functions, rather than each individual function, is to be represented in terms of the number of each type of function. The associated Transaction Functions, illustrated in Figure 5.3, must therefore be represented in a manner that enables the number of functions, and hence the functional size, to be determined. The approach taken to representing this functionality is to classify the Transaction Functions according to the type of functionality they represent, the number of functions required and the complexity of these functions. Each instance of a transaction function type corresponds to an instance of a Transaction Class that will be applied to the Data Functions. The initial parameters for a Transaction Class are therefore:

(1) Transaction Function Type – [EI or EO or EQ]

Records the BFC type represented by the Transaction Class

(2) Transaction Level – [ILF or EIF]

Records whether the Transaction Class is applied to each ILF or each EIF

(3) Transaction Complexity – [Average]

Records the complexity level of the functions represented by the Transaction Class

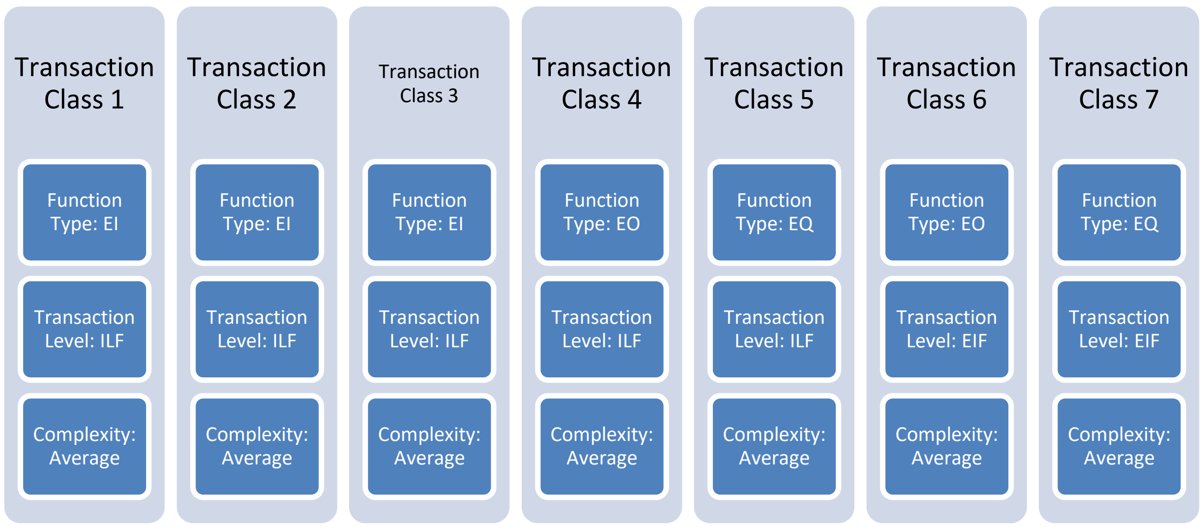

Figure 5.4 - Initial Default Transaction Classes

This would therefore correspond to seven initial Transaction Classes for the initial size estimate, as shown in Figure 5.4. The initial default values for each Transaction Class parameter, illustrated in Figure 5.4, are derived from the assumptions of the simplified NESMA methods. The Indicative NESMA method assumptions determine the initial Transaction Function Type, Transaction Level and Transaction Proportion. The Estimated NESMA method assumptions determine the initial Transaction Complexity. For the simplified sizing method, the Data Functions are assumed to be of low complexity and the Transaction Functions to be of average complexity, with the standard weightings applied. This approach removes the issue of selecting an appropriate weighting by transparently applying Transaction Classes to the Data Functions. This Level I adaptation assumes that the data model used in the requirements documentation is normalised. For use at an earlier stage of a project lifecycle it may be appropriate to incorporate additional initial Transaction Classes, such as another EO, in order to reflect a less refined data model.

Provision of the functional profile requires that the functional size for each Transaction Function Type is calculated from the detail recorded in the Transaction Classes. This calculation is made using the present NESMA weighting values. The functional size of a Transaction Class can be calculated using the formula:

*Number of FP = (Number of Transaction Level Components * Complexity Weighting of Function Type)*

For example, using a project with 20 ILF and an EI Transaction Class of Average complexity, the functional size would be:

*Number of FP = (20*4) = 80*

To obtain the functional size for the EI component, the functional sizes of each of the Transaction Classes of this type would be added together. The complexities for each Transaction Class are assigned from the Estimated NESMA assumptions i.e. 'Average' complexity. This approach is also applied to the EO and EQ Transaction Classes in order to obtain the functional size for these components.

The development progress for the simplified sizing method at this point is summarised in Table 60. The software sizing activities have been limited to following the existing NESMA guidelines for Data Functionality. The adaptations for sizing the Transaction Functionality are essentially automated as they are derived from the Data Functionality sizing data without any intervention from the estimator. SMR1 has therefore been maintained during this iteration of the development process. The estimation effort required is identical to that incurred from using the Indicative NESMA method, so SMR6 has also been maintained during this iteration. The adapted approach to obtaining the Transaction Functionality functional size has removed the use of weightings applied to the Data Functionality sizing data, fully satisfying SMR2. The Functional Profile can be obtained from the adaptations, fully satisfying SMR3, as functional sizes are provided for each BFC. The functional sizes for the Transaction Functionality components can be considered to be a 'default' Functional Profile derived from the Data Functionality. The next iteration is focused on estimating the specific profile of functionality for a project by incorporating consideration of the specification of Transaction Functionality in the project documentation.

Table 60 - Status of Sizing Method Requirements (First Iteration Completed)

Sizing Method Requirement	Status	Next Action
SMR1 (NESMA Consistency)	Maintained	Maintain
SMR2 (Weighting Free)	Completed	No Action
SMR3 (Provide Functional Profile)	Completed	No Action
SMR4 (Determine Specific Profile)	Remaining	Address
SMR5 (Refine Specific Profile)	Remaining	No Action
SMR6 (Low Estimation Effort)	Maintained	Maintain

5.6.3.3 Sizing Method Review (Second Iteration)

In this second iteration the initial default profile is amended to consider the Transaction Functionality required for the specific project. The completion of software sizing using NESMA in the preceding chapters provided insight into how the projects included in the research dataset are not consistent with the assumptions of the simplified NESMA methods. The necessary adaptations can therefore be developed from these inconsistencies.

(i) The analysis of the number of transaction functions assumed by the Indicative NESMA method, and those identified from the more detailed methods revealed significant differences. The initial default profile provided by the Level I sizing is representative of the Indicative NESMA assumptions. It is therefore necessary to enable these initial Transaction Classes to be amended. The first step is to allow new Transaction Classes to be added to the initial set, and for existing Transaction Classes to be removed if they are not required. For example, additional EO Transaction Classes may be required due to more extensive reporting requirements. Some types of functionality, such as 'Delete' functionality may not be required, which would result in the removal of an EI Transaction Class.

(ii) The Indicative NESMA method assumes that the Transaction Functions that would be associated with each Data Function are applied at the level of the Logical File. These Logical Files would themselves consist of a varying number of Record Element Types (RET). The previous completion of software sizing using the project documentation revealed that this assumption did not hold for some projects. In these cases, some aspects of the requested functionality were being specified at the RET level as opposed to the ILF level. This in itself would potentially lead to a significant under estimation of the project's size by the Indicative NESMA method, as more Transaction Functions would be required for each ILF. The

Transaction Level parameter should therefore accommodate the allocation of the RET level in addition to the ILF level for Internal Data Functionality.

(iii) The Estimated NESMA method provided the assumed Transaction Complexity for each Transaction Class i.e. 'Average'. The relative accuracy of this assumption is dependent upon the degree to which projects conform to this overall pattern. In order to remove this dependency, it is necessary to enable the Transaction Complexity of each Transaction Class to be modified. The initial size estimate for the Transaction Classes can be refined by identifying the overall profile of the specific transaction functionality required. Table 61 shows the boundary values for determining the complexity of EI components, with a similar table used for EO and EQ components.

Table 61 - Complexity Boundaries for EI Components (adapted from NESMA, 2004)

File Types Referenced	**Data Element Types**		
	1-4	5-15	>15
0-1	Low	Low	Average
2	Low	Average	High
3 or more	Average	High	High

The project documentation includes a detailed data model, listing the Data Element Types (DETs) within each RET. The transaction functionality is specified in a consistent structure within the Functional Specification. For example, each functional area in a project, corresponding to specific Data Functions, is outlined within the Functional Specification in the form of recognisable patterns such as 'Add', Modify', 'Delete' etc. The general profile of the functionality associated with a Transaction Class can consequently by examining the 'typical' specification of this functionality. It is necessary to assess the functionality associated with each Transaction Class, in order to determine the general complexity of the functionality. This assessment is focused on where the functionality generally fits into the boundaries for Low, Average or High complexity. It can be guided by using suitable parameters such as:

- Transaction Functions with in excess of 15 DETs
- Transaction Functions with 3 or more File Types Referenced
- Transaction Functions consisting of 'minimal' DETs (i.e. less than 5)

These parameters provide an indication of the types of Transaction Functions which are likely to differ from the initial assumption of Average complexity. The functionality associated with a

Transaction Class may thus be determined to generally be of Low or High complexity, and modified accordingly.

These adaptations result in the following amended parameters for a Transaction Class:

(1) **Transaction Function Type** – [EI or EO or EQ]
Records the BFC type represented by the Transaction Class

(2) **Transaction Level** – [ILF/RET or EIF]
Records whether the Transaction Class is applied to internal or external Data Functionality
Internal Data Functionality can be recorded/modified at either the ILF or RET level

(3) **Transaction Complexity** – [Low or Average or High]
Records the complexity level of the functions represented by the Transaction Class
Complexity level can be modified

The commercial projects included in this dataset do not provide scope for refining the estimates of the EIF component and this BFC was an insignificant factor in the overall functional size. In practice, Transaction Classes associated with External Data Functionality could be assigned at contrasting levels of detail where project documentation provides for this determination. This research phase does not provide the opportunity to evaluate this approach, restricting the modification of the Transaction Level of a Transaction Class to those associated with Internal Data Functionality. The definition of this parameter therefore does not provide for the EIF Transaction Level to be modified.

Preliminary usage of the sizing method on the initial projects indicated that modifications made broadly, according to the current parameters, could demonstrate relatively large changes in functional size from the initial default profile estimate. The value of the updated functional profile would be dependent, to some degree, on the relative inaccuracy of the initial default functional profile. The modifications possible with the current iteration of the sizing method are, however, limited in terms of their effectiveness due to their broad assessment of the Transaction Functionality. In some cases, the relative inaccuracy of the initial default functional profile could not be improved upon through the use of such broad modifications. It is therefore necessary to focus on facilitating a more fine-grained refinement of the Transaction Functionality in the subsequent iterations of the development process.

Table 62 shows the updated development progress for the simplified sizing method after the second iteration. The software sizing activities now incorporate a high level consideration of the required Transaction Functionality in order to determine the specific Functional Profile. The sizing method allows for broad modifications to the Transaction Classes, which would incur a minimal degree of estimation effort to complete. The assessment of the Transaction

Functionality has been limited to the identification of the existing NESMA components, taking the form of their overall profile rather than individual functions. For this iteration of the development process, it can therefore be concluded that SMR1 and SMR6 have therefore both been maintained. The adapted approach now provides a specific Functional Profile, in terms of the functional sizes for the Transaction Functionality components calculated from the Transaction Classes. SMR4 can be considered to be partially complete at the conclusion of this iteration. The next iteration is focused on refining the specific profile of functionality for a project by incorporating a more detailed consideration of the specification of Transaction Functionality in the project documentation. SMR5 can now be addressed, which may result in refinements in the defined parameters of the Transaction Classes. In the event that fundamental changes are made to the Transaction Classes, progress addressing SMR4 is considered ongoing in the next iteration.

Table 62 - Status of Sizing Method Requirements (Second Iteration Completed)

Sizing Method Requirement	Status	Next Action
SMR1 (NESMA Consistency)	Maintained	Maintain
SMR2 (Weighting Free)	Completed	No Action
SMR3 (Provide Functional Profile)	Completed	No Action
SMR4 (Determine Specific Profile)	Partially Complete	Address
SMR5 (Refine Specific Profile)	Remaining	Address
SMR6 (Low Estimation Effort)	Maintained	Maintain

5.6.3.4 Sizing Method Review (Third Iteration)

In the third development iteration the application of Transaction Classes to the identified Data Functions should be refined to enable a finer profile of Transaction Functionality to be estimated. Specific types of Transaction Functionality may not be necessary for every Data Function e.g. not every ILF may require 'Delete' functionality to be provided. Each specific Transaction Class should therefore only be applied to the approximate proportion of Data Functions that require this functionality. This proportion should be assigned in broad terms e.g. half, third, quarter etc. in order to maintain the simplicity of the sizing method.

An additional parameter is therefore added to the definition of a Transaction Class to record this proportion detail:

(1) Transaction Function Type – [EI or EO or EQ]
Records the BFC type represented by the Transaction Class

(2) Transaction Level – [ILF/RET or EIF]
Records whether the Transaction Class is applied to internal or external Data Functionality
Internal Data Functionality can be recorded/modified at either the ILF or RET level

(3) Transaction Complexity – [Low or Average or High]
Records the complexity level of the functions represented by the Transaction Class
Complexity level can be modified

(4) Transaction Proportion – [0.0 to 1.0]
Records the relative proportion of the Transaction Level components requiring this Transaction Class

Examination of the usage of the updated sizing method on the initial projects indicated that the more fine-grained modifications through the allocation of a Transaction Proportion demonstrated improvements in the relative accuracy of the functional size estimates. Assessing the relative Transaction Proportion for each Transaction Class presents challenges in maintaining SMR1 and SMR6.

- Does the assessment of the Transaction Proportion require specific knowledge outside of the existing NESMA guidelines?
- Does the estimation effort required extend the method beyond the simplified sizing methods?

The approach to assessing the Transaction Proportion should not require additional types of component be identified by the estimator. The ease with which the proportion can be estimated is affected both by how the different types of Transaction Functionality are classified, and by the clarity of the requirements documentation. The identification of the required Transaction Classes should therefore be focused on how an appropriate classification can be utilised to adhere to the software projects to be estimated.

Table 63 shows the updated development progress for the simplified sizing method after the third iteration. The refinement of the specific Functional Profile has focused on an assessment of the Transaction Proportion for each Transaction Class. This approach must itself be refined to ensure that SMR1 and SMR6 can be satisfied effectively. The next iteration is focused on refining the use of Transaction Proportion for enabling a fine-grained size estimate to be developed. As fundamental changes to the definition of Transaction Classes are still possible, progress addressing both SMR4 and SMR 5 is considered ongoing in the next iteration.

Table 63 - Status of Sizing Method Requirements (Third Iteration Completed)

Sizing Method Requirement	Status	Next Action
SMR1 (NESMA Consistency)	Under Review	Maintain
SMR2 (Weighting Free)	Completed	No Action
SMR3 (Provide Functional Profile)	Completed	No Action
SMR4 (Determine Specific Profile)	Partially Complete	Address
SMR5 (Refine Specific Profile)	Partially Complete	Address
SMR6 (Low Estimation Effort)	Under Review	Maintain

5.6.3.5 Sizing Method Review (Fourth Iteration)

The determination of the required Transaction Proportion for each Transaction Class involves the identification of the specific types of Transaction Functionality. NESMA recognises three different types of Transaction Functionality with the EI, EO and EQ components. In order to develop a fine-grained profile of this functionality a more detailed classification of the types of functionality should be considered. In the previous chapter such a classification was developed to facilitate the analysis of the relative sizes of specific types of functionality. This classification provides the basis for facilitating the estimation of a fine-grained level of Transaction Functionality. The suitability of the classification can be assessed in relation to SMR1 and SMR6.

NESMA Consistency: The current design of the Transaction Classes defines the type of functionality in terms of the NESMA BFCs. Enabling the estimator to identify suitable types of functionality for allocating Transaction Proportions should maintain consistency with these NESMA BFCs. This ensures that no additional types of transaction component are required to be identified by the estimator, limiting the additional knowledge required to develop the size estimate. It also restricts the nature of the types of functionality that may be associated with a specific Transaction Class. This serves to encourage an equivalent degree of consistency in the simplified sizing method as can be found with the existing NESMA approach. Each of the types of functionality included in the classification is directly related to a NESMA BFC. The classification is therefore considered to provide an appropriate template for the development of more fine-grained size estimates that remain consistent with the existing NESMA approach.

Low Estimation Effort: The use of the classification of transaction functionality types should not incur undue estimation effort when using the simplified sizing method. The ease of identifying the distinct types of transaction functionality is central to the degree of simplicity of the method. The process of developing a fine-grained level of transaction functionality incorporates the previous steps of the sizing method. An example of this process is illustrated in Figure 5.5.

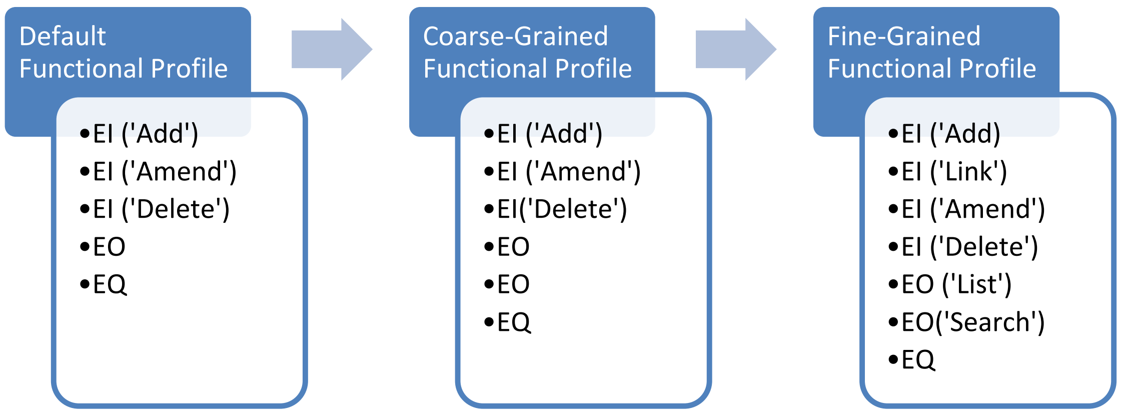

Figure 5.5 - Example of Refinement of Estimate of Transaction Classes

Figure 5.5 shows an example of the Transaction Class identified at each stage of the estimation process. The Transaction Classes identified at each stage of the process are listed with the BFC component identified for each Class, along with a more detailed categorisation of the functionality type where possible. The Default Functional Profile is derived from the Indicative NESMA assumptions, with the EI component Transaction Classes related to ‘Add’, ‘Amend’ and ‘Delete’ functionality respectively. The refinement of this initial estimate to obtain the Coarse-Grained Functional Profile generally prevents the identification of more specific types of functionality. The allocation of the Transaction Class to all of the Data Functions does not enable the Transaction Functionality to be decomposed into a more considered Functional Profile. The use of Transaction Proportion enables a Fine-Grained Functional Profile to be developed, whereby more specific types of functionality can be estimated. In Figure 5.5, examples of additional Transaction Classes for specific types of functionality from the classification are shown. EI related functionality such as ‘Link’ can be recognised at this level, while EO related functionality can be decomposed into such types as ‘List’ and ‘Search’. The classification did not designate different types of EQ related functionality due to the nature of the sample projects, although in practice appropriate distinctions could be drawn where feasible in other projects.

In relation to estimation effort the issue is how easily can more specific types of functionality be identified, and to what level of this division of functionality should the estimate consider. The flexibility of this simplified sizing method should enable the estimator to consider as great a level of detail as is appropriate for each estimate. The designation of the functionality, beyond the level of the BFC, for each Transaction Class should therefore be optional. The estimation of specific types of functionality should only be considered when it can be developed without incurring a significant additional effort. The ease of identifying relevant types of functionality is affected by the structure and format of the requirements documentation. For the sample projects the requirements documentation generally followed the same general structure. The format of the documentation became more uniform from the earlier projects to the more recent projects. Transaction Functionality associated with each Data Function was often presented in the same format, with distinct types of functionality such as ‘Add’, ‘Amend’ and ‘Delete’ easily identified from the section headings. In these cases, a detailed table of contents at the beginning of the documentation greatly reduces the time taken to ascertain the extent to which these types of functionality are performed. The specification of functionality within the main body of the documentation was also characterised by a uniform approach to their representation. The presence of commonly required distinct functionality e.g. ‘List’ functionality, within other functionality can be ascertained from the uniformity in how it is specified. The provision of screenshots assists in enabling overall patterns of functionality to be readily identified by communicating the main functionality contained within each functional area visually.

Defining a specific Classification of functionality would provide potential benefits for the simplified sizing method. Firstly, it guides the estimator as to how to recognise appropriate types of functionality when estimating the overall patterns of functionality. Secondly, it serves as a measure of consistency by ensuring that estimators are focused solely on the same types of transaction functionality.

There are, however, disadvantages to the use of a specific Classification of functionality. Firstly, it reduces the flexibility of the sizing method to adapt to the project that is to be estimated. Secondly, it further restricts the applicability of the sizing method to the specific application domain of the development organisation. Thirdly, it potentially increases the estimation effort required by guiding the estimator to consider more specific types of functionality than are necessary for the project. This can therefore be considered to adversely affect the satisfaction of SMR6.

SMR1 specifies that the simplified sizing method should be consistent with the main NESMA approach to software sizing. The types of functionality included in the Classification do not correspond to different types of component as defined by NESMA. Each type of functionality can be directly mapped to a NESMA BFC, ensuring that the sizing output detail is equivalent. The use of the Classification is therefore as a means of identifying overall patterns of required functionality, in terms of NESMA Transaction Functions, rather than identifying different types of component. The Classification of functionality is therefore not considered integral to ensuring consistency with the NESMA approach. In conclusion, this Classification is provided as an optional guideline for the simplified sizing method, but is not a specific feature of the method.

Each development organisation may benefit from developing a Classification that characterises their development projects. No formal Classification shall be incorporated into the sizing method in order to preserve the applicability of the method to that of FPA in general. The definition of a Transaction Class is now as follows:

(1) **Transaction Function Type** – [EI or EO or EQ]
Records the BFC type represented by the Transaction Class
Records the nature of the Transaction Functionality (Optional)

(2) **Transaction Level** – [ILF/RET or EIF]
Records whether the Transaction Class is applied to internal or external Data Functionality
Internal Data Functionality can be recorded/modified at either the ILF or RET level

(3) **Transaction Complexity** – [Low or Average or High]
Records the complexity level of the functions represented by the Transaction Class
Complexity level can be modified

(4) **Transaction Proportion –** [0.0 to 1.0]
Records the relative proportion of the Transaction Level components requiring this Transaction Class

Each Transaction Class can now be associated with a more specific type of transaction functionality. This should enable the Transaction Complexity for each Transaction Class to be more easily determined. Table 64 shows the status of each SMR at the end of this iteration of the development. The nature of the sizing process and the output produced by the simplified method are consistent with the overall NESMA approach. SMR1 is consequently considered to have been maintained at this stage of the development. The effort required to develop the size estimate is not anticipated to exceed the expectation of a simplified sizing method. This aspect of the method can only be formally evaluated after the completion of the development process. At this point SMR6 is considered to have been satisfied to the extent that such as assessment can be made prior to the formal evaluation in the next stage of the research. The refinement of the specific Functional Profile, in terms of Transaction Functionality, has been facilitated through the use of appropriate types of functionality with which to determine overall patterns for a project. This aspect of the simplified sizing method can now be considered for evaluation by formally defining specific levels of estimation detail in order to assess the trade-off between estimation effort and estimation accuracy. The degree to which the estimation of the Data Functionality can be refined within the simplified sizing method has not been addressed at this point in the development. The next iteration is therefore focused on how the initial estimate of the Data Functionality can be effectively refined to increase the accuracy of the Functional Profile.

Table 64 - Status of Sizing Method Requirements (Fourth Iteration Completed)

Sizing Method Requirement	Status	Next Action
SMR1 (NESMA Consistency)	Maintained	Maintain
SMR2 (Weighting Free)	Completed	No Action
SMR3 (Provide Functional Profile)	Completed	No Action
SMR4 (Determine Specific Profile)	Partially Complete	Address
SMR5 (Refine Specific Profile)	Partially Complete	Address
SMR6 (Low Estimation Effort)	Maintained	Maintain

5.6.3.6 Sizing Method Review (Fifth Iteration)

The estimation of the Data Functionality at this point in the development of the simplified sizing method follows the NESMA guidelines. The functional sizes produced for the ILF and EIF components are identical to those of the Estimated NESMA method i.e. each individual Data Function is identified and assumed to be of Low complexity. The potential for improving upon the accuracy of the size estimate for these components is therefore limited to the complexity each Data Function.

The relative accuracy of the Estimated NESMA method, and in particular the previous analysis of the Functional Profile size results, indicates that for the sample projects ‘most’ of the Data Functions conform to the assumption of Low complexity. The focus is therefore on assessing approximate proportions of Data Functions that are of Low, Average or High complexity. Such an assessment would require sufficient detail about the composition of each logical file i.e. the number of RETs and DETs. This level of detail would typically not be available at the initial stage of a project, but a detailed data model should be completed by the Functional Specification stage. The availability of a complete data model facilitates an efficient assessment of the approximate proportions of each level of complexity present in the Data Functions. As with the assessment of Transaction Functionality the use of complexity boundary values enables the Data Functions to be 'filtered' into appropriate complexity levels. Table 65 shows the boundary values for determining the complexity of Data Functions.

The initial estimated complexity, from the NESMA assumptions, for each Data Function is Low. In establishing the approximate proportions for each complexity level it is only necessary to identify those Data Functions that have greater than Low complexity.

Table 65 - Complexity Boundaries for Data Functions (adapted from NESMA, 2004)

Record Element Types	**Data Element Types**		
	1-19	20-50	>50
1	Low	Low	Average
2-5	Low	Average	High
>5	Average	High	High

The initial size estimate established how many RETs each ILF is comprised of, and the detailed data model lists the DETs within each RET. It is therefore possible to conduct a simple search of which Data Functions cross into boundaries for Average or High complexity, without requiring a detailed count of every DET. This is, in effect, the use of a 'threshold' procedure of allocating Average or High complexity by following suitable parameters such as:

- Look for RETs with in excess of 50 DETs
- Look for Data Functions with 5 or more RETs
- Look for Data Functions consisting of multiple 'small' RETs (i.e. less than 20 DETs in total)

This simple 'threshold' procedure produces a quick count of the number of Average and High complexity Data Functions, and enables suitable proportions to be calculated for each complexity level. The allocation of different complexities for Data Functions can be represented using an equivalent structure as found within the Transaction Classes. The definition of the structure of a Data Class is as follows:

(1) Data Function Type – [ILF or EIF]
Records the BFC type represented by the Data Class

(2) Data Complexity – [Low or Average or High]

Records the complexity level of the Data Functions represented by the Data Class

Complexity level can be modified

(3) Data Proportion – [0.0 to 1.0]

Records the relative proportion of the Data Function components represented by this Data Class

The initial size default estimate assumes a Data Complexity of Low and a Data Proportion of 1.0 for each Data Class. The Data Complexity of a Data Class can be modified during the refinement of the size estimate. The composition of the sample projects does not allow for the EIF component estimate to be refined beyond the initial default estimate. In practice, should sufficient detail be available for external data functionality the initial size estimate could also be refined in the same manner as the internal data functionality. For the initial size estimate there can be at most two initial Data Classes, shown in Figure 5.6, representing the entire internal and external data functionality respectively. During the refinement of the size estimate additional Data Classes can be added to enable different levels of complexity to be represented. For each Data Function Type there can be at most three Data Classes, corresponding to the three different levels of Data Complexity. The Data Proportion of the complete set of Data Classes for each Data Function Type should add up to 1.0, ensuring that the entire required data functionality is recorded within a Data Class.

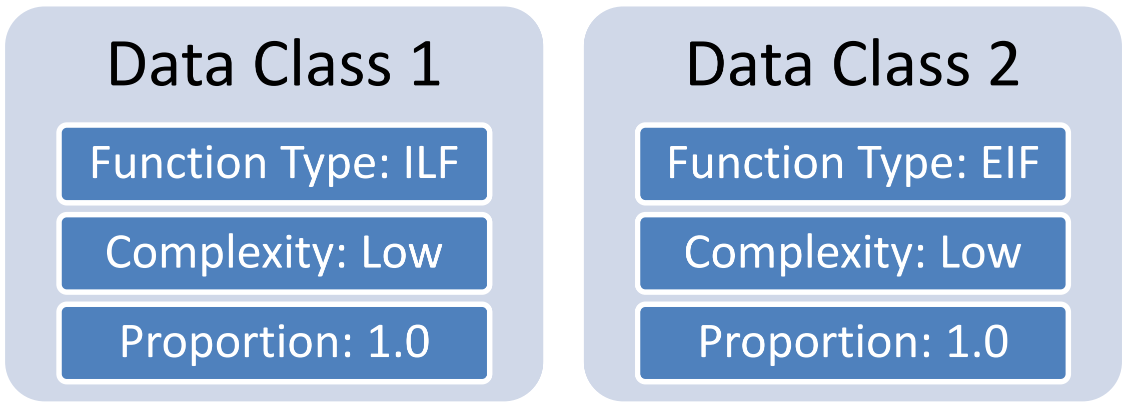

Figure 5.6 - Initial Data Classes

Table 66 shows the status of each SMR at the conclusion of this iteration of development. Initial assessment of the use of the Data Classes to refine the initial size estimate indicated that such refinement could be completed with an acceptable level of additional estimation effort. The potential for improving upon the accuracy of estimating data functionality is considered to be limited without completing the full NESMA assessment of complexity for each Data Function. SMR6 is considered to have been satisfied during the development of the sizing method. The estimation of the Functional Profile has been facilitated to the extent appropriate for a simplified sizing method. SMR3 and SMR4 are therefore also now considered to have been satisfied. The introduction of Data Classes has maintained consistency with the NESMA approach, as it corresponds to recording the profile of different levels of complexity but continues to consider the nature of data functionality using the same component types. No further action is required for any SMR, thus concluding the development process for the simplified sizing method. The final component of this Research Stage is to formally define the different software sizing levels with which to evaluate the use of the simplified sizing method.

Table 66 - Status of Sizing Method Requirements (Fifth Iteration Completed)

Sizing Method Requirement	Status	Next Action
SMR1 (NESMA Consistency)	Maintained	No Action
SMR2 (Weighting Free)	Completed	No Action
SMR3 (Provide Functional Profile)	Completed	No Action
SMR4 (Determine Specific Profile)	Completed	No Action
SMR5 (Refine Specific Profile)	Completed	No Action
SMR6 (Low Estimation Effort)	Maintained	No Action

5.6.3.7 Simplified Sizing Levels

The simplified sizing method defines Data Classes and Transaction Classes for representing the Functional Profile of a project. The initial size estimate derives the Functional Profile from the identification of the required Data Functions. This initial estimate can be refined using the dimensions of the Data Classes and Transaction Classes. The three dimensions of the Data Classes facilitate the following modifications:

(i) The complexity of a Data Class to be modified.

(ii) The use of multiple Data Classes to represent the relative proportion of Data Functions corresponding to each level of complexity.

The four dimensions of the Transaction Classes facilitate the following modifications:

(i) The addition or removal of instances of Transaction Classes.

(ii) Amending the level at which each instance of a Transaction Class is applied.

(iii) Amending the complexity to assign for each instance of a Transaction Class.

(iv) Amending the relative proportion of logical files or RETs to which a Transaction Class should be applied.

The extensibility of this sizing method is provided by facilitating the incorporation of increasing levels of input detail from the specification of user functionality. The degree of rigour incorporated into the size estimate can be varied according to the circumstances, enabling its application at any stage of the development lifecycle when size estimation is to be performed. The use of this simplified sizing method, when incorporating every aspect of the refinements, satisfies RQ1 and RQ2. In order to satisfy RQ3 and RQ4, different simplified sizing levels need to be defined and utilised so that the trade-off between estimation accuracy and estimation effort can be evaluated. These levels should incorporate increasing levels of input detail representing an increased degree of rigour in the estimation process.

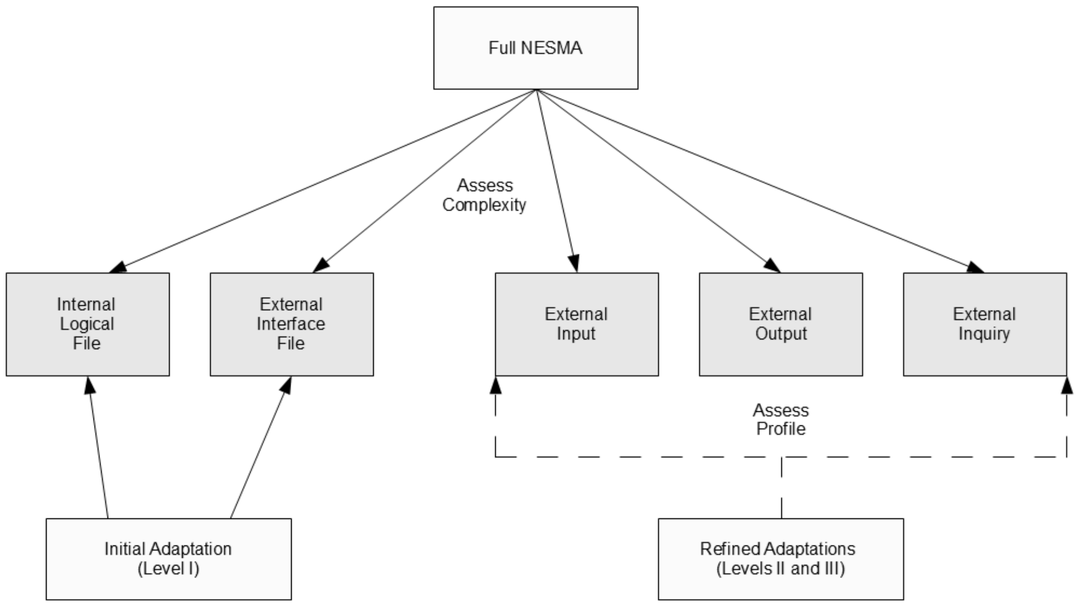

Figure 5.7 - Scope of Refined Adaptation for Simplified Sizing Method

The subsequent refined adaptations (Levels II and III) incorporate aspects of NESMA functional sizing beyond those involved in the Indicative NESMA Method. Figure 5.7 illustrates how this approach may be related to the different NESMA methods in terms of the sizing activities performed. In contrast to the full NESMA method, in which each of the Transaction Functions are identified, the refined levels of adaptation assess the overall profile of these Transaction Functions. The assessment of this profile is concerned with the 'average' transaction functionality associated with each Data Function. The classification of Level II and

Level III sizing enables an assessment to be made on the limits of size estimation performance when the same average transaction functionality is assigned equally across each data function.

Figure 5.8 outlines the estimation refinements made at each simplified sizing level. In Level I the estimator is focused solely on identifying the Data Functions (ILF and EIF) required by the system, represented as Data Classes. The associated Transaction Classes are essentially automatically derived, using NESMA assumptions, thus no additional effort is required relative to the Indicative NESMA method.

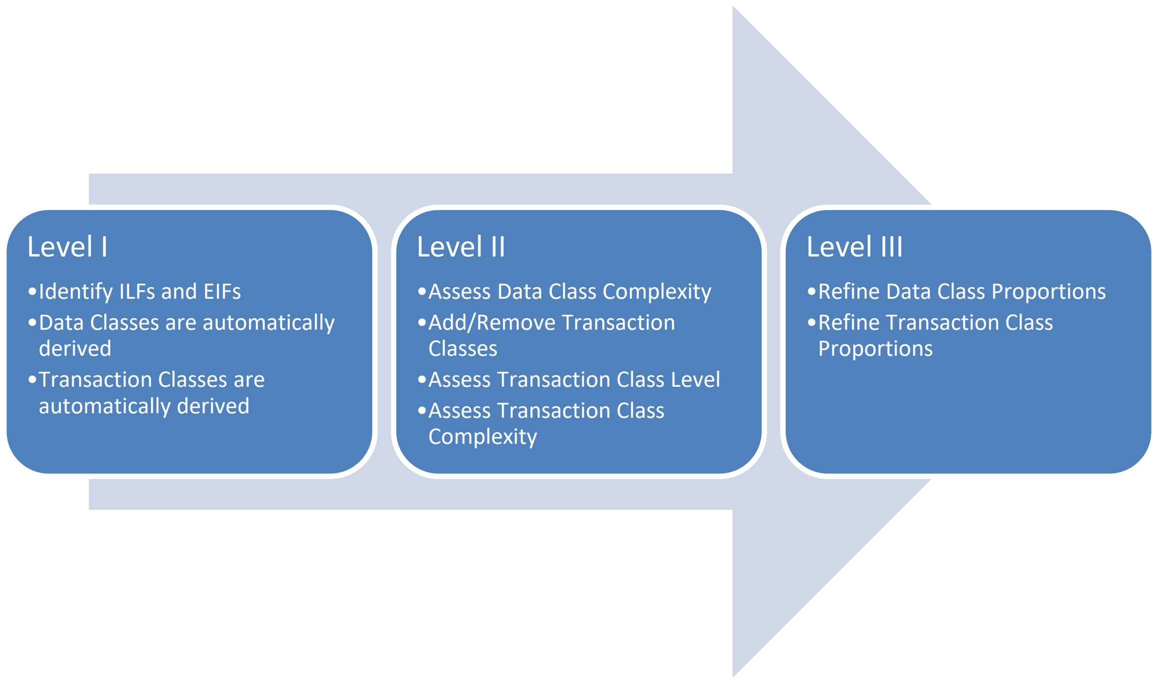
Level I
•Identify ILFs and EIFs
•Data Classes are automatically derived
•Transaction Classes are automatically derived
Level II
•Assess Data Class Complexity
•Add/Remove Transaction Classes
•Assess Transaction Class Level
•Assess Transaction Class Complexity
Level III
•Refine Data Class Proportions
•Refine Transaction Class Proportions

Figure 5.8 - Simplified Sizing Levels

None of the modifications outlined above are available at Level I. In Level II the estimator is concerned with the profile of the Transaction Functions, by assessing which Transaction Classes are required by the system. This determines their associated Transaction Level and Transaction Complexity. The first three modifications for Transaction Classes are available at Level II. The first modification for Data Classes is available in Level II, although no refinements were made to the Data Classes at this level as improvements could not be made by altering the complexity for all Data Functions. This reflects the characteristics of the sample projects, although for other types of applications it could in theory be appropriate to alter the complexity for all Data Functions. In Level III the estimator completes a more detailed assessment of the Data Classes by refining the complexity of required functions. The Transaction Classes are also refined in terms of the proportion of Data Functions the class should be applied to. For both Data Classes and Transaction Class all of the modifications are available at Level III. The key difference between Level II and Level III, in terms of proportional functionality, enables an assessment of the limit of applying the Estimated NESMA assumptions. At Level II the initial estimate of the Data Classes is maintained as the majority of Data Functions are of Low complexity. The Transaction Classes are applied across all of the Data Functions, in effect applying the same ‘typical’ functionality to all Data Functions. At Level III these estimates are refined so that proportions of functions can be estimated separately. Data Classes may be decomposed into smaller classes representing different levels of complexity. Transaction Classes may be applied more selectively at a specific proportion of the Data Functions.

The following steps are involved in obtaining the initial functional size (Level I) for each of the sample projects:

1. The identification of the required ILF and EIF for the project being analysed, including the individual RETs they are comprised of. The ILF and EIF components are each represented by their own Data Class.
2. Formulation of the initial 'default' Transactions Classes from the raw counts of ILF and EIF. These Transaction Classes are used to produce corresponding raw counts for EI, EO and EQ components.
3. Application of the assumptions of the Estimated NESMA method to the raw counts obtained in the previous stage i.e. ILF and EIF have low complexity, while EI, EO and EQ have average complexity. This calculation produces the complete Functional Profile for the project.

In terms of the estimator tasks, Level I is identical to that of the existing Indicative NESMA method. The formulation of the Data Classes and the Transaction Classes can essentially be automated once the individual Data Functions have been identified by the estimator.

In the next level of simplified sizing (Level II) the general profile of Transaction Functions associated with the Data Functions is assessed. This level enables the Transaction Level and Complexity for the Transaction Classes to be modified, but the functionality is still applied to all of the Transaction Level components. The following steps are involved in Level II of the simplified sizing for each of the sample projects:

1. Verify whether the initial Transaction Classes are typically required for most of the Data Functions. Any unnecessary Transaction Classes are deleted at this step.
2. Determine whether any additional Transactions Classes are found to be required for most Data Functions. Specify the necessary parameters for any Transaction Classes added at this step.

3. Through consideration of a detailed data model, listing DETs for each RET, determine whether the assumption of low complexity for the Data Functions is generally valid for the project.
4. Assess the Transaction Level and Transaction Complexity of the individual Transaction Classes by considering the number of RETs and DETs that are 'generally' involved in such functions.

The assessment of complexity does not involve any detailed counting of DETs, instead focusing on the identification of the general ‘boundary’ each type of function fell within. This enables an appreciation of the 'typical' complexity boundary for the Transaction Function associated with each Transaction Class to be developed. For the Data Functions there is no deviation from the overall assumption of low complexity in this dataset, so step 3 has no effect on the size estimates.

Level III sizing addresses the fine-grained assessment of the required functionality for both Data Classes and Transaction Classes. This level incorporates the following steps:

1. Assess the relative proportion of Data Functions, if any, which do not conform to the assumption of low complexity. If necessary, add Data Classes to represent a complexity of 'average' or 'high' as required.
2. Assess the relative proportion of Data Functions to which each Transaction Class should be applied. This assessment is made at a broad level, e.g. half, quarter etc. in order to maintain the desired reduction in estimation effort. This step may result in a current Transaction Class being split into a number of separate Transaction Classes, distinguished by considering more specific types of functionality or by different levels of complexity.

Step 1 utilises the ‘threshold’ approach to determining the relative proportion of Data Functions which conform to each level of complexity. For step 2, it is the 'general' characteristics of a Transaction Class that are being assessed based on the availability of appropriate detail in the requirements documentation. The degree to which Transaction Classes are split into smaller classes representing more specific types of functionality varies according to the ease of distinguishing and assessing such functionality.

In practice, the extent to which each aspect of this simplified sizing method can be utilised may be limited by the level of detail contained in the requirements documentation available at the time the size estimate is completed. For example, at the initial stages of a project there may not be sufficient detail about the DETs present in the Data Functions to facilitate even a broad assessment of the complexity, as required in steps ‘3’ and ‘4’ in Level II. However, assessing the relative proportion of Data Functions affected by each Transaction Class, from step '2' of Level III, may still be feasible at this initial stage of a project. The specific design of the simplified sizing levels is therefore a reflection of the research objectives, rather than specific guidelines as to how the sizing method should be utilised in practice. An example calculation using this simplified sizing method is provided in Appendix A.

5.6.4 Research Stage 3: Testing of Simplified Sizing Method

Stage 3 of the research comprises the testing of the simplified sizing method on the complete dataset, (Projects 'A' to 'K'), of commercial projects provided by Equiniti-ICS.

5.6.4.1 Size Estimation Measurements

Stage 2 of the research used the dataset, (Projects 'A' to 'E') in order to develop the simplified sizing method. The size estimates for the NESMA methods had therefore been previously developed. The effort required to use the NESMA methods was recorded as the size estimates were completed. The simplified sizing method was developed, and refined, during Stage 2 and applied to this dataset during Stage 3. Due to the process of refining the method and designing the sizing levels, the recorded effort for development of these size estimates represents 'judged' effort. The effort to complete Level I sizing is identical to that for the Indicative NESMA method, as it automatically derives the BFC values from the output of this method. The effort value for Level I sizing will therefore automatically be the same as that for the Indicative NESMA method for each project.

For the remaining projects in the dataset (Projects 'F' to 'K') no previous size estimates had been made. The simplified sizing levels, as previously defined, were used to develop size estimates for each of the projects in this dataset. In this case the actual effort required to develop each size estimate was recorded for each project, by updating an effort timesheet at the end of each estimation session. Each of the NESMA methods (Indicative, Estimated and Full) were also used to develop size estimates for these same projects, and the estimation effort recorded in the same manner.

The previous NESMA method size estimates and all of the size estimates in Research Stage 3 were completed by the same estimator. The use of different estimation methods in this way on the same project documentation introduces the possibility of biasing the estimation effort figures recorded. The estimator may in effect 'learn' about the project with one method, which enables the subsequent method to be completed more quickly. For the initial projects (‘A’ to ‘E’) as the estimation effort figures for the simplified sizing method were ‘judged’ the issue of bias is offset as the ‘learning’ effect does not impact the judgement. In order to counter this ‘learning’ effect for the remaining projects (‘F to ‘K’) the order in which the different estimation methods were used was designed to minimise a ‘learning’ advantage for any one method as follows:

(1) The Indicative NESMA method and Level I simplified sizing are identical in terms of only requiring the Data Functions to be identified from the requirements documentation. This task therefore only needs be completed once to provide the results for both approaches. The identification of these Data Functions was therefore completed first for each of the projects.

(2) The remaining NESMA methods (Estimated and Full) and the remaining simplified sizing levels (II and III) both require consideration of the Transaction Functions required in a project. The estimation activities differ in that the NESMA methods require each individual Transaction Function to be identified, while the simplified sizing method only requires the general profile of the Transaction Functions to be identified. The order of completion of these methods was varied evenly across the projects in order to ensure that each method was given an equal opportunity to benefit from previously acquired knowledge about a project. Any bias would therefore be cancelled out across the projects enabling the average estimation effort for each method to be compared more reliably.

In practice, performing functional sizing may require clarification from the system users in order to verify understanding of the requirements documentation. The software sizing in this research phase involved the use of documentation from completed projects, rendering clarification with the customer as beyond the scope of this research. The estimation effort incurred with each method was therefore not inclusive of this aspect of functional sizing. The figures reported in Tables 63 and 64 represent the time taken to analyse the requirements documentation and determine the functional size according to each specific method.

5.7 Empirical study results

Table 67 shows the number of Transaction Functions derived using the highest level of simplified sizing (Level III) and the associated percentage difference when they are compared with the counted number obtained using the Full NESMA method. Tables 68 and 69 show the effort required in using each of the variants of the NESMA standard to complete sizing for each of the commercial projects studied, together with comparable data from the three simplified sizing levels. Due to the difference in how the effort figures for the simplified sizing method were produced for the initial projects (‘A’ to ‘E’) and the remaining projects (‘F’ to ‘K’), the average percentage reductions in effort achieved by each method, compared to the Full NESMA method, are only provided separately for each subset of projects. Table 70 shows the average relative accuracy of the overall functional sizes and project BFC profiles obtained from the NESMA (Indicative and Estimation) methods and the simplified sizing method, compared to the results from the Full NESMA method. In Tables 67 to 70, the statistical significance of the results was assessed using the paired samples t-test (2-tailed). This procedure determines whether the mean difference between the results for each sizing method was statistically significant. For results that violated the assumption of a normal distribution that is required for the paired samples t-test, an alternative appropriate procedure was used as indicated.

Table 71 shows the correlations (using the Pearson correlation coefficient) between the BFC sizes produced by each estimation method and the overall functional size produced by the Full NESMA method. The Indicative NESMA method is not included in this final table as it does not provide sizes for each individual BFC type. The functional sizing results for each individual project are provided in Appendix A.

Table 67 - Comparison of Level III and (Full NESMA method) Counted NESMA Profiles

Project	Counted ILF	Counted EIF	Level III EI	Counted EI	Percentage Difference	Level III EO	Counted EO	Percentage Difference	Level III EQ	Counted EQ	Percentage Difference
A	16	3	47	41	14.63%	60	60	0.00%	18	14	28.57%
B	29	4	75	69	8.70%	142	164	-13.41%	10	9	11.11%
C	51	2	159	167	-4.79%	160	151	5.96%	52	51	1.96%
D	20	1	89	85	4.71%	69	77	-10.39%	20	25	-20.00%
E	27	1	77	75	2.67%	73	68	7.35%	6	5	20.00%
F	9	1	17	17	0.00%	14	13	7.69%	10	9	11.11%
G	23	1	87	75	16.00%	37	37	0.00%	22	20	10.00%
H	21	4	87	87	0.00%	77	82	-6.10%	18	15	20.00%
I	18	1	35	31	12.90%	27	30	-10.00%	1	1	0.00%
J	54	1	117	118	-0.85%	95	103	-7.77%	12	16	-25.00%

Project	Counted ILF	Counted EIF	Level III EI	Counted EI	Percentage Difference	Level III EO	Counted EO	Percentage Difference	Level III EQ	Counted EQ	Percentage Difference
K	53	3	171	173	-1.16%	147	146	0.68%	37	38	-2.63%
Overall Dataset	**Projects ‘A’ - 'K’**	**Average Difference (Overall Direction)**			4.80%			-2.36%			5.01%
		Average Difference			6.04%			6.31%			13.67%
		Level 1 Average Difference			25.66%			56.57%			268.40%
		Level I - Level III Mean Difference (%)			19.62%			50.26%			254.72%
		Level I - Level III Paired samples t-test *p* value (2 tail)			0.010			< 0.001			0.016 (Sign)

Table 68 - Comparable effort performance for the size estimation methods (Projects 'A' to 'E')

	Indicative NESMA	Level I	Level II	Level III	Estimated NESMA	Full NESMA
Effort (Hours) to Complete (% Reduction Relative to Full NESMA)						
Project A	2 (85.71%)	2 (85.71%)	3 (78.57%)	3.5 (75.00%)	6 (57.14%)	14 (0%)
Project B	4 (84.00%)	4 (84.00%)	5 (80.00%)	6 (76.00%)	9 (64.00%)	25 (0%)
Project C	6 (85.00%)	6 (85.00%)	7 (82.50%)	8 (80.00%)	16 (60.00%)	40 (0%)
Project D	4 (80.95%)	4 (80.95%)	5 (76.19%)	5.5 (73.81%)	12 (42.86%)	21 (0%)
Project E	2 (88.24%)	2 (88.24%)	3 (82.35%)	3.5 (79.41%)	7 (58.82%)	17 (0%)
Average Effort (Hours)	3.6	3.6	4.6	5.3	10	23.4
Average % Reduction Relative to Full NESMA	84.78%	84.78%	79.92%	76.84%	56.56%	-
Mean Difference (Hours) with Full NESMA	-19.80	-19.80	-18.80	-18.10	-13.40	-
Paired samples t-test *p* value (versus Full NESMA)	0.007	0.007	0.008	0.008	0.011	-
Mean Difference (Hours) with Estimated NESMA	-6.4	-6.4	-5.4	-4.7	-	-
Paired samples t-test *p* value (versus Estimated NESMA)	0.005	0.005	0.009	0.012	-	-

Table 69 - Comparable effort performance for the size estimation methods (Projects ‘F’ to ‘K’)

	Indicative NESMA	**Level I**	**Level II**	**Level III**	**Estimated NESMA**	**Full NESMA**
Effort (Hours) to Complete (% Reduction Relative to Full NESMA)						
Project F	1 (77.78%)	1 (77.78%)	1.25 (72.22%)	1.5 (66.67%)	2.5 (44.44%)	4.5 (0%)
Project G	1 (92.86%)	1 (92.86%)	1.5 (89.29%)	2 (85.71%)	5.5 (60.71%)	14 (0%)
Project H	1.5 (86.96%)	1.5 (86.96%)	2 (82.61%)	2.5 (78.26%)	5.5 (52.17%)	11.5 (0%)
Project I	1.5 (80.00%)	1.5 (80.00%)	1.75 (76.67%)	2 (73.33%)	3.5 (53.33%)	7.5 (0%)
Project J	4 (87.30%)	4 (87.30%)	5.5 (82.54%)	6.5 (79.37%)	15 (52.38%)	31.5 (0%)
Project K	5 (87.18%)	5 (87.18%)	6 (84.62%)	7 (82.05%)	16 (58.97%)	39 (0%)
Average Effort (Hours)	1.8	1.8	2.4	2.9	6.4	13.8
Average % Reduction Relative to Full NESMA	85.35%	85.35%	81.32%	77.57%	53.67%	-
Mean Difference (Hours) with Full NESMA	-12.0	-12.0	-11.4	-10.9	-7.4	-
Paired samples t-test *p* value (2 tail) (versus Full NESMA)	0.026	0.026	0.027	0.028	0.030	-
Mean Difference (Hours) with Estimated NESMA	-4.6	-4.6	-4.0	-3.5	-	-
Paired samples t-test *p* value (versus Estimated NESMA)	0.023	0.023	0.024	0.027	-	-

Table 70 - Summary of relative sizing inaccuracies of each sizing method compared to Full NESMA method (Overall Dataset – Projects 'A' to 'K')

Estimation Type	**ILF**	**EIF**	**EI**	**EO**	**EQ**	**Overall**
Percentage Inaccuracy Relative to Full NESMA method						
Indicative NESMA	-	-	-	-	-	26.30%
Level I	10.43%	0.00%	28.03%	54.29%	209.19%	22.19%
Level II	10.43%	0.00%	14.50%	11.29%	42.30%	4.86%
Level III	3.71%	0.00%	4.33%	4.91%	14.66%	2.38%
Estimated NESMA	10.43%	0.00%	6.78%	5.43%	11.67%	4.50%
Level I - Level II						
Mean Difference (%)	-	-	13.53%	43.00%	166.89%	17.33%
Paired samples t-test *p* value (2 tail)	-	-	0.066	<0.001	0.063 (Sign)	0.001
Level I - Level III						
Mean Difference (%)	6.72%	-	23.70%	49.38%	194.53%	19.81%
Paired samples t-test *p* value (2 tail)	0.001	-	0.003	<0.001	0.004 (Sign)	<0.001
Level II - Level III						
Mean Difference (%)	6.72%	-	10.17%	6.38%	27.64%	2.48%
Paired samples t-test *p* value (2 tail)	0.001	-	0.002	0.030	0.014	0.004
Level I - Indicative NESMA						
Mean Difference (%)	-	-	-	-	-	-4.11%
Paired samples t-test *p* value (2 tail)	-	-	-	-	-	0.104
Level I - Estimated NESMA						
Mean Difference (%)	-	-	21.25%	48.86%	197.52%	17.69%
Paired samples t-test *p* value (2 tail)	-	-	0.009	<0.001	0.001 (Sign)	0.001
Level II - Indicative NESMA						
Mean Difference (%)	-	-	-	-	-	-21.44%
Paired samples t-test *p* value (2 tail)	-	-	-	-	-	0.001
Level II - Estimated NESMA						
Mean Difference (%)	-	-	7.72%	5.86%	30.63%	0.36%
Paired samples t-test *p* value (2 tail)	-	-	0.031	0.089	0.013	0.688
Level III - Indicative NESMA						
Mean Difference (%)	-	-	-	-	-	-23.92%
Paired samples t-test *p* value (2 tail)	-	-	-	-	-	<0.001
Level III - Estimated NESMA						
Mean Difference (%)	-6.72%	-	-2.45%	-0.52%	2.99%	-2.12%
Paired samples t-test *p* value (2 tail)	<0.001	-	0.201	0.733	0.364	0.010

Table 71 - Correlations of BFCs with Overall Full NESMA Size for each Sizing Method (Overall Dataset – Projects ‘A’ to ‘K’)

Factor	**Correlation with Overall Size (Full NESMA)**	**Correlation with Overall Size (Estimated NESMA)**	**Correlation with Overall Size (Level I)**	**Correlation with Overall Size (Level II)**	**Correlation with Overall Size (Level III)**
Internal Logical Files	**0.897**	**0.923**	**0.923**	**0.923**	**0.897**
Statistical Significance (1-Tailed Test) (Benjamini-Hochberg)	***p<0.001***	***p<0.001***	***p<0.001***	***p<0.001***	***p<0.001***
External Interface Files	0.366	0.366	0.366	0.366	0.366
Statistical Significance (1-Tailed Test) (Benjamini-Hochberg)	*p=0.134*	*p=0.134*	*p=0.134*	*p=0.134*	*p=0.134*
External Inputs	**0.940**	**0.940**	**0.923**	**0.951**	**0.953**
Statistical Significance (1-Tailed Test) (Benjamini-Hochberg)	***p<0.001***	***p<0.001***	***p<0.001***	***p<0.001***	***p<0.001***
External Outputs	**0.891**	**0.925**	**0.934**	**0.936**	**0.926**
Statistical Significance (1-Tailed Test) (Benjamini-	***p<0.001***	***p<0.001***	***p<0.001***	***p<0.001***	***p<0.001***

Hochberg)					
External Queries *Statistical Significance (1-Tailed Test)* (Benjamini-Hochberg)	**0.765** ***p=0.003***	**0.751** ***p=0.004***	**0.934** ***p<0.001***	**0.752** ***p=0.004***	**0.704** ***p=0.008***

5.7.1 Number of Derived Transaction Functions

Each of the Transaction Classes identified during the sizing is applied to a specific number of Data Functions, thus enabling a specific number of Transaction Functions to be derived from these Transaction Classes. This enables the number of derived Transaction Functions to be compared with the number of actual Transaction Functions identified using the Estimated (or Full) NESMA method as a means of validating our results. The earlier results from Table 57, which showed the percentage difference for the 'expected' number of Transaction Functions, may be considered to be the same as the Level I sizing used in this research. The results presented in Table 62 show the derived Transaction Functions for the Level III sizing, thus enabling the improvements made through increasing the requirements detail to be assessed. The Level I average differences for the overall dataset are included for comparison.

Each of the types of Transaction Function demonstrated overall improvements, in terms of deriving the number of functions, from Level I to Level III sizing. The use of the paired samples t-test demonstrated statistically significant results for both the EI and EO Transaction Functions. Due to the distribution of the differences between the paired results for the EQ Transaction Functions being neither normal nor symmetrical, the paired samples sign test was used to test the statistical significance for this component. The paired samples sign test, denoted by (sign) in the results table, determines the statistical significance of the median difference between the paired results. The results for the EQ Transaction Functions were also found to be statistically significant, at *p*=0.016. Despite showing the most substantial improvement, in terms of the average difference, this was the highest *p* value across the Transaction Function types. This may be due to the effect of some individual projects demonstrating extreme levels of relative inaccuracy for Level I sizing. It is intuitive in those cases that suitable consideration of the requirements documentation can provide a more accurate indication of the number of EQ

Transaction Functions required in a project. The dataset demonstrated a lower relative overall improvement in terms of the average percentage difference for the other Transaction Function types, but the improvements were more consistently demonstrated across the individual projects. The smallest initial average difference at Level I for the EI Transaction Functions may reflect its least potential for relative improvement.

5.7.2 Performance of Simplified Sizing Method - Estimation Effort

The effort figures presented in Table 68, shown previously in Chapter 3, for the three variants of NESMA were recorded as the sizing methods were performed. As a result of the refinement made to the simplified sizing method as it was performed on the first five sample projects, the effort figures reported for these projects (‘A’ to ‘E’), with the exception of Level I, reflect a judgement of the time taken to complete the sizing at each level of adaptation on each project. The effort figures in Table 69 for the remaining projects (‘F’ to ‘K’) were all recorded through the maintenance of a timesheet, and therefore the figures for the simplified sizing levels represent actual rather than judged values. From the results it is clear to see that the Indicative NESMA method requires significantly less effort to complete than the Estimated NESMA method, which in turn requires considerably less effort than the Full NESMA method. These findings are consistent with those reported by van Heeringen et al. (2009). It is also clear to see that the Level I sizing presents no additional effort overhead when compared with the Indicative NESMA method, as explained in the Research Method. Table 68 shows that the Level II sizing was judged to provide, on average, approximately an 80% reduction in effort compared to the Full NESMA method with only a modest deviation observed across the sample projects. The Level III sizing still provided a significant reduction of 76.84% on average, which generally represents half of the effort incurred with the Estimated NESMA method. Table 69 shows that, where the effort values were accurately recorded, the average reduction in estimation effort was generally similar to the judged effort values for each level of sizing. There were more substantial deviations across the sample projects in this case, which may to some extent, be explained by the use of actual recorded effort. The use of the paired samples t-test demonstrated that the results were statistically significant in each case for both tables. The level of statistical significance was slightly lower for recorded effort, typically at $p<0.03$, compared to for judged effort that was typically at $p<0.01$.

For each project in Table 68, approximately one additional hour above the Level I sizing (effort was judged to the nearest half hour for each level) was incurred in completing the Level II sizing. In Table 69, with the more accurate recording of estimation effort, a variation in effort for Level II sizing was observed, ranging from 0.25 to 1.5 hours. This variation may be explained both by the effect of the size of each project and the clarity and structure of the requirements documentation. For Level III sizing the increased detail of assessment led to some variation in the effort required, with the larger projects (‘B’, ‘C’ and ‘F’) requiring approximately double the additional effort, above Level II sizing, compared to the other projects. This may indicate that as the level of input detail considered by the sizing approach increases, the size of the project has a more significant impact on the estimation effort required.

5.7.3 Performance of Simplified Sizing Method - Estimation Size Accuracy

The results in Table 70 show that the Indicative NESMA method offers insufficient accuracy to enable it to be used as a replacement for the Full NESMA method. For the complete dataset the relative average inaccuracy of 26.03% of the Indicative NESMA method would be undesirable. The Estimated NESMA method offered an attractive alternative, however, with an average inaccuracy relative to the Full NESMA method of less than 5% achievable with the previously noted effort reduction. These findings are consistent with those reported by van Heeringen et al. (2009) and suggest that there is limited value in performing the Full NESMA method.

The use of the paired samples t-test showed that each successive level of the developed simplified methods demonstrated statistically significant improvements in overall relative accuracy over the preceding levels. The results for the Level I sizing offer a slight improvement on the existing Indicative NESMA method, but they would still provide an unsatisfactory degree of accuracy overall. The application of the Level II sizing led to a statistically significant improvement in overall relative accuracy compared to the Indicative NESMA method, with the relative average inaccuracy close to that of the Estimated NESMA method for the overall dataset. For Level III sizing a statistically significant improvement, at p=0.01, in overall relative accuracy over the Estimated NESMA method was achieved. The performance at this level would leave limited scope for further improvement in terms of estimating the overall functional size.

The value of these results is further explored through assessment of the project profiles produced by each method in the following section.

5.7.4 Performance of Simplified Sizing Method - Project Profile Accuracy

The project profile produced by each method may be assessed by the relative accuracy of the functional sizes produced for each BFC type, against those produced using the Full NESMA method. The results in Table 70 show that the current Estimated NESMA method provides BFC sizes that are on average within 12% of those produced by the Full NESMA method. The effect of relative inaccuracies in different BFC types cancelling each other out, to some degree, resulted in a smaller difference in overall functional size. The Level I sizing provides an illustration of the BFC sizes produced by the assumptions of the Indicative NESMA method. The relative sizes of these BFC specific inaccuracies are, however, much higher on average at Level I than with the Estimated NESMA method. The results for Projects E and J (see Appendix A), in particular, illustrate that even when the Level I sizing produces a similar overall size to the Full NESMA method, it may be more due to chance than through providing an accurate indication in the BFC profile of the relative contributions to the total functional size.

For each of the projects there was insufficient detail on the EIF component to provide any scope for reconsidering the initial assumptions of Level I. For the ILF component, deviations from the assumption of low complexity derived from the Estimated NESMA method were only evident in a minority of ILFs. The initial relative inaccuracy of ILF component ranged from 5.58% to 17.83%. These results indicate that the extent to which the ILF component generally conforms to the low complexity assumption may vary across projects. Improvements in the ILF component were only available at Level III sizing where complexity could be applied to proportions of the ILFs. The use of the paired samples t-test demonstrated that these improvements were statistically significant at $p<0.001$. For the EQ component, any significance comparisons with Level I sizing results required the paired samples sign test to be used due to the characteristics of their distribution discussed in Section 5.7.1.

The Level II sizing shows statistically significant improvements in the relative accuracy of the EO component sizes compared to Level 1 for the overall dataset. Clear improvements were evident with the EI and EQ components for some projects, but they were not consistently found across the dataset. In general, these components were more difficult to refine using the steps in Level II sizing. Overall, while the total project functional size is more precise at this level, the degree of inaccuracy within the component types, in particular the EQ component, limits the confidence in these estimates. Indeed, the profiles at this level remained less accurate than those produced by the Estimated NESMA method.

Adopting a further degree of input detail in Level III provided statistically significant improvements over Level II in the relative accuracy of the EI, EO and EQ components. At this level the overall project profile produced was now slightly more accurate than that produced by the Estimated NESMA method, with only the EQ component exhibiting a higher relative inaccuracy. For the EI, EO and EQ components none of the mean differences between Level III and the Estimated NESMA methods were found to be statistically significant. With the exception of the EQ component, the average relative inaccuracy for each BFC type with Level III was under approximately 5%, which would provide a satisfactory level of confidence in the size estimate produced. The accuracy of the overall project profile at Level III would offer increased potential, relative to the initial assumed profile of Level I, for deriving a more accurate estimation of development effort.

5.7.5 Correlation between BFC sizes and Overall Functional Size

Table 71 shows the correlations between the individual BFC sizes for each estimation method, and the overall functional size produced by the full NESMA method. For the analysis of multiple correlation significance test each sizing method was considered to represent a separate ‘family’ of 5 results. The EIF component demonstrated a weak correlation with the overall functional size, reflecting the insignificance of this component within this dataset. For each of the other BFC types, highly significant correlations were found with the overall full NESMA size for each of the estimation methods. The application of the Benjamini-Hochberg procedure did not affect the significance of these results. The EQ component demonstrated a slightly lower correlation than the other BFC types in most of the estimation methods. It was also evident that the use of a more detailed estimation method did not necessarily lead to higher correlations. For some components the Estimated NESMA method produced a higher correlation than the full NESMA method. Likewise, the Level I sizing produced higher a correlation than both the Level II and Level III sizing in some components. These results suggest that the estimation of an individual BFC type can provide a satisfactory indication of the overall functional size. There was no ‘best’ estimation method, however, to use in order to provide this indication.

5.8 Evaluation of Research

For the standard NESMA methods, the Estimated NESMA method provides the most attractive trade-off between estimation accuracy and effort required. The inaccuracy of the Indicative NESMA method suggests that, despite the significant effort savings to be had, this approach is inadequate for the projects studied.

The estimation accuracy performance of the simplified sizing method shows a significant improvement over the existing Indicative NESMA method, while at the highest level (Level III) it provided performance similar to that of the Estimated NESMA method. The estimation effort incurred at Level III is, however, approximately 50% of the effort required by the Estimated NESMA method. Through increasing the level of sizing, a ‘smoothing’ effect on the relative inaccuracies across each of the BFCs can be observed. The benefits of implementing the most detailed level are twofold: (1) obtaining a more accurate BFC profile provides greater scope for supporting project management activities such as cost/effort estimation; (2) the effect of individual inaccuracies cancelling each other out is significantly reduced, allowing for greater confidence in the overall functional size obtained.

The results of this research phase show the potential of adapting simplified functional sizing methods for achieving performance closer to that of full scale sizing approaches while maintaining the lower levels of estimation effort required. By comparing performance with the existing NESMA methods, the ability to accommodate additional input detail has demonstrated the potential for extending the applicability of simplified sizing methods to the provision of a project profile comparative to that produced by a full function count. When size estimation must be conducted at an earlier lifecycle stage, the extent to which a size estimate may be refined in this manner is limited by the level of detail available in the requirements documentation at that stage.

At the functional specification stage the simplified sizing method is aimed at emulating the performance of the full NESMA method. In terms of refining the accuracy of the simplified sizing method, there is a trade-off between the relative improvement in accuracy and the additional estimation effort incurred in refining the estimate. The development of a more accurate functional profile using the simplified sizing method is also limited by the structure and detail of the functional specification. In order to more easily discern the overall functional profile, it is desirable for the functional specification to provide a high level overview of the system functionality. In practical terms this would generally take the form of the table of contents at the beginning of the document. For the projects used in this research, there was found to be variation across projects in terms of the level of detail included in the table of contents in the functional specification. Less detail in the table of contents results in greater effort in examining the main content of the functional specification. When the description of required functionality is consistent throughout the document, the overall pattern of functionality can be more effectively discerned in terms of the reliability and the effort required in doing so. Repetition of ‘common’ functionality across different sections of the document can adversely impact the ease at which the functional profile is derived. Inconsistently specifying similar functionality can also obscure the overall pattern of functionality e.g. ‘Maintain Investigation’ is broken into separately headed sections ‘Add Investigation’ ‘Modify Investigation’ and ‘Delete Investigation’ in the functional specification, but ‘Maintain Complaint’ is self-contained in one headed section in which distinct ‘Add’, ‘Modify’ and ‘Delete’ functionality is required.

5.8.1 Simplified Sizing Research Issues

This investigation into the simplification of functional sizing has demonstrated the potential for improving such methods, while recognising the limitations that remain. The main research issues that should be considered are as follows:

(1) The potential for adopting simplified sizing methods may be determined both by the nature of the system and of the requirements documentation available. The Estimated NESMA method will always provide the more accurate identification of individual Transaction Functions since this is a fundamental requirement of the method. Refining Data Classes and Transaction Classes will 'close in' on indicating an equivalent profile to the extent that the documentation enables a clear pattern to be identified without incurring a comparable effort. The Estimated NESMA method assumes an average complexity for each Transaction Function, which may not be suitable within some application domains. Cândido and Sanches (2004) demonstrated how altering this assumption may be suitable in such situations, but this approach is still limited by applying the same complexity across all of the Transaction Functions. The Estimated NESMA method may therefore be least effective when there is significant variation across different Transaction Functions. Altering the complexity at a finer level of detail, as carried out in this research, may therefore provide a more suitable alternative in such situations. The improvements in the relative accuracy of the functional profile obtained at Level III sizing, in comparison to Level II, suggest there is value in enabling a more selective allocation of ‘typical’ functionality.

(2) This research has examined how the utility of simplified sizing methods can be enhanced, increasing their relevance further into the development lifecycle. Simplified sizing methods have generally been examined in the context of early phase estimation, where more detailed estimation methods are not feasible. The use of simplified sizing methods when more detailed documentation is available should be examined in order to determine the potential value of such methods.

(3) The simplified sizing method developed in this research has been evaluated using detailed Functional Specifications. The potential for approximating the estimates produced by more detailed estimation methods may be dependent on the quality of the requirements documentation. Early phase bid estimation presents a challenge because of the absence of sufficient requirements detail for the Full NESMA method to be used. Consequently, it may be that there is effectively no more detailed estimation method for a simplified method to attempt to approximate. Studying the limitations of bid stage estimation, rather than simplifying the sizing process at this stage, may offer more potential for future improvement.

(4) The sizing approach developed in this research represents a simplified and considerably less detailed process of identifying Transaction Functions than that involved in the more detailed NESMA methods. The consideration of the Data Classes and the Transaction Classes facilitates amending their associated complexity in a finer grained manner than the Estimated NESMA method, but the broad level at which this is considered ensures it remains a more simplified approach in terms of the estimation effort required. In practice, the Transaction Functions associated with specific Data Functions would involve more complicated scenarios than catered for by these Transaction Classes. In this regard, the simplified sizing method corresponds to refining the 'average' transaction functionality associated with each Data Function.

5.8.2 Response to Research Questions

RQ1: To what level of requirements detail can the FPA approach be simplified while achieving an acceptable level of accuracy (i.e. within 5% of Full NESMA method)?

Level I simplified sizing provides a slight improvement in terms of overall accuracy compared to the Indicative NESMA method. It is necessary to use Level II simplified, however, in order to obtain an acceptable level of accuracy. In terms of requirements detail, the implication is that considering only the Data Functions is insufficient. Transaction Functionality should be assessed in a coarse-grained manner, where the functionality is considered broadly across all of the Data Functions.

RQ2: To what level of requirements detail can the FPA approach be simplified while providing the functional profile to an acceptable level of accuracy (i.e. within 5% of Full NESMA method)?

Estimating the full functional profile at both Level I and Level II sizing does not provide an acceptable level of accuracy. Level III sizing, however, both provides an acceptable level of accuracy and exceeds the Estimated NESMA method in this regard. Detailed consideration of both Data Functionality and Transaction Functionality is required to obtain a reliable indication of the functional profile. Assessing this functionality in a fine-grained manner, where the proportional requirement of specific functionality is estimated, requires the same level of detail in the requirements documentation as the more detailed estimation methods.

RQ3: What level of utility can a simplified sizing method provide?

The developed simplified sizing method has demonstrated that it can equal, and in some cases exceed, the accuracy of the Estimated NESMA method at the detailed functional specification stage. It can effectively provide the complete functional profile with an acceptable degree of accuracy. The provision of an equivalent level of output detail as the full NESMA method extends the utility of simplified sizing methods beyond that of a rough indicator of size. For example, the use of simplified sizing methods for subsequent project planning activities can be considered. The increased flexibility offered by the method enables sizing to be adapted to different project profiles, without requiring similarity to 'typical' projects. The increased transparency facilitates comparison with expert-based estimates by providing a clearer indication of the estimated functionality.

The developed method provides a more attractive alternative at later stages in the project lifecycle than the more detailed estimated methods due to the significantly lower effort overhead. Simplified software sizing can therefore facilitate an expansion in the relevance of software sizing to the development process.

RQ4: Where is the optimal trade-off between estimation accuracy and estimation effort?

The simplified sizing method, even at Level III, demonstrated a statistically significant reduction in estimation effort compared to the Estimated NESMA method. Comparisons of the results between different methods revealed that each subsequent level of simplified sizing provided a statistically significant improvement in accuracy over the preceding level. For overall size accuracy the Level III sizing provides a statistically significant improvement over the Estimated NESMA method. For the individual BFC types, it is only the ILF component for which Level III sizing offers the same degree of improvement over the Estimated NESMA method. The preceding two levels are outperformed by the Estimated NESMA method. These results suggest that the simplified sizing method, at the most detailed level used in this research, provides the optimal trade-off in comparison with each of the NESMA methods.

Identifying the optimal trade-off, in practice, is dependent upon what is trying to be achieved when developing a size estimate. Level II sizing would provide an acceptable balance if the estimate was only required to provide the overall functional size. Provision of an accurate functional profile would, however, necessitate Level III sizing. The estimation effort incurred at Level III sizing would suggest that there could be value in using this level of detail regardless of the expectations of the estimate. The results are, however, limited to the extent of the largest projects in the dataset used. The effect on estimation effort of substantially larger projects is uncertain, so the recommendations are limited to the range of sizes evident in the dataset.

In addition, the degree to which the level of detail used by the simplified sizing method can be increased effectively is dependent upon the quality of the requirements documentation. In this respect the flexibility of the simplified sizing method to accommodate different levels of detail enables the optimal trade-off to be similarly flexible.

5.9 Conclusions

The limited use of formal software sizing within industry has coincided with research on developing new estimation methods. Supporting, rather than replacing, the existing estimation methods utilised in organisations may represent a more realistic goal for model-based sizing estimation. The use of simplified functional sizing methods has predominantly been focused on either addressing the unavailability of the necessary requirements detail early in the lifecycle, or on providing a low effort indication of the system size. Deriving an overall size from a subset of the BFCs requires significant groundwork to establish reliable correlations for a specific organisation and provides limited transparency and flexibility. As such, the potential benefits of utilising such methods are insufficient to offset the need for a more rigorous sizing method to be completed where possible. The commitment to undertake such a process may not be justifiable to many software development organisations. Simplified methods may therefore be more palatable provided that they can demonstrate sufficient benefits.

The aim of this study was to assess the feasibility of adapting existing simplified functional sizing methods to provide value in a commercial context, in terms of size estimation accuracy, estimation effort and degree of utility. The results obtained have highlighted the potential benefits that adapting existing simplified NESMA methods can provide, while maintaining suitability to be incorporated into existing development practices. These benefits include: (i) significantly reduced effort with minimal degradation in sizing results when compared with the Estimated NESMA method, enabling completion within a limited time frame without disrupting other project activities; (ii) flexibility to accommodate the level of information available in requirements documentation through more detailed refinements to the Transaction Classes, ensuring no additional demands are placed on the project documentation produced; and (iii) providing detailed BFC profiling, which is useful in subsequent project management activities such as cost/effort estimation.

The full functional sizing approach provides the most benefit to the software development process, but incurs the most significant cost in achieving such benefits. Simplifying the software sizing process reduces the estimation effort required, but also the benefits that may be obtained. The inevitable compromises resulting from using simplified functional sizing may, however, be preferable to the downside of forgoing any form of software sizing altogether. The question may therefore be one of finding value from an acceptable trade-off between benefits and cost. This chapter has shown how the developed simplified sizing method aims to provide this value to the development process by simplifying the sizing process, reducing the estimation effort, while maintaining the provision of output detail comparable in quality to that obtained with complete functional sizing methods.

6. Thesis Evaluations and Conclusions

The research in this thesis has addressed the issue of how the focus of software estimation research has, in general, not converged with developments in estimation practice within the software industry. The context for researching the use of software size estimation in this thesis is summarised in the next section. The research consisted of three phases investigating how software sizing can provide value to the development process in practice. The conclusions from each of the three phases are summarised in the subsequent sections. The case study oriented nature of this research incorporated consideration of the actual process of developing software size estimates on real world project documentation. This provided valuable insight into the practicalities of using functional sizing in a commercial environment. The opportunity to use functional size estimation in an active project bid context is firstly discussed. This is followed by a discussion of the lessons learned from the practical application of functional sizing on commercial project documentation. The research contributions made by this thesis are then presented, followed by consideration of the generalisability and limitations of this research. Future research issues arising from this thesis are then discussed before finishing with the overall conclusions.

6.1 Rationale for investigating Software Size Estimation

Industry surveys on the use of estimation methods present a pattern that suggests that the value of functional software sizing has not been sufficiently demonstrated to encourage the uptake of such an approach within industry. Yang et al. (2008) reported that the main reasons for not using model-based cost estimation methods, such as FPA metrics, were the insufficient benefits to justify the additional effort, and therefore cost, required. The number of estimates made in projects throughout their development lifecycle was reported to range between one and six, in a survey of the Norwegian software industry by Moløkken-Østvold et al. (2004). The value, which may be associated with an estimate, may vary throughout the project lifecycle. This research thesis provided an investigation into the value of software sizing, in terms of benefits and costs, incorporating the perspectives of a local software development organisation, Equiniti-ICS. The investigation involved assessing the value of the size estimate at two distinct stages of the project lifecycle.

The case for the adoption of formal model-based estimation methods within industry would have been more convincing if superior performance, over the expert judgement-based approaches used in industry, had been demonstrated. Evidence in favour of either approach had proven to be inconclusive thus far (Jorgensen, 2004). This was in spite of considerable focus within the research community towards improving the estimation accuracy. Research may therefore be approaching the limits of what is achievable for the estimation accuracy of model-based approaches. This research thesis provided an investigation into the use of more detailed functional size data, as an alternative to focusing on overall size accuracy, for achieving greater benefits. This investigation focused on providing greater insight into bid stage estimates, which are generally regarded as being the most crucial to the success of a project.

More detailed specifications of system functionality are available in later stages in the project lifecycle, enabling more accurate size estimates to be developed. Such estimates are generally considered to provide insufficient value at this stage to justify the effort to develop them. Fully realising the benefits from using software sizing, however, may require detailed and accurate estimates to be developed using completed specifications of the functionality. This research thesis investigated how simplified sizing methods can be adapted to provide sufficiently accurate and detailed estimates without incurring the prohibitive estimation effort associated with full methods.

6.2 Research Phases

This research was conducted in three phases, each of which addressed its own Research Aims and Objectives as specified in Chapter 1. The conclusions reached for each of these phases are summarised in the following sections. The results from each phase provided greater understanding of the potential role of software sizing, while also identifying issues to be addressed in the subsequent phase. The research issues arising from each phase are also summarised to highlight how the consecutive phases formed a cohesive investigation into adapting software sizing to provide value for commercial project development.

The research in each phase utilised the NESMA approach to functional sizing. Research Phase 1 used the three levels of the NESMA method on commercial project documentation to investigate the value of size estimation. The sizing data from that phase was subsequently decomposed into greater detail in Research Phase 2, in order to analyse the differences between the bid stage and functional specification stage estimates. Research Phase 3 adapted the NESMA approach to more efficiently develop detailed size estimates.

6.2.1 Research Phase 1 (Value of Software Sizing) Conclusions

This phase was focused on addressing Research Aim RA1:

To assess the effectiveness of software size estimation in terms of business value at differing stages of the software lifecycle.

In a commercial development environment, the effectiveness of software size estimation had to consider both the estimation accuracy and the estimation effort required in order to justify the investment in time and personnel in completing the estimate. Software size estimates were developed in this phase for eleven commercial projects at the bid and the functional specification stages. The conclusions and observations reached for each specific Research Objective in this phase were:

(RO1) - *To evaluate the impact on the relative sizing of projects of a sizing process where (a) sizing methods incorporating different levels of sophistication are used and (b) sizing is performed on documentation from different stages in the software lifecycle.*

- Overall performance of the NESMA estimation methods was generally similar to the existing expert judgement-based method used by Equiniti-ICS.
- Considering relative sizes of projects did not yield any demonstrable advantage over using the specific functional size estimate.
- Functional Size estimates developed at the later functional specification stage were generally larger than the equivalent initial bid stage estimates, and provided a stronger relationship with actual development effort.
- In terms of estimation accuracy at the functional specification stage, the Indicative NESMA was not considered sufficiently reliable. The accuracy of the Estimated NESMA method was sufficiently close to that of the Full NESMA method to render the additional effort required to perform the Full NESMA method unnecessary.

(RO2) - *To assess the value of software sizing to assist the associated project management activities within the local software development organisation through consideration of tangible/intangible benefits and costs.*

- The primary concern within Equiniti-ICS was with obtaining a level of assurance in the initial bid stage estimates.
- Management would assign value to having an additional perspective in the estimation process, at the bid stage, but would not replace their existing approach.
- Only individual ‘sprint’ estimates were currently undertaken subsequent to completion of the functional specification rather than a complete estimate of the overall system. Management were keen to see how the functional size changed between the two stages,

suggesting that such sizing data was of merit at the latter stage. The benefits were not sufficiently apparent, however, to accommodate the associated cost of obtaining the data.

(RO3) - *To assess the potential for incorporating software sizing within the existing estimation practices of the local software development organisation.*

- At the bid stage the Estimated NESMA method was considered to require an acceptable amount of effort to perform. Insufficient requirements detail was available at the bid stage to enable the use of the full NESMA method.
- The use of the Estimated or the Full NESMA methods at the subsequent functional specification stage was not seen as justifiable due to the estimation effort required. The Indicative NESMA method was considered acceptable at the functional specification stage in terms of estimation effort, but would not provide sufficient size data to be of use to the organisation.

In terms of addressing RA1 overall, software size estimation was found to provide a similar performance to the existing expert judgement-based approach at the bid stage at an acceptable level of effort. The more accurate size estimates possible from the more detailed functional specification stage were offset by the more limited justification that could be made for completing further size estimates at that stage. The effectiveness of software size estimation was therefore found to be better at the bid stage due to the greater business value that could be derived at that stage. This posed the question of how the business value for using software size estimation could be enhanced at either lifecycle stage.

The resulting issues to be addressed in the next research phase were:

- Increase the understanding of the differences in functional size between the two distinct lifecycle stages.
- Contrast the size estimation data with the existing effort/cost estimation data obtained by Equiniti-ICS.
- Identify whether additional insight is provided by size estimation data at the later functional specification stage.
- Determine whether the respective sizing data at each stage can provide a basis for assessing the reliability of bid stage estimates on subsequent projects.
- Assess the suitability of existing requirements documentation for facilitating functional size estimation.

6.2.2 Research Phase 2 (Bid Stage Estimation) conclusions

This phase was focused on addressing Research Aim RA2:

To investigate the use of detailed software size measures to address the limitations of project documentation at the initial 'bid' stage of the software lifecycle.

The reliance on customer developed bid stage documentation limits the thoroughness of the description of required functionality. The interest is therefore on attempting to discern what required functionality the documentation for the eleven commercial projects does not include. The conclusions and observations reached for each specific Research Objective in this phase were:

(RO4) - *To assess the potential for using a detailed profile of the required software functionality components identified using both bid stage project documentation and the functional specification in order to validate bid stage estimation.*

- The overall functional size estimated was greater at the Functional Specification stage, but the relative magnitude of this increase from the bid stage varied significantly across projects.
- Some specific types of functionality were found to be consistently less commonly identified at the initial bid stage – typically generic functionality for processing associated data for main entities e.g. 'Add', 'Link' functionality.
- Using the more detailed consideration of types of Transaction Functionality can therefore enable patterns to be determined and used to facilitate identification of some underspecified types of functionality at the bid stage.

(RO5) - *To evaluate the accuracy of different levels of functional size data at the bid stage for indicating both overall functional size and actual development effort.*

- Data Functionality was strongly correlated with overall functional size, although the accuracy at the initial bid stage was slightly less than at the functional specification stage. Transaction Functionality was strongly correlated with overall functional size, with similar results found at both the bid stage and the functional specification stage.
- Only Data Functionality demonstrated a statistically significant correlation with actual development effort at the bid stage, with the more detailed RET count producing a stronger correlation. The RET count could be considered to be providing a more effective assertion of estimation uncertainty by providing an indication of 'associated' functionality.

- Using more detailed size measures than the standard BFC types generally did not provide superior results in reflecting overall functional size or actual development effort.

(RO6) - *To examine the degree to which the size of required functionality not specified in bid stage project documentation can be assessed at the bid stage.*

- Explaining the subsequent overall functional size at the functional specification stage was generally found to be more effective than attempting to explain the actual growth in functional size.
- Data Functionality in particular at the bid stage was found to be strongly correlated with the subsequent overall functional size at the later stage. Consideration of the complete BFC profile sizes further improved this level of correlation, but using even more detailed sizing measures did not demonstrate any relative improvements.
- Some ratios between Data Functionality types and Transaction Functionality types did demonstrate moderately strong correlations with the growth in functional size between the two stages.

In terms of addressing RA2 overall, the use of decomposed software size measures at the bid stage highlighted the limitations of early documentation produced by the customer. Nevertheless, the application of a more detailed software sizing measure (RET) for Data Functionality proved useful for revealing associated Transaction Functionality not explicitly specified in the documentation, and consequently for explaining the actual effort for a project. The recognition from company management of the potential value of developing ratios to determine future increases in functional size later on in a project suggested that (1) a more detailed composition of the functional profile of a project provided additional insight, and (2) developing complimentary size estimates at the functional specification stage could enhance the value of software sizing at the bid stage.

- Company management were keen on establishing specific ratios that could be used at the bid stage to determine the anticipated functional size at the functional specification stage.
- Issues raised by the management concerned the expertise required to develop the necessary sizing data and whether employees could readily be trained to fulfil this activity.
- The effort required to develop NESMA estimates at the functional specification stage represented a significant barrier – relatively simple measures were found to provide a crude assessment although without any insight into how the size has changed.

The resulting issues to be addressed in the next research phase were:

- For the full value of software sizing to be obtained it is necessary to obtain an accurate measure of the size of the system.
- The measurement should therefore be made on the final functionality implemented in the system.
- The benefits of this 'final' measurement may vary according to the lifecycle stage at which it is made:
    - At the completion of the project the benefits are limited to reviewing the completed project and informing future projects
    - Obtaining the measurement at an earlier stage can also benefit the subsequent progress of the current project by providing an indication of how accurate the initial estimates were and aiding the management of the remaining lifecycle stages
    - The functional specification stage provides an attractive option by having a relatively stable completed view of the required functionality prior to the development stage of the lifecycle. The additional benefits accrued at that stage can be considered to more than offset any relative inaccuracy of estimating the functional size at this stage compared to measuring the final completed system.
- The most significant barrier to making any 'final' measurement of the system size is the effort required.
- The objective is therefore to establish an acceptable trade-off between estimation accuracy and estimation effort at the functional specification stage.
- In order to maintain the same overall benefits of the estimation at this stage the sizing output detail should be equivalent to that produced by using the full NESMA method.

6.2.3 Research Phase 3 (Simplified Software Sizing) conclusions

This phase was focused on addressing Research Aim RA3:

To investigate the effectiveness of enhancing simplified software sizing approaches in contrast to full software size estimation.

Simplified sizing methods have thus far been concerned with approximating the overall functional size for a system. In order to provide greater benefit, the objective is to maintain sufficient estimation accuracy at a greater level of granularity than just the overall size. The dataset of eleven commercial projects was used to develop and evaluate the enhanced simplified sizing method in this phase. The conclusions and observations reached for each specific Research Objective in this phase were:

(RO7) - *To assess the extent to which software sizing can be simplified while maintaining an acceptable level of accuracy in the overall size estimate.*

- The simplest level of simplified software sizing requiring only consideration of Data Functionality, and ‘assuming’ Transaction Functionality did not provide an acceptable level of accuracy in estimating overall size.
- The intermediate level of simplified software sizing, incorporating a coarse-grained consideration of Transaction Functionality, was therefore representative of the extent to which the process can be simplified without compromising overall size estimation accuracy.

(RO8) - *To assess the extent to which software sizing can be simplified while maintaining an acceptable level of accuracy in the estimated functional profile.*

- The highest level of simplified software sizing, incorporating a fine-grained consideration of Transaction Functionality, was necessary to obtain an acceptable level of accuracy in the estimated functional profile.
- The individual identification of Transaction Functions, as required by the Estimated NESMA method, was not found to be necessary to obtain a satisfactory estimated functional profile.

(RO9) - *To assess the utility of the simplified software sizing method in the context of the local software development organisation*.

- By effectively approximating the performance of the Estimated NESMA method, the developed simplified sizing method provides at a minimum the same level of utility.
- The simplified sizing method can be considered for use at the later functional specification stage due to the lower estimation effort required.
- The flexibility of incorporating different levels of requirements detail enables the use of the approach to be adapted according to estimation needs.
- The expertise required to use this simplified approach to functional sizing can be considered broadly similar to the full FPA approach as the same BFC types must be understood and recognised from the requirements documentation.
- The process of utilising the simplified approach to functional sizing is influenced by the structure of the requirements documentation analysed:
    - Consistency in the structure and level of detail provided enables the pattern of specific types of functionality to be more easily estimated.

  - The provision of a detailed table of contents can be readily supported through electronic requirements documentation, providing an appropriate basis for an initial overview of the required functionality.

(RO10) - *To evaluate the relationship between estimation accuracy and estimation effort using different levels of software sizing.*

- Three different levels of size estimation were completed using this simplified method and evaluated against the different NESMA methods:
  - Statistically significant improvements were achieved at each subsequent sizing level.
  - The estimation accuracy at the most detailed sizing level was comparable with that of the Estimated NESMA method.
  - Even at the most detailed sizing level, the reduction in estimation effort required relative to the Estimated NESMA method was statistically significant.
- The most detailed level of the simplified sizing method demonstrated the most effective trade-off between estimation accuracy and estimation effort.

In terms of addressing RA3 overall, the scope for enhancing simplified software sizing approaches was effectively demonstrated against the use of the different NESMA methods at the functional specification stage. The Full NESMA method was considered to be prohibitive at the functional specification stage due to the estimation effort required. By providing comparable performance to the Estimated NESMA method at significantly reduced estimation effort, the developed simplified sizing method could be considered to provide a successful alternative option to the Full NESMA method. This Research Phase addressed the main barrier to the use of software size estimation at this later stage in the project lifecycle. It also raised the possibility of commercial organisations adopting a similar estimation approach at this stage that may not necessarily involve the specific FPA concepts.

6.3 Practical Application of Functional Sizing

The utility of functional size estimation in a commercial context was demonstrated as the result of a request made from Equiniti-ICS during the completion of the research. The organisation was in the process of formulating a bid for a new project based on a tender document outlining the requirements for the proposed software system. A request was received, from the Technical Director within Equiniti-ICS, to provide an assessment of the size of this new project. Internal discussions concerning the feasibility of completing the project within the required timescale had led to the interest in obtaining an independent review of the size of the project. The main parameters of this request were as follows:

- A tender document that was 291 pages in length was provided as the basis for developing a size estimate.
- The project management staff were 'extremely interested' in learning how the size of the new project compared to the sample projects that had already been subjected to functional size estimation. No indication was provided from the project managements of their view of the size of the project. There was therefore no prior expectation of the outcome of the size estimation process.
- The deadline provided for this request was that the size estimate must be provided by the close of business two days from the principle researcher receiving notification of the request. The amount of time available for completing the work was therefore approximately two and a half working days.

The imposition of a constraint on the time available in which to develop the estimate represented a significant difference in how the sizing process could be approached. This exercise represented an example of the difficulties of applying the use of size estimation in a real world commercial context. In order to complete this exercise, it was necessary to consider the feasibility of what could be achieved within the available timeframe. This involved a preliminary examination of the tender document in comparison with the project documentation that had been used for previous projects. This would be considered in combination with the estimation effort figures recorded for the use of each size estimation method on the previous projects. The results of this preliminary analysis are summarised in Table 72. This comparison was based on the projects on which size estimation had been performed at that point on time. The largest figures for project documentation and estimation effort correspond to those projects, and not the subsequent projects included in the main research phases.

Table 72 - Preliminary Analysis Results

Tender Document Characteristics	Comparison with previous Projects
Around 200 of the 291 total pages are concerned with discussion of functional requirements.	Largest bid stage documentation for previous projects was 90 pages, with the next largest being 55 pages. Largest functional specification stage documentation was 210 pages, with the next largest being 160 pages.
No data model is provided for	The need for the estimator to develop a data model from the

the project.	tender document is consistent with other projects.
Level of requirements detail is greater than that typically found within tender documentation.	In terms of identifying transaction functionality, the tender document was more equivalent to functional specification document. In terms of assessing complexity of functions the tender document was more equivalent to bid stage documentation.

The page size indicated in Table 72 for the tender document under consideration suggested that the estimation effort required may be expected to exceed that found for previous projects. The most detailed estimation method used at the bid stage on the previous projects was the Estimated NESMA method. The largest estimation effort incurred using this method at the bid stage was 16 hours, although it should be noted that this did not coincide with the largest documentation page size. In contrast, the largest estimation effort incurred using the Estimated NESMA method at the functional specification stage was also 16 hours, which in this case this did coincide with the largest documentation page size. The page size alone cannot therefore be considered to be a reliable indicator of the required estimation effort for a project. An important distinction between the two documentation stages was that a data model was not available at the initial bid stage. As Table 72 indicates, this was again the case for the tender document under consideration. As the Indicative NESMA method is focused on the estimation of Data Functionality, the estimation effort recorded for the use of that method was of most interest here. The recorded estimation effort was consistently greater at the initial bid stage, with a maximum of 7 hours, reflecting the need for the estimator to essentially develop a data model from the textual description of the required functionality. As with the Estimated NESMA method, the largest estimation effort recorded for the Indicative NESMA method at the bid stage did not coincide with the largest documentation page size. The level of requirements detail in the tender documentation provides an indication of both the feasibility of using a particular estimation method and the estimation effort that may be required. As Table 72 shows, the identification of transaction functionality, mainly associated with the Estimated NESMA method, may incur estimation effort similar to that found with previous functional specification stage estimates. The assessment of function complexity, associated with the Full NESMA method, would not be possible with the tender document.

Overall consideration of all of the characteristics of the tender document would suggest that the need to develop the data model would have the most significant impact on the estimation effort compared to the previous projects. The much larger document size and greater level of requirements detail, compared with bid documentation for previous projects, would be expected to increase the estimation effort required for the use of the Indicative NESMA method. The subsequent estimation of the transaction functionality required by Estimated NESMA method would present a risk, within the constraints of this estimation request, as it would be expected to expand the overall estimation effort beyond 16 hours. The possibility of commencing estimation of the Transaction Functionality, but not being able to complete the estimation by the provided deadline, necessitated the consideration of an alternative approach. The simplified sizing method investigated in Chapter 5 provides the flexibility to adapt the level of requirement detail considered by the method to the specific circumstances of an estimate. This method enables an initial estimate to be refined to consider greater requirements detail, providing a 'complete' size estimate at the completion of each refinement. This reduces the potential for only having developed an incomplete size estimate at the time of reaching the deadline for the exercise. At the time of receiving the request to undertake this estimation task the simplified sizing method had not been completely refined. The specific sizing levels for the method described in Chapter 5 had not been designed, and as such there were no specific estimation efforts established for the use of this method on the previous projects. The relative accuracy of the simplified sizing method, relative to the various NESMA methods, had also not been formally established at this stage. Informal use of the simplified sizing method indicated that it was, relative to the full NESMA method, more accurate than the Indicative NESMA method and similar to the Estimated NESMA method. The accuracy relative to the Estimated NESMA method would be dependent upon the level of refinement from the initial simplified size estimate.

Both the Estimated NESMA method and the developed simplified sizing method incorporate the specific estimation activities performed when using the Indicative NESMA method. This enabled the choice of which approach to adopt to be made upon completion of the Indicative NESMA estimate. This choice was made based upon the time taken to develop the Indicative NESMA estimate and the greater understanding of the proposed system established at that point. Equiniti-ICS management had expressed interest in obtaining the relative size of the proposed project in comparison with the previous projects. The emphasis was therefore on the overall size of the project rather than a more detailed consideration of size. Upon completing the development of the Indicative NESMA estimate it was decided that the use of the Estimated NESMA method was unlikely to be completed within the remaining time. The simplified sizing method was therefore used to refine the initial estimate derived from the Indicative NESMA method. In the context of the overall research in this thesis, the estimation request described in this section was independent of the main research phases. There was therefore no requirement to conform the use of the simplified sizing method in this instance to how it would subsequently be investigated.

6.3.1 Results of estimation exercise

Estimates using both the Indicative NESMA method and the simplified sizing method were successfully developed within the available timeframe. The results provided by these methods were considered as a whole rather than considering any one estimate to be the most accurate. The results indicated that the proposed project would exceed the previous projects in size, with the relative size estimated to be 70% greater than the largest project that had previously been estimated. This relative size was reported to the development organisation in accordance with their request. As this estimate was outside the range of the previous projects, it was advised to Equiniti-ICS management that they should not to extrapolate an effort estimate from the functional size.

Feedback was then provided from Equiniti-ICS on how this information was utilised. Two experienced personnel within the organisation had independently assessed the percentage of reuse of existing core product code that could cover the stated system requirements. The intuition of both staff was that the project was 'big', but they desired an independent review of the size. The effort estimate developed by the organisation incorporated an uplift of the actual effort required for the previous largest project that had been sized using functional sizing as part of this research. This value was then multiplied by the percentage of the project that they anticipated would not be reused from existing core product code. A bid for the project was consequently submitted by Equiniti-ICS.

An update on the outcome of the bidding process was requested at a later date as part of this research. The organisation reported that their bid for the project had been successful. Much of the development of the project was, however, subsequently outsourced to a third party. The reason stated for this decision was technology variances from the normal skillset of the organisation. In these circumstances no figures for the actual effort required to develop the project were available to the organisation. The relative accuracy of the estimates developed for the project could not therefore be assessed.

6.3.2 Practical Recommendations

The main recommendations from completing this exercise are:

- An initial feasibility analysis for the size estimate should be conducted to enable a realistic view to be developed regarding how detailed the estimation method used should be.

- Detailed data on estimation effort and documentation characteristics should be gathered, in addition to the size estimate, for each project to aid the feasibility analysis for future projects.
- The use of software sizing on the functional specification stage of projects provides insight into size estimation using more detailed requirements documentation. This can inform bid estimation involving atypical levels of requirements detail.
- A flexible size estimation method that can refine simpler initial estimates may provide insurance against any potential failure to develop an estimate for the complete system functionality.

6.4 Software Size Estimation in Practice

Section 6.2 focused on drawing conclusions from analysis of the results obtained from each Research Phase. Section 6.3 provided discussion of a practical application of software size estimation on an actual real world bid submission. This section presents a summary of the insights gained during this research on the general suitability of functional sizing based approaches to software estimation in practice.

6.4.1 Practical Considerations for Bid Stage Estimation

The bid stage documentation has insufficient detail to enable the full functional sizing approach to be adopted. Simplified functional sizing methods are therefore a necessity at this stage, with the Estimated NESMA method representing the most detailed method feasible. In the bid stage documentation there was more variation in terms of the level of detail provided for each of the projects. This may be attributed, in part, to the fact that such ‘tender’ documentation was generally produced by the customer. Whereas the approach to specifying the requirements adopted by Equiniti-ICS was generally consistent, the different customers adopted an approach suited to their requirements. A fundamental difference compared with the later Functional Specification, which is primarily concerned with describing the required functionality, is that the ‘tender’ documentation addresses additional wider concerns, such as the bidding process, non-functional requirements, and integration with existing practices etc. The description of required functionality, within the ‘tender’ documentation, is itself problematical for functional sizing. The focus is often on describing the business processes involved rather than specifically focusing on the user functionality required. Repetition of required functionality across different segments of the documentation is present at times. This issue is further complicated by more significant variation in the terminology present in customer documentation. In particular, discerning whether two similar terms are in fact referring to the same system entity necessitates the judgement of the estimator, as enquiring about each individual occurrence of such situations would generally be impractical in the bidding process. Functional size estimates are consequently more affected by subjectivity, arising from the greater uncertainty, at the initial bid stage of a project.

As identified in Chapter 4 on bid estimation, the expert judgement-based estimation approach employed within Equiniti-ICS, does not consider the data functionality requirements to the same level of detail as the functional sizing approach. In contrast the functional sizing approach does not account for significant complexity in the business processes underlying the user functionality. The expert-based approach relies more on the expertise of the 'expert' as it considers any such significant complexity implicitly in estimating the required effort. In effect, there is a degree of compensation offered by each approach wherein the functionality that may be underestimated by one approach may be more appropriately estimated by the other approach. There is therefore value in adopting both approaches to estimating software development projects to offset relative weaknesses in either case.

6.4.2 Practical Considerations for Functional Specification Stage Estimation

For the more detailed documentation available in the form of the functional specification, the user functionality required is specified in a less ambiguous and more focused manner than in the 'tender' documentation. For some projects the scope of the data model extended beyond that of the current project, which slightly complicated the development of this aspect of the size estimate. This was not a significant issue, however, and was not recorded as estimation effort as this was an atypical aspect of a minority of projects. As the specifications were produced within the development company there was a higher level of consistency evident at this stage of the project. There was still a degree of inconsistency, at times, in the level of detail provided in different sections for the description of functionality within the same project. For some sections of a specification there was consequently not sufficient detail to provide a full function count for a specific function. This also necessitated the use of some subjective judgement from the estimator in identifying the exact user functionality present. Some 'common' functionality is also repeated in different sections of the specification, and in some cases it is not readily apparent that this is the case. This approach to documenting the functionality provides a greater degree of clarity for developing the system, but complicates the process for the estimator as it is necessary to recognise 'common' functionality so that it is not counted multiple times in the size estimate.

6.4.3 Practical Considerations for Simplified Size Estimation

The use of a simplified sizing method on the detailed functional specification has been evaluated in this study. The quality of the documentation available at this stage directly impacts on the effort required to develop the simplified size estimate. The presence of a data model at this stage of a project enables the assessment of the Data Functionality to be more straightforward. This compensates for the greater detail available at this stage, compared to the bid stage, by limiting potential increases in the estimation effort required to assess the Data Functionality. Consistency in the structure and level of detail provided in the requirements documentation will enable the general profile of functionality to be identified more easily, and provide for greater reliability in conforming to expected ‘typical’ profiles. In particular, specific functionality should be decomposed to equivalent levels of functionality e.g. ‘Maintain’ functionality separated into individually headed sections for ‘Add’, ‘Modify’ and ‘Delete’ functionality where applicable. Provision of a detailed overview of the structure of the requirements documentation e.g. table of contents, may reduce the time needed to complete the simplified software sizing. Electronic copies of the documentation e.g. Word document, enable the automatic generation of this overview at the desired level of detail. The utility of the table of contents in supporting the simplified size estimate would be affected by the consistency in the section heading structure used throughout the functional specification. Greater consistency in this regard would facilitate the generation of a consistently detailed overview in the table of contents. This would therefore not introduce any significant overhead on the existing development of requirements documentation.

Assessment of the Transaction Functionality for the Estimated NESMA method at the functional specification stage demonstrated an increase in the estimation effort required compared to the initial bid stage. The required functionality was described in a more detailed fashion, which resulted in a lower level assessment of the functionality. Even though the Estimated NESMA method does not require assessment of complexity, it was still necessary to consider the specific data elements involved in the Transaction Functions in order to distinguish separate functions from each other. This meant that the identification of the functionality was more complicated compared to the higher level assessment at the bid stage, but there would also be less scope for subjective judgements by the estimator due to the more detailed and precise description of the functionality. This greater detail enables the assessment guidelines to be utilised more extensively. In terms of required estimator expertise, the initial bid stage would require more domain experience to compensate for the less detailed documentation. At the functional specification stage, the estimator would be less reliant on domain experience and more on their ability to apply the method guidelines correctly.

For the use of the developed simplified sizing method, the objective was to approximate the sizing results obtained from the more detailed NESMA functional sizing methods applicable at the project stage under consideration, without incurring the same degree of estimation effort. The results from Research Phase 3 demonstrated the effectiveness of such an approach when detailed requirements documentation is available. The simplified method developed in this research allows for complexity to be considered more accurately than the Estimated NESMA method, where the complexity of the required functionality is assumed. At the bid stage, however, the limited detail available in 'tender' documentation generally offers insubstantial insight into the complexity of the functionality. This consequently limits the potential for refining this aspect of estimates developed using the simplified sizing method. The limited requirements detail also provides a relatively broad indication of the overall profile of functionality for a system. There is therefore also less scope for refining the initial assumed functionality produced by the developed sizing method. Such refinements would be more influenced by the domain experience of the estimator in identifying 'expected' functionality not accurately outlined in the documentation.

6.4.4 Overall Practical Recommendations for Size Estimation

Considering the importance of the bid estimate, and the relative estimation effort required, it is recommended to use the Estimated NESMA method when the size of the project documentation is not excessive in relation to the time constraints of developing the estimate. The developed simplified sizing method will suit projects where time constraints are more restrictive, since it will reduce the estimation effort required. For projects with overly brief requirements documentation at the bid stage the developed simplified sizing method may also be preferred in order to extrapolate the required functionality from the insufficiently detailed documentation. At the functional specification stage, where estimation effort is of primary concern, the use of the developed simplified sizing method is recommended. The level of simplification used could be tailored to the circumstances of each individual project. For example, where the functional specification is particularly long or where the time constraints are more significant, a less refined size estimate could be developed.

This research has analysed the potential of deriving 'expected' functionality from the detailed comparison of functionality specified at the two distinct stages of the project lifecycle. The findings indicate that, for the project data analysed, there is limited scope for applying a general pattern of 'expected' functionality to new projects. The benefits from such a comparison were found to be in identifying possible indicators of future growth in functional size. This increase in functional size may be predicted at a broad level, rather than in specific areas of functionality.

In terms of the feasibility of automating the estimation process, documentation from both stages presents challenges. The significant variation in the bid stage documentation, discussed in Section 6.3.1, would likely limit the suitability of any automated approach to a case by case basis. The functional specification documentation would provide a more consistent basis for automation, but the subjectivity issues discussed for human estimators in Section 6.3.2 would likely be even more significant in this case. The generalisability of any automated approach would be limited by the need to accommodate the specific characteristics of the project documentation.

6.5 Research Contributions

Each Research Phase provided individual contributions to the area of software estimation research. The work represents an overall contribution towards addressing the gap between industry practice and academic research in estimation.

6.5.1 Research Phase 1

- The benefits of software sizing from the perspective of Equiniti-ICS were concerned with the additional insight that could be provided, rather than as a replacement for their existing estimation practices. This investigation therefore supported the broadly documented view that formal estimation methods, such as functional sizing, are suitable only as a complementary approach to, rather than as a superior alternative to, more informal expert judgement-based methods.
- Consideration of the value of software sizing indicated that standard functional sizing methods are not sufficiently focused on providing such value. This highlighted the need to identify more clearly where value can be provided to the software development process and how the sizing approach can be adapted to provide such value. The acceptability of the costs of software sizing differed markedly between the bid stage and the functional specification stage of the project lifecycle. Software sizing can play a complementary role to existing expert judgement-based approaches only if the additional effort it entails can be carefully controlled.

6.5.2 Research Phase 2

- An insight was provided into the limitations of estimating project size from customer developed bid documentation. In order to address the insufficient requirements detail available at the bid stage, it is necessary to evaluate earlier size estimates against those from a complete specification of the functionality. An understanding of the nature of the differences between the two stages can thus be developed.

- The more accurate sizing data gathered at the functional specification stage provide benefits in the following ways:
    - Indication of evolving functional size from the initial estimate, both in terms of the overall size and in the sizes of specific types of functionality.
    - Assisting in the assessment of the uncertainty (risk) for bid stage estimates in subsequent projects by:
        - Identification of types of functionality that typically cause the functional size to increase.
        - Facilitating the development of appropriate ratios for functional size metrics between the two stages.

6.5.3 Research Phase 3

- The feasibility of simplifying a functional sizing method while maintaining an acceptable level of detail and accuracy was demonstrated:
    - The relative estimation accuracy of the overall functional size and functional component sizes achieved by the more detailed Estimated NESMA method can be achieved with a significant reduction in the estimation effort required. This addresses the constraints of estimating at the functional specification stage highlighted in Research Phase 1.
    - A more detailed functional profile than the conventional BFC types can be obtained, enabling sizes of more specific types of functionality to be estimated. This demonstrates the potential for using a simplified sizing approach to obtain the benefits of more detailed functional sizing demonstrated in Research Phase 2.
- The developed simplified method for functional sizing provides flexibility to accommodate varying levels of requirements detail in developing the estimate:
    - The method focuses on a detailed consideration of the required Data Functionality, which is both more readily available and more strongly correlated with development effort than the required Transaction Functionality.
    - The flexibility lies predominantly with how thoroughly the Transaction Functionality is assessed in developing the estimate.
    - The level of detail to be assessed during the size estimation is determined either by the available documentation or by the purpose of the estimate.

6.5.4 Overall Contributions

- Software sizing has been proposed as a unifying factor between software estimation in industry and in academic research.
- For the software industry:
    - The size of a system is explicitly considered during the estimation process.
    - The formal functional sizing methods used provide a more methodical approach than expert judgement-based methods.
    - Software sizing can complement the existing effort and cost based estimation approaches used in industry.
- For academic research:
    - The need to focus on business value rather than basic estimation accuracy would greater serve the needs of the software industry.
    - Focusing on the level of sizing data provided by the estimation method may hold more potential for providing such value.
    - Enhancing simplified sizing methods to provide more than just an approximation of the overall size would enable research to be directed towards achieving industry acceptance.
- Completing this work with the assistance of Equiniti-ICS has highlighted the importance of industry participation in this type of research:
    - Identifying and acknowledging what industry professionals perceive the most important issues for estimation to be enables the research to focus on the most beneficial aspects of software sizing.
    - Performing software sizing on real world commercial project documentation provides a more realistic experience than with using historical datasets or requirements documentation authored specifically for research. This helps identify the limitations of sizing methods.

6.6 Generalisability and Limitations of Research

The design of this research study involved consideration of its general applicability. The following generalisation mechanisms were adopted within this research:

- The use of the ISO standard NESMA functional sizing method ensured that the size estimation was conducted in accordance with established guidelines. The NESMA method is essentially identical to the most widely used functional sizing method, the ISO standard IFPUG method, and is based on the original FPA approach. The nature of the projects is also consistent with those that are explicitly supported by FPA based methods. This research is therefore highly applicable to the field of FPA related research.

- The use of bid stage documentation involved took the form of Request for Tender documents provided to Equiniti-ICS. This served to replicate the circumstances in which functional size estimation could be used to inform the development of a bid for a software project. This research is therefore applicable to the field of bid related size estimation.
- The detailed classification of required functionality in Chapter 4 was derived from the standard BFC of the FPA based methods. The developed simplified sizing method in Chapter 5 was constructed using the standard BFC of the FPA based methods. The research therefore did not introduce any new functional components to be identified by the estimator. The approaches adopted in these Chapters are therefore applicable to individuals with expertise in FPA based methods.

The completion of this research involved limitations on the extent to which it could be generalised. The main limitations of this overall research thesis can be summarised as follows:

- The case oriented focus of the research involved the use of a single software organisation. This enabled an in depth investigation within this organisation, but the outcomes of this research can therefore not facilitate inferences about estimation practices within the wider software development industry. The presence of an initial bidding process is common to development organisations at this level, requiring estimates to be developed at this point. In the lifecycle, it is considered likely that a more detailed description of the required functionality will be developed prior to the implementation of the system in any software development organisation. The nature of the project documentation developed by this company reflects standard formats such as Use Case descriptions, Data Models etc.
- The significant time required to complete the sizing for each project using the NESMA methods, and the availability of complete documentation from Equiniti-ICS, limited the Research Phase to eleven commercial projects. The use of a dataset of eleven projects from a single software organisation limits the generalisability of the statistical analysis of the sizing results. The extent to which the results would be replicated with other software projects from other software organisations is unclear.
- The development of a simplified sizing method has been internally validated on eleven commercial projects, which precludes it being considered as an established method. This research therefore serves as a validation of the process of simplifying existing functional sizing methods in such a manner as to preserve the breadth of estimation data produced by the fuller methods. The investigation into simplifying the sizing approach

has been externally validated via the peer review process resulting in journal publication.

- The use of a single human estimator to complete the software sizing activities does not address the potential subjectivity of utilising functional sizing. The degree to which such an approach can be generalised for consistent use by other software estimators is therefore undetermined. The design of each research phase did, however, limit any potential negative impact on the process of completing the sizing activities using a single estimator.

6.7 Future research issues

The results of this research have highlighted a number of issues that would require further investigation. In addition, ongoing developments within software estimation research provide further opportunities for extending the work presented in this thesis. The main future research issues are as follows:

- **Investigating the software estimation approaches adopted, both within the local software development industry and the wider software industry in general.** Research into the more informal expert judgement-based estimation approaches used within industry has primarily been focused on estimation accuracy, or on how to account for internal and external factors that can lead to bias in an estimate. Focusing specifically on the precise detail considered by and recorded as part of expert judgement-based estimation approaches would provide more clarity between how such approaches differ from functional software sizing in terms of the project data they consider. This investigation would provide insight into the potential for adopting some degree of software sizing within industry.
- **The use of software sizing to assess the estimation uncertainty (risk) at the bid stage should be investigated across a greater number of commercial projects**. The strength of the use of size ratios as indicators of potential future growth in functional size depends upon establishing stable ratios. Increasing the size of the dataset used to derive such ratios would provide the basis for determining if the ratios are sufficiently stable.
- **The simplified sizing method should be evaluated further in order to facilitate an assessment of its objectivity.** The initial default functional profile produced by this sizing method is derived from following conventional NESMA guidelines. It is the subsequent refinement of this default functional profile that is not based on an

established sizing method. Firstly, the method should be used on further commercial projects to provide greater confidence in the repeatability of the approach. Secondly, estimates developed using this method should be completed using multiple human estimators to establish an understanding of the level of subjectivity that could be inherent in refining the size estimates from the initial default functional profile.

- **The scope of the simplified sizing method should be widened to assess its general applicability.** The research in this thesis has utilised software projects drawn from the same application domain. The sizing method should be applied to other application domains recognised as suitable for the conventional FPA methods (IFPUG and NESMA) in order to determine how its suitability compares with those established methods. The general approach adopted within this research should be investigated for the simplification of other functional sizing methods to determine whether a similar simplified approach could be developed for a wider range of methods.
- **The use of software sizing within industry should be investigated by evaluating its use by current software development professionals:**
    - The use of students in software estimation experiments is quite common within research, in large part due to the greater availability of participants. Studies such as Pow-Sang (2017) have even looked at applying learning techniques from other domains for teaching functional sizing to students. The value of such research in a wider context may be limited, however, as it focuses mainly on just the conceptual complexity of functional sizing.
    - Robiolo (2011) conducted an experiment on how simple it is for software professionals to perform size estimation. This study included the use of the FPA method, with only one of the participants having any experience with functional sizing. Training provided for the experiment consisted only of a few lecture slides, with use case textual descriptions used for the application to be sized. The mean functional size of the measurements was 131.60 FP, with a standard deviation of 67.94 FP. The mean measurement time was 33.20 minutes, with a standard deviation of 25.07 minutes. The relatively high level of variation in both the size estimates and the estimation effort required would not be encouraging to a software development organisation. The use of only 5 participants from different backgrounds at an IT workshop, as in this experiment, would not have provided the most accurate reflection of introducing functional sizing to industry. Longer term experiments within an organisation on real world project documentation would be valuable in assessing its viability.
    - Each phase in this research thesis presents an opportunity to investigate the use of software sizing in practice:

        - The use of both the standardised NESMA method, and the simplified sizing approach developed in Research Phase 3, could be evaluated on the condition that development staff receives appropriate training in the methods. This would provide an indication of the relative investment an organisation would have to make in developing sufficient expertise to adopt a standardised software sizing method. It would also provide a comparison of the relative ease of use and reliability of the different approaches to be utilised by current software development professionals.
        - The potential for reducing the expertise required for software sizing could be investigated by considering the approaches adopted in Research Phases 2 and 3. The scope of software sizing could be scaled back to focus on the core aspects that provided the most beneficial data about a project. The use of sizing ratios for bid stage estimation could be adapted to reduce the learning curve required to perform the sizing activities. The simplified sizing approach could also be adapted to reduce the level of understanding required of the full NESMA method for its use, by reducing the range of fundamental components that must be identified and assessed by the human estimator.

- **The use of software sizing may have wider applicability beyond the realm of estimation.** This research thesis considered the use of characteristics of requirements documentation, such as page size and functional size per page, for assessing the level of detail present about the system functionality:
    - Zhi et al. (2015) conducted a review into the published research on the costs, benefits and quality of software documentation. Only 10% of the research papers were reported as being industry based, suggesting that there is a need to incorporate real world project documentation into such research. Research Phase 2 highlighted how software sizing may implicitly provide a measure of the quality of requirements documentation by identifying whether expected functionality has been included and at a sufficient level of detail. Further research should examine whether software sizing can explicitly measure the quality of requirements documentation. Only 17% of the research papers in the Zhi et al. (2015) review addressed the issue of the cost of software documentation. The authors noted the lack of systematic methods or models to measure these costs. Research into whether software sizing provides an indication of the subsequent costs of software documentation could contribute to addressing the limited coverage of this specific research topic.

  - The Yoshigami et al. (2017) study included the use of the ISBSG dataset to investigate the use of the ratios of individual BFCs for projects. The relationships between these ratios and other project characteristics, such as software quality, productivity and lifecycle phase ratios were investigated. For new projects the ratio of EI components was found to be significantly correlated with the implementation phase ratio. The EO and EQ component ratios were found to be significantly correlated with the fault ratio for projects. For maintenance projects, which include data on Add and Modify components, the ratios for those components were found to be significantly correlated with the test phase ratio. The approach adopted in Research Phase 2 could be extended to investigate whether using specific types of functionality are strongly correlated with specific phases of a project. This research could establish the potential for using such data to further assist with project scheduling.
- **The potential of software sizing as a complementary estimation measure may be extended beyond its use alongside other estimation methods.** Combining the use of functional sizing with the actual early phase effort for projects was investigated by Tsunoda et al. (2013). In their study they used projects from the ISBSG dataset that included data for planning and requirements analysis alongside the functional size estimates. Combining the size estimates with those early phase effort data values was found to provide the most accurate estimation model. When evaluating the variables within the estimation models, the partial regression coefficients were larger for the early phase activities, indicating that they had a more significant effect on the estimated development effort. The use of the more detailed types of functionality, such as in Research Phase 2, should be investigated as a complement to the use of early phase activity effort for overall effort estimation. Early phase activity effort data could also potentially provide additional validation of the relationship between the detailed estimate of the functional profile at the bid and functional specification stages.

6.8 Overall Conclusions

The divide between how software estimation is used in industry and the focus of research in this area was investigated by concentrating on what has been neglected on each side. For the software industry this was the lack of consideration for obtaining a methodical and detailed estimate of the system size. For the software estimation research community this was an insufficient focus on what value software sizing can add within the constraints of real world project development. The case for adopting more formalised approaches in industry needed to be made on business value terms, recognising the trade-off between benefits and costs associated with differing levels of rigour in the estimation process. The commercial application of a formalised estimation approach necessitates that (i) it can be readily incorporated into existing software development practices and (ii) it provides additional information beyond that produced by existing estimation practices. The main conclusions formed from this research thesis are:

- Software size estimation can only assume the position of primary estimation approach if it demonstrates a clear superiority in predicting actual effort and cost.
- The limited requirements detail available at the bid stage excludes the possibility of obtaining any such superior estimation accuracy.
- The effort required to get the most accurate size estimate possible at the functional specification stage cannot be justified on business value terms.
- Software size estimation can play a complementary role to existing expert judgement-based approaches only if it minimises the estimation effort required.
- The focus of this research has been on developing the fundamental sizing data that is beneficial to project management and doing so in an efficient manner.
- At the bid stage the development of a detailed estimate of Data Functionality in particular provides the most efficient approach to deriving the fundamental sizing data.
- At the functional specification stage the use of an enhanced simplified sizing approach provides the most efficient approach to obtaining the necessary level of estimate detail.

Appendix A – Sizing Results for each Individual Project

Table 73 - Individual Project Sizing Results

Estimation Type	**Total Function Count Comparison (Function Points)**						
	ILF (% Inaccuracy Relative to Full NESMA)	**EIF (% Inaccuracy Relative to Full NESMA)**	**EI (% Inaccuracy Relative to Full NESMA)**	**EO (% Inaccuracy Relative to Full NESMA)**	**EQ (% Inaccuracy Relative to Full NESMA)**	**Total Function Points**	**Overall Percentage Inaccuracy Relative to Full NESMA**
Project A							
Indicative NESMA	-	-	-	-	-	430	-31.64%
Level I	112	15	184	90	72	473	-24.80%
	(-6.67%)	(0.00%)	(26.90%)	(-68.97%)	(22.03%)		
Level II	112	15	175	281	72	655	-4.13%
	(-6.67%)	(0.00%)	(20.69%)	(-3.10%)	(22.03%)		
Level III	112	15	142	281	72	622	-1.11%
	(-6.67%)	(0.00%)	(-2.07%	(-3.10%)	(22.03%)		
Estimated NESMA	112	15	164	300	56	647	2.86%
	(-6.67%)	(0.00%)	(13.10%)	(3.45%)	(-5.08%)		
Full NESMA	120	15	145	290	59	629	-
Project B							
Indicative NESMA	-	-	-	-	-	765	-45.55%

Level I	203 (-5.58%)	20 (0.00%)	340 (22.30%)	160 (-81.22%)	128 (220.00%)	851	-39.43%
Level II	203 (-5.58%)	20 (0.00%)	372 (33.81%)	710 (-16.67%)	16 (-60.00%)	1321	-5.98%
Level III	212 (-1.40%)	20 (0.00%)	300 (-7.91%)	756 (-11.27%)	41 (2.50%)	1329	-5.41%
Estimated NESMA	203 (-5.58%)	20 (0.00%)	276 (-0.72%)	820 (-3.76%)	36 (-10.00%)	1355	-3.56%
Full NESMA	215	20	278	852	40	1405	-
Project C							
Indicative NESMA	-	-	-	-	-	1295	-31.55%
Level I	357 (-12.29%)	10 (0.00%)	604 (0.67%)	260 (-62.91%)	208 (19.54%)	1439	-23.94%
Level II	357 (-12.29%)	10 (0.00%)	681 (13.50%)	800 (14.12%)	208 (19.54%)	2056	8.67%
Level III	388 (-4.67%)	10 (0.00%)	608 (1.33%)	721 (2.85%)	183 (5.17%)	1910	0.95%
Estimated NESMA	357 (-12.29%)	10 (0.00%)	668 (11.33%)	755 (7.70%)	204 (17.24%)	1994	5.39%
Full NESMA	407	10	600	701	174	1892	-

Project D							
Indicative NESMA	-	-	-	-	-	510	-45.34%
Level I	140	5	232	100	80	557	-40.30%
	(-10.83%)	(0.00%)	(-25.40%)	(-72.60%)	(-15.79%)		
Level II	140	5	378	345	80	948	-1.61%
	(-10.83%)	(0.00%)	21.54%)	(-5.48%)	(-15.79%)		
Level III	152	5	336	345	80	918	-1.61%
	(-3.18%)	(0.00%)	(8.04%)	(-5.48%)	(-15.79%)		
Estimated NESMA	140	5	340	385	100	970	3.97%
	(-10.83%)	(0.00%)	(9.32%)	(5.48%)	(5.26%)		
Full NESMA	157	5	311	365	95	933	-
Project E							
Indicative NESMA	-	-	-	-	-	685	-15.01%
Level I	189	5	316	140	112	762	-5.46%
	(-9.13%)	(0.00%)	(22.48%)	(-55.70%)	(489.47%)		
Level II	189	5	289	320	8	811	0.62%
	(-9.13%)	(0.00%)	(12.02%)	(1.27%)	(-57.89%)		
Level III	211	5	243	320	22	801	-0.62%
	(1.44%)	(0.00%)	(-5.81%)	(1.27%)	(15.79%)		

Estimated NESMA	189 (-9.13%)	5 (0.00%)	300 (16.28%)	340 (7.59%)	20 (5.26%)	854	5.96%
Full NESMA	208	5	258	316	19	806	-
Project F							
Indicative NESMA	-	-	-	-	-	235	7.80%
Level I	63 (12.50%)	5 (0.00%)	100 (56.25%)	50 (-16.67%)	40 (21.21%)	258	18.35%
Level II	63 (12.50%)	5 (0.00%)	68 (6.25%)	58 (-3.33%)	40 (21.21%)	234	7.34%
Level III	63 (12.50%)	5 (0.00%)	68 (6.25%)	57 (-5.00%)	36 (9.09%)	229	5.05%
Estimated NESMA	63 (12.50%)	5 (0.00%)	68 (6.25%)	65 (8.33%)	36 (9.09%)	237	8.72%
Full NESMA	56	5	64	60	33	218	-
Project G							
Indicative NESMA	-	-	-	-	-	585	-24.61%
Level I	161 (-8.52%)	5 (0.00%)	268 (-18.29%)	120 (-34.07%)	96 (12.94%)	650	-16.24%
Level II	161	5	347	145	96	754	-2.84%

	(-8.52%)	(0.00%)	(5.79%)	(-20.33%)	(12.94%)		
Level III	175	5	306	185	95	766	-1.29%
	(-0.57%)	(0.00%)	(-6.71%)	(1.65%)	(11.76%)		
Estimated NESMA	161	5	300	185	80	731	-5.80%
	(-8.52%)	(0.00%)	(-8.54%)	(1.65%)	(-5.88%)		
Full NESMA	176	5	328	182	85	776	-
Project H							
Indicative NESMA	-	-	-	-	-	565	-41.21%
Level I	147	20	244	125	100	636	-33.82%
	(-5.77%)	(0.00%)	(-26.51%)	(-68.51%)	(78.57%)		
Level II	147	20	372	280	100	919	-4.37%
	(-5.77%)	(0.00%)	(12.05%)	(-29.47%)	(78.57%)		
Level III	160	20	343	376	65	964	0.31%
	(2.56%)	(0.00%)	(3.31%)	(-5.29%)	(16.07%)		
Estimated NESMA	147	20	348	410	60	985	2.50%
	(-5.77%)	(0.00%)	(4.82%)	(3.27%)	(7.14%)		
Full NESMA	156	20	332	397	56	961	-

Project I							
Indicative NESMA	-	-	-	-	-	460	7.23%
Level I	126	5	208	95	76	510	18.88%
	(-15.44%)	(0.00%)	(69.11%)	(-34.93%)	(1166.67%)		
Level II	126	5	140	125	4	400	-6.76%
	(-15.44%)	(0.00%)	(13.82%)	(-14.38%)	(-33.33%)		
Level III	152	5	122	130	4	413	-3.73%
	(2.01%)	(0.00%)	(-0.81%)	(-10.96%)	(-33.33%)		
Estimated NESMA	126	5	124	150	4	409	-4.66%
	(-15.44%)	(0.00%)	(0.81%)	(2.74%)	(-33.33%)		
Full NESMA	149	5	123	146	6	429	-
Project J							
Indicative NESMA	-	-	-	-	-	1360	-9.51%
Level I	378	5	640	275	220	1518	1.00%
	(-17.83%)	(0.00%)	(31.42%)	(-42.11%)	(189.47%)		
Level II	378	5	424	437	166	1410	-6.19%
	(-17.83%)	(0.00%)	(-12.94%)	(-8.00%)	(118.42%)		
Level III	439	5	489	448	54	1435	-4.52%

	(-4.57%)	(0.00%)	(0.41%)	(-5.68%)	(-28.95%)		
Estimated NESMA	378	5	472	515	64	1434	-4.59%
	(-17.83%)	(0.00%)	(-3.08%)	(8.42%)	(-15.79%)		
Full NESMA	460	5	487	475	76	1503	-
Project K							
Indicative NESMA	-	-	-	-	-	1355	-29.83%
Level I	371	15	628	275	220	1509	-21.85%
	(-10.17%)	(0.00%)	(-8.99%)	(-59.56%)	(65.41%)		
Level II	371	15	739	735	167	2027	4.97%
	(-10.17%)	(0.00%)	(7.10%)	(8.09%)	(25.56%)		
Level III	408	15	656	690	132	1901	-1.55%
	(-1.21%)	(0.00%)	(-4.93%)	(1.47%)	(-0.75%)		
Estimated NESMA	371	15	692	730	152	1960	1.50%
	(-10.17%)	(0.00%)	(0.29%)	(7.35%)	(14.29%)		
Full NESMA	413	15	690	680	133	1931	-

Appendix B – Example of Simplified Sizing Method Calculation

To obtain the same output detail as provided by a full function count method the size of each BFC must be calculated from the Data Classes and the Transaction Classes. For this example, assume that after examination of requirements documentation a project was determined to contain 20 ILFs, comprised of 35 RETs, and 2 EIFs. Figure B.0 shows the initial Transaction Classes derived automatically for each ILF (1 to 5) and EIF (6 and 7). In this case the three EI Transaction Classes correspond to the 'Add', 'Amend' and 'Delete' functionality, in accordance with the Indicative NESMA method assumptions. The initial total functional count, corresponding to Level I, for the project is calculated using the NESMA complexity weighting values from Table 55.

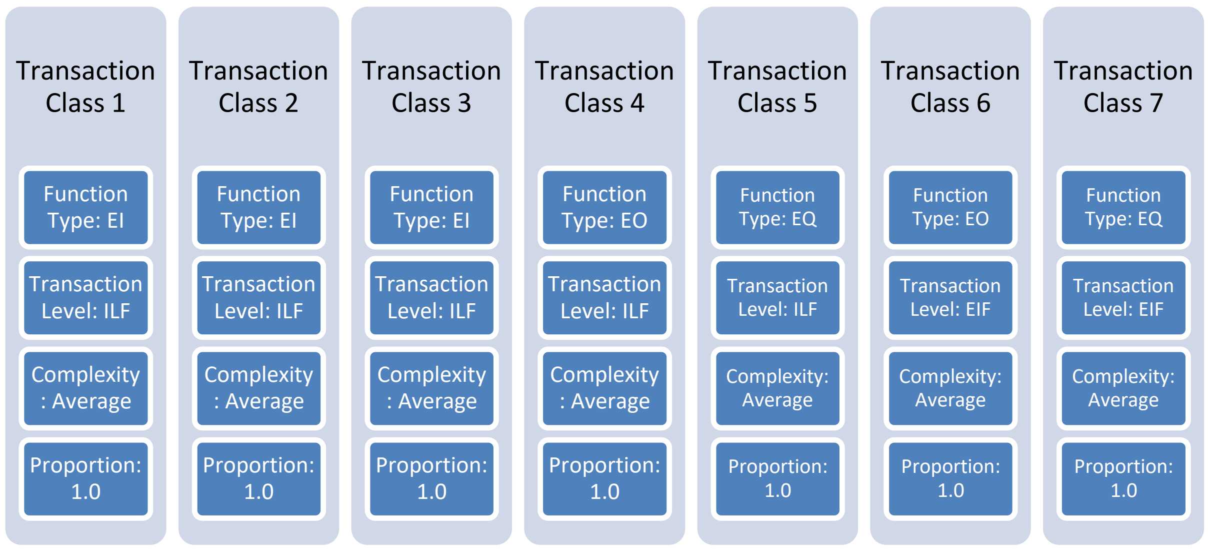

Figure B.0 - Example of Initial Transaction Classes (Level I)

The functional size of each Data Class is calculated using the following formula:

$$\textit{Number of FP} = (\textit{Number of Data Functions} * \textit{Proportion} * \textit{Complexity Weighting of Function Type})$$

For the example project the functional size for each Data Class is as follows (Note: ‘Low’ Complexity Weighting for Data Functions = 7):

$$\textit{Data Class 1} = (20*1*7) = 140\ \textit{FP}$$

*Data Class 2 = (2*1*7) = 14 FP*

The functional size of each Transaction Class is calculated using the following formula:

*Number of FP = (Number of Transaction Level Components * Proportion * Complexity Weighting of Function Type)*

For Transaction Classes, which are applied across all their respective Transaction Level components, the 'Proportion' variable takes the value of 1. The functional size for each Transaction Class would therefore be as follows (Note: ‘Average’ Complexity Weighting for EI and EQ Types = 4, and for EO Type = 5):

*Transaction Class 1 = (20*1*4) = 80 FP*

*Transaction Class 2 = (20*1*4) = 80 FP*

*Transaction Class 3 = (20*1*4) = 80 FP*

*Transaction Class 4 = (20*1*5) = 100 FP*

*Transaction Class 5 = (20*1*4) = 80 FP*

*Transaction Class 6 = (2*1*5) = 10 FP*

*Transaction Class 7 = (2*1*4) = 8 FP*

The functional count for the each BFC type is calculated by adding the size of each of their respective Data Classes and Transaction Classes. This provides the same functional profile detail as the full NESMA method. For this example, the size for each BFC type would therefore be as follows:

ILF = 140 FP (Data Class 1)

EIF = 14 FP (Data Class 2)

EI = 80 + 80 + 80 = 240 FP (Transaction Classes 1,2,3)

EO = 100 + 10 = 110 FP (Transaction Classes 4,6)

EQ = 80 + 8 = 88 FP (Transaction Classes 5,7)

The total functional count for the project would therefore be the sum of these values:

$$\textit{Total Functional Count} = 140 + 14 + 240 + 110 + 88 = 592\ FP$$

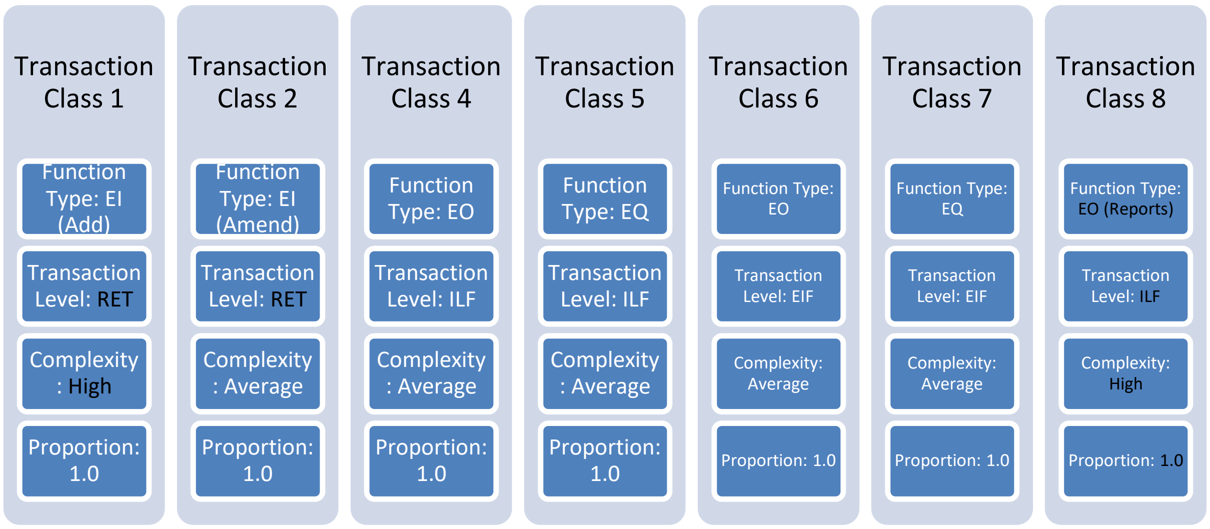

Figure B.1 - Example of Refined Transaction Classes (Level II)

Consideration of the transaction functions described in the project documentation would enable these initial Transaction Classes to be refined. For example, assume Level II assessment of the project documentation reveals that:

(Refinement 1) - The ‘Add’ and ‘Amend’ functionality has generally been requested at the level of the RET.

(Refinement 2) - The complexity of the ‘Add’ functionality may be generally high

(Refinement 3) - ‘Delete’ functionality may not be required in the completed application.

(Refinement 4) - The presence of more extensive, and complicated, reporting requirements may require the addition of a second EO Transaction Class of high complexity.

The set of Data Classes remains unchanged since no relevant refinements were made at this level. The set of amended Transaction Classes produced by these refinements is shown in Figure B.1. The changes made from the Transaction Classes in Figure B.0 are shown in black text. In Transaction Classes '1' and '2' the Transaction Level has been amended to RET (Refinement 1). In Transaction Class ‘1’ the Transaction Complexity has been amended to

High (Refinement 2). The EI Type Transaction Class '3' from Figure 4 has been removed (Refinement 3) and a new EO Type Transaction Class '8' has been added (Refinement 4).

The functional sizes for the Data Classes and Transaction Classes would be as follows:

*Data Class 1 = (20*1*7) = 140 FP*

*Data Class 2 = (2*1*7) = 14 FP*

*Transaction Class 1 = (35*1*6) = 210 FP*

*Transaction Class 2 = (35*1*4) = 140 FP*

*Transaction Class 4 = (20*1*5) = 100 FP*

*Transaction Class 5 = (20*1*4) = 80 FP*

*Transaction Class 6 = (2*1*5) = 10 FP*

*Transaction Class 7 = (2*1*4) = 8 FP*

*Transaction Class 8 = (20*1*7) = 140 FP*

The functional sizes for each BFC component would be as follows:

*ILF = (20*7) = 140 FP* (Data Class 1)

*EIF = (2*7) = 14 FP* (Data Class 2)

EI = 210 + 140 = 350 FP (Transaction Classes 1,2)

EO = 100 + 140 + 10 = 250 FP (Transaction Classes 4,6,8)

EQ = 80 + 8 = 88 FP (Transaction Classes 5,7)

The total functional count for the project would now be:

Total Functional Count = 140 + 14 + 350 + 250 + 88 = 842 FP

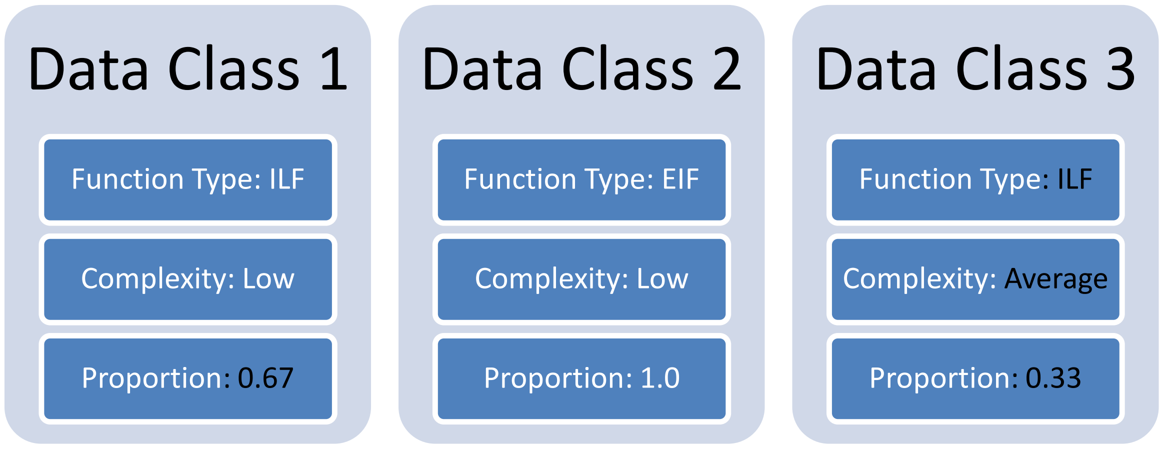

Figure B.2 - Example of Refined Data Classes (Level III)

Further refinements can then be made allowing the relative proportion of the required functionality to be estimated. In Level III, assume it was determined that:

(Refinement 5) – Approx. one third of the ILFs counted in this project are of Average complexity. The other two thirds of the ILFs should remain at Low complexity

(Refinement 6) - The 'Amend' Transaction Class can be divided into one 'Amend' class and one 'Link' class. The 'Amend' class should be at the ILF level and be of Average complexity. The 'Link' class should be at the RET level and be of Low complexity.

(Refinement 7) - EQs would only be required for approximately half of the ILFs counted in this project

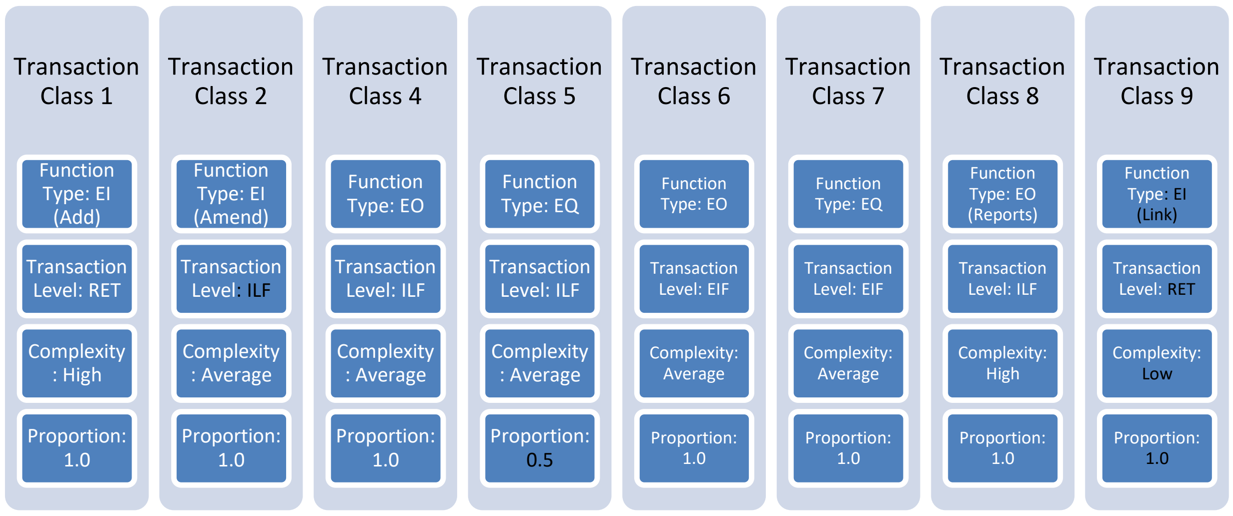

Figure B.3 - Example of Refined Transaction Classes (Level III)

The set of amended Data Classes and Transaction Classes produced by these refinements are shown in Figure B.2 and Figure B.3. Refinement 5 leads to all of the changes in Figure B.2. The proportion of Low complexity ILFs in Data Class 1 has been reduced to 0.67 (two thirds). Data Class 3 has been added to represent the one third of ILFs which are of Average Complexity. In Figure B.3, Transaction Class 9 has been added to represent the 'Link' functionality (Refinement 6). The remaining 'Amend' functionality represented by Transaction Class 2 has been modified to the ILF Transaction Level (Refinement 6). The Transaction Proportion has been amended to 0.5 in Transaction Class '5' (Refinement 7).

The functional sizes for these Transaction Classes would be as follows:

*Data Class 1 = (20*0.67*7) = 93 FP*

*Data Class 2 = (2*1*7) = 14 FP*

*Data Class 3 = (20*0.33*10) = 67 FP*

*Transaction Class 1 = (35*1*6) = 210 FP*

*Transaction Class 2 = (20*1*4) = 80 FP*

*Transaction Class 4 = (20*1*5) = 100 FP*

*Transaction Class 5 = (20*0.5*4) = 40 FP*

*Transaction Class 6 = (2*1*5) = 10 FP*

*Transaction Class 7 = (2*1*4) = 8 FP*

*Transaction Class 8 = (20*1*7) = 140 FP*

*Transaction Class 9 = (35*1*3) = 105 FP*

The number of transaction level components for the first two Transaction Classes has increased to 35 due to the change in transaction level from ILF to RET. The proportion variable in Transaction Class 5 has changed to 0.5 to reflect that approximately only half of the ILF require this associated transaction functionality.

The functional count for each of the BFC types would now be:

ILF = 93+67 = 160 FP (Data Classes 1,3)

*EIF = (2*7) = 14 FP*

EI = 210 + 80 + 105 = 395 FP (Transaction Classes 1,2,9)

EO = 100 + 140 + 10 = 250 FP (Transaction Classes 4,6,8)

EQ = 40 + 8 = 48 FP (Transaction Classes 5,7)

The total functional count for the project would now be:

Total Functional Count = 160 + 14 + 395 + 250 + 48 = 867 FP

Appendix C – Glossary of Terms

Base Functional Components: The function types that an application is comprised of, as viewed by the specific sizing method. For methods based upon the original Function Point Analysis approach there are five Base Functional Component types.

Data Element Type: In terms of Data Functions, it refers to the individual data attributes within those functions. In terms of Transaction Functions, it also includes any user recognisable information that crosses the application boundary e.g. user prompt.

Data Functions: The Base Functional Component types that represent a logical grouping of permanent data. For the Function Point Analysis based methods this corresponds to Internal Logical Files and External Interface Files.

Expert Judgement: An approach to software estimation that relies upon the expertise of a human estimator in order to develop the estimate. This approach may vary significantly in regards to the level of formality employed in its use.

External Input: A unique function, as recognised by the user, in which data is entered into the application for the purpose of maintaining the data. It may also include control information supplied by the user to facilitate the completion of the transaction function.

External Inquiry: A unique function, as recognised by the user, in which data is provided to the user that crosses the application boundary. The size of the output is predetermined, and the data retrieved from logical files does not require any further processing.

External Interface File: A logical grouping of permanent related data as recognised by the user of an application. The file must be both directly available to and used by the application being sized. The file must be maintained only by an external application.

External Output: A unique function, as recognised by the user, in which data is provided to the user that crosses the application boundary. The output will vary in size and/or the data retrieved from logical files requires further processing.

Function Complexity: A measure of the size allocated to an individual function. The complexity considers the number of Data Element Types and Logical Files that are involved with a function.

Function Point Analysis: The original development of a functional approach to size estimation. This approach provides the fundamental basis for the NESMA sizing method.

Functional Profile: The functional sizes for each of the Base Functional Component types associated with a specific sizing method.

Functional Size Measurement: An approach of measuring of the size of an application in terms of the functionality provided to the user. This size is expressed in the units associated with the specific sizing method e.g. Function Points.

Internal Logical File: A logical grouping of permanent related data as recognised by the user of an application. Each file must be both used by and maintained by the application being sized.

NESMA: The ISO Standard functional size measurement approach used in this research, consisting of three individual methods. These are the Indicative, Estimated and Full NESMA methods.

Record Element Type: A user recognisable sub-grouping of data within a Logical File. An individual Logical File may consist of one or more Record Element Types.

Simplified Sizing Method: Any sizing method that reduces the steps required, relative to a full method, to develop an estimate. For example, only a subset of Base Functional Component Types may be required to be identified as part of developing the estimate.

Transaction Functions: The Base Functional Component types that represent a distinct function of actions performed for the user. For the Function Point Analysis based methods this corresponds to External Inputs, External Outputs and External Inquiries.

List of References

Abrahão, S., Poels, G. and Pastor, O. (2006) A functional size measurement method for object-oriented conceptual schemas: design and evaluation issues. *Software & Systems Modeling*, 5 (1), 48-71.

Abran, A., Gil, B. and Lefebvre,E. (2004) Estimation Models based on Functional Profiles. *In: International Workshop on Software Measurement – IWSM/MetriKon*, 2-5 November 2004, Kronisburg. Shaker Verlag, 195-211.

Adem, N.A. Z. and Zarinah, M.K. (2010) Automating Function Points analysis based on functional and non functional requirements text. *In: The 2nd International Conference on Computer and Automation Engineering*, Volume 5, 26-28 February 2010, Singapore, 664-669.

Agrawal, M. and Chari, K. (2007) *Software Effort, Quality, and Cycle Time: A Study of CMM Level 5 Projects.* IEEE Transactions on Software Engineering, Vol. 33, No.3, March 2007, 145-156.

Agresti, W.W, Evanco, W.M and Thomas, W.M. (2010) Models for Improving Software System Size Estimates during Development. *Journal of Software Engineering and Applications*, 3, 1-10.

Albrecht, A.J. (1979) Measuring Application Development Productivity. *In: Proceedings of the Joint SHARE, GUIDE, and IBM Application Development Symposium*, 14–17 October 1979, IBM Corporation, 83–92.

Albrecht, A.J and Gaffney, J.E. (1983) Software function, source lines of code, and development effort prediction: A software science validation. *IEEE Transactions on Software Engineering*, 9.(6), 639-648.

Almakadmeh, K. and Abran, A. (2013) Experimental Evaluation of an Industrial Technique for the Approximation of Software Functional Size. *International Journal of Computers & Technology*, 10 (3), 1459-1474.

Alves, R., Valente, P. and Nunes, N. (2013) Improving Software Effort Estimation with Human-Centric Models: a comparison of UCP and iUCP accuracy. *In: '13 Proceedings of the 5th ACM SIGCHI symposium on Engineering interactive computing systems*, 24-27 June 2013, London. New York : ACM, 287-296.

Anda, B., Benestad, H.C. and Hove, S.E. (2005) A multiple-case study of software effort estimation based on use case points. *In: 2005 International Symposium on Empirical Software Engineering*, 17-18 November 2005, Noosa, Australia. IEEE Computer Society, 407-416.

Antoniol, A., Lokan, C., Caldiera, G. and Fiutem, R. (1999) A function point-like measure for object-oriented software. *Empirical Software Engineering*, 4 (3), 263-287.

Austin, P.C. and Steyerberg, E.W. (2015) The number of subjects per variable required in linear regression analyses. *Journal of Clinical Epidemiology*, 68 (6), 627-636.

Bagriyanik, S. and Karahoca, A. (2016) Automated COSMIC Function Point measurement using a requirements engineering ontology. *Information and. Software Technology*, 72 (C), 189–203.

Basten, D. and Sunyaev, A. (2014) A Systematic Mapping of Factors Affecting Accuracy of Software Development Effort Estimation. *Communications of the Association for Information Systems*, 34, 51-86.

Berlin, S., Raz, T., Glezer, C., and Zviran, M. (2009) Comparison of estimation methods of cost and duration in IT projects. *Information and Software Technology*, 51 (4), 738-748.

Bock D. B. and Klepper R. (1992) FP-S: a simplified function point counting method. *Journal of Systems and Software*, 18 (3), 245-54.

Boehm, B. (1981) *Software engineering economics*. Englewood Cliffs, NJ: Prentice-Hall.

Boehm, B., Abts, C., Brown, A. W., Chulani, S., Clark, B. K., Horowitz, K., Madachy, R., Reifer, D., and Steece, B. (2000) *Software cost estimation with COCOMO II*. Upper Saddle River, NJ: Prentice Hall.

Boetticher, G. D. and Lokhandwala, N. (2008) Using correlation and accuracy for identifying good estimators. *In: Proceedings of the 4th international workshop on Predictor models in software engineering*, 12-13 May 2008, Leipzig. New York : ACM, 33-38.

Braz, M.R., and Vergilo, S.R. (2004) Using fuzzy theory for effort estimation of object-oriented software. *In: 16th IEEE International Conference on Tools with Artificial Intelligence*, 15-17 November 2004, Boca Raton, FL. IEEE, 196-201

Buglione, L. and Gencel, C. (2008) Impact of Base Functional Component Types on Software Functional Size based Effort Estimation. *In:* Jedlitschka, A. andSalo, O. (eds.) *PROFES 2008, Lecture Notes in Computer Science (LCNS),* Volume 5089, Springer: Heidelberg, 75–89.

Bundschuh, M. (2006) Early Project Estimation with Early Function Point Prognosis. *In: 1st Annual International Software Measurement & Analysis Conference*, 10-15 September 2006, San Diego.

Cândido, E.J.D. and Sanches, R. (2004) Estimating the size of web applications by using a simplified function point method. *In: Proceedings of IEEE Webmedia and LA-Web conference,* Volume 1, 12-15 October, 2004, IEEE Computer Society, 98-105.

Carroll, E.R. (2005) Estimating software based on use case points. *In: Companion to the 20th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications*, 16-20 October 2005, San Diego. New York : ACM, 257–265.

Chen, X. and Staples, M. (2007) Using practice outcome areas to understand perceived value of CMMI specific practices for SMEs. In: Proceedings of the 14th European conference on Software Process Improvement, 26-28 September 2007, Potsdam, Germany. Berlin: Springer-Verlag, 59-70.

Chen, Y., Boehm, B. W., Madachy, R., and Valerdi, R. (2004) An empirical study of eservices product UML sizing metrics. *In: Proceedings of the 2004 International Symposium on Empirical Software Engineering,* 19-20 August 2004, Redondo Beach, CA.Washington: IEEE Computer Society Press, 199–206.

Chrissis, M. B., Konrad, M. and Shrum, S. (2003) *CMMI® Guidelines for Process Integration and Product Improvement,* Boston: Addison-Wesley.

Cohn, M. (2005) *Agile Estimating and Planning*. Upper Saddle River, NJ: Prentice-Hall.

COSMIC (2015) *Guideline for Early or Rapid COSMIC Functional Size Measurement by using approximation approaches*. COSMIC, Available from: https://cosmic-sizing.org/2015/07/guideline-for-early-or-rapid-cosmic-fsm/ [Accessed 20 July 2016].

Cuadrado-Gallego, J.J., Abran, A., Rodríguez-Soria, P. and Lara, M.A. (2014) An experimental study on the conversion between IFPUG and UCP functional size measurement units. *Journal of Zhejiang University - Science C*, 15 (3), 161-173.

Czarnacka-Chrobot, B. (2010, The economic importance of business software systems size measurement. *In: Computing in the Global Information Technology (ICCGI), 2010 Fifth International Multi-Conference on*, 20-25 September 2010, Valencia. IEEE, 293-299.

de Freitas Junior, M., Fantinato, M. and Sun, V. (2015) Improvements to the Function Point Analysis Method: A Systematic Literature Review. *IEEE Transactions on Engineering Management*, 62 (4), 495-506.

De Marco, L., Ferrucci, F. and Gravino, C. (2013) Approximate COSMIC size to early estimate web application development effort. *In: 39th Euromicro Conference Series on Software Engineering and Advanced Applications Approximate*, 4-6 September 2013, Santander, Spain. IEEE, 349–356.

Demirors, O. and Gencel, C. (2004) A Comparison of Size Estimation Techniques Applied Early in the Life Cycle. *In: European Software Process Improvement Conference (EurSPI 2004), Springer Lecture Notes in Computer Science (LNCS)*, Volume 3281, Springer Verlag, 184-194.

Demirors, O. and Gencel, C. (2009) Conceptual Association of Functional Size Measurement Methods. *IEEE Software*, 26 (3), 71-78.

Desharnais, J.M. (1988) Analyse Statistique de la Productivite des Projets de Developpement en Informatique a Partir de la Technique des Points de Fonction, maîtrise informatique de gestion, Univ. Quebec Montreal.

Desharnais J. and Abran A. (2001) Applying functional measurement method: Cognitive issues. *International Workshop on Software Measurement - IWSM 2001, in Current Trends in Software Measurement*, 28-29 August 2001, Montréal. Aachen, Germany: Sharker Verlag, 26–50.

Desharnais, J.M. and Abran, A. (2003) Approximation techniques for measuring function points. *Proceedings of the 13th International Workshop on Software Measurement (IWSM 2003)*, 23-25 September, Montréal. Aachen, Germany: Shaker Verlag, 270-286.

Desharnais, J.M., Buglione, L. and Kocaturk, B. (2011) Improving Agile Software Projects Planning Using the COSMIC Method. *In: VALOIR 2011*, 20-22 June 2011, Torre Cane, Italy.

Diab, H., Koukane, F., Frappier, M., and St-Denis, R. (2005) cROSE: automated measurement of COSMICFFP for Rational Rose RealTime. *Information and Software Technology*, 47 (3), 151-166.

DPO (2012) *The Early & Quick Function Points Manual for IFPUG method Release 1.3.1, Reference Manual 1.1*. DPO, Available from: www.dpo.it/eqfp/downloads/eq&fp-ifpug-31-rm-11-en-p.pdf [Accessed 20 July 2016]

Efe, P. (2006*) Software Size Estimation - A Survey*. Ankara, Turkey: METU Informatics Institute.

El Emam, K. and Koru, A. G. (2008) A replicated survey of IT software project failures. *IEEE Software*, 25 (5), 84-90.

Equiniti-ICS (2009) *Equinti acquires ICS Computing*. Equiniti-ICS, Available from: http://www.equiniti-ics.com/news-events-equiniti-acquires.html [Accessed July 2015]

European Commission (2003) Commission Recommendations of 6th May 2003 concerning the definition of micro, small and medium-sized enterprises. *Official Journal of the European Union*, 20 May 2003, (2003/361/EC) L124, 36 -41.

Faria, P., and Miranda, E. (2012) Expert Judgment in Software Estimation During the Bid Phase of a Project--An Exploratory Survey. *Joint Conference of the 22nd International Workshop on Software Measurement and the 2012 Seventh International Conference on Software Process and Product Measurement (IWSM-MENSURA),* 17-19 Oct 2012, Assisi, Italy. IEEE, 126-131.

Farr, L., and Zagorski, H. J. (1964) *Factors that Affect the Cost of Computer Programming. Volume II. a Quantitative Analysis*. Santa Monica, California: System Development Corp.

Fernández-Diego, M. and González-Ladrón-de-Guevara, F. (2014) Potential and Limitations of the ISBSG Dataset in Enhancing Software Engineering Research: A Mapping Review. *Information and Software Technology*. 56 (6), 527-544.

Fetcke, T., Abran, A., and Nguyen, H. (1997) Mapping the OO-Jacobson approach into function point analysis. *In: International Conference on Technology of Object-Oriented Languages and Systems (TOOLS-23).*1 August 1997, Santa Barbara. IEEE Computer. Society, 192-202.

Ferrucci, F., Gravino, C. and Lavazza, L. (2016) Simple function points for effort estimation: a further assessment. *In: SAC '16 Proceedings of the 31st Annual ACM Symposium on Applied Computing*, 4-8 April 2016, Pisa, Italy. New York: ACM, 1428-1433.

Forselius, L. (2006) Faster and more accurate functional size measurement by KISS – keeping it simple, *IFPUG FSS*, 28-29 March 2006, Cambridge, MA.

Fraternali, P., Tisi, M., and Bongio, A. (2006) Automating function point analysis with model driven development. *In: Proceedings of the 2006 conference of the Center for Advanced Studies on Collaborative research*, 16-19 October 2006, Toronto, Canada. Riverton, NJ: IBM Centre for Advanced Studies Conference.

Frohnhoff, S. and Engeroff, T. (2008) Field study: influence of different specification formats on the use case point method. *In: Software Process and Product Measurement. International Conferences IWSM 2008, MetriKon 2008, and Mensura 2008*, 18-19 November 2008, Berlin, Germany: Springer-Verlag, 62-75.

Furnham, A. and Boo, H. C. (2011). A literature review of the anchoring effect. *The Journal of Socio-Economics*, 40 (1), 35–42.

Gaffney, J.E. and Werling R. (1993) Estimating Software Size from Counts of Externals, a Generalization of Function Points. *In:* Gulledge, T.R. and Hutzler, W.P. eds. *Analytical Methods in Software Engineering Economics*, Berlin, Heidelberg: Springer, 193-203.

Gencel, C. and Demirors, O. (2008) Functional Size Measurement Revisited, *ACM Transactions on Software Engineering and Methodology*, 17(3), 1-36.

Gencel, C., Heldal, R., and Lind, K. (2009) On the Relationship between Different Size Measures in the Software Life Cycle. *In: Proceedings of the 2009 16th Asia-Pacific Software Engineering*, 1-3 December 2009, Batu Ferringhi, Malaysia. Washington: IEEE COomputer Society, 19-26.

Gibson, D.L, Goldenson D.R. and Kost, K. (2006) *Performance Results of CMMI® –based Process Improvement.* Technical Report from the Software Engineering Institute, Pittsburgh, USA: Carnegie Mellon University, CMU/SEI-2006-TR-004.

González-Ladrón-de-Guevara, F., Fernández-Diego, M. and Lokan, C. (2016) The usage of ISBSG data fields in software effort estimation: A systematic mapping study. *The Journal of Systems and Software*, 113, 188–215.

Green, S. (1991) How many subjects does it take to do a regression analysis. *Multivariate Behavioral Research*, 26 (3), 499-510.

Heemstra, F. J. (1992) Software Cost Estimation. *Information and Software Technology*, 34 (10), 627-639.

Hericko, M. and Živkovič, A. (2008) The size and effort estimates in iterative development. *Information and Software Technology*, 50 (7-8), 772-781.

Horgan, G., Khaddaj, S. and Forte, P. (1998) Construction of an FPA-type metric for early lifecycle estimation. *Information and Software Technology*, 40(8), 409-415.

Huijgens, H., Bruntink, M., van der Storm, T., Vogelezang, F. and van Deursen, A. (2016) An exploratory study on functional size measurement based on code. *In: ACM Proceedings of the International Conference on Software and Systems Process (ICSSP),* 14-22 May 2016, Austin, Texas. New York: ACM, 56-65.

Huijgens, H. and van Solingen, R. (2014) A Replicated Study on Correlating Agile Team Velocity Measured in Function and Story Points. *In: WETSoM 2014 Proceedings of the 5th International Workshop on Emerging Trends in Software Metrics*, 3 June 2014, Hyderabad, India. New York: ACM. 30-36.

Huijgens, H. and Vogelezang, F. (2016) Do Estimators Learn? On the Effect of a Positively Skewed Distribution of Effort Data on Software Portfolio Productivity. *In: Proceedings of the 7th International Workshop on Emerging Trends in Software Metrics*, 14-22 May 2016, Austin, TX. New York: ACM, 8-14.

IFPUG (2010) *The Function Point Counting Practices Manual version 4.3.1*. IFPUG), January 2010, Available from: www.ifpug.org.

International Organization for Standardization (2002) *ISO/IEC IS 20968:2002 Software engineering — Mk II Function Point Analysis — Counting practices manual*. Genève, Switzerland: International Organization for Standardization.

International Organization for Standardization (2005) *ISO/IEC IS 24570:2005 Software Engineering - NESMA functional size measurement method version 2.1 - Definitions and counting guidelines for the application of Function Point Analysis*. Genève, Switzerland: International Organization for Standardization.

International Organization for Standardization (2006) *ISO/IEC IS 14143-6:2006 Information technology -- Software measurement -- Functional size measurement -- Part 6: Guide for use of ISO/IEC 14143 series and related International Standards*. Genève, Switzerland: International Organization for Standardization.

International Organization for Standardization (2007) *ISO/IEC 14143-1:2007 Information technology -- Software measurement -- Functional size measurement -- Part 1: Definition of concepts*. Genève, Switzerland: International Organization for Standardization.

International Organization for Standardization (2009) *ISO/IEC IS 20926:2009 Software and systems engineering - Software Measurement - IFPUG Functional Size Measurement Method*. Genève, Switzerland: International Organization for Standardization.

International Organization for Standardization (2010) *ISO/IEC IS 29881:2010 Information Technology – Systems and software engineering – FISMA 1.1 Functional Size Measurement Method*. Genève, Switzerland: International Organization for Standardization.

International Organization for Standardization (2011) *ISO/IEC 19761:2011 Software Engineering - COSMIC: A Functional Size Measurement Method*, Genève, Switzerland: International Organization for Standardization.

ISBSG (2009) *ISBSG Estimating, Benchmarking and Research Suite 11*. International Software Benchmarking Standards Group (ISBSG), Available from: http://www.isbsg.org

Jamieson, D., Vinsen, K. and Callender, G. (2005) Agile Procurement: New Acquisition Approach to Agile Software Development. *In: EUROMICRO Conference on Software Engineering and Advanced Applications*, 30 August- 3 September 2005, Porto, Portugal. IEEE, 266-273.

Jones, C. (1995) Backfiring: converting lines of code to function points. *Computer*, 28 (11), 87-88.

Jones, C. (2008) *A new business model for function point metrics*. Available from: http://www.itmpi.org/assets/base/images/itmpi/privaterooms/capersjones/FunctPtBusModel2008.pdf

Jørgensen, M. (2004) A review of studies on expert estimation of software development effort. *Journal of Systems and Software,* 70 (1-2), 37-60.

Jørgensen, M. (2013) Relative Estimation of Software Development Effort: It Matters with What and How You Compare. *IEEE Software,* 30 (2), 74-79.

Jørgensen, M. (2014) The Ignorance of Confidence Levels in Minimum-Maximum Software Development Effort Intervals. *Lecture Notes on Software Engineering* 2.(4), 327-330.

Jørgensen, M. (2016) Unit effects in software project effort estimation: Work-hours gives lower effort estimates than workdays. *Journal of Systems and Software*, 117, 274-281.

Jørgensen, M. and Boehm, B. (2009) Viewpoints Software Development Effort Estimation: Formal Models or Expert Judgment?. *IEEE Software*, 26 (2), 14-19.

Jørgensen, M. and Carelius, G.J. (2004) An Empirical Study of Software Project Bidding. *IEEE Transactions On Software Engineering*, 30 (12), 953-969.

Jørgensen, M. and Grimstad, S. (2005) Over-optimism in Software Development Projects: ‘The Winner's Curse’. *In: Proceedings IEEE Conference in Electronics, Communications, and Computers (CONIELECOMP)*, 28 February - 3 March 2005, Puebla, Mexico. New York: IEEE Computer Society Press, 280–285.

Jørgensen, M. and Grimstad, S. (2008) Avoiding Irrelevant and Misleading Information when Estimating Development Effort. *IEEE Software,* 25 (3), (2008), 78-83.

Jorgensen, M. and Gruschke, T. M. (2009) The Impact of Lessons-Learned Sessions on Effort Estimation and Uncertainty Assessments. *IEEE Transactions on Software Engineering*, 35 (3), 368-383.

Jørgensen, M. and Shepperd, M. (2007) A systematic review of software development cost estimation studies. *IEEE Transactions on Software Engineering*, 33 (1), 35-53.

Jørgensen, M., Teigen, K. H., and Moløkken, K. (2004) Better sure than safe? Over-confidence in judgement based software development effort prediction intervals. *Journal of Systems and Software*, 70 (1-2), 79-93.

Kappelman, L. A., McKeeman, R., and Zhang, L. X. (2006) Early warning signs of IT project failure: The dominant dozen. *Information Systems Management*, 23 (4),31-36.

Karner, G. (1993) *Resource Estimation for Objectory Projects*. Torshamnsgatan, Sweden: Objective Systems SF AB.

Kashyap, D., Shukla, D. and Misra, A. K. (2014) Refining the Use Case Classification for Use Case Point Method for Software Effort Estimation. *In:* Proceedings of the *International Conference. on Recent Trends in Information, Telecommunication and Computing,* 21 March 2014, Chandigarh, India. Association of Computer Electronics and Electrical Engineers, 183-191.

Kassab, M., Neill, C. and Laplante, P. (2014) State of practice in requirements engineering: contemporary data. *Innovations in Systems and Software Engineering,* 10 (4), 235-241.

Kemerer, C.F. (1987) An empirical validation of software cost estimation models. *Communications of the ACM,* 30 (5), 416–429.

Kemerer, C. F. (1993) Reliability of Function Points Measurement - A Field Experiment. *Communications of the ACM*, 36(2), 85-97.

Keung, J. W., Kitchenham, B. A., and Jeffery, D. R. (2008) Analogy-X: providing statistical inference to analogy-based software cost estimation. *IEEE Transactions on Software Engineering,* 34 (4), 471-484.

Khan, P.M. and Beg, M.M.S. (2013) Application of Sizing Estimation Techniques for Business Critical Software Project Management. *International Journal of Soft Computing and Software Engineering*, 3 (6), 19-38.

Kitchenham, B. and Kansala, K. (1993) Interitem Correlations Among Function Points, I*n: Proceedings of the 15th International Conference on Software Engineering*, 21-22 May 1993, Baltimore, MD. IEEE, 11-14.

Kitchenham, B.A., MacDonell, S.G., Pickard, L.M. and Shepperd, M.J. (2001) What Accuracy Statistics Really Measure. *IEE Proceedings Software*, 148(3), 81-85.

Kitchenham, B. and Mendes, E. (2009) Why comparative effort prediction studies may be invalid. *In: Proceedings of the 5th International Conference on Predictor Models in Software Engineering,* 18-19 May 2006, Vancouver, Canada. New York: ACM, 1-5.

Kitchenham, B., Mendes, E. and Travassos, G.H. (2007) Cross Versus Within-Company Cost Estimation Studies: A Systematic Review. *IEEE Transactions on Software Engineering*, 33(5), 316-329.

Kocaguneli, E., Menzies, T. and Keung, J (2012) On the Value of Ensemble Effort Estimation. *IEEE Transactions on Software Engineering,* 38 (6), 1403-1416.

Kralj, T., Rozman, I., Hericko, M., and Zivkovic, A. (2005) Improved standard FPA method-resolving problems with upper boundaries in the rating complexity process. *Journal of Systems and Software*, 77 (2), 81-90.

Kusumoto, S., Matukawa, F., Inoue, K., Hanabusa, S., and Maegawa, Y. (2004) Estimating effort by use case points: method, tool and case study. *In: Proceedings of 10th International*

Symposium on Software Metrics, 14-16 September 2004, Los Alamitos, CA. Washington: IEEE Computer Society, 292-299.

Laird, L. M. (2006) The limitations of estimation [software estimation]. *IT Professional*, 8 (6), 40-45.

Lavazza, L. (2015) Automated Function Points: Critical Evaluation and Discussion. *In: IEEE/ACM 6th International Workshop on Emerging Trends in Software Metrics*, 16-24 May 2015, Florence, Italy. Piscataway, NJ: IEEE Press, 35-43.

Lavazza, L. (2016) An Investigation on the Relative Cost of Function Point Analysis Phases. *In: ICSEA 2016 : The Eleventh International Conference on Software Engineering Advances*, 21-25 August 2016, Rome, Italy. IARIA, 16-21.

Lavazza, L. (2017) On the Effort Required by Function Point Measurement Phases. *International Journal on Advances in Softwar*e, 10 (1&2), 108-120.

Lavazza, L. and Liu, G. (2013) An Empirical Evaluation of Simplified Function Point Measurement Processes. *International Journal on Advances in Software*, 6 (1&2), pp 1-13.

Lavazza, L. and Meli, R. (2014) An Evaluation of Simple Function Point as a Replacement of IFPUG Function Point. *In: Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement*, 6-8 October 2014, Rotterdam, Netherlands. IEEE, 196-206.

Lavazza, L., Morasca, S. and Robiolo, G. (2013) Towards a simplified definition of function points. Information and Software Technology, 55 (10), 1796–1809.

Lederer, A. L. and Prasad, J. (1992) 9 Management Guidelines for Better Cost Estimating. *Communications of the ACM*, 35 (2), 50-59.

Lederer, A. L. and Prasad, J. (1995) Causes of Inaccurate Software-Development Cost Estimates. *Journal of Systems and Software*, 31 (2), 125-134.

Lent, J.S. and Qu, Y. (2016) On XML Based Automated Function Point Analysis: An Effective Method to Assess Developer Productivity. *Lecture Notes on Software Engineering*, 3 (4), 245-250.

Lester, N., Wilkie, F. G., and McCaffery, F. (2004) Using a Ruler to Size Software. *In: 30th Euromicro Conference*, 31 August - 3 September 2004, Rennes, France.

Lima, O..S., Farias, P.P.M. and Belchior, A.D. (2003) Fuzzy Modeling for Function Points Analysis. *Software Quality Journal*, 11 (2), 149-166.

Little, T. (2006) Schedule estimation and uncertainty surrounding the cone of uncertainty. *IEEE Software*, 23 (3), 48-54.

Liu, G., Wang, X. and Fang, J. (2016) A UML-based Simple Function Point Estimation Method and Tool. *In: The Eleventh International Conference on Software Engineering Advances*, 21-25 August 2016, Rome, Italy. IARIA, 39-45.

Lokan, C. J. (1999) Empirical study of the correlations between function point elements. *In: Proceedings of the 1999 6th International Software Metrics Symposium*, 4-6 November 1999, Boca Raton, FL. IEEE, 200-206.

Lokan , C. J. (2000) An empirical analysis of function point adjustment factors. *Information and Software Technology*, 42 (9), 649-660.

Low, G.C. and Jeffery, D.R. (1990) Function Points in the Estimation and Evaluation of the Software Process. *IEEE Transactions on Software Engineering*, 16 (1), 64-71.

Magazinovic, A. and Pernstal, J. (2008) Any other cost estimation inhibitors?. *In: 2nd International Symposium on Empirical Software Engineering and Measurement*, 9-10 October2008, Kaiserslautern, Germany. ACM, 233-242.

Mair, C. and Shepperd, M. (2005) The consistency of empirical comparisons of regression and analogy-based software project cost prediction. *In: 2005 International Symposium on Empirical Software Engineering*, 17 November - 18 November 2005, Queensland, Australia. IEEE, 509-518.

Mair, C. and Shepperd, M. (2011) Human judgement and software metrics: vision for the future. *In: Proceedings of the 2nd International Workshop on Emerging Trends in Software Metrics*, 21-28 May 2011, Waikiki, Hawaii. New York: ACM, 81-84.

Mair, C., Martincova, M. and Shepperd. (2009) A literature review of expert problem solving using analogy. *In: 13th International Conference on Evaluation & Assessment in Software Engineering*, 20-21 April 2009. Durham. BCS.

Marinelli, V. (2008) *An analysis of current trends in requirements engineering practice*. In: Master of Software Engineering Professional Paper, Pennsylvania State University.

Maxwell , K.D. (2002) *Applied statistics for software managers*. Upper Saddle River, New Jersey: Prentice Hall.

McCaffrey, F., Wilkie, F. G., McFall, D., and Lester, N. (2004) Northern Ireland Software Industry Survey. *In: Fourth International SPICE Conference*, 28-29 April 2004, Lisbon, Portugal. 159-161.

MacDonell, S.G. and Shepperd, M.J. (2007) Comparing local and global software estimation models – reflections on a systematic review. *In: Proceedings of the 1st International Symposium on Empirical Software Engineering and Measurement*. 20-21 September 2007, Madrid, Spain. IEEE Computer Society Press, 404-409.

McConnell, S. (1996) *Rapid Development: Taming Wild Software Schedules*. Redmond, WA: Microsoft Press

McConnell, S. (2006), *Software Estimation: Demystifying the Black Art*. Redmond, WA: Microsoft Press.

Meli, R. (1997) Early and Extended Function Point: A New Method for Function Points Estimation, *In: IFPUG-Fall Conference*, 15-19 September 1997, ScottsDale, Arizona.

Meli, R. (2011) Simple Function Point! A new method for functional size measurement fully compatible with the IFPUG method 4.x. *In: UKSMA/COSMIC Intl Conf. on Software Metrics & Estimating*, 27-28 Oct. 2011, London.

Meli, R. (2015) Early & Quick Function Point Method An empirical validation experiment. *In: The First International Conference on Advances and Trends in Software Engineering*, 19-24 April 2015, Barcelona, Spain. IARIA, 14-22.

Meli, R and Santillo, L. (1999) Function Point Estimation Methods: a Comparative Overview. *In: FESMA '99 Conference Proceedings*, 4-8 October 1999, Amsterdam.

Minkiewicz, A. (2004) Use case sizing. *In: 19th International Forum on COCOMO and Software Cost Modeling*, 26-29 October 2004, Los Angeles, CA.

Minkiewicz, A. F. (2009) The evolution of software size: A search for value. *CrossTalk*, 22 (3-4), 23-26.

Miranda, E. (2001) Improving subjective estimates using paired comparisons. *IEEE Software*, 18 (1), 87-91.

Miranda, E., Bourque, P., and Abran, A. (2009) Sizing user stories using paired comparisons. *Information and Software Technology*, 51 (9), 1327-1337.

Moløkken-Østvold, K., Jørgensen, M., Tanilkan, S.S., Gallis, H., Lien, A.C. and Hove, S.E. (2004) A Survey on Software Estimation in the Norwegian Industry. *In: Proceedings of the 10th IEEE International Symposium on Software Metrics*, 11-17 September 2004, Chicago, USA. Washington: IEEE Computer Society, 208-219.

Nasir, M. and Ahmad, H.F. (2006) An Empirical Study to Investigate Software Estimation Trend in Organisations Targeting CMMI. *In: Proceedings of 5th IEEE/ACIS International Conference on Computer and Information Science*, 10-12 July 2006, Honolulu, USA. IEEE, 38-43.

Neill, C. J. and Laplante, P. A. (2003) Requirements engineering: the state of the practice. *IEEE Software,* 20 (6), 40-45.

NESMA (2004) *NESMA - Definitions and counting guidelines for the application of function point analysis. NESMA Functional Size Measurement method compliant to ISO/IEC 24570 version 2.1*. NESMA. Available from: https://nesma.org/downloads/countingpracticesmanual-en-v21/ [Accessed 20 July 2016].

Nguyen-Cong, D. and Tran-Cao, D. (2013) A review of effort estimation studies in agile, iterative and incremental software development. *In: The 2013 RIVF International Conference on Computing & Communication Technologies - Research, Innovation, and Vision for Future*, 10-13 November 2013, Hanoi, Vietnam. IEEE, 27-30.

Niazi, M. and Babar, M. A. (2009) Identifying high perceived value practices of CMMI level 2: An empirical study. *Information and Software Technology,* 51 (8), 1231-1243.

Nunes, N.J., Constantine, L. and Kazman, R. (2011) iUCP: Estimating Interactive-Software Project Size with Enhanced Use-Case Points. *IEEE Software*, 28 (4), 64–73.

Object Management Group (2014) *Automated Function Points (AFP) Version 1.0*, Object Management Group, OMG Document Number: formal/2014-01-03, Available from: http://www.omg.org/spec/AFP [Accessed 10 July 2017]

Ochodek, M. (2016) Functional size approximation based on use-case names. *Information and Software Technology*, 80, 73-88.

Ochodek, M., Nawrocki, J. and Kwarciak, K. (2011) Simplifying effort estimation based on Use Case Points. *Information and Software Technology,* 53 (3), 200-213.

Ohiwa, S., Oshino, T., Kusumoto, S. and Matsumoto, K. (2014) Towards an Early Software Effort Estimation Based on the NESMA Method (Estimated FP). *In: The IT Confidence 2014 conference – 2nd International Conference on IT Data collection, Analysis and Benchmarking*, 22 October 2014, Tokyo, Japan.

Ohlsson, M. C., Wohlin, C., and Regnell, B. (1998) A project effort estimation study. *Information and Software Technology*, 40 (14), 831-839.

Popović, J. and Bojić, D. (2012) A Comparative Evaluation of Effort Estimation Methods in the Software Life Cycle. *Computer Science and Information Systems*, 9 (1), 455-484.

Pow-Sang, J.A. (2017) Experiences using the Jigsaw learning technique to teach IFPUG function points, *In: World Engineering Education Conference*, 19-22 March 2017, Santos, Brazil. IEEE, 76-79.

Putnam, L. H. (1978) A general empirical solution to the macrosoftware sizing and estimating problem. *IEEE Transactions on Software Engineering*, SE- 4(4), 345-361.

Quesada-Lópeza, C. and Jenkins, M. (2016) Function Point Structure and Applicability: A Replicated Study. *Journal of Object Technology*, 15, 1-16.

Ram, D.J., and Raju, S.V.G.K. (2000) Object oriented design function points. *In: Proceedings of the First Asia-Pacific Conference on Quality Software*, 30-31 October 2000, Hong Kong. IEEE, 121–26.

Robiolo, G. (2011) How Simple is It to Measure Software Size and Complexity for an IT Practitioner?. *In: International Symposium on Empirical Software Engineering and Measurement*, 22-23 September 2011, Banff, Canada. Washington: IEEE Computer Society, 40-48.

Royce, W. (1970) Managing the Development of Large Software Systems. *In: Proceedings of IEEE WESCON*, 25-28 August 1970, IEEE Computer Society, 328-339.

Santana, C., Leoneo, F., Vasconcelos, A. and Gusmão, C. (2011) Using Function Points in Agile Projects. *In: Agile Processes in Software Engineering and Extreme Programming - Lecture Notes in Business Information Processing*, Volume 77, Berlin Heidelberg: Springer-Verlag, 176-191.

Santillo, L. (2012) Easy Function Points – 'Smart' Approximation Technique for the IFPUG and COSMIC Methods. *In: Joint 22nd International Workshop on Software Measurement and 7th International Conference on Software Process and Product Measurement*, 17-19 October 2012, Assissi, Italy. IEEE, 137–142.

Santillo, L., Conte, M. and Meli, R. (2005) Early & Quick Function Point: Sizing More with Less. *In: Proceedings of the 11th IEEE International Software Metrics Symposium*, 19-22 September, 2005, Como, Italy. Washington: IEEE Computer Society, 41.

Sauer, C. and Cuthbertson, C. (2003) *The State of IT Project Management in the UK 2002-2003*. Oxford: Templeton College, University of Oxford.

Savolainen, P., Ahonen, J.J. and Richardson, I. (2012) Software development project success and failure from the supplier's perspective: A systematic literature review. *International Journal of Project Management*, 30 (4), 458-469.

Sehra, S. K., Brar, Y. S., Kaur, N. and Sehra, S. S. (2017) Research Patterns and Trends in Software Effort Estimation. *Information and Software Technology*, 91, 1-21.

Sheetz, S.D., Henderson, D. and Wallace, L. (2009) Understanding developer and manager perceptions of function points and source lines of code. *Journal of Systems and Software*, 82 (9), 1540-1549.

Shepperd, M. and Cartwright, M. (2001) Predicting with sparse data. *IEEE Transactions on Software Engineering*, 27 (11), 987-998.

Shepperd, M. and MacDonell, S. (2012) Evaluating prediction systems in software project estimation. *Information and Software Technology*, 54(8), 820-827

Shepperd, M., Schofield, C., and Kitchenham, B. (1996) Effort estimation using analogy. *In: Proceedings of the 18Th International Conference on Software Engineering*, 25-29 March 1996, Berlin, Germany. Washingtwon: IEEE Computer Society, 170-178.

Silva A., Pinheiro P. and Albuquerque A. (2016) A Brief Analysis of Reported Problems in the Use of Function Points. *In:* Silhavy R., Senkerik R., Oplatkova Z., Silhavy P., Prokopova Z., eds. *Software Engineering Perspectives and Application in Intelligent Systems. Advances in Intelligent Systems and Computing*. Volume 465. Springer, Cham, 117-126.

Symons, C. R. (1988) Function Point Analysis - Difficulties and Improvements. *IEEE Transactions on Software Engineering*, 14 (1), 2-11.

Tichenor, C. (2008) How to Develop an ILF Model to Reduce the Time and Costs Required for Function Point Sizing. *IFPUG MetricViews Winter 2008*, [online] Available from: http://www.ifpug.org/members/newsletter/2008MetricViews/MV_Winter_2008.pdf [Accessed 10 January 2011].

Total Metrics (2007) *Methods for Software Sizing – How to Decide which Method to Use*, TotalMetrics, Available from: http://www.totalmetrics.com/function-point-resources/downloads/R185_Why-use-Function-Points.pdf [Accessed 10 January 2011].

Tsunoda, M., Kamei, Y., Toda, K. and M. Nagappan (2013) Revisiting software development effort estimation based on early phase development activities. *In: Proceedings of the 10th IEEE Working Conference on Mining Software Repositories*, 18-19 May 2013, San Francisco CA, Piscataway, NJ: IEEE Press, 429-438.

Tunalılar, S. and Demirors, O. (2011) An Exploration of Functional Size Based Effort Estimation Models. *International Journal of Software Engineering and Knowledge Engineering*, 21 (3), 413-429.

Uemura, T., Kusumoto, S., and Inoue, K. (2001) Function-point analysis using design specifications based on the unified modelling language. *Journal of Software Maintenance Evolution: Research Practice*, 13 (4), 223–243.

Usman, M., Mendes, E., Weidt, F. and Britto, R. (2014) Effort Estimation in Agile Software Development: A Systematic Literature Review. *In: Proceedings of the 10th International Conference on Predictive Models in Software Engineering*, 17 September 2014, Turin, Italy. New York: ACM, 82-91.

Valdés-Souto, F., and Abran, A (2012) Case study: COSMIC approximate sizing approach without using historical data. *In: Joint 22nd International Workshop on Software Measurement and 7th International Conference on Software Process and Product Measurement*, 17-19 October 2012, Assisi, Italy. IEEE, 178–189.

Valdés-Souto, F. and Abran, A. (2015) Improving the COSMIC approximate sizing using the fuzzy logic EPCU model. *In:* Kobyliński, A., Czarnacka-Chrobot, B. amd Świerczek, J., eds. *Joint International Workshop on Software Measurement and International Conference on Software Process and Product Measurement*, 5-7 October 2015, Krakow, Poland. Springer International, 192–208

Valdés-Souto, F. (2017) Analyzing the performance of two COSMIC approximation sizing techniques at the functional process level. *Science of Computer Programming*, 135, 105-121.

van den Berg, K., Dekkers, T. and Oudshoorn, R. (2005) Functional size measurement applied to UML-based user requirements. *In: Proceedings of the 2nd Software Measurement European Forum*, 16-18 March 2005, Rome, Italy. 69-80.

van Heeringen, H.S., van Gorp, E.W.M. and Prins, T.G. (2009). Functional size measurement – accuracy versus costs – is it really worth it?. Unpublished conference paper at: *Software Measurement European Forum*, 27-29 May 2009, Roma, Italy.

Verner, J.M. and Evanco, W.M. (2005) In-House Software Development: What Project Management Practices Lead to Success?. *IEEE Software*, 22 (1), 86-93.

Vinsen, K., Jamieson, D., and Callender, G. (2004) Use case estimation - the devil is in the detail. *In: Proceedings of 12th IEEE International Requirements Engineering Conference*, 6-11 September 2004, Los Alamitos, CA. Washington: IEEE Computer Society, 10-15.

Walkerden, F. and Jeffery, R. (1999) An empirical study of analogy-based software effort estimation. *Empirical Software Engineering*, 4 (2), 135-158.

Wang, X., Li, J. and Yu, F. (2008) Simplified Function Point Analysis Method Aiming at Small-to-medium-sized Software. *Jisuanji Gongcheng / Computer Engineering*, 34 (9), 103-105.

Wen, J. Li, S., Lin, Z., Hu, Y. and C. Huang (2012) Systematic literature review of machine learning based software development effort estimation models. *Information and Software Technology*, 54 (1), 41–59

Wilkie, F.G, McChesney, I.R., Morrow, P., Tuxworth, C. and Lester, N.G. (2011) The Value of Software Sizing. *Information and Software Technology,* 53 (11), 1236-1249.

Wilkie, F. G., McFall, D., and McCaffery, F. (2005) An Evalutation of CMMI Process Areas for Small-to-Medium-Sized Software Development Organistations. *Journal of Software Process Improvement and Practice*, 10 (2), 189-201.

Wilson, T. D, Houston, C. E., Etling, K. M..and Brekke, N. (1996) A new look at anchoring effects: Basic anchoring and its antecedents. *Journal of Experimental Psychology: General,* 125 (4), 387–402.

Wu, J. and Cai, X. (2008) A Software Size Measurement Model for Large-Scale Business Applications, *In: Proceedings of the International Conference on Computer Science and Software Engineering - Volume 2,* 12-14 December 2008, Hubei, China. Washington: IEEE Computer Society, 39-42.

Wydenbach, G. and Paynter. J. (1995) Software Project Estimation: a Survey of Practices in New Zealand. *New Zealand Journal of Computing*, 6 (1B), 317-327.

Yang, D., Wang, Q., Li, M. S., Yang, Y., Ye, K., and Du, J. (2008) A Survey on Software Cost Estimation in the Chinese Software Industry. *In: Proceedings of the 2008 Acm-Ieee International Symposium on Empirical Software Engineering and Measurement,* 9-10 October 2008, Kaiserslautern, Germany. New York: ACM, 253-262.

Yoshigami., K., Tsunoda, M., Yamada, Y. and Kusumoto, S. (2017) Should Function Point Elements Be Used to Build Prediction Models?. *In: 8th International Workshop on Empirical Software Engineering in Practice*, 13 March 2017, Tokyo, Japan. IEEE, 41-46.

Zhi, J., Garousi-Yusifoğlu, V., Sun, B., Shahnewaz, S. and Ruhe, G. (2015) Cost, benefits and quality of software development documentation: A systematic mapping. *Journal of Systems and Software*, 99 (C), 175–198.

Živkovič, A., Heričko, M., Brumen, B., Beloglavec, S. and Rozman, I. (2005a) The Impact of Details in the Class Diagram on Software Size Estimation. *Informatica*, 16(2), 295-312.

Živkovič, A., Rozman, I. and Heričko, M. (2005b) Automated software size estimation based on function points using UML models. *Information & Software Technology*, 47 (13), 881–890.