SOFTWARE COSTING ANALYSIS MODEL FOR EVALUATING VALUE FOR MONEY

Guy Callender, Diane Jamieson, and Kevin Vinsen*

ABSTRACT. Key procurement challenges include the definition of value and the identification of Value for Money principles. In the case of software development, the failure of contemporary systems of costing and project management to create reliable, replicable cost estimation that links to the requirements of project planners and managers represents a significant deficiency in procurement theory and practice. This paper explores the covert problems of software development procurement and its multi-disciplinary facets to arrive at a theoretical structure and methodology for estimating costs and value within unique software development procurement proposals. The methodology aims to provide a conceptual link between competing professional disciplines within a multi-disciplinary evaluation model.

INTRODUCTION

The comprehensive pursuit of business process outsourcing (Hodge, 1996), in which the presence of an external party complicates project or functional dynamics, has potentially resulted in greater involvement of specialist procurement practitioners and increased the visibility of poor Value for Money (VfM) decisions. Multiple contracting organizations and competing professional disciplines may be fundamentally involved, but existing methodologies, including those commonly applied in software development procurement, do not normally acknowledge the multi-disciplinary nature of large-scale or complex procurement.

---

* *Guy Callender, Foundation Chair in Leadership of Strategic Procurement, Curtin University of Technology. His research interests center on procurement policy, business intelligence, and public sector reform. Diane Jamieson, Senior Research Fellow, Curtin University of Technology. Her research interests include project management, ICT procurement policy, and software development. Kevin Vinsen, Research Associate, Murdoch University. His research interests include software engineering and software evaluation.*

Additionally, when the complete contracting process from planning through to delivered outcomes is considered, current methodologies often represent a seemingly rational approach in an environment where decisions may be based on ideological or conformist considerations.

Public sector procurement in many jurisdictions is typically undertaken using the principles of competitive tendering and contracting, with a public tender threshold of around $50,000. A traditional twelve step contracting process (SSC, 2002) requires budget approval for the entire life of a contract to be obtained in the contract planning process. The quality and honesty of information and cost estimates in the procurement plan become critical to contract outcomes and the necessity for subsequent procurement activities. Where contract costs have been poorly understood or have been arbitrarily reduced to meet political budget agendas, contracts are seen to have failed, exceeded budget or not met expectations. Predicting final costs is particularly difficult in the field of software development, where techniques and technologies undergo rapid evolution. Customers have tended to move away from requiring fixed prices for large-scale software development, but are still held accountable for managing to the budget specified in the procurement plan. This need for an accurate early estimate frequently causes major long term procurement dilemmas when software development is being purchased. The manner in which costs are represented often vary between the various parties involved in a software development contract, also creating ongoing conflict regarding exactly what has been delivered for what price.

The research outlined in this paper represents the first stage of a larger longitudinal, multi-disciplinary study of software development procurement to be undertaken over the next four years. The general aim of the project is the formation of more sophisticated costing models that build upon existing software development costing and management frameworks to create more reliable practitioner-usable and managerially-defensible approaches to software procurement and contract management. In particular the project aims to formalise a common representation of requirements and their costs to be used by the procurers and suppliers of software development. Enforcing a common approach through the procurement process is not common, although it has been attempted in some jurisdictions for less complex software development (Multimedia Victoria, 2003)

The paper will present the project objectives and outline the principal methodologies to be applied to deconstruct existing costing models in software development within the context of a major procurement activity. A review of the diverse, cross-disciplinary literature relevant to this procurement problem suggests very limited engagement with this conundrum at the level of the academic, practitioner, and manager. Simply put, the problem is seemingly too large and thus it remains unresolved. Furthermore, once problems emerge in a particular software development project, the difficulties are viewed as a general management failure, rather than being considered in the context of a highly complex procurement activity with multiple stakeholders across multiple disciplines.

The responsibility for addressing the problem is not usually referred back to those undertaking the detailed planning and specification for the project, such as the procurement specialists who could be assumed to have been involved in the contract release and evaluation. Moreover, specialized contracts of this nature are also likely to have been evaluated by a number of people with a specialist professional knowledge in only one dimension of the proposal, when a multi-disciplinary review is also required. For example, individual specialists in software development, general cost estimation, or project management might evaluate the costing estimates in isolation, without due consideration to the elements of Total Cost of Ownership (TCO) or Life Cycle Cost (LCC), or the impact of unintended complications as the software development unfolds.

Perhaps a more holistic, procurement-based process, centered around the achievement of VfM in relation to the TCO or over the LCC, is required. Specialized costing models could be integrated through common parameters to provide a core evaluative mechanism for software development procurement.

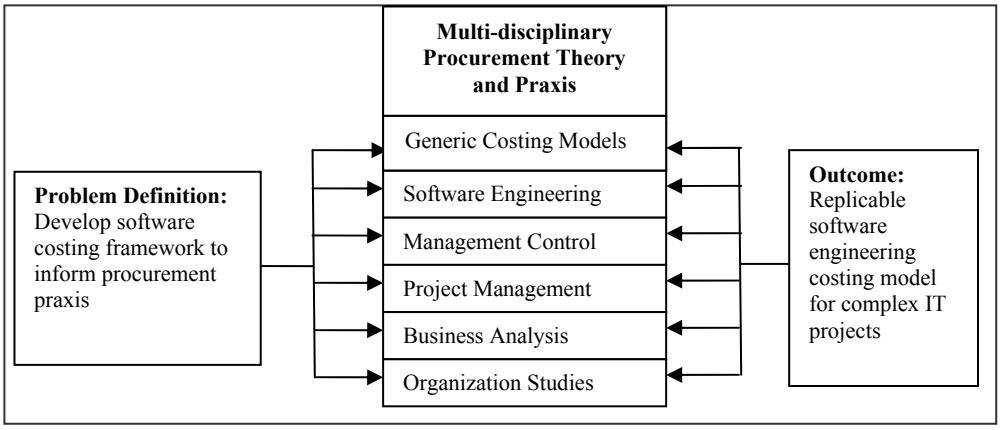
CHALLENGES IN SOFTWARE DEVELOPMENT PROCUREMENT

A detailed exploration of the challenges inherent in the creation of a replicable, standardised model for software development project costing is overdue. The ICT industry, in its current form, is around forty years old. Yet the principles of software costing are only just being elaborated. This research therefore commences with an iteration of two procurement

challenges: establishing specialized, yet replicable, costing models, and probing the problems of post-contract management. Both issues were identified as global purchasing challenges by the International Research Study of Public Procurement (IRSPP, 2003). The breadth of the literature review and the inter-relatedness of the literature to be considered in this research problem are illustrated in Figure 1 below.

In a recent review of the literature outlining the limitations of Information Technology (IT) investment, Suwardy, Ratnatunga, and Speight (2003, p. 328) reported the principal problems included "poor project management … lack of understanding of the new technology … and … insufficient software evaluation". These management-related shortcomings are compounded by the limited productivity improvements found to be associated with IT developments. For example, McGuckin and Stiroh (1998, p. 41) report that US productivity growth, in a macro sense, fell from an annual average of "3.4 percent between 1948 and 1973 to 1.2 percent between 1979 and 1997", inferring that the decline occurred despite the level of investment in ostensibly productivity-improving IT.

FIGURE 1
Developing a Procurement-Centered Software Engineering Costing Model



Furthermore, at the micro level, the Auditor-General of Victoria (2003) recently investigated the cost and project management problems arising from the implementation of an Academic Management System at RMIT University (RMIT) in Australia. The report drew upon an OECD Public Management Policy Brief (2001) which also reported on widespread experiences of large IT projects with budget excesses, deadline failures, and quality deficiencies. The Auditor-General's report found that these three issues, nominated by the OECD, had occurred at RMIT: the project budget had been exceeded by 370% (over AUD$46 million), deadlines were not met, and the software development project outcomes did not meet the requirements of RMIT. There was evidence that the project management mirrored some of the other deficiencies identified by the OECD (2001, p. 4), namely that IT project implementation needs to be "led by top management and not IT experts". RMIT management also suffered from the absence of an adequate software costing evaluation tool of the type proposed in this research.

The RMIT experience, while extreme, is not atypical. The Standish Group's CHAOS Report (1995, p. 895) "found that 32% of projects terminate before delivery and only 11% are completed on budget. Of the remaining 57% of projects that are completed, the average budget overrun is 87%". Lederer and Prasad (1995) reported that only one in four systems development projects included in their study was completed within cost estimates. Lederer and Prasad (1995, p. 132), researching the causes of inaccurate software cost estimates, recommended the adoption of an estimating methodology, and a need to "reassess the project control system for accountability and management commitment". Their research reaffirmed the link between the representation of software costs at the requirements phase and that used by the project managers.

As a consequence of the well-established nature of the negative management and practitioner experiences reported earlier, it could be expected that the problem would have been more widely researched in the areas of organization studies, costing, project management, procurement, and software engineering. However, the literature discussed below suggests that adequate software cost control tools do not exist and managers are therefore seemingly reliant on project management techniques that have been developed independently of procurement or software development requirements. Furthermore, preliminary interviews at senior management level suggest a focus on relationship management, and overall project deliverables may have been

substituted, at least in some cases, for detailed project breakdown during software development.

LITERATURE REVIEW

The first hurdle to be encountered in this type of research is addressing the multi-disciplinary range of communities of knowledge that surround the process of software development in order to arrive at a methodology that makes sense to procurement and other professionals. First, techniques for achieving profitability goals through the management of cost efficiency and procurement practice have led, over the past decade, to attempts to conceptualize the nature of VfM in a more sophisticated fashion. VfM has been linked to notions of lower costs, improved quality, shorter delivery time, technically superior products and services, and a targeted return for shareholders (Lasiter, 2001; Callender & Johnston, 2002). A quantitative evaluation of VfM has been developed as the TCO model which, when including Net Present Value (NPV) estimates, creates a sophisticated financial evaluation of purchase alternatives.

The VfM literature would be expected to make assumptions about the nature of value. Perhaps because it is frequently related to the economists' notion of utility, which "is essentially ignored in modern economic discourse" (Kahnement & Wakker, 1997, p. 375), the inherent nature of this concept is largely unexplored. The concept of value is also widely used in the organization studies literature (Clegg & Hardy, 1996; Childerhouse & Towill, 2002). However, while the related field of organizational psychology has, over the past half century, paid considerable attention to the issue of needs and needs satisfaction, the extent of this research provides a substantial warning to researchers about the difficult nature of subjective evaluations of 'value' inherent in concepts such as Use Value (Bowman & Ambrosini, 2000), or prescriptive value (Parasuraman, Zeithaml & Berry, 1988; Anderson & Narus, 1998).

Second, the notion of Earned Value has found popularity with project management practitioners, although it is also not widely researched in scholarly literature. Earned Value is defined by the Project Management Institute (PMI) (2000, p 201) as:

> The physical work accomplished plus the authorized budget for this work. The sum of the approved cost estimates (may include overhead allocation) for activities (or portions of activities) completed during a given period (usually project-to-date). Previously called the budgeted cost of work performed (BCWP) for an activity of group of activities.

Furthermore the PMI (2000, p 201) defines Earned Value Management as "a method for integrating scope, schedule, and resources, and for measuring project performance. It compares the amount of work that was planned with what was actually earned with what was actually spent to determine if cost and schedule performance are as planned."

While there are a number of textbooks in this field, the researched relationship between value and Earned Value has not been well established, apart from the loose adaptation of value models provided by Fleming and Hoppelman (1996), Anderson and Narus (1998), and Bowman and Ambrosini (2000). However, these Earned Value approaches do create the necessary link between value and costing which is integral to the proposed research.

Third, costing philosophy and practice has been enhanced by the emergence of Activity Based Costing (Brandt, Levine & Gourdous, 1999; Back & Maxwell, 2000). Once again, there are only limited links between the scholarly literature and the proposed research (Brandt *et al.,* 1999). There are historical links between the problems addressed in this research and earlier approaches to solving complex costing problems. For example, the early Nineteenth Century contribution of Mark Huish to the development of accounting and costing methodologies suited to the operation of complex railway systems (Gourvish, 1970) is relevant to the current research. Gourvish (1970) records that Huish made one of the first attempts to adequately separate capital and recurring expenses in a way that then enabled a railway to estimate the cost of running a rail system, as distinct from building one.

Fourth, work study (ILO, 1978) and organization and methods (MSD, 1965) also provide a basic conceptual grounding for the present study. As in the case of costing, there have been numerous attempts to measure work and, as a consequence, develop large-scale production techniques. An early attempt to demonstrate some of the necessary

requirements in the UK is recorded in engineer Brunel's era (Rolt, 1970, p. 33):

> It was claimed that with his [Marc Brunel's block making] machines six men could do what sixty had done before. Certainly they represented what was perhaps the first example of mechanized production in the world and they would soon become a showpiece for visiting notabilities [(sic)].

Later work by engineers, such as Taylor (1911) and Gantt (1919), provided the basis for measuring and costing labor output.

Fifth, the planning process is enhanced by the adoption of a contract management methodology. However, the development of a body of knowledge to support the contract management process over the life of a contract is limited, although a number of project management process tools have emerged (OGC, 2004).

The proposed research is designed to commence the building of procurement tools for enabling more comprehensive project planning and costing of IT software development projects, with an approach that bridges these multi-disciplinary boundaries.

DILEMMAS IN SOFTWARE DEVELOPMENT PROCUREMENT

Software development procurement involves a number of practitioner groups interacting simultaneously, the most prominent of which include a client's procurement specialists, a supplier's project manager, and a supplier's software engineers. The procurement lifecycle is normally based upon a standard twelve-step contracting process recommended by a number of professional and regulatory organizations (State Supply Commission, 2002). However this process requires expansion in the earlier phases to incorporate some of the major information flows required from the field of software engineering prior to the award of a contract.

Where the personnel with the skills to produce this information do not work for the customer organization, a common occurrence, a contractor specializing in procurement activities may be employed to assist with the procurement process, thus becoming part of the overall procurement lifecycle. However the multiplicity of potential arrangements ensures that the situation cannot be diagrammatically represented in a meaningful manner. The procurement workflow may

include procurement generalists who attempt to undertake the entire procurement process, and the full range of organizations specializing in software metrics, probity audit, and contract law. Nevertheless, an appreciation of the entire complex web or network of interactions, all relevant to the software development procurement process, is integral to the problem explored in this paper.

A business analysis activity is ideally undertaken in the early phases of a software development procurement to assist the customer in defining the project scope and budget. This activity could be shown as part of either the procurement or software engineering lifecycle. The software engineering lifecycle is also called upon prior to contract award in terms of the preliminary requirements analysis and system design that must be undertaken by the supplier to enable the offer and price to be formulated. Both the customer and the supplier need a reasonable degree of understanding of the user requirements, system design, and unit costs before a contract price and scope can be agreed upon.

These demands create a major contracting dilemma: the need for reliable information to form the basis of a contract long before that information would normally be produced as part of a contracted project. Even if a contract has a labor rate basis rather than a fixed price and scope, most customers expect to be able to achieve a guaranteed outcome for an agreed budget. Similarly, suppliers need a definite idea of their costs even when the contract basis is variable since failing to meet customer expectations, regardless of the cause, reflects on their perceived professionalism.

THE RESEARCH

The marketplace outcomes outlined earlier in this paper point to the need for accurate software cost estimates to be created early in the procurement lifecycle for budgeting and evaluation purposes. A range of contract cost estimation methodologies have already evolved to predict software costs in the procurement planning process since the detailed system design activities which could produce this information as part of the normal software engineering lifecycle will not have commenced. At this stage, cost estimation is normally independently undertaken by the customer and all potential suppliers, with varying degrees of competence. A supplier with poor estimation skills may win a contract over a more

capable supplier because the buyer's VfM analysis does not recognise that the low offer has resulted from a faulty estimation process. Estimation methodologies follow rational processes but VfM analyses are not as well described and do not necessarily allocate an appropriately increased risk component to negate poor estimates. Besides the negative effects of a poor contract on the viability of the supplier's and the customer's budget, there is the additional opportunity cost of failing to select the best supplier.

These dilemmas, along with varying computations of individual requirement costs by estimators, buyers, software developers, and project managers, mean that there can be a lack of accord between all the potential parties in regard to the price of the software development.

This research will develop principles to connect a number of existing academic and practitioner research outcomes. The operational aim is to develop a comprehensive and rigorous representation of software costs that is consistent with all three of the following approaches: in procurement, southernSCOPE method (Multimedia Victoria, 2003); in project management, Work Breakdown Structure and Earned Value Management; and in software metrics, Function Point Analysis and COCOMO model. These principles will form the basis of a new methodology that will facilitate a reduction in the cost of software development through greater reliability in cost estimation. This is particularly valuable in the case of multi-million dollar, complex, and mission-critical software systems used by organizations such as departments of Defence.

Public procurement is often contentious, especially in the complex and costly area of military procurement. For example, the purchase by the Australian Department of Defence, at AUD$4.5 billion, of the Collins Class Submarines resulted in extensive problems, including significant and irreparable flaws in the combat system software (COA, 1999). Defence officials find themselves required to, on the one hand "purchase products that are not yet designed or for which production experience is lacking and at prices for which there is little precedent" (Weidenbaum, 1998, p. 76) while, on the other hand, are being told that there is a possibility that "procurement 'drives' strategy rather than serving it" (Etzioni, 1984, p. 29). The Australian Defence Procurement Review (2003) implicitly acknowledged some of these challenges by establishing Defence Materiel Organisation (DMO) as an agency that is

financially autonomous from the Department of Defence (DoD) with a requirement to become more business-focused and outcomes-driven.

Early functional techniques for capturing software requirements have been largely superseded by Use Case based techniques (Jalote, 2002), forcing an evolution of cost estimation methodologies without solving their underlying shortcomings. This project aims to build upon and extend the existing techniques for costing contemporary software systems to develop a link between cost metrics at the requirements definition phase and the Earned Value of completed requirements during the development phase of the project. Analysis of the scholarly literature, supported by interviews with a number of leading practitioners, indicates that this link has not yet been made. The result is a chronic inability to consistently and reliably determine the ongoing value of the developed system against initial requirements.

As many large organizations, including the DoD, have outsourced their software development, a common view of the initial and ongoing value of requirements for procurement specialists, software developers, and project managers will contribute to a reduction in budget blowouts and contract conflict. This will improve VfM outcomes for software systems.

Specifically, this research aims to:

1. Investigate existing cost estimation techniques and procurement methods for software development arising from Use Case descriptions (see below for description of Use Case);
2. Develop an additional level of Use Case decomposition that gives detailed information on system functionality, ensures that no cost elements are missed or duplicated, and is either in or can be converted to Work Breakdown Structure format (see below for description of Work Breakdown Structure);
3. Apply or develop costing techniques for the additional Use Case decomposition level that are suitable for a Work Breakdown Structure format, which will allow the Earned Value, as discussed above, to be calculated against delivered requirements; and
4. Apply this research to the resolution of the shortcomings in procurement approaches to multi-million dollar software systems

to ensure that expenditure on software development will become a greater VfM proposition.

BACKGROUND TO THE METHODOLOGY

Several concepts are drawn from the fields of software development, software cost estimation, and project management to provide the foundation for this project. First, the concept of Use Cases, formalised by Jacobson (1992), provides a popular technique for capturing requirements for object-oriented software analysis and design. A Use Case is commonly described as a sequence of actions an IT system performs that yields an observable result of value to a particular Actor. An Actor is someone or something outside the system that interacts with the system. Use Cases capture who (Actor) does what (Interaction) with the system for what purpose (Goal), without dealing with system internals. A complete set of Use Cases specifies all the different ways to use the system and behaviour required of the system, thus bounding its scope.

Second, Function Point Analysis (FPA) represents a commonly used method for undertaking cost estimation technique of software. FPA was first published in 1979. "In 1984, the International Function Point Users Group (IFPUG) was set up to clarify the rules, set standards, and promote their use and evolution. FPA provides a standardised methodology for measuring the various functions of a software application. FPA measures functionality from the user's point of view, that is, on the basis of what the user requests and receives in return" (Abran & Robilliard, 1996, p. 895). ISO/IEC 14143-1 (1998, p. 1) "defines the fundamental concepts of Functional Size Measurement (FSM) and describes the general principles for applying an FSM Method". It "does not provide detailed rules on how to measure Functional Size of software using a particular method, use the results obtained from a particular Method, or select a particular Method". IFPUG 4.1 Unadjusted (ISO/IEC 20926) and COSMIC FFP (ISO/IEC 19761), from the COmmon Software Measurement Consortium, are two FSM methods conforming to ISO/IEC 14143 that have been adapted to be compatible with Use Case techniques. Both of these methods contribute to the International Software Benchmarking Standards Group's (2003) repositories of software project metrics.

Third, Karner (1993) modifies the function point approach of Albrecht (1979), to propose the concept of Use Case Points. Karner's (1993) approach has contributed to an intermediate elaboration of the costing problem without adding to its specificity. It fails to enhance contemporary approaches to costing software development, although it hints at further developmental opportunities.

Fourth, the Work Breakdown Structure (WBS) is a key component of project management methodologies, as evidenced by the PMI's (2001) production of a WBS Standard. The DMO (2003) strongly supports the application of effective project management principles and states that in line with Activity Based Costing and Earned Value principles there should be no cost outside of the WBS.

THE PROBLEM DEFINITION

Research undertaken by the investigators, to date, has uncovered little evidence that the academic community has scientifically investigated the link between the costs associated with Use Cases and those costs used to represent requirements in WBSs. A common approach is to directly map Use Cases into the WBS but the investigators argue that this is not valid for large projects, particularly if there are significant numbers of 'include' and 'extend' Use Cases. 'Include' indicates a relationship between two Use Cases and 'extend' allows capture of a variant to a Use Case. In these situations there is a many-to-many relationship between Use Cases and functionality, even for simple functionality.

Software estimation techniques vary from those using estimated lines of code, such as Constructive Cost Model (COCOMO) (Boehm, 1981; Boehm *et al*., 2000) through to those using functions or modules (such as FPA). The fact that ISO/IEC has approved four separate functional size counting methods standards, because it cannot currently select the most reliable, demonstrates the lack of global agreement on the effectiveness of current techniques. This combined with the necessity for project managers to map Use Cases into a WBS to enable them to report upon cost and schedule status highlights some of the difficulties in understanding software costs. Woodings' (2001) unpublished research on software metrics indicates that requirements or scope creep have a large influence on project cost and schedule, further illustrating the

complexities of software projects. The problem may not be that software projects exceed budget, but rather a lack of understanding of the costs from the outset.

Function Points (FPs) and WBSs are both based upon functional requirements, and although both Use Cases and FPs view a system from a user perspective, Use Cases are deliberately not functionally decomposed. The estimation of 'functions' from Use Cases is therefore fraught with problems. One of the problems with specifying requirements as use cases is that many of the functional requirements are implied, or even worse, hidden. It is necessary to flush out the detailed requirements to enable accurate estimation. COSMIC FFP has moved some way towards sizing use cases but typically the available information will be insufficient for the desired accuracy in sizing. The COSMIC FFP measurement method "involves applying a set of models, rules and procedures to a given piece of software as it is perceived from the perspective of its Functional User Requirements" (COSMIC, 2003, p. 15).

The term Functional User Requirements (FUR) "is a sub-set of the user requirements. The FUR represent the user practices and procedures that the software must perform to fulfil the user's needs" (COSMIC, 2004, p. 9).

Typically only identification of the COSMIC processes is possible, and these are given a complexity of low, average or high. It is possible that a use case may map to one or more COSMIC processes. It may also be necessary for the use case to be detailed to a level that would also allow the identification of entries and exits (Morris, 2004). An 'Entry' Type is "a data movement that moves a data group type from a user across the boundary into the functional process type where it is required" and an 'Exit' Type is "a data movement that moves a data group type from a functional process type across the boundary to the user that requires it" (COSMIC, 2003, 9).

Consider a use case that modifies addresses of clients as an example of these problems. This might state: "The actor will enter the new address. The address will be validated and saved."

Hidden in this use case is the implicit requirement that the suburb field will need some way of reading a list of suburbs to allow the field to auto-complete. The address will need to be checked against a list of

approved addresses (in Australia a Data Point Identifier has been issued by Australia Post for every address). This could be a new piece of software which requires a new series of entries and exits.

What is generally missing from existing techniques is a mapping layer that maps the Use Cases to all of the functional requirements. This second layer of functional decomposition of Use Cases could provide the basis for FP counting and also map directly into the WBS. It the context of COSMIC FFP the research would be focussed upon the principles for extracting FURs from use cases to be passed into the COSMIC FFP Mapping Phase in a manner that is consistent with a WBS. The Mapping Phase "takes as input a statement of FUR of a piece of software, …, and, using a defined set of rules and procedures, produces a specific software model … suitable for measuring functional size" (COSMIC, 2003, p. 18).

This approach would allow the complexity of a Use Case to be described rather than implied, to map elements of disparate Use Cases to single WBS elements, and ensure all functional areas are counted. It would provide a more comprehensive cost breakdown at a level understood by the procurers, project managers, and software engineers. Thus, Earned Value of requirements could be determined and the VfM implicit in the procurement of software development would become more explicit. Figure 2 below illustrates how the proposed mapping layer could resolve the many-to-many relationships between uses cases and functionality, producing a WBS that is logical and complete. The example shows a sub-set of elements of a rates processing system.

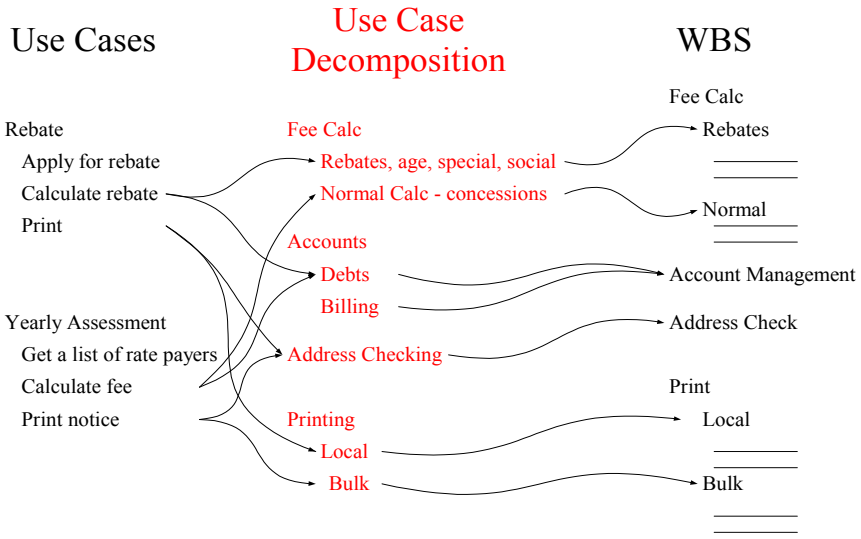
PROPOSED METHODOLOGY

In terms of the aims of this research and the background to the research problem, the researchers have developed a six-stage methodology centered around a "collective case study" approach (Stake, 2000, p. 437), with triangulation being developed on the basis of results arising from the literature review, the deconstruction of a number of small cases, and concept testing of static and live data.

Stage 1: Initial Project Scoping and Literature Review

A preliminary literature review has been completed to ensure the novelty of this proposed research. However a much wider, global search

FIGURE 2
Mapping Example



of the literature is being undertaken to uncover the range of concepts relevant to the research across the discipline areas of software engineering, project management, accounting, and procurement. The preliminary review has revealed that much of the work in this field is more likely to be found in "textbooks, lectures, and laboratory exercises" (Kuhn, 1970, p. 43), which reflects a general pursuit of solutions in this field by practitioners in the absence of active involvement of scholars (for example, Glass, 1998). Thus the contributions of Hines, et al. (2000) is typical of the literature guiding the second and third stages of the research.

Stage 2: Search for Project Exemplars

This area has been identified as a separate part of the methodology because the existence of exemplars is not readily apparent in the public

domain. However, the researchers have a number of sources which are prepared to provide data to permit concept-building to take place within the objectives of the project. The exemplars will provide basic model-building data.

Stage 3: Refinement of the Problem Definition and Prototype Development

Data from exemplars and the completed literature review will permit elaboration and refinement of the research aims and structure. This stage will result in the deconstruction of the potential cost metrics embedded in the exemplars. A range of interrogative techniques will be used to re-evaluate the links between pre-contract costing metrics and post-contract monitoring mechanisms. A prototype Earned Value model focused upon software requirements is the anticipated result.

Stage 4: Development and Administration of Tests Sets to the Earned Value Model

Using the prototype developed in Stage 3, the researchers will develop a series of case studies to test the consequences of its application on real project data, recording and reporting on exceptions arising and generating supporting instructions for the application of the model.

Stage 5: Live Testing Against a Completed Software Project

A proof of concept testing and review on a completed project will be independently conducted by parties not involved in the research.

Stage 6: Live Predictive Testing on a Forthcoming Software Development Project

This step replicates Stage 5 and provides a predictive test. This part of the research is highly speculative, but as a consequence of the interest already generated by this proposal, it is anticipated that a number of part projects will be available to the researchers to enable live testing against current methodologies for testing reliability, capacity for generalisation, and 'value in use'.

Stage 7: Completion of Results and Dissemination and Elaboration of Future Related Research

Results will be progressively published throughout the project and during this final stage the researchers will develop appropriate documentation to support the Earned Value methodology for software development projects. Further outcomes may include specifying possible software systems to support the model and the dissemination of the methodology within areas of sophisticated procurement cost estimation, unrelated to software development.

CONCLUSION

The sizeable impact of procurement practices upon organizational operational and financial performance is only beginning to be understood, as evidenced by the extensive, non-standard public and private sector procurement reform activities in progress around the world. However procurement is not supported by established professional practice, particularly for difficult procurement such as complex software development.

A significant number of budget problems in software projects originate in the early phases of the procurement process. When the extent of these budget excesses is examined it is clear that this is an area in urgent need of attention. This paper presents an outline of some original research designed to better conceptualize and structure the costing and project management of unique, software development. The outcomes are designed to assist the evaluation and management of procurement activities associated with complex, bespoke projects and to ensure that this type of measurement is seen as a procurement, rather than general, management responsibility

While the research outcomes will be published during the period of the research, the purpose of publication, at this stage, is to encourage debate among academics and practitioners engaged in the discipline areas of procurement, software engineering, and costing. The research problem represents a neglected area of research, perhaps because of its multi-disciplinary nature. A number of disciplines have tackled elements of the problem, but inconsistent application of current techniques have created divergent views on their accuracy and usefulness. As both scholarly and more populist literature has shown, software development projects have a reputation for failing to meet budgetary, timetable, and quality expectations. Here is an opportunity to apply a range of multi-

disciplinary energy and knowledge to resolve some of these problems, provide some new insights into the larger VfM debate and contribute to improved organizational performance.

REFERENCES

Abran, A., & Robilliard, P.N. (1996). Function point analysis: An empirical study of its measurement and processes. *IEEE Transactions on Software Engineering, 22* (12), 895-910.

Albrecht, A.J. (1979, October). Measuring application development productivity. *Proceedings of the IBM Application Development Symposium*. Monterey, CA, 83-92.

Anderson, J.C., & Narus, D.J. (1998). Business marketing: Understanding what customers value. *Harvard Business Review, 77* (6), 53-63.

Auditor-General of Victoria. (2003). Implementation of RMIT University's Academic Management System. In *Report on Public Sector Agencies* (pp. 58-88). Melbourne: Auditor General of Victoria.

Back, W.E., & Maxwell, D.A. (2000). Activity-based costing as a tool for process improvement evaluations. *Journal of Management in Engineering, 16* (2), 44-58.

Boehm, B.W. (1981). *Software Engineering Economics*. Englewood Cliffs, NJ: Prentice Hall.

Boehm, B.W., Horowitz, E., Madachy, R., Reifer, D., Clark, B.K., Steece, B., Brown, A.W., Chulani, S., & Abts, C. (2000). *Software cost estimation with Cocomo II*. Englewood Cliffs, NJ: Prentice Hall.

Bowman, C., & Ambrosini, V. (2000). Value creation versus value capture: Towards a coherent definition of value and strategy. *British Journal of Management, 11* (1), 1-15.

Brandt, M.T., Levine, S.P., & Gourdous, J.R. (1999). Application of activity-based cost management. *Professional Safety, 44* (1), 22-27.

Callender, G., & Johnston, J. (2002). Best value approaches to infrastructure management: For the good of the shareholders? In

Delener, N. and Chao, C-N. (Eds.). *Beyond the Boundaries* (pp. 162-170). New York, NY: GBATA.

Childerhouse, P., & Towill, D.R. (2002). Analysis of the factors affecting real-world value stream performance. *International Journal of Production Research, 40* (15), 3499-3519.

Clegg, S., & Hardy, C. (1996). Organizations, organization and organizing. In Clegg, S., Hardy, C., & Nord, W. (Eds.). *Handbook of Organisational Studies* (pp. 1-28). Thousand Oaks, CA: Sage.

Common Software Measurement International Consortium (COSMIC) (2003) COSMIC FFP Measurement Manual Version 2.2. [Online]. Available: http://www.lrgl.uqam.ca/cosmic-ffp/manual.jsp. (Accessed September 27, 2004)

Commonwealth of Australia (COA) (1999). *Report to the Minister of Defence on the Collins Class Submarine and Related Matters*. [Online]. Available: http://www.minister.defence.gov.au/1999/collins.html. (Accessed October 3, 2004).

Defence Materiel Organisation. (DMO) (2003). *ASDEFCON Strategic Materiel*. [Online]. Available: http://www.defence.gov.au/dmo/lsd/asdefcon/asdefcon.cfm. (Accessed November 10, 2003).

Defence Procurement Review. (2003). *Defence Procurement Review 2003*. [Online]. Available: www.defence.gov.au/publications/dpr180903. (Accessed November 10, 2003).

Etzioni, A. (1984, Fall). Do defence contractors map our military strategy? *Business and Society Review, 51*, 29-34.

Fleming, Q.W., & Hoppelman, J.M. (1996). *Earned value project management*. Upper Darby, PN: Project Management Institute.

Gantt, H. (1919). *Organising for work*. New York, NY: Harcourt, Brace and Howe.

Glass, R.L. (1998). *Software runaways - Lessons learned from massive software project failures*. Upper Saddle River, NJ: Prentice Hall.

Gourvish, T. (1970). Captain Mark Huish: A pioneer in the development of railway management. *Business History, 12* (1), 46-58.

Hines, P., Lamming, R., Jones, D., Cousins, P., & Rich, N. (2000). *Value stream management: Strategy and excellence in the supply chain*. London: Prentice Hall.

Hodge, G. (1996). *Contracting-out government services: A review of international evidence*. Melbourne: Montech Pty Ltd.

ILO. (1978). *Introduction to work study* (3rd ed.). Geneva: International Labour Office.

IRSPP. (2003). *International Research Study of Public Procurement*. Budapest.

International Software Benchmarking Standards Group. (2003). [Online]. Available: http://www.isbsg.org/html/index2.html. (Accessed November 1, 2003).

ISO-IEC. (1998). *International ISO/IEC Standard 14143-1, Information Technology - Software measurement - Functional size, Part 1: Definition of Concepts*.

Jacobson, I. (1992). *Object-oriented software engineering - A use case driven approach*. Reading, MA: Addison-Wesley Publishing.

Jalote, P. (2002). *Software project management in practice*. Boston, MA: Addison-Wesley.

Kahnement, D., & Wakker, P. (1997). Back to Bentham: Explorations of experienced utility. *Quarterly Journal of Economics, 112* (2), 375-406.

Karner, G. (1993). *Use Case points - Resource estimation for objectory projects*. MSc thesis.

Kuhn, T. (1970). The structure of scientific revolutions. *International Encyclopaedia of Unified Science II.* Chicago, IL: The University of Chicago Press.

Lasiter, T. (2001). Balanced sourcing the Honda way. In Woods, J., & Marien, E. (2001). *The Supply Chain Yearbook: 2001 Edition* (pp. 205-213). New York, NY: McGraw-Hill.

Lederer, L., & Prasad, J. (1995). Causes of inaccurate software development cost estimates. *Journal of Systems Software, 31*, 125-134.

McGuckin, R., & Stiroh, K. (1998). Computers can accelerate productivity growth. *Issues in Science and Technology, 14* (4), 41-48.

Morris, P., Total Metrics (2004, October 8). Personal Communication.

MSD. (1965). *The practice of O & M*. London: Her Majesty's Stationery Office.

Multimedia Victoria. (2003). *SouthernSCOPE*. [Online]. Available: www.mmv.vic.gov.au/southernscope. (Accessed November 1, 2003).

OECD Public Management Policy Brief. (2001). The hidden threat to e-government - Avoiding large government IT failures. Paris: OECD.

Office of Government Commerce. (OGC) (2004). *Successful delivery toolkit, contract management*. [Online]. Available: www.ogc.gov.uk/sdtoolkit/workbooks/contracts/index. (Accessed June 15, 2004).

Parasuraman A., Zeithaml, V.A., & Berry, L.L. (1988). SERVQUAL: A multiple-scale for measuring consumer perceptions of service quality. *Journal of Retailing, 64*, 12-40.

Project Management Institute. (PMI) (2001). PMI *Practice for work breakdown structures - Excerpts*. [Online]. Available: http://www.pmi.org/info/search_result.asp?query=work+breakdown+structure&image1.x=24&image1.y=6. (Accessed November 1, 2003).

Project Management Institute (PMI) (2000) *A Guide to the Project Management Body of Knowledge PMBOK Guide 2000 Edition*. [Online]. Available: http://www.pmi.org/info/search_result.asp?query=earned+value+management&image1.x=20&image1.y=7. (Accessed October 2, 2004).

Rolt, L. (1970). *Isambard Kingdom Brunel*. London: Penguin Books Ltd.

Stake, R. (2000). *Case studies*. In Denzin, N., & Lincoln, Y. (Eds.). *Handbook of Qualitative Research* (2nd ed.). London: Sage.

Standish Group, The. (1995). *The Standish Group Report - CHAOS*. [Online]. Available: http://spinroot.com/spin/Doc/course/Standish_Survey.htm. (Accessed October 15, 2003).

State Supply Commission. (2002). *The Contract Process Guidelines*. [Online]. Available: www.ssc.wa.gov.au/publications.asp?id=63. (Accessed March 20, 2004).

Suwardy, T., Ratnatunga, S., & Speight, G. (2003). IT projects: evaluation, outcomes and impediments. *Benchmarking: An International Journal, 10* (4), 325-342.

Taylor, F. (1911). *The principles of scientific management*. New York, NY: Harper and Brothers Publishing.

Weidenbaum, M. (1998). Government as a market. *Business and Society Review, 100/101*, 75-85.

Woodings, T. (2001). *Increase in workload under requirements creep*. Personal communication, September 2003.