The copyright of this thesis vests in the author. No quotation from it or information derived from it is to be published without full acknowledgement of the source. The thesis is to be used for private study or non-commercial research purposes only.

Published by the University of Cape Town (UCT) in terms of the non-exclusive license granted to UCT by the author.

University of Cape Town

Managing economic value and uncertainty on software projects; an empirical study with the CASSE Framework

By

Joseph Kibombo Balikuddembe

In partial fulfilment of the requirements for the degree of

Doctor of Philosophy in Computer Science, Faculty of Science - University of Cape Town

Advised by

Dr. Antoine Bagula

Department of Computer Science, University of Cape Town

University of Cape Town

December 2009

Declaration

This thesis is a presentation of my original research work. Wherever contributions of others are involved, every effort is made to indicate this clearly, with due reference to the literature, and acknowledgement of collaborative research and discussions. This work was done under the guidance of Dr. Antoine Bagula, at the Department of Computer Science, University of Cape Town.

Joseph Kibombo Balikuddembe

Signature:_____

Date:_____

In my capacity as supervisor of the candidate's thesis, I certify that the above statements are true to the best of my knowledge.

Dr. Antoine Bagula

Signature:_____

Date:_____

University of Cape Town

Acknowledgments

I would like to thank the numerous people who have been very supportive during my research. My utmost gratitude goes to my supervisor Dr. Antoine Bagula for his constant criticism, which has led to a better research product. I would also like to thank my colleagues at work for their support in this research.

My sincere thanks go to the management of Complex Adaptive Systems (Pty) Ltd and Cell-life for their cooperation and for granting me the opportunity to use their organisational business and project data.

I also want to thank all the people who participated in the survey.

My sincere thanks specifically go to Dr. Anet E. Portgieter for attracting me to this research area in Software Engineering and for the interesting discussions, financial support and inspirations given during the inception stages of this research. I am indeed very grateful.

I owe my utmost gratitude to Professor Michael Kyobe for the quality assurance and criticism given throughout this project. Thank you very much indeed.

During this work I have collaborated with many colleagues for whom I have great regard, and I wish to extend my warmest thanks to all those who have helped me with my work. For example, Isaac, Kurt, Nemanja, Werner; without your technical help, it would have been extremely difficult to reach this far.

I owe my loving thanks to my wife Dr. Barbara Ndagire and my son Anthony. They have lost a lot due to my research abroad. Without their encouragement and understanding it would have been impossible for me to finish this work. My special gratitude is due to my mother Josephine, my brothers Emmanuel and Charles and my sisters, Pacis, Anita, Josephine and Triza together with their families for their loving support. My loving thanks are due the Ugandan friends in Cape Town, Kate inclusive, for you have let me own a happy family in Cape Town.

It has been a pleasure to work with you all.

- Joseph Kibombo Balikuddembe

University of Cape Town

Abstract

Lack of adaptive-predictor models in software development renders the decision-making process complex, principally when evaluating investment options. Prior work has presented various approaches which are still non-integrated, requiring rigour to embrace overall economic value and uncertainty management.

In this study, a Complex Adaptive Software Engineering Framework which uses the Actor Object Dependency Analysis (AOD) technique is proposed as a feasible option. It is aimed at providing a useful technique for monitoring and controlling value propositions in terms of cost, schedule, and progress of complex issues in software development. Specifically, examination was made on how:

- AODs and their properties can enhance the estimation of the probability distribution of losses and earnings on software project portfolio;
- economic concepts, models and tools can help advance our understanding and improvement of the development of software and process that produce them; and also
- if the project selection and requirements engineering processes employed in industry today align appropriately with organizational business strategies of supporting value creation on software projects.

A business and process view dimension were formulated to test this phenomenon. Empirical data tests from industry revealed that overall predictions of economic value is possible in the solution domain and also planning and control of software projects can greatly be enhanced with use of such value-oriented approaches. Secondly, the ranking metric utilized therein is fundamentally important in communicating economic value to stakeholder as project unit costs change during the project lifetime; thereby empowering the portfolio office to decide which projects to initiate, which on-going projects to continue to fund, and which projects to terminate. On the other hand, the process view dimension revealed that the need to realign engineering and project selection processes with business objectives is still evident. In instances where business objectives are not well integrated into the processes that produce the software, certain business elements may be sacrificed, such as productivity measurement, quality and risk detection in business selection and overall management.

These results thus reinforced the view that it is a significant business security to exploit adequate project selection techniques that increase the rate of successful yet profitable projects.

University of Cape Town

Extended Abstract

The current market forces in the software development business today require that software vendors select only those projects increasing competitive advantage over their competitors. Thus, an opportunity cost decision of selecting one project against the other is dependent on how best a given project meets the business goals and the overall company's competitive strategy. Accordingly, the software project value differentiator in a market place is immensely dependent on the process engineering approaches utilized to optimize schedule, cost and budget constraints on the project – while at the same time ensuring that at project completion, the stakeholder benefit is realized.

However, viable project selection is still proving to be complex. It is still difficult to determine the probability distribution of profits and losses on software projects before investment commitment is made on the projects. Equally, determining realistic implementation time estimates at project inception while sizing with system functional points is still a major challenge in the software industry. Although prior work presents software development lifecycle and business decision models in effort estimation and product release planning using feature dependency analysis, economic models such as these cannot be created and assessed in a single step. Rather, they must be refined and improved over time by incorporating new techniques and approaches that make the models more extensible. The use of emergent engineering approaches as a fundamental decision making guideline in analyzing project value has not been largely integrated in these models.

The goal of this research is to develop a unifying Framework for providing useful techniques for monitoring and controlling the cost, schedule, and progress of complex issues in software development. This work draws from a value-based context while at the same time taking into consideration the emergent engineering principles.

This work answers the following research questions.

1. Based on Actor Object Dependency Analysis (AODA) and their properties, can we estimate the probability distribution of losses and earnings that can be incurred by a software vendor on any project given his/her software project portfolio?
2. Can economic concepts, models and tools help to advance our understanding and improvement of the development of software and process that produce them?
3. In particular, do the project selection and requirements engineering processes employed in industry today align appropriately with organizational business strategies of

University of Cape Town

supporting value creation on software projects?

While using both the qualitative and quantitative research design, this work draws on principles from economics as well as computer science to develop new predictors in software development.

To evaluate the validity of the CASSE engineering approach, case study evaluations of this approach in industry were undertaken with two software vendors in Cape Town and a leading Non Government Organisation developing mobile health systems. Results obtained from these case study assessments assisted with answering the first two research questions. These results were presented and measured against the company baselines.

It was observed that overall predictions of economic value is possible in the solution domain and also planning and control of software projects can greatly be enhanced with use of such value-oriented approaches. Secondly, the ranking metric utilized therein is fundamentally important in communicating economic value to stakeholder as project unit costs change during the project lifetime; thereby empowering the portfolio office to decide which projects to initiate, which on-going projects to continue to fund, and which projects to terminate. On the other hand, the survey undertaken to answer the third research question revealed that the need to realign engineering and project selection processes with business objectives is still evident. In instances where business objectives are not well integrated into the processes that produce the software, certain business elements may be sacrificed, such as productivity measurement, quality and risk detection in business selection and overall management.

This work contributes to knowledge in the following ways.

- The overall contribution is in demonstrating that alternative engineering approaches that analyse Actor Object Dependencies (AOD) can enhance the way we select software development projects as well as schedule planning so as to increase value derivation on projects. Available literature suggests that previous work in this area has firstly approached such value related issues in isolation, not embracing the overall value management on software projects. Secondly, most studies seem to have largely concentred on optimising development efforts for release planning purposes but not overall value management as actual development time is uncured. Consequently, this renders the AOD technique as an alternative engineering approach, capable of handling uncertainty in project selection as well as overall value prediction and management over time.

University of Cape Town

- This study results in formulating and analysing the performance of a Complex Adaptive System Software Engineering Framework (CASSE). This framework implements the AODs technique as a mechanism to predict properties of the finished software system before the expense of software development is incurred. This in turn supports the evaluation of favourable, yet risk-balanced operating points on these projects; thereby improving the decision making process significantly.

University of Cape Town

List of Acronyms

ASD: Adaptive Software Development

ARV*:* Antiretroviral

ART*:* Antiretroviral Treatment

AO*:* Actor Object

AOD*:* Actor Object Dependency

AOOS*:* Actor Object Operation Structures

CASSE*:* Complex Adaptive Systems Software Engineering

CPM: Critical Path Method

FP: Functional Points

GNU: General Public License

iDART: Intelligent Dispensing of Antiretroviral Treatment

NGO: Non Government Organization

MDL: Minimum Description Length

OOSAD: Object Oriented Systems Analysis and Design

PPM: Project Portfolio Management

RFP: Request For Proposal

SPIN: Software Process Improvement Network

UML: Universal Modeling Language

VBSE: Value-Based Software Engineering

XML: Extensible Markup Language

University of Cape Town

List of Figures

Figure 1: Value ladder 7
Figure 2: Methodology flow chat 13
Figure 3: Technical inputs to the business strategy 18
Figure 4: Research question flow chat 23
Figure 4: Project profit triangle 26
Figure 6: value-based software development transformation process 36
Figure 7: NB Model example 53
Figure 8: Association relationship 58
Figure 9: Generalization relationship 59
Figure 10: Aggregation relationship 59
Figure 11: Composition relationship 60
Figure 12: Actor Object relationships 60
Figure 13: Explicit node dependency 62
Figure 14: Resulting implicit dependency 64
Figure 15: Actor Object model 66
Figure 16: Resulting model 68
Figure 17: Model beliefs when evidence is set 69
Figure 18: Framework mode 78
Figure 19: Project value evaluator 81
Figure 20: Algorithm flow in the framework 83
Figure 21: Added time implications 92
Figure 22: RFP project mapping 93
Figure 23: Profitability/AO graph 94
Figure 24: Threshold implications 95
Figure 25: Hourly project difference 96
Figure 26: Quadrant shifting 97
Figure 27: iDART Logical Model 99
Figure 28:iDARD AOD model 100
Figure 29: Common AOD scenario 103
Figure 30: Bird's-eye view of analysis decomposition levels 107
Figure 31: Use-case diagram 108
Figure 32: Object interaction calls of user actions 110
Figure 33: Actor Object Logical Model 111
Figure 34: Implicit model 111
Figure 35: Data Proxy Actor Object graphical model 115
Figure 36:A1 Application Actor Object graphical model 116
Figure 37: Library Application Actor Object graphical model 117
Figure 38: Project evaluator - application shifting patterns 118
Figure 39: Budget evaluation 119
Figure 40: Pattern generation on project schedules 122
Figure 41 Software Development Project example 124
Figure 42: Explicit dependency model 127
Figure 43 Task attributes 128
Figure 44: Task explicit model 129
Figure 45: Task strengths in the model 131
Figure 46: Task project budget assessment 133

University of Cape Town

Figure 47: Project acceptance pattern 136
Figure 48: Project 2 acceptance 137
Figure 49: Completion rate analysis for project 1 138
Figure 50: Completion analysis for project 2 139
Figure 51: Profitability on project 1 140
Figure 52: Project 2 profitability analysis 141
Figure 53: Targeted Sample Characteristics 144
Figure 54: Risk detection in requirements engineering 146
Figure 55: Project selection basis 149
Figure 56: Business strategy formation 170
Figure 57: Strategic requirement engineering 171
Figure 58: Project selection parameters 172

University of Cape Town

List of Tables

Table 1: Operation structure matrix....65
Table 2: Generated matrix example....67
Table 3: Project viability status....91
Table 4: AO value matrix in the iDART system....102
Table 5: System rule....109
Table 6: AO dependencies....112
Table 7: Task dependency relationships....126
Table 8: Assertions in requirements engineering....146
Table 9: Assertions in project selection....148
Table 10: Project choice & development procedure....152
Table 11: No dependency assessment undertaken & selection based on strategic business objectives....154
Table 12: No dependency assessment undertaken & reliance on feasibility assessment reports....156
Table 13: Standardized process utilization & analytical tools usage....158
Table 14: Systems modeling & projects being in-house....160
Table 15: R&D Projects & traceability formulation importance....162
Table 16: Standardized process & UML modeling....164
Table 17: Standardized procedure &selection based on profitability levels....166
Table 18: Assessment for risk & analytical tool usage....168
Table 19: Key for trends` scale....169
Table 20: Key parameters of interest....170

University of Cape Town

Publications

- Joseph Kibombo Balikuddembe, Antoine Bagula. **Business viability assessment of potential software projects: an empirical study with the CASSE Framework**. Paper accepted for the 2009 International Conference on Advanced Software Engineering & Its Applications (ASEA 2009) , December 10 ~ 12, Jeju, Jeju Island, Korea.
- Joseph Kibombo Balikuddembe, Antoine Bagula. **Aligning the software project selection process with the business strategy; a pilot study**. Paper accepted for the 2009 International Conference on Advanced Software Engineering & Its Applications (ASEA 2009) , December 10 ~ 12, Jeju, Jeju Island, Korea.
- Joseph Kibombo Balikuddembe, Antoine Bagula. **Aligning the software project selection process with the business strategy: a pilot study – extended version.** To appear in the International Journal of Software Engineering and Its Applications, 2009.
- J.K. Balikuddembe, A. Bagula**. Requirements negotiation on software projects; a value-based Actor Object Dependency Analysis Technique.** Proceedings of the 4th IFIP TC2 Central and Eastern European Conference on Software Engineering Techniques, Krakow, Poland 2009.
- J.K. Balikuddembe, A. Bagula, U. Rivets. **Complexity management of customization requests on software products; A value-based Approach.** Proceedings of the 2009 International Conference on Software Technology and Engineering (ICSTE 2009), Chennai, India. June 27-29, 2009.
- J.K. Balikuddembe, A. Bagula, Isaac.O.Osunmakinde, **Software Project Profitability Analysis Using Temporal Probabilistic Reasoning; an Empirical Study with the CASSE Framework**. Advances in Security Technology - Communications in Computer and Information, Springer Berlin Heidelberg, 2009, pp.138-150
- J.K. Balikuddembe, A.E. Potgieter, Isaac.O.Osunmakinde: **Software profitability analysis using temporal probabilistic reasoning.** Proceedings of the 2008 Advanced Software Engineering and Its Applications. pp.99-102 (2008)
- J.K. Balikuddembe and A.E. Potgieter: **Using Actor Object Operations Structures to Understand Project Requirements Complexities.** Proceedings of the 2008 The Third International Conference on Software Engineering Advances. pp. 290-294 (2008)
- J.K. Balikuddembe and A.E. Potgieter: **Predicting Value-Creation in Software Project Complexity**, Proceeding of the 10th IASTED International Conference on Software Engineering and Applications **–** Dallas, 2006

University of Cape Town

Table of Contents

DECLARATION ..... **I**
ACKNOWLEDGMENTS ..... **II**
ABSTRACT ..... **III**
EXTENDED ABSTRACT ..... **IV**
LIST OF ACRONYMS ..... **VII**
LIST OF TABLES ..... **X**
PUBLICATIONS ..... **XI**
TABLE OF CONTENTS ..... **XII**
INTRODUCTION AND BACKGROUND TO THE STUDY **1**
1.1 Problem definition 1
1.2 Motivation 3
1.3. Rationale 5
1.4 Study Contribution and Thesis Outline 6
1.5 Differentiation factor of the CASSE Framework 8
RESEARCH DESIGN AND METHODOLOGY **11**
2.1 Introduction 11
2.2 Research strategy 11
2.3 Rationale for using both process and the business dimensions 14
2.4 Business View Investigation Dimension 15
2.4.1 Case Selection *15*
2.4.2. Motivation for selecting the various case studies *16*
2.4.3. Case study protocol *18*
2.5 Process view investigation dimension 18
2.5.1. Background *18*
2.5.2 Survey *19*
2.5.3 Variables *19*
2.5.4 Data Collection *20*
2.5.5. Data Collection Procedures *22*
2.5.6 Data analysis *22*
2.6. Design Science approach 23
2.7. Approach to answering the research questions 23
2.8. Validity of study results 24
2.9. Data presentation 24
DETERMINING AND SUSTAINING PROJECT VALUE ON SOFTWARE PROJECTS **25**
3.1 Introduction 25
3.2. Choosing valuable projects 25
3.3 Value drivers within software development 30
3.3.1 Organisational performance *30*
3.2.2 Growth and competition *30*
3.3.3 Strategy and strategic alignment *31*
3.4 Agile paradigm 32
3.4.1 Traditional approaches to software development *32*
3.4.2 The difference with agility *33*
3.4.3 Rationale for taking an agile approach *33*
3.4.4 Software development agility – a determining factor for success *34*
3.4.5. How CASSE Compares to SCRUM methodology *35*
3.5 The sustainable chain of value creation on software projects 36

University of Cape Town

3.5.1 Understanding economics of software development *37*
3.5.2 Leveraging value on software projects *39*
3.5.3 Appropriate techniques *46*
3.6. Baseline strategy 54
3.6.1 Industrial trends *54*
3.6.2 The business need *55*
3.6.3 The research gap *55*

THE CASSE FRAMEWORK ENGINEERING APPROACH 57

4.1 Engineering Approach 58
4.1.1 Background to deriving Actor Object dependencies *58*
4.1.2 Realizing the estimative mechanism *61*
4.1.3 Modelling facets *62*
4.2 Key facets/aspects in the approach 70
4.2.1 Project profitability function *70*
4.2.2 AO properties as inference attributes *72*
4.2.3 Analyzing AO causal dependencies in the model *73*
4.2.4 Calculating the critical path *74*
4.2.5 Explicit risk of the model *74*
4.2.6 Prioritization *74*
4.2.7 Handling complexity in the model *75*
4.2.8 Profitability tracking on projects *75*
4.3 Project measurement and evaluation 76
4.4 CASSE Framework Architecture 77
4.5 Project value evaluator 80
4.6 Algorithm flow 82
4.8. Scalability of the CASSE Framework 84
4.9 CASSE Framework Scope 85

CASE STUDY PRESENTATION, ANALYSIS AND INTERPRETATION 87

5.1 Overview 87
5.2 CASE 1: Business viability assessment of potential software projects 87
5.2.1 Case study objective *87*
5.2.2 Case Study Description *87*
5.2.3 Project characteristics *88*
5.2.4 Case study procedure *88*
5.2.5 Case study analysis results *91*
5.2.6 Case study implications *94*
5.3 Case two: Bottom-lining functionality packaging on customable software products 97
5.3.1 Case study objective *97*
5.3.2 Case study description *98*
5.3.3 iDART product characteristics *98*
5.3.3 Case study procedure *100*
5.3.4 Case study analysis results *101*
5.3.5. Case study implications 104
5.4 Case three: Supporting the requirements clarification and negotiation process 105
5.5 Case four: Project performance evaluation 113

TASK PERFORMANCE ANALYSIS AND MEASUREMENT ON PROJECT SCHEDULES 121

6.1. Introduction 121
6.1.1 Analysis objective *121*
6.1.2 How it all fits together *121*
6.1.3 Basis for empirical analysis *123*
6.2. General Project Management 123
6.3 Illustrative example 124
6.3.1 Step 1: Task relationship definition *125*
6.3.2 Step 2: UML Modelling *127*

University of Cape Town

6.3.3 Set 3: Analyze the model in the Framework *130*
6.4 Analysis implications 133
6.5. Case study: Tracking project profitability levels 134
6.5.1: Case Study Description *134*
6.5.2. Task acceptance patterns *135*
6.5.3 Task completion patterns *138*
6.5.4 Profitability analysis patterns *140*
6.6. Case study implications 141

ALIGNING THE SOFTWARE PROJECT SELECTION PROCESS WITH THE BUSINESS STRATEGY: ANALYSIS OF THE PROCESS VIEW INVESTIGATION 143

7.1 Introduction 143
7.1.1 Purpose *143*
7.1.2 Investigation procedure *143*
7.1.3 Target sample *143*
7.1.4 Chapter overview *144*
7.2 Sample Characteristics and Frequencies 144
7.3 Study assertions 145
7.3.1 Risk assessment in requirement process *145*
7.3.2 Project selection process *148*
7.4 Cross tabulations results of study assertions 150
7.4.2 Anticipated project profitability & dependency assessment for risk *151*
7.4.3 No dependency assessment done & business objectives driving selection *153*
7.4.4 No dependency undertaken & reliance on feasibility reports *155*
7.4.5 Standardized process utilization & analytical tool usage *157*
7.4.6 System modelling & projects being in-house *159*
7.4.7 R&D projects & less consideration of traceability formulation *161*
7.4.8. Standard process & UML modelling *163*
7.4.9. Standard procedure & selection based on profitability levels *165*
7.4.10. Assessment for risk & analytical tool usage in project selection *167*
7.5 Results summary 169
As illustrated in the table above, for each parameter of interest, a scale ranking was attached arising from the statistical results obtained from the data. *170*
7.5.1 Overall strategy formation *170*
7.5.2 Requirements engineering *171*
7.5.3 Project selection *172*
7.6 Study Implications 172

CONCLUSION AND FUTURE WORK 177

8.1 Discussion 177
8.2 Addressing validity threats in the study results 178
8.2.1 Business view investigation dimension *178*
8.2.2 Process view investigation dimension *179*
8.3 Answering the research questions 180
8.4 Conclusion 182
8.5 Limitations of the CASSE framework 184
8.6 Future work 186

REFERENCES 189

APPENDIXES 200

APPENDIX 1: SAMPLE SCREEN SHOTS OF CASSE 200

APPENDIX 2: REQUEST FOR PROPOSAL DOCUMENT 206

APPENDIX 3: QUESTIONNAIRE 216

University of Cape Town

CHAPTER ONE

Introduction and Background to the Study

1.1 Problem definition

The increasing complexity of making business decisions with minimal certainty is not only limited to the Software Engineering community per se (Burge et al, 2008), but is also a problem being experienced in other disciplines too (Lhuillery & Pfister, 2009). However, from a software engineering perspective, this recurring issue raises two key research elements requiring redress both academically and industrially.

- Firstly, on the global scale, the level of failing IT projects is on the rise and this failure is largely being attributed to wrong decisions taken at a strategic or even at technical levels within the software development lifecycle (Armour, 2005).
- Secondly, making business decisions about viability of software projects, without a substantive degree of certainty has led various software firms into legal disputes with their clients. In most cases revenue mishaps haven't been avoided which ultimately has led to firms being edged out of the software market (Baccarini *et al*, 2004; Wilton, 2005).

The driving factor here is that addressing these challenges requires examining and evaluating the software production process as a value-creating activity. However, improving the performance of software development projects requires firstly measuring the key dimensions of project complexity as well as understanding how they affect project outcomes (Xia & Lee**,** 2004). Several studies show that the development process currently is not well understood as one might think nor as it should be (Coleman & O'Connor, 2008). This is particularly true with respect to the economics of the development life-cycle which is increasingly becoming complex by day. Some of the factors at play include the costing, scheduling and how profitable undertaken projects are. Managers are still in search for mechanisms that can enable them understand these elements with the aim of making informed decisions especially in conditions of uncertainty. This would in part minimize the level of failing IT projects and also lead to business growth and sustainability.

Although past studies for example (Wieczore, 2002; Gruschke, 2005) have presented several approaches for evaluating these project parameters, to date there is no model for estimating the cost of software projects with any degree of certainty. Furthermore, existing approaches do not consider the value-based software engineering aspects in this process. An attempt to link software development and the development process with value creation is therefore vital. Equally,

University of Cape Town

mechanism that attempt to link the software development process with strategic decision making by examining the profit motive would provide more business benefit for key decision makers in the software industry. In the business context, the project that generates the most value to the business forms the best development strategy.

As Kiper & Feather (2005) maintain, in any software development process, the most important strategic decisions are by definition. In most cases these definitions are made early in the lifecycle. They require consideration of multiple interacting alternatives such as cost, schedule, flexibility, complexity, quality control and so on. However, these decisions have to be based on historical information. This is favorable in instances where the current project is sufficiently similar to the past project and where the judgment of domain experts exists. Unfortunately, these decisions have often been made in a *data-starved* environment lacking past similarity correlation.

This lack of correlative efforts coupled with insufficient estimation mechanisms as well as project management bottlenecks have resulted in various failed IT projects (Cooke *et al*, 2001). In most case these failed projects often fail to meet the required need for which they were intended, or they overrun their allocated budgets, worse still getting delayed at times. Therefore, evaluating their true value from both the producer's and the consumer`s perspectives remains a daunting research challenge. The central questions that arise from this challenge span various dimensions of the production process. For example, were requirements well understood at project inception? Were the requirements poorly managed as they evolved during the development life-cycle? Was it a problem of inadequate estimates which were made on the project? Could it have happened due to poor project management mechanisms or was it just a wrong project choice?

Several industrial reports indicate that billions of dollars have already been spent in the past nursing such IT wounds without forth-coming success (Hougham, 1996**;** Sillitti *et al*, 2005). From the employee perspective, this failure has largely been attributed to external factors, whilst success is being attributed to internal in-house risk management efforts [Standing, 2006]. In a nutshell, no one wants to take the blame.

Software is a strategic differentiator in many markets so it is not surprising that software engineering is of strategic importance in many companies. Stamelos & Angelis (2001) observe that although typically a software development organization is involved in more than one project simultaneously, the available tools in the area of software cost estimation deal mostly with single software projects. In order to calculate the possible cost of the entire project portfolio, one must combine the single project estimates taking into account the uncertainty involved. Invariably, as identified by McGregor (2009), this challenge continues to be a focal research interest worth pursuing. Equally, given that industrially decision support systems, specifically in software

University of Cape Town

development are also not well understood or integrated for that matter, further reinforces the view that rational business decisions are bound to be made on development project which may result in revenue loss in the long run or even loss of customer trust and project disasters.

Therefore, it is on the basis of these elements that this research hinges. In particular, the fundamental business strategy alignment question which arises is that what can be done to minimize such occurrences? Specifically, can the use of adaptive techniques that minimize the level of uncertainty in business decision-making be utilized to ensure that software businesses are sustained over time? If so, how can project mitigating mechanisms enable optimization of our production processes as practitioners? Can such mechanisms perhaps enable us increase customer commitment to our products and services as software practitioners?

1.2 Motivation

Software vendors today are increasingly facing the challenge of uncertainty management especially when it comes to determining which software projects to take on and which ones to drop as investment options. This is further complicated by the need to ensure that whichever projects selected fit within the overall organisation`s competitive strategy. As observed by Jamieson & Hyland, (2006) and also by Martin, (2007), this in turn accelerates the need for having innovative techniques of assessment and evaluation to guide the decision making process. At the same time, measuring the probability distribution of profits and losses on any given software project, necessitates tuning various value indicators on these projects. These may range from project portfolio process optimisation to system functional points modelling and understanding as mechanisms for obtaining worthwhile results. The system functional points are often provided in the project requirements as analysis stage artefacts and are very instrumental in determining the overall project investment and value (Costa *et al*, 2007; Barney *et al*, (2008).

Vendors have different business goals which govern their future business direction and operations, including increased return on investment. Available literature suggests that a proactive business strategy must ensure that any project added to the project portfolio must hold business value, if not in the short term, then in the medium or long term (Ross, 2004). Detecting such value, where value is being determined from a vendor`s perspective, is only achievable if a net comparative analysis is made on the entire project right at the project selection stage. This can be through a combinatorial usage of both heuristic knowledge, models, tools and techniques that support project value mapping (Dong *et al.*, 2008).

However, the practice in industry today is overly reliant on tools that are sometimes used in isolation and are unfit to assess net investment in projects intended to create greater business benefit and a

University of Cape Town

sustainable competitive edge (Fadi *et al*., 2005). This approach may be inadequate in ensuring that a project portfolio aligns fully with an established business strategy (Ross, 2004). Rather, what would be optimal is to integrate all heuristic and formal value mapping techniques into one integrated model that is capable of giving an overall big picture for effective decision making. With this, project complexities often arising as risks, will have been controlled. Equally, improved feedback to decision makers will have been ensured in that various business value indicators would now be transparent from all perspectives which in turn substantially support uncertainty management (Martin *et al.*, 2005).

Appropriate tools adopted thus, must be able to handle these project complexities and uncertainty arising out of project properties or artefacts so as to guide decision making in times of uncertainty. Stefan (2006) maintains that all projects have some degree of complexity. However, anticipating this degree of complexity and planning to manage actions to meet such situations effectively requires an understanding of project details and the project implementation strategy (Slater & Olson**,** 2001). The range of complexities that need to be assessed prior to accepting the project span various dimensions, including technical and business scoping.

Making such assessments therefore results in providing information for planning and anticipated actions by which to address the various situations impinging on the value of a given project. The technical aspects of the project imply the degree of difficulty that could be required in building the product while business scoping in this regard would imply issues such as schedule, cost, risk, communication, among others. Whereas the technical scope revolves around building the product desired by the client, it is often the business scope on the other hand that adversely affects the project through appropriate decision making.

Complexity can span various dimensions, including project boundary definition and management, requirement volatility, resource bottlenecks, change management and project miscommunications among others (Shaban-Nejad & Haarslev, 2007); (De Souza & Redmiles, 2008). If this complexity is not well modelled, assessed or even detected uncertainty on these projects will resultantly increase.

Yet, the available project information at hand could be useful in assisting with managing and decreasing these complexity levels, thereby leading to appropriate decisions that result in quantifiable valuable outcomes. The tangible information assets on the project, in this regard, are the system actors and their properties such as the system functional points that form the basis for obtaining business value (Al-Hajri *et al*., 2005). However, it is largely dependent on how well this initial project information is analyzed and interpreted technically into an intangible project value yardstick required to make appropriate business decisions.

University of Cape Town

For example, systems engineering provides various ways of analyzing this information, most of which is model-based. Various modelling techniques such as Object Oriented Systems Analysis and Design (OOSAD) using the Universal Modelling Language (UML) mapping can help in refining this information into valuable assets that are capable of supporting the business decision making process. However, extensively utilizing such techniques requires innovative ways directed towards minimizing uncertainty while increasing stakeholder benefit.

Several authors such as Andersen *et al*., (2000) and Polack & Stepney, (2005) observe that the emergent engineering approach as a critical enabler for understanding and managing project complexity can help. This approach however, is largely dependent on representation. It borrows inspiration from both biological systems and social systems such as self-repair, self-management and self-adaptation to changing environment. The belief in this thesis is that any given project is a complex adaptive system with required emergent properties. These emergent properties comprise the following.

- elements described by simple state behaviour (Actor Objects and their properties),
- detectable surprises (patterns, constructions etc), described in terms of the physical reality or physical properties (the way objects interact and evolve to achieve project value), and
- a representation or environment, supporting mappings between, and possibly interacting with the other parts (dependency relationships).

1.3. Rationale

Whereas delivering software on time and within budget, and satisfying all of its requirements pose significant technical challenges for researchers, managers, and practitioners in industry today, equally ensuring that business creates value (in all time runs) on projects undertaken remains largely a business scoping challenge. As Peppard (2007) recons, businesses have some measure or metric used to assess value on projects. However, what would be fundamental in guiding the decision making process, especially when faced with uncertainties, is the ability to measure the exact business shape or value status which a given project is likely to take. Often such project status indicators are presented as strategic quadrants representing different business interests.

Worse still, this seemingly recurring uncertainty is additionally making it difficult to quantify the relative probabilities and sizes of loss in software development (Ward & Chapman, 2003). Erdogmus *et al.*, (2004), Costa *et al.*, (2005) and Huang & Boehm, (2005) agree that practical approaches for determining risk-balanced favorable operating points as value-related aspects in software development processes are still inadequately developed. In the past various researchers such as Costa *et al*. (2005) have been applying economical concepts to confront and resolve these challenges.

University of Cape Town

However, there is still more research required to provide alternative approaches. As Boehm (2003) observes, many verification techniques exist and are being actively researched. These are often directed towards validating investment viability of potential projects as well as operational competency implications required to deliver the required project within time, scope and budget. These, for example include: static analysis, testing and formal verification as proposed by Nigel et al. (2001). However, they have largely been applied as isolated technologies.

This therefore raises three questions, which are addressed in this study:

- Based on Actor Object Dependency Analysis (AODA) and their properties, can we estimate the probability distribution of losses and earnings that can be incurred by a software vendor on any project given his software project portfolio?
- Can economic concepts, models and tools help to advance our understanding and improvement of the development of software and process that produce them?
- In particular, do the project selection and requirements engineering processes employed in industry today align appropriately with organizational business strategies for supporting value creation on software projects?

This work is grounded in the Adaptive Software Development (ASD) (HighSmith, 2000) branch of the Agile theory (Highsmith, 2002); (Cockburn, 2002) and is also based on the Value-Based Software Engineering (VBSE) framework initially proposed and developed by Boehm (2003). It has an overall aim of providing a useful technique for monitoring and controlling value propositions in terms of cost, schedule, and progress of complex issues in software development, specifically by applying decision analysis models and techniques in software engineering so as to support the value-based paradigm. A detailed definition of value adapted in this research is given chapter 3 of this thesis.

1.4 Study Contribution and Thesis Outline

As summarized by the value ladder given in Figure 1, this empirical study introduces a theoretical foundation for addressing value-related challenges that confront software engineers and their organizations.

University of Cape Town

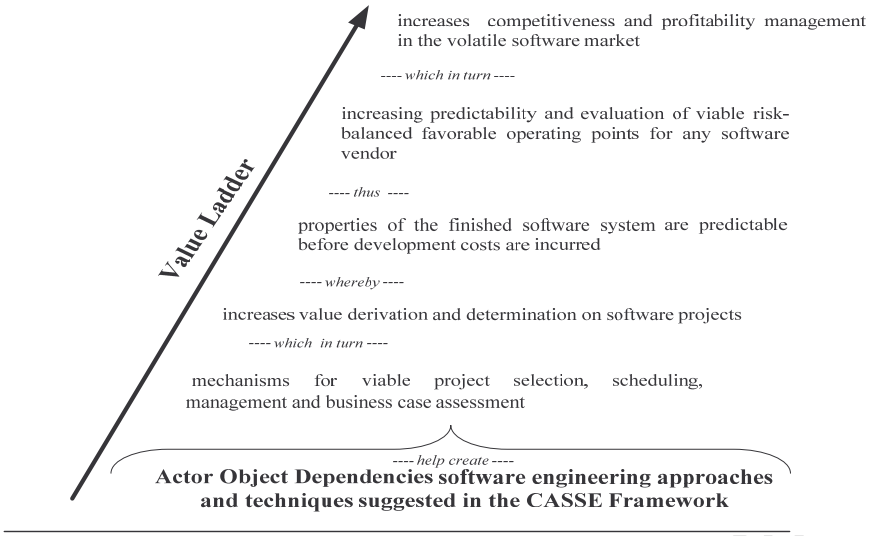


Figure 1: Value ladder

This study is geared towards performing a comparative analysis of modern software management and development methods. This theoretical foundation is built around the Complex Adaptive Systems theory in order to support comparative analysis strategies on software projects. It thus results in the formulation and introduction of the Complex Adaptive Systems Software Engineering Framework (CASSE) as an alternative engineering option to economic value prediction and management in software development. This is achieved through supporting the decision making process around software projects and applications especially in conditions of uncertainty and change.

As summarised in the value ladder above, the CASSE framework introduces the Actor Object Dependency (AOD) technique. This technique help create mechanisms for business case assessment on software projects. In this approach viable projects are selected, scheduled and managed while at the same time tracking anticipated profitability on these projects is made possible. This in turn increases value derivation and determination on software projects since properties of a finished software system can be predictable before development costs are incurred. This thus increases predictability and evaluation of viable risk-balanced favourable operating points for any software vendor. This in turn increases competitiveness and profitability management in this volatile software market.

This approach complements the traditional cost, schedule, and product planning and control techniques of the value delivered to stakeholders. It uses emergent engineering approaches initially suggested by Linton *et al*, (2000) to enhance project selection and value tracking during project execution. It also adapts principles from economics, including market and technicality impact

University of Cape Town

assessment described by Dey, (2006). It predicts project value before the actual cost of developing a product is incurred. It is also extensible in tracking the anticipated profitability on the project during the development life-cycle.

This work thus contributes to knowledge and industrial practice in the following ways:

- **Theoretical contribution**:

This study has resulted in formulating a CASSE framework which uses AOD techniques to embrace the practical application of firstly, software engineering management and, secondly, the economics that accompany the software life-cycle. Even in its preliminary state, this framework is sufficiently expressive to be useful in explaining and characterizing project evaluation techniques.

- **Practical contribution**:
    - The study will help software engineers learn how to decompose systems into "chunks" of revenue-generating functionality to carefully sequence them in order to maximize the overall value of the project and also to manipulate other project metrics such as reducing the initial funding investment, or decreasing the time to reach break-even status.
    - This approach will also equip managers to analyse and understand the impact of other requirement prioritization decisions on the financial returns of a project.
    - This approach is equally effective in both traditional and agile software development environments and is scalable enough to handle large projects. With such a technique, users can perform sensitivity analyses of the most appropriate investment benefit levels for different cost-of-delay situations. One practical example is the role of CASSE in supporting organisations engaged in determining budget and schedule estimates of proposed systems especially when guiding their clientele on "*Request for Proposal (RFP)*" specifications expected from preferred IT product suppliers. Alternatively, CASSE can support a client requesting a new system and facing the challenge of adjusting the desired functionality to the available budget and choosing a software vendor requesting a realistic price.
- **Core contribution:** The core contribution of this work lies in risk detection and management in software development using Actor Object evolving interrelations throughout the development life-cycle. This is achieved through consolidating and supporting value derivation linkages during the business (project) selection process as well as during the project tracking stage.

1.5 Differentiation factor of the CASSE Framework

The differentiating factor of the CASSE Framework, lays in its ability to manage uncertainty on any given software project. Uniqueness to this framework is its ability to learn from past experiences in comparison to the present situation so as to adapt to the current situation and also predict the

University of Cape Town

likeness of the future situation with significant certainty. Ideally, as more projects are added to the project portfolio, the model prediction accuracy adapts and improves. Accuracy prediction is based on historical data. Equally, as new requirements are added to the project scope or changes made to the scope and model therein, the project profitability patterns and properties evolve and update accordingly.

Therefore, it offers ground for adequate project estimation based on the past project history so as to project a valuable evaluation status. Thus, this approach strengthens the mechanism for eliminating data-starved environments typical in estimations and at the same time increases correlative efforts in project comparisons. Previous work in this realm, for example in Suliu & Rule, (2007), has not largely addressed this adaptiveness. It has only embraced optimisation of development efforts for release planning purposes but not overall value management as actual development time is uncured. Although bi-object modelling theory is applied in both instances, this work presents an extended alternative approach. It addresses overall value prediction and management over time. This in turn ensures that uncertainty on projects is well tracked and managed. Secondly, that the changing project patterns are adaptively learned so as to improve value prediction and management for future projects. Thirdly, in the event that this approach is well tuned, it can considerably aid the decision making process in real time.

The rest of the thesis is structured in the following way:

Chapter 1: Introduction and Background to the study: This first chapter presents an introduction and an overview to the research field. It elaborates the rationale and motivation for the study. Research questions and aim of the study are set together with expected results, as well as the focus and expected results of the study.

Chapter 2: Methodology and Research Design: The research strategy is laid out based on the basic assumptions about research, the characteristics of the phenomenon in focus and the dependent and independent variables being investigated. The chapter also highlights the research strategy adopted in this study in addressing sample size and selection procedure adopted. Data collection techniques used, as well as data analysis and presentation techniques used in communicating the results are also described in this chapter.

Chapter 3: Determining and sustaining project value on software projects. This chapter highlights fundamental building blocks in establishing a valuable project portfolio process for software projects. It details the adaptive techniques that can be used in revolutionising the software development process so that successful yet profitable projects are accomplished by any commercial vendor. Such value propositions can aid decision making under conditions of uncertainty and also

University of Cape Town

assist with the evaluation of project externalities that hamper project value derivation especially when scored to the value threshold often stated in the business operational model.

Chapter 4: The CASSE Framework engineering approach: This chapter introduces the CASSE framework and also details the constructs of the framework. It elaborates the value-based matrices upon which the framework functions. The different blocks forming the framework are also explained.

Chapter 5: Case study presentation, analysis and interpretation

Empirical data from industry and results are presented in this section. Analysis of each case is made and case-based conclusions are made. It is upon this cross-case examination that the research questions for the study are answered.

Chapter 6: Framework applicability in general project management

This chapter details how the framework can be applied to general project management complexities. Specifically, it demonstrates how anticipated profitability levels on the project can be measured against project tasks at the initial stages of the project or constantly tracked as the project progresses. In this chapter, an illustrative example is given on how task interdependences can be used to achieve this objective. This is complemented by results from an empirical study conducted on two commercial software projects in industry particularly looking at temporal task acceptance patterns, task completion rates and project profitability levels at different time steps during the project lifetime.

Chapter 7: A process view survey analysis and interpretation

This chapter presents findings from a survey conducted in Cape Town though the SPIN Network with practitioners in industry. These results are instrumental in establishing baseline data for future comparison and problem analysis and secondly for identifying trends, issues and concerns unique to software development process management and challenges.

Chapter 8: Business and process views alignment with overall project value

The conclusions discussed in this chapter are drawn from the outcome of the analysis of the cases selected in this study. A discussion about the validity and reliability of the conclusions is also detailed in this chapter together with specific contributions and future research needs.

Appendices: All other documents such as the data collection tools used in the study as well as screen shots of the CASSE analytics tool are given in this section.

University of Cape Town

CHAPTER TWO

Research Design and Methodology

2.1 Introduction

This study addresses an operation research problem and was primarily empirical, designed to gather information from practitioners in industry about how best economic concepts, models and tools can enhance the improvement of the software development process. It is both analytically qualitative and quantitative in nature aimed at formulating a theoretical framework that supports the value-based software engineering paradigm especially in the project selection process and in the overall project management spectrum.

The rest of this section describes the methodology that was utilized in accomplishing this goal. It essentially answers the four focal research areas emphasized by Silverman (2005):

- The overall strategy adopted and why that particular strategy was used;
- The reasons for the selection of these methods and not any others;
- How the research was conducted; and
- The design and techniques used.

The unique contribution given in this chapter is the combination of the case study reach technique as well as the survey research approach in addressing the research problem as illustrated in Figure 2. The case study approach was adopted to address the business view perspective in relation to value-based software engineering in industry. The survey technique was adopted to address the process view perspective currently prevailing in industry today. This perspective aimed at examining whether software projects undertaken generate business value to management and the organisation as a whole. This blend of techniques therefore resulted in a highbred multi-technique research methodology enabling a congruent assessment and validation of the theory under investigation.

University of Cape Town

2.2 Research strategy

In answering the research questions raised in this study, two investigative dimensions were used, that is, the process view investigation dimension against the business view investigation as illustrated in Figure 2. The left pane illustrates the business view investigation dimension while the right pane illustrates the process view investigation dimension in defining, analyzing and collecting data about the proposed theory in this research. Key steps included:

1. Building an understanding of the problem domain by reviewing literature surrounding uncertainty and economic value management on software project,

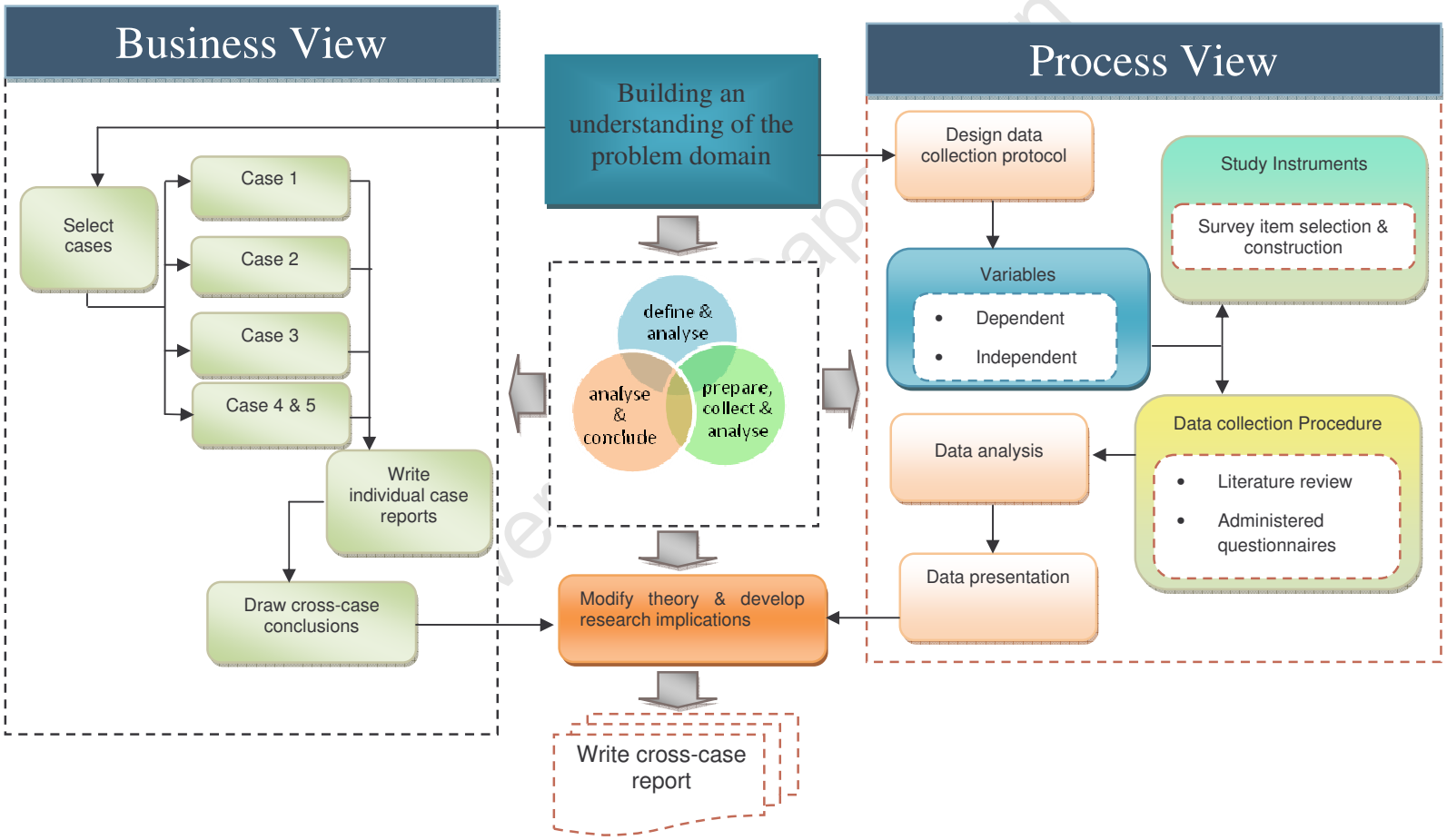
2. preparing data collection instruments,
3. collecting data, and
4. analysing and interpreting this data.

Based on the interpretation and documentation of the findings from both the survey and the case study analysis, conclusions were made and research implications were drawn. Accordingly, the theory was modified according to the findings. This formed the basis for the cross-case findings presented in this research that validate the theory.

The highlighted steps above were executed in parallel in both the business and the process view dimensions. The initiating step for both dimensions was the theoretical development step as illustrated in the figure below.

University of Cape Town

Figure 2: Methodology flow chat



There was a need to measure how each dimension impacted on the value-based software project selection process and the profitability tracking technique proposed in this study. A longitudinal technique therefore was applied since the phenomenon in focus was likely to be subjected to heavy change, given the agility of software development. Abercrombie *et al*., (1994) define a longitudinal study technique as an approach which involves collecting data from the same sample at intervals over time. This technique is useful for studying trends and the effects of particular changes. The sample in this case constituted the software engineering community engaged in the business of software development in Cape Town, South Africa.

A combination of selected-response survey items in conjunction with a qualitative review of sample empirical project data from industry was employed. The major goal was to investigate the possibility of estimating the probability distribution of losses and earnings that can be incurred by a software development organization. This would be evaluated based on its software project portfolio and its project requirements risk analysis and management strategies. Subsequently, while the case study approach was employed to embrace the business view investigation dimension, the survey approach was utilized to embrace the process view investigation dimension.

2.3 Rationale for using both process and the business dimensions

In the agile market, certain business decisions have to be taken without much delay in consultation; otherwise, an opportunity is lost. This is especially true for new entrants into the market that are competing for a market share. Too often companies enter markets where they do not have sufficient capabilities to compete effectively (Singh, 2008). Making a foot print in the market requires that one adopts market penetrating strategies such as price-cuts so as to out-compete their competitors. Therefore, chances are that unknowingly or knowingly certain unprofitable projects are taken on for the sake of increasing the business profile especially for the small firms.

In instances where such blind decisions are taken, especially where some projects appear profitable on the surface during the initial stages, once a clear understanding of the requirements sets in, project disaster symptoms start to emerge. At times it is often too late to walk away from such projects given the legalities involved. Firms therefore have to complete such projects, regardless. At the end of the day, these projects tend to be executed at a loss and someone has to account to top management and even to the shareholders if necessary about this situation. This pressure is often felt at the management level other than at the technical level.

Therefore, the need for tools that increase visibility into assessment and identification of risky business becomes very critical in order to avoid repeatability of undertaking non profitable projects. At the same time company bottom-lines would be set at a project level before commitment is made

University of Cape Town

on these projects. Essentially, this would support tactical decision-making to ensure that past mistakes do not reoccur. The business view dimension was therefore ideal in capturing this notion. This would demonstrate that profitability as well as viability bottom-line establishment can be achieved with adaptive tools that keep track of what is happening in the process at every level within the development lifecycle.

The employees on the other hand often execute what management dictates without much question or debate. They only worry of whether they will be paid at the end of the day. As firms struggle to stay afloat in the market, desperate measures are always adopted. At times projects taken on may not necessarily be in line with the initial business objectives or competing strategies, but rather survival mechanisms. For example, a firm starting out as a research and development company focusing on the production of a specific technology may end up taking on web development jobs in the short term to keep afloat. In case the web development jobs turn out profitable, some of these firms may change business strategy. In most cases resources have to be juggled around between research and development and also between the *bread and butter* projects.

Therefore, by adopting the process view investigation dimension, such instances would be captured. Specifically, the technical people would have an opportunity of evaluating the business selection procedures against company strategies. Of course on the assumption that the business strategy is well diffused to the lower levels within the company and that it is well understood. Case-based evaluation of projects against profitability and project success would not span to this level. Therefore, a process view dimension had to be devised to address this challenge.

Thus, combining both approaches would address both the employee evaluation of business decisions and at the same time capture managerial view about aided decision support mechanisms. In this regard, the business view investigation dimension, cases would be selected based on unique scenarios. On the other hand the process view dimension would examine and test the general practice in industry in relation to process management and optimization against the perceived business strategy.

2.4 Business View Investigation Dimension

2.4.1 Case Selection

To validate the approach, four case studies with empirical data from industry were analyzed. These cases were illustrative of the various perspectives within which CASSE can be practically valuable. Case 1 represented a vendor using CASSE to respond appropriately to a request for proposal call by fitting the proposed functionality into a projected budget and development portfolio. Case 2 represented baseline benchmarking for viable determination of the functional baselines of system feature customization on an evolving health management system developed by a South African

University of Cape Town

vendor. Case 3 illustrated the usefulness of CASSE in the requirement negotiation process. Case 4 on the other hand demonstrated how CASSE can be used in project portfolio performance measurement and evaluation. Case 5 looked at temporal profitability analysis of two projects for a software vendor in Cape Town. A report detailing the results of each study was written.

2.4.2. Motivation for selecting the various case studies

Cases used in the study were motivated by various research needs as described below.

- **Case 1: Business viability assessment of potential software projects:** This case study was motivated by the need to increase the visibility of identifying risky projects earlier on in the process before commitment is made on such projects. In industry today, remaining competitive in the agile software market requires selecting only those projects that position a business strategically in the market place and that render it competitive over time(Balikuddembe & Bagula, 2009). Thus the need for mechanisms that would eventually enhance evaluation of the probability distribution of profits and losses on a software project before investment costs on such a project are incurred was imperative. This work draws from available literature in value-based software engineering, such as (Boehm& Huang, 2003; Saliu & Ruhe, 2007) to provide an integrated approach for assessing overall value on a software project. This case study thus illustrated that CASSE can be valuable in addressing this challenge in the operational environment.

  The independent variable in this analysis was ***favorable operating point for a given project*** while the dependant variable was ***uncertainty management in project selection process***.

- **Case 2: Bottom-lining functionality packaging on customable software products:** This case study was motivated by the fact that the pricing of requested customizable product features against existing features remains critical in the software business. The underlying cause being the general lack of adequate techniques that facilitate gauging customer sensitivity to product pricing against vendor profitability derivation on such requests (Liao, 2005; Karandikar & Nidamarthi, 2007). As Kettunen, (2009) observes, many current decisions taken on customization requests for software products seem to be made from a functional perspective, with little consideration for overall cross-functional implications. This therefore raises two critical factors for ensuring competitiveness of the product line, namely: implementing the right tools and techniques to meter and measure complexities associated with a given customization request against the overall system functionality; and assessing the business value associated with accepting a given customization request (Balikuddembe *et al*, 2009). Thus, by applying the CASSE framework to this scenario, bottom-lining of key system functionality that are used by all

University of Cape Town

system elements in the system would be identified. The independent variable in this analysis was AOD technique while the dependent variable was functionality packaging on customizable products.

- **Case 3: Supporting the requirements clarification and negotiation process:** Requirements negotiation on software projects necessitates use of value-driven approaches that guarantee a win-win situation for all stakeholders, thereby permitting agreement on desired functionality and stakeholder commitment on projects(Balikuddembe& Bagula, 2009). Such approaches must be particularly edifying in order to motivate stakeholders to set realistic budget estimates, schedules and scope while benchmarking on the complexity of project requirements. Previous work in this area has emphasized that inadequate ways of managing and interpreting project requirements, as a primary scoping facet, results in various consequences, most of which considerably impact on business resources and competitiveness (Croll & Croll, 2007; Kutsch, , 2008). This research challenge formed the basis for investigating this case study.

  The dependent variable in this analysis was ***requirement clarification and negotiation process*** while the independent variable was ***clarification mechanisms used to illustrate complexity***.

- **Case 4: Project performance evaluation:** This particular case study was motivated by the fact that the software business is continually faced with the challenges of tracking multiple-project progress and establishing a methodology for evaluating, selecting, and prioritizing these projects Equally, identifying potential projects and determining their value in supporting organizational strategy is still a daunting task (Killen *et al*, 2008; Thomas & Fernándeza, 2008). Therefore, by using this case study, this research problem was explored and relevant conclusions were drawn.

  The key interest here was to validate the notion whether the successful management of value on the software development process is dependent on the evaluation mechanism adapted to evaluate project performance.

- **Case 5: Tracking project profitability levels:** This case study was motivated by the need to evaluate profitability performance of undertaken projects. In particular, providing viable estimates, understanding project requirements and doing proper risk management on software projects require extensive application and sophisticated techniques of analysis and interpretation (Zafra-Cabeza *et al*, 2008; Alba & Chicano, 2007; Balikuddembe *et al*, 2009). There is still a lack of informative techniques and feedback mechanisms that help to assess

University of Cape Town

how well and efficiently a specific development methodology is performing. Therefore, by using the CASSE Framework in analyzing project tasks, enhancement of how well individual tasks are estimated, how well they are defined, and whether items are completed on-time and on-budget would be ensured.

The key evaluation criterion here was to evaluate if temporal evaluation of task completion rates on projects affects the actual and anticipated profitability levels on projects.

2.4.3. Case study protocol

One central instrument for conducting case studies, and in particular multiple case studies, is a case study protocol. The case study protocol contains an overview of the case study project, field procedures, case study questions and also a guide for the case study report. In this study, however, no explicit case study protocol was made. A notebook for planning, executing and reviewing the work with the case studies was used instead. The notebook does to a large extent cover the previously mentioned content and thus serves as a case study protocol.

2.5 Process view investigation dimension

2.5.1. Background

The value-based software engineering premise perceives value derivation on software projects as an integral notion, embracing a full range of existing and emerging software engineering principles and practices (Egyed *et al.*, 2005). Pivotal to this are requirements engineering, project planning and control, as well as risk management mechanisms utilized in the development processes. Similarly, achieving project value and success requires that the engineering approaches utilized on such projects support both the project selection process and the business strategy (Wu *et al*, 2009). In this work, this is perceived as a three-in-one cycle summarized in Figure 3 below.

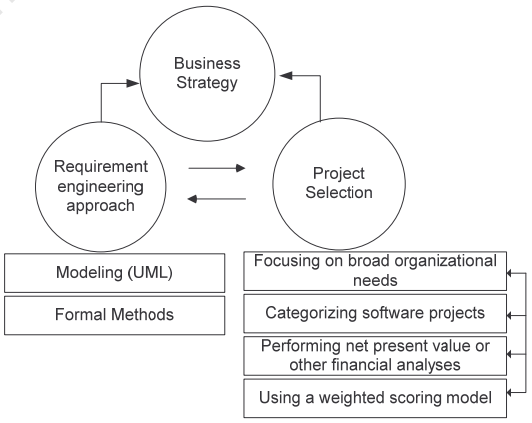


Figure 3: Technical inputs to the business strategy

University of Cape Town

The project selection process is fundamentally driven by four principles, including: focusing on broad organizational needs, categorizing software projects according to organizational needs, performing net present value or other financial analyses on these projects and using a weighted scoring model to obtain a clear project value. Of equal importance, it is inferred here that the engineering approaches utilized in understanding project needs and requirements must support these principles as bi-directional business functions (that is, requirements engineering and project selection processes) impinging on the business strategy. This process can either be '*formal-method*' driven or carried out by use of modelling approaches such as UML, thereby supporting the planning and control process on projects as well as identifying risk factors on the project. If well adapted (as project risk precaution drivers), support for identification of any risky business early on in the process is ensured and even before resources are committed on such projects. If not, in the absence of such business strategy- supporting mechanisms, profit-driven software businesses are likely to be edged out of the market easily (Gregor *et al*., 2007).

2.5.2 Survey

With this business view as a focal point and also to complement the case study findings, an empirical study with software practitioners in Cape Town, South Africa was conducted. The main objective was to examine whether project selection and requirements engineering processes (as technical inputs to the project selection process) employed in industry today align appropriately with organizational business strategies in general. The purpose of this research was focused on understanding the phenomena of prevailing project selection processes in industry and their alignment with the business strategy so as to gain some insight in what is going on in business strategy formation and why it is indeed happening.

This survey constituted a sample of 70 software companies in Cape Town through the Software Process Improvement Network (SPIN), Cape Town chapter. The SPIN network is made up of more than 200 members. All these members are practitioners in industry while others are from academia. This specific survey only targeted practitioners. They were included in the survey to evaluate whether this approach is feasible in the real world. Based on the findings, the approach was redefined to suit software process improvement. When selecting participating companies in this survey, there was no specific consideration of size in sales, size in employment or industrial sector. Rather, it was a random selection.

2.5.3 Variables

2.5.3.1 Dependent variables

The dependent variable in this study was software development process improvement. In this study software development process is defined as the approach followed in producing software. It is

University of Cape Town

inferred that the software development process is affected by many factors, most of which are economic such as personnel involved in the development process, methodologies and tools utilized, as well as the design criteria followed. Therefore, adopting an optimal approach to software development requires strategic mechanisms of controlling these economic factors so as to estimate the probability distribution of losses and earnings that can be incurred by a software development organization according to its software project portfolio.

2.5.3.2 Independent variables

Independent variables that were considered in this study are economic concepts, models, and tools. In this study, it is inferred that economic concepts, models, and tools can help us apply decision analysis and techniques in software engineering so as to improve software development processes.

2.5.4 Data Collection

In this research, documentation, direct observation, and participant-observation were the main vehicles for data collection. For the business view dimension, direct observations were conducted to evaluate what the case companies had done. Having been a consulting contractor on one of projects and also an employee of two case companies during the time this research was conducted, a lot of scenarios were observed to refine the theory and also to extend the understanding of value linkages on software projects.

For the process view dimension, data collection was based on two focal areas: understanding interrelationships in project matrix factors affecting value creation in software engineering and the feasibility of such matrices on overall business strategy evaluations.

Since the results were relevant for software industry, conferences papers were published on the findings for further scrutiny by both academia and industry. Copies of these publications are available from the publishers` websites as indicated in the publications section of this thesis.

2.5.4.1 Literature review

In developing a theory for this study, literature was reviewed from journals and other relevant sources in order to address the software economics issues that support value creation in software engineering, specifically, process improvement. This review enabled the value creation mechanisms underlying software development to be examined in relation to this paradigm.

2.5.4.1. Observation

In the business view investigation dimension, observations were used in some instances. This combined *ethnographic* and *interview* approaches to obtain clarity on the number of factors affecting economic value management on software projects. Spradley, (1979) defines ethnography

University of Cape Town

as the work of describing a culture with the goal of understanding another way of life from the native point of view. Therefore, this rendered this approach useful in understanding how other people see their experience. This particular approach strengthened the data collection process by supporting the understanding of practitioner`s needs, experiences, viewpoints, and goals about value creation. Such information enabled me to redesign parts of framework so as to render it useful in the business context and ultimately improve the value creation paradigm in software engineering. Interviews were conducted in parallel to strengthen the view point and experience. From a business context, this mechanism helped me uncover and also interpret potential consumer`s point-of-view of this framework in practice and the hidden rules of the operational environments.

2.5.4.2 Study Instruments

2.5.4.2.1 Survey Item Selection and Construction

The survey instrument was designed as an exploratory tool to gather a large data set of information relevant to software process improvement in relation to the economics of software development. The results from the survey were used to:

- establishing baseline data for future comparisons and problem analysis, and
- identifying trends, issues, and concerns unique to software development process management and challenges.

The survey instrument items (contained in the appendix section of this thesis) were divided into subscales of selected-responses centered on economic aspects of software development. A systematic process for survey development, as suggested by Sanchez *et al*. (2000), which is grounded in consideration of the basic purposes for which the test scores would be used, was employed to develop the present survey instrument.

Subscale 1: Role and Professional experience:

The first section of the survey consists of two items that gather nominal data about participants' roles and experience in managing software development projects. From these items it was anticipated that various correlations would be drawn with other items in the other subscale sections. The main assertion here was that given the experience of the respondents and their role or roles in engineering and planning information technology projects, there is a high likelihood that throughout their experience, they have come across project selection, change management as well as monitoring challenges and complexities on IT projects. Such renowned challenges often compel a need for viable solutions, in terms of models, applications and techniques that can aid informed decision making in project management.

University of Cape Town

Subscale 2: Project management and implementation:

The 27 items in subscale 2 of the survey were designed to gather ordinal data on requirement engineering, project selection, project monitoring and implementation and project completion and signoff. A 5-point Likert-type scale (i.e., 1 = Strongly Disagree, 2 = Disagree, 3 = Not sure, 4 = Agree, 5 = Strongly Agree) previously used by Guan *et al*. (2006) in a study of the relationship between competitiveness and technological innovation capability based on DEA models was employed.

The requirement engineering subsection aimed at understanding current software development processes used today, by specifically analyzing the depth and breadth of requirement engineering techniques used in industry. Given the agility of software development today, it is highly likely that certain important aspects of requirement engineering are ignored, yet insight into such aspects would enable risk factors on the project to be identified, which could consequently influence project profit levels.

The project selection subsection aimed at understanding how projects are selected in the various participating developing houses. The selection criteria used were instrumental in refining the study assertions and answering the research question thereof.

2.5.5. Data Collection Procedures

In demonstrating the capability of the CASSE framework to the practitioners in industry, a CASSE tool demo was set up. This was conducted during a presentation on VBSE at one of the SPIN meetings. As a follow-up of the presentation an anonymous, 40-item survey instrument was administered to participants via email, as well as a near-equivalent paper-based, or hardcopy, version of the instrument. Invitations to participate were conducted telephonically and on the SPIN discussion list in which participants were encouraged to participate.

Voluntary participation was assured because participants had to consciously decide to complete the questionnaire and return it. There was little direct pressure on individuals to participate in the survey. Participants were not required to provide any identifying information as a part of the survey process. However, for those study participants who wanted to receive a report on the results of the investigation, their contacts were logged. However, no attempt was made to link this identifying information with the data provided in the study.

2.5.6 Data analysis

Data analysis consists of examining, categorizing, tabulating, or recombining the evidence to address the research question or the study propositions. In this case, data were recorded thematically according to the research question in the study. It was interpreted and presented with the help of direct quotations with regard to the extent of the problem in comparison with the available data or

University of Cape Town

information. For the purposes of analysis, the structured data in the questionnaires were matched according to the category of respondents. The frequencies of the various responses were tallied for analysis purposes, and averages and rankings of discrete responses were made. Descriptive research analysis was used for closed-ended questions that demand discrete responses. All this was achieved with the help of SPSS 17.0 which is a statistical package widely used in social research.

2.6. Design Science approach

The approach used in undertaking this study was design science and the research design involved five case studies. According to Gero (1999), design science research aims at developing general knowledge which can be used by professionals in the field in question to design solutions to their specific problems. Given that software development faces unique challenges when it comes to value creation on projects, this approach was suitable. Ken et al (2008) emphasise that whilst design science strives for a theoretical foundation, its utility lies in its ability to use design computing and design cognition to represent both designing situations and designing processes using concepts of varying theoretical rigour. Consequently, in this research, this approach would offer creation of successful artifacts in the solution space. This aspect included problem identification and motivation, definition of the objectives for a solution, design and development, demonstration, evaluation, and communication. This was all achieved in the case-study approach devised in the business view investigation dimension.

University of Cape Town

2.7. Approach to answering the research questions

The figure below illustrates how the identified research questions of the study were investigated, answered and also validated.

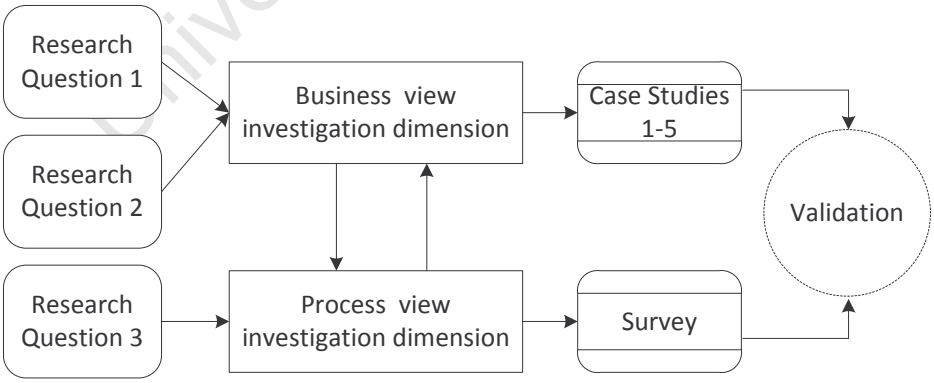


Figure 4: Research question flow chat

Research question one and two were answered using the business view dimension. On the other hand, research question three was answered using the process view dimension. Although different approaches were adopted in achieving the results, these approaches complimented each other in the problem domain exploration. Results obtained from the survey conducted in the process view

investigation dimension as well as those from the business view investigation dimension were validated using validity constraints as elaborated in the validity section of this thesis.

2.8. Validity of study results

Kvale, (1996) maintains that validity is often defined by asking the question: "*Are you measuring what you think you are measuring*?". In other words it is the best available approximation to the truth of a given proposition, inference, or conclusion. Wohlin *et al*, (2000) maintains that every empirical study in software engineering must be validated. Therefore, in order to answer the research questions raised in this study, validation of the results had to be done. As suggested by Walter & Padberg, (2007), this theory of validation and the many lists of specific threats, provide a useful scheme for assessing the quality of research conclusions. These validity threats include: internal, external, conclusion as well as construct. It was projected that various inferences or conclusions could be made while conducting this study; most of which being related to the process of doing research and not the major hypotheses of the study. Thus, the results obtained from each investigation dimension were evaluated on these lines in order to draw meaningful conclusion of the study.

2.9. Data presentation

After the data had been collected, they were sorted, analyzed and presented according to the research question focal areas. Before the data was presented, however, it had to be inspected and edited to avoid distortion of the message from the respondents. Anonymous quotations were noted from questionnaires, actual interviews, casual discussions and open-ended questions, and observational descriptions were made.

University of Cape Town

CHAPTER THREE

Determining and sustaining project value on software projects

3.1 Introduction

This chapter highlights fundamental building blocks in establishing a valuable project portfolio process for software projects. It details the adaptive techniques that can be used in revolutionising the software development process so that successful yet profitable projects are accomplished by any commercial vendor. Such value propositions can aid decision making under conditions of uncertainty. They can also assist with the evaluation of project externalities that hamper project value derivation. This is specifically attainable in situations when such projects are scored to the value threshold often stated in the business operational model.

Unique in this chapter is the notion of perceiving each software project as a value generating activity dimensionally triangulated; whereby, adapting to agile trends in the market place, internal process realignment as well as competitive advantage leveraging is fundamental in achieving a big market share in the software business. This view is detailed further in the next section and is also illustrated in Figure 4.

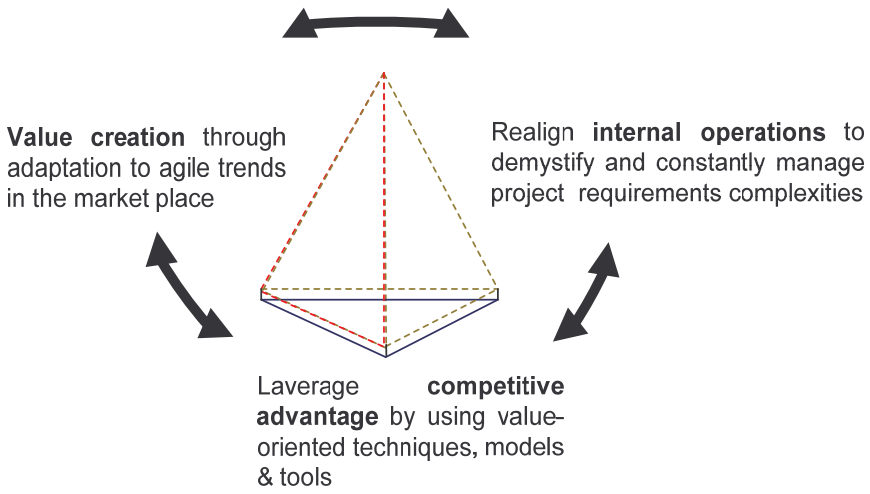
The other contribution to the realm of value determination and sustainability management is the new value-based software development management process illustrated in Figure 5. This is developed as a probable solution to the problem of sustainable value-chain creation on software projects.

3.2. Choosing valuable projects

Value proposition is driven by the notion that projects taken on must be profitable or have long-term strategic business value (Walter *et al.*, 2004); (Czuchry & Yasin, 2003). The strategic project profitability drivers require that business dynamically adapts its processes as the market needs evolve (Palanisamy, 2005). This raises three adaptation pattern approaches for business evaluation. In here these patterns are visualised using a project profit triangle illustrated in Figure 4.

University of Cape Town

Figure 5: Project profit triangle



The three sides of the profit triangle are value creation, internal operations and competitive advantage; all of which are important for a business to be profitable. Business must excel in all these three areas particularly if it intends to strategically hold a big market share over its competitors. Like a triangle, all three aspects are linked – a company's performance on one side will have an impact on the other sides. If its internal operations are not well structured, a company will find it difficult to create value and at the same time be highly competitive.

3.2.1. But what is value?

Value creation is a central concept in the management and organization literature for both micro-level (individual, group) and macro-level (organization theory, strategic management) research. Yet there is little consensus on what value creation is or on how it can be achieved and the mechanisms that allow the creator of value to capture the value (Lepark *et al.*, 2007). This variance in definition differs based on whether value is created by an individual, an organization, or society. Several authors have emphasized that in business terms, value is often determined by the customer where the vendor sets a price and the customer decides if that price represents value. If the price is too high, the associated value may be too low and not attractive; hence there will be no sale (Davey *et al*. 2006; Braido, 2008). This creates a challenge for the vendor to make the product or service available at a price that is attractive to the potential customer by allowing the development costs as well as the ongoing sales and marketing costs to be amortised over the sales lifetime. Conversely, from an economic perspective, value creation theory revolves around calculations of the net present value of the future benefits associated with the investment (Carlucci & Schiuma, 2007; Joaquin, 2001).

University of Cape To

On the whole, software value can be seen as consisting two main categories of value aspects; that is, user based value perspective and business based perspective. The business value, for instance can include for example, production value, differentiation value, shareholder value, intellectual capital value among others (Lin & Tang, 2008). Thus it can be perceived as being more than just cost and profits and can also be created in many ways. Therefore, not every project needs to be profitable but must be strategic in a way or must support the business strategy to which the company was established.

From the customer perspective however, value could span issues such as whether the intended business need of the software is fulfilled or not. In other words, answering the question whether the software product is enabling business or not. If it falls short of this, then its true value would be questionable.

3.2.1.1 Value focus in this research

In this research value is perceived as ***the benefit realized in terms of profit or long-term business benefit that the software developer/vendor derives from a project, given the project overheads***. From an accounting perspective, profit can be defined as the amount by which revenues exceed costs (Ruppel**,** 2002). On the other hand, economists perceive earnings as the amount by which cash inflows exceed the costs associated with all of the factors of production (Polachek & Siebert, 1993). In essence, investment is about putting funds at risk with the hope of receiving a greater amount in return.

Charan, (2008) defines revenue as the total amount of money received by the company for goods sold or services provided during a certain time period. It also includes all net sales, exchange of assets; interest and any other increase in owner's equity. It is often calculated before any expenses are subtracted. Net income/profit can be calculated by subtracting expenses from revenue. Shareholders in any business are most interested in this net revenue since it is distributed as return on their investment. If there is no revenue coming into the company or if the there is revenue but there is no sufficient profits accrued from these revue streams, then management would be in trouble with the key stakeholders in the business. Therefore, to ensure that shareholders receive their return on investment and that also the business stays afloat, revenue has to be generated from most of the projects undertaken. While doing this, market opportunities must be balanced against organizational capabilities (Singh, 2008).

For those projects that are not revenue generating but may be strategic in positioning a company differently in the market, then adequate resource management on these projects must be ensured. They should also align appropriately with the business strategy at the operational level especially if

University of Cape Town

stakeholder/shareholder buy-in has to be solicited for future investment. Their true value in this regard would be in form of options to adapt in the future when currently uncertain conditions become better understood (Claude& Hobbs, 2006). Although, activities with proven ability to improve the bottom line should be pursued even if it warns of expensive impacts, risks on these projects need to be well managed. Otherwise, there will be minimal investment interests from the shareholders` perspective. The idea here is to help developers and managers to think more effectively about the business value of such options, and, in particular, how to increase the net present value of products, projects, and processes by designing in the right options as project success factors. Controlling risk in software projects is considered to be a major contributor to project success (Bannerman, 2008).

Therefore, intangible value benefits such as company image as well customer value is not measured or evaluated in this work. Rather, profitability as well as a strategic value benefit (as a company business value perspective) is evaluated. Various interrelationships between value drivers of difference perspectives in software engineering exist (Gorschek & Davis, 2008). These for example include: project, product and organization perspectives which influence the value drivers of a project directly or indirectly. This research approaches this challenge from a requirements engineering perspective.

In this work it is inferred that by using a valuable requirements engineering approach, risky businesses can be identified earlier on during the project selection process even before investment commitment is made on these projects. More still, by extending this value paradigm to support planning and control, anticipated value on projects can be well tracked and managed in real time. This is turn can give management the prospect of making informed decisions during times of investment uncertainty. This in turn ensures that adaptive risk mitigation is enhanced and that the anticipated project value at project inception is sustained throughout the development life-cycle.

The conviction here is that investment in any software project will be successful in creating business value only if there is improved organizational effectiveness and business competitiveness. This is driven by the view that the right project investment is chosen. Secondly, that this investment is linked to the right combination of process redesign, people skills and commitments and the like; whereupon this mix is well managed so that an effective organizational system emerges.

Software businesses therefore must place first priority to projects that may improve profit margins and create value. As literature suggests, various metrics of such measurement exist such as measures of internal rate of return or payback period (Newell, 1986). With such measures, investments that have managerial approval can then be pursued based on their future benefits and

University of Cape Town

strategic importance, with the view that they are expected to deliver economic profits and increase value creation.

3.2.1.2 Value creation

As Curriea & Mihir, (2006) observe, value creation in the software business is all about understanding the market space in which the business is operating, adapting to the changing trends in that market, consolidating the development processes (by understanding the economics that follows the software development lifecycle) and also by using appropriate techniques, tools and models that leverage value creation in its operations. Similarly, Boehm (2003) proposes a theory on Value-Based Software Engineering (VBSE) to support the above argument. In this theory, value is perceived as an integral aspect of the software development life-cycle where every activity is value generating. In particular addressing value derivation in requirements engineering**,** architecting, design and development, verification and validation, planning and control, risk management, quality management and people management.

These value principles are applied in this work to formulate the theory and also establish a definition for value creation. In particular, interest is ingrained in the value-based requirements engineering, planning and control as well as the risk management perspective. In this regard, value-based requirements engineering entails embracing principles and practices for identifying a system's success-critical stakeholders; eliciting their value propositions with respect to the system; and reconciling these value propositions into a mutually satisfactory set of objectives for the system. On the other hand, value-based planning and control includes principles and practices for extending traditional cost, schedule, and product planning and control techniques to include planning and control of the value delivered to stakeholders. Value-based risk management however, includes principles and practices for risk identification, analysis, prioritization, and mitigation as maintained by Boehm (2003).

3.2.1.3. Fix, grow, exit and exploit

The belief in this work is that when considering valuable projects for any given project portfolios, four courses of actions occur, that is, fix, grow, exit and exploit. When choosing any projects to add to a portfolio, projects that show significant return on investment and have significant opportunity for growth must be given first priority as such projects hold the greatest potential for value creation. On the other hand, projects envisaged having high returns but limited growth opportunities should be managed and exploited for the fact that cash flow from these projects can be used to fund more attractive investments. However, projects that produce poor returns but have significant promise should be fixed. Invariably, projects that demonstrate symptoms of low return yields and have limited promise should be exited and potential resources that would have been invested on such

University of Cape Town

projects be channelled to other investment options. Consequently, having a metric that offers this kind of visibility for effective decision making creates opportunities for executives to assess the merits of existing corporate strategy and also to form optimal strategies for future business direction.

3.3 Value drivers within software development

Executives today are faced with various questions when it comes to choosing valuable projects. As Wortman, (2005) suggests, the most prominent being "Should this project be carried out?" often followed by "Is it profitable?" This presents a binary decision-based predicament of whether to invest or not to invest. Answering such economic questions involves taking careful consideration and assessment on the project to alienate any uncertainty that may result in wrong decision taking and ultimately averting taking up risky business. In the absence of appropriate tools, models and techniques, it is often extremely difficult getting the appropriate answers to such strategic business questions. The practice in industry today however, is to overly apply heuristic expertise to gauge project value which may often have various limitations.

Similarly, in software development the need to identify and assess key value drivers on projects by use of tested models that determine the sensitivity of the company's performance, and hence its value, to changes in each area of its operations is inevitable.

3.3.1 Organisational performance

A wide range of views exists on the actual definition of organisation performance. Commercial software development however, advocates for organisational performance to be measured in terms of its business objectives realisation over time. If the internal processes driving these business objectives are not fully aligned with the business objectives, chances are that set targets may not be achieved, thus compromising the competitiveness strategy. Competitiveness is about cost and differentiation advantage leveraging. Cost leadership can be achieved through economies of scale and better relationship with suppliers while differentiation can be achieve through innovative products, improved streamlined production processes and better customer service among others. Both of which are fundamental in ensuring that a company succeeds in business.

3.2.2 Growth and competition

According to Porter's (1985) theory of competitive advantage, a firm's value growth and competitiveness is dependent on the value it is able to create for its buyers which exceed the firm's cost of creating it. The cost of creating this value entails realigning internal processes to render them effective so that a successful business strategy is ensured. Strategy can be understood as a management (planning) concept that is defined as an elaborated and systematic long term plan of

University of Cape Town

action designed to achieve the basic long-term objectives or a particular goal of an organization or an enterprise (Mintzberg et al. 2003). On the other hand, alignment has been defined as the extent to which the IT mission, objectives and plans support and are supported by their business counterparts (Reich and Benbasat, 2000). Alignment is an enabler in ensuring that the firm maximizes return on investment; achieves competitive advantage and also providing direction and flexibility to react to new opportunities.

3.3.3 Strategy and strategic alignment

Strategy is defined as the direction and scope of an organisation over the long-term (Liu *et al*, 2009). Its aim is to achieve advantage for the organisation through its configuration of resources within a challenging environment, to meet the needs of markets and to fulfill stakeholder expectations (James *et al*, 2005).Business strategy can be seen as the focal foundation of a successful business. Difference types of business strategy exist. On the whole, they should be meaningful and easily understandable by all employees and at the same time relevant to the organisation. However, as noted by Smaczny (2001), the most favorable business strategies must steer a course between the inevitable internal pressure for business continuity and the demands of a rapidly changing world for revolutionary business strategies.

Past studies such as Parnell, (2002) and Silvius, (2007), have highlighted the importance of middle and lower level managers in strategy formulation especially in ensuring that the strategy effectively "diffuses" throughout the organization. The remaining challenge though is whether prevailing participative management styles in an organization can enhance this process. Organizations must ensure that all the employees understand what the company mission and strategic objectives are since it is on these fundamental factors that the business strategy hinges. Without such mechanisms in place, the strategy will be rendered ineffective. Even if the firm uses contracted workers, ideally they must know that their absolute role is to support the success of this business. Therefore, their contribution must fit within what the prevailing business strategy suggests.

For instance if the organization`s business strategy is to ensure that they control a significant market share in the next couple of months in future, all production processes and the products thereof must provide those qualities that drive this strategy. It is the onus of the staff to ensure that this is achieved. For example, those in marketing must do their part while those in development must ensure that such quality issues are taken care of. Project managers on the other hand must ensure that all these qualities are achieved within the time and budget limits specified by business. This way, IT will successfully enable business other than being just a cost centre in the organization.

University of Cape Town

Smaczny (2001) maintains that a key success factor for a successful company is effective and efficient IT usage in supporting business strategies and processes. The alignment between business needs and IT capabilities is therefore still a prominent area of concern.

Although literature, for example Henderson & Venkatraman (1993), presents varying definitions of strategic alignment, in this work the definition suggested by Luftman (2000) is adopted. He defines strategic alignment as the application of Information Technology in an appropriate and timely way, in harmony with business strategies, goals and needs. It is also often understood to mean integration, cohesion, fusion, fit, and match and linked (Luftman, 2005). It is largely borrowed from military and adapted for use in business to achieve the firm's short term objectives and long term plans (Mintzberg, 1987; Sadler, 2003). However, in all cases, it concerns the integration of strategies relating to the business and its IT processes.

One of the current challenges facing most software vendors is the prioritization and selection of Project Portfolios that fully align with strategic business goals. Broadly interpreted, software development is driven by economics and it frequently requires economic models to describe and guide it, especially at this time of paradigm shifts of agile software development and also in bid to optimize the business project portfolio process. Business project portfolio is largely measured with the success rate and failure rate of the projects undertaken (Reyck *et al.*, 2005). Success would imply that a given project yielded viable benefit to all key stakeholders (Barclay, 2008). Taking the vendor perspective, it could imply that relative return on investment was realized, customer trust was established or even new technology was successfully integrated in real business applications. Failure in this regard would be the reverse of success, the main causes of this being scope creep problems, inaccurate size estimates, budgetary inflexibilities, development capacity and competency to deliver the project on time and within budget as well as unforeseen and uncontrollable circumstances arising during the project execution lifecycle (Verner *et al.*, 2005). This therefore creates a need for cost effective production approaches geared towards project failure mitigation such as the agile methodology.

University of Cape Town

3.4 Agile paradigm

3.4.1 Traditional approaches to software development

Traditional software development approaches advocate for extensive planning, codified processes, and rigorous reuse to make development an efficient and predictable activity (Boehm, 2002). In this traditional approach, emphasis is directed towards a rationalized engineering-based approach where problems are perceived to be fully specifiable and that optimal and predictable solutions exist for every problem (Dyba, 2000; Nerur *et al.*, 2005).

3.4.2 The difference with agility

By contrast, agile processes address the challenge of an unpredictable world by relying on people and their creativity rather than on heavy processes per say. As Agerfalk & Fitzgerald (2006) maintain, methods for agile software development constitute a set of practices for software development that have been created by experienced practitioners. However, these measures have not gone without criticism from both the practitioners and the academic community. There is still ongoing debate on issues such as, the instances where Agile makes sense compared to traditional methods and also how the combination of Agile with traditional process can be aligned to better address a specific situation.

With such criticism however, highbred approaches of the agile development methodology often referred to as lightweight approaches are beginning to emerge. In this investigation, early feedback and change management is perceived as being pivotal in achieving project success and ultimately realizing value creation on projects. Equally, appreciating the need to react to mistakes and the fact that requirements may well change during the development life-cycle is instrumental in guiding the development process.

3.4.3 Rationale for taking an agile approach

In this framework therefore, the Adaptive Software Development method proposed by HighSmith, (2000) and the Lean development approach (Womack *et al.*, 2007), which is rooted in the Toyota Production System (Ohno, 1998) from the 1950s, was adapted to formulate a highbred approach.

The argument advanced is that we need to eliminate waste, achieve quality first time, and focus on problem solving rather than heavy emphasis on the design and the heavy development processes. Some of the founding principles of agile manufacturing are the integration of customer-supplier relationships, managing change, uncertainty, complexity, utilizing human resources, and information (Gunasekaran, 1999; Sanchez & Nagi, 2001). Agile methodologies are developed to embrace, rather than reject, higher rates of change (Williams and Cockburn, 2003). The observation here is that in most cases, increasing feedback, not decreasing it, is the single most effective way to deal with troubled software development projects and environments.

With the rapidly changing market environment, characterized by evolving requirements and tight schedules, software developers ought to take pragmatic and adaptive engineering approaches such as the agile approach (Cao & Ramesh, 2007); (Grace *et al.*, 2008). Agility ultimately, is about creating and responding to change. It addresses two pressures that characterize today's business and technology world: the need for dynamic, innovative approaches and the desire to build dynamic development processes. Agile development excels in exploratory problem domains—extreme, complex, high-change projects—and operates best in a people-cantered, collaborative,

University of Cape Town

organizational culture (Highsmith & Cockburn, 2001). With an overall aim to optimize the internal economies of scale and subsequently influencing market forces, agile development implements the highest priority features early so that customers can realize the most business value (Cao & Ramesh, 2008).

A guiding principle behind agile software development is the idea of keeping things simple and taking advantage of experience (Hunt & Thomas, 2003). Highsmith & Cockburn (2001) maintain that agile processes are designed to capitalize on each individual and each team's unique strengths. The emphasis on people and their talent, skill, and knowledge is important. The market trends today demand that we produce products under a tight schedule and usually under a tight budget (Ooms, 2008). Typical in small development teams, time and again project managers achieve this by making a compromise. They overload experienced programmers, since these are better able to complete the tasks than the inexperienced programmers would be, a choice that usually leads to productivity bottlenecks in that resource overloads may arise. Resultantly, project managers are now advocating for Extreme Programming techniques to improve the skills of programmers and to match the agility in software development. More still, others are aiming at maintaining a qualified team so as to increase resilience in production.

3.4.4 Software development agility – a determining factor for success

Past studies in this area for example Shaw, (2005) and Jiamthubthugsin & Sutivong, (2006), have validated the notion that software engineering entails making decisions under constraints of limited time, knowledge, and resources. Therefore, there is a need for improving our ability to systematically and predictably analyze and develop sophisticated software in the shortest time possible so as to satisfy complex, ill-specified requirements that evolve during the project development lifetime as suggested by Deng *et al*.,(2003).

Several authors have noted that the greatest pain with traditional software engineering approaches is that they offer limited support for the development of intelligent systems that support value creation (Raybould, 1995); (Lawrie & Gace, 2002). Sterling & Juan, (2005) maintain that to handle the tremendous complexity and the new engineering challenges presented by intelligence, adaptiveness and seamless integration, developers need higher-level building blocks, or development constructs. These constructs must support value creation. Intelligent control is a critical enabler for realizing increases in productivity and quality during production (Balakirsky & Lacaze, 2002).

In this era of agile software development, new methods geared towards value-based software engineering are of prime importance. Industry needs frameworks created for evaluating early-stage designs to predict the value a user will see in an implementation. Early design evaluations predict

University of Cape Town

properties of artifacts that will result from a proper implementation of the design; the predicted properties can include costs as well as functionality, performance, and quality measures (ITRD, 2001). In this case, consideration is made of the cost property against other properties of the project artifacts. Some software engineering approaches include evaluation techniques such as these, however, there is still a lack of predictors for many properties of interest, and generally, we lack a systematic way to explain, compare, develop, and apply them.

3.4.5. How CASSE Compares to SCRUM methodology

Ambler, (2008) maintains that SRUM supports business value driven software development. In this approach functionality is delivered iteratively and incrementally by self-organizing teams (Schwaber & Beedle, 2001). This is all driven by business priorities. Often, within the first thirty days of the start of the project, useful functionality is delivered. SCRUM lets requirements, architectures and designs emerge throughout the project instead of waiting for requirements finalization or when detailed architectural designs and validations are finalized. Dingsøyr, (2006) maintains that SCRUM provides the manager direct visibility into the ongoing development of a product so that one can steer the process in real time, making tradeoffs between cost, time, quality, and functionality to deliver the most important customer functionality as soon as possible. This thus enhances managing the very complex process of product development and control.

However, SCRUM has been found to have major limits especially when it comes to vision and strategy. It fails to illustrate how it fits into company's vision or how they work together. Due to this limitation, it always therefore requires a strategist –someone responsible for creating a level playing field where all stakeholders and participating parties in business can contribute and add more value (Henrik & Nielsen, 2008). This role ensures that the ever-changing market trends, sales, and the organization as a whole is evaluated on a continuous basis to make sure that what engineering builds is not just feasible but valuable.

Singh (2008) observes that in SCRUM, product goals are set without an adequate study of the user's needs and context. The user stories selected may not be good enough from the usability perspective. Secondly, user stories of usability importance may not be prioritized high enough (Marchenko & Abrahamsson, 2008). Thirdly, given the fact that a product owner thinks in terms of the minimal marketable set of features in a just-in-time process, it is difficult for the development team to get a holistic view of the desired product or features (Diaz et al, 2009).

CASSE Framework strives at bridging this gap. It doesn't require a strategist role. Given that the strategic project evaluation parameters are set on the project evaluator implies that company vision is put into context. Project performance is monitored and observed in the different quadrants. In

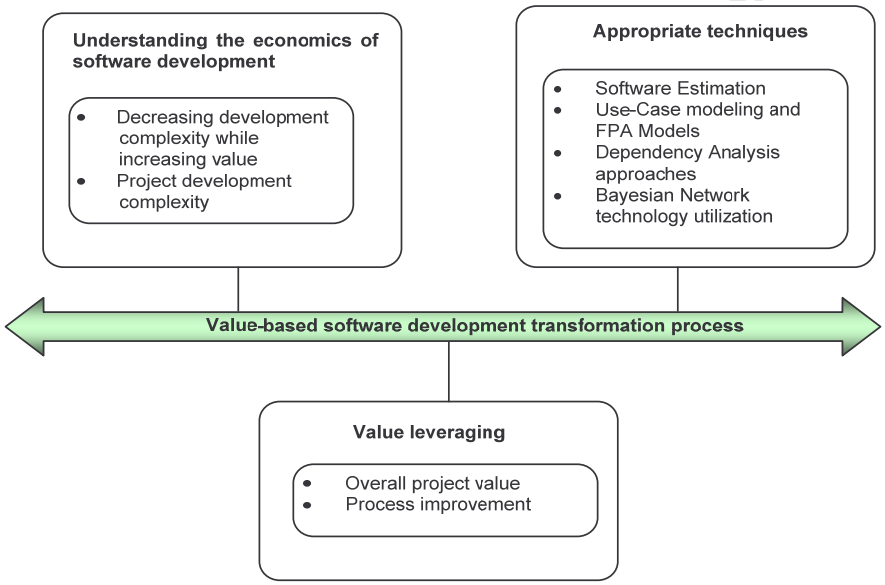
University of Cape Town

case the company strategy and vision is changed or revised, this can easily be mapped to the project evaluator by configuring the scoring parameters of project evaluator. This changing vision and strategy will be visible to all stakeholders at every level of the development process. The CASSE Framework doesn't presuppose that it is the role of one person in the organization to monitor this, rather a collective effort unless a particular firm chooses to do so.

3.5 The sustainable chain of value creation on software projects

In order to match with the market demands therefore, this work views the value-based software development transformation process raising three focal nodes of sustainable project value derivation and management as illustrated in Figure 6 below. These include: understanding the economics of software development, use of appropriate techniques and value leveraging.

Figure 6: value-based software development transformation process



In this analogy, it is argued that understanding the economics of software development entails, understanding the development complexity and reducing this complexity while increasing project value. On the other hand, value leveraging is about exploring the overall project value and improving the process with agile methods such as Extreme Programming, use of management-oriented development models for effective feedback management and control, coordinating projects appropriately and deriving value-oriented tasks on projects. The last block looks at appropriate techniques that can be adopted in this value mapping strategy. This involves use of value-based models that enhance accurate software estimation. At the same time any surprises on projects can be handled. This handling process can be through adaptive techniques such as Bayesian Network technology or even undertaking dependency analysis of key project properties, in this case requirements.

3.5.1 Understanding economics of software development

Software Engineering Economics approximates concepts from economic sciences or corporate finance theories to the software development context. It supports managers and investors who work in the software industry in making better decisions about their projects, increasing profits or minimizing losses (Costa *et al*., 2007). However, the economics governing most of the software development projects still leave much to be desired. We still do not understand adequately the economic parameters under which investments in the use of formal methods create value. Past work on software economics (Boehm & Sullivan, 2000) addressed mainly the issues of cost and schedule estimation and control but not overall value creation. As observed by Sullivan (2007), a large part of the problem is that software developers continue to make key technical decisions without an adequate ability to reason about their economics. This problem is rooted in the lack of formal, testable and tested theories, methods, and tools for supporting economic-based analysis and decision-making -more broadly, value-based analysis.

In considering economics in managing and developing software in the agile environment, two underlying issues arise:

a. We have to simultaneously decrease the development complexity on projects while increasing project value by utilizing viable estimation approaches on these projects.
b. If we are to take on projects, development is critical to the success of projects: we must coordinate our projects well by utilizing appropriate scheduling techniques, and deliver project tasks that derive value for key stakeholders on the project, while mitigating project risks iteratively.

3.5.1.1 Decreasing development complexity while increasing value

Over a decade ago, Babu (1996) projected that the rise in demand for software in turn requires increases in productivity of the development process and quality of the software product, and the general success of the software development project. This belief has been taken on actively in the research circles, for example Dossani (2006). However, productivity can only be achieved if self-adaptive project management is utilized. Furthermore, while effective project management is important to successful development of software, the challenge is to develop more complicated software products within the constraints of time and resources without the sacrifice of quality and value.

Most proponents of software economics, such as Sullivan (2007) and Harrison, (2006) contend that software is the source of great economic value, but also of much economic pain, especially if the economics surrounding its development are not dynamic or self-adaptive.

University of Cape Town

Mantyla, (2004) observes that most accounts of software problems focus on flaws in technical design and usability. Surely better project management and requirement understanding can help resolve software development problems. Specifically, this can be achieved by firstly adaptively managing the structure of the technical and organizational processes that produce the software. Secondly, by also ensuring that the technical and organizational infrastructures and constraints under which it is built and maintained are well managed.

3.5.1.2 Complexity concerns in software development

Several studies, for example Solomon (2004); (Emam & Koru, 2008), have shown that it is a common occurrence for software projects to fall behind schedule in terms of validating requirements, completing the preliminary design, meeting weight targets, or delivering software releases that met the requirements baseline. For most projects, in the initial stages of project development, some of these adverse conditions are not described as problems or issues. They are rather classified as *risks* towards achieving subsequent objectives.

Reyck *et al*. (2005) maintain that there are still major challenges making software development and management very complex. These challenges include:

- providing a centralised view of all the projects in an organisation by enabling financial and risk analysis of projects;
- modelling interdependencies between a family of projects by incorporating constraints on resources shared between projects;
- enabling prioritisation and selection of projects by ensuring accountability and governance at the portfolio level, allowing for portfolio optimisation; and
- providing support in the form of standardised processes and software tools.

The trend towards increasing use of software products and Information Technology (IT) in general, across all sectors continues. The challenge however remains how to better manage IT projects in order to maximise their economic benefits. Part of that challenge can be tackled by ‘‘doing projects right’’ and partly ‘‘doing the right projects’’ (Linenberg, 2003).

Indeed software development is a difficult activity and the software industry is notorious for project failure (Jackson, 2003). The main culprit is a complex environment and an insufficient understanding of business requirements (Belmiro *et al*., 2000). Traditional models of software development try to solve this problem with a heavy emphasis on upfront design (Coopera *et al*., 2005). Proponents of this approach believe that if every detail of the proposed system is accurately captured and documented, the successful development of the system will be ensured. Possibly, the problem with

University of Cape Town

this approach is that it unrealistically expects that the final system is well understood by both the client and the developer at the start of the project. In addition, it demands or expects that requirements remain fixed throughout the development life-cycle. However, requirements often change as the understanding of the system evolves (Woit, 2005).

3.5.1.3 The prevailing limitations

As a software engineering community, we are less able to build systems from specialized, efficiently produced, volume-priced third-party components than is possible in many other fields. The inability to manage risk by the use of market mechanisms is a major hindrance to efficient production. The use of formal methods, or the shape of a given architecture, is a technical issue. For a system to create value, the cost of an increment should be proportional to the benefits delivered (Benjaoran, 2008). But if a system has not been designed for change, the cost will be disproportionate to the benefits. Design for change is thus promoted as a value-maximizing strategy, provided one can anticipate changes correctly (Boehm & Sullivan, 2000).

3.5.1.4 Probable solution

Dynamic monitoring and control mechanisms therefore are needed to better guide decision-makers through the design space in search of value added over time. These mechanisms have to be based on models of links between technical design and value, and on system-specific models. These system- specific models are based on better cost and payoff models and estimation and tracking capabilities, at the center of which is a business-case model for a given project, program or portfolio**.**

Costa *et al*. (2007) agree that economic concepts can greatly support the development process in the effort to better quantify the uncertainties of either a single project or even a project portfolio. Such concepts must be generic enough to embrace both traditional and agile methods of software development.

3.5.2 Leveraging value on software projects

3.5.2.1. Overall project value

Eisenhardt & Brown, (1998) argue that it is essential for a company engaged in software production business to understand how to link business management and software development. Equally, Vähäniitty, (2005) asserts that such companies must also employ a solid, value-based approach in their decision making. Madachy (2006) suggests that business value goals should be considered when making software process and product decisions, but it is usually difficult to integrate the perspectives quantitatively. Optimal policies depend on various stakeholder value functions, opposing market factors and business constraints. To achieve real earned value, business value attainment must be a key consideration when designing software products and processes.

University of Cape Town

Both parties involved in system development, that is, the client and the vendor, must benefit at the end of the development cycle. For the client, the system must offer the expected requirement as initially agreed at project inception. For the vendor, the project must offer strategic business options either in the medium or the long term. Otherwise it must have considerable return on investment. With regard to business principles, it is uncommon to find a business whose business model and target are only to break even. Rather, in order to stay afloat, a business would always make a priority of selecting valuable and profitable projects that validate its existence.

The current trend in industry seems to suggest that we are now looking outside traditional boundaries and developing ways of achieving maximum business value from our investment in requirements (Favaro, 2002); (Linetsky, 2008). Good business decisions are based on facts elicited through careful evaluation of key elements of the business situation—for example, market, competitors, technical feasibility, strategy, intellectual property, product quality, and resource availability (Wallin *et al*., 2002). Without a thorough understanding of the business context, we tend to operate in a fog, believing things happen arbitrarily by chance and cannot be predicted (Harrison, 2005). Consequences of project overruns frequently drive the quest for repeatability and predictability. Erdogmus (2005) asserts that predictability refers to the much sought-after property whereby inputs determine cost and schedule.

University of Cape Town

3.5.2.2 Business value

Patton (2008) maintains that business value is something that delivers profit to the organization paying for the software in the form of an increase in revenue, an avoidance of costs, or an improvement in service. If we share a common idea of what is valuable, then we will not pull in opposite directions. As Sullivan (2005) notes**,** software development is not only driven by economics, but it increasingly drives economics. Yet, we lack models that account for or that can effectively form software engineering as an economic activity. In particular, we lack models that can help us decide which choices will be most productive of overall value.

Software engineering is a decision-intensive discipline. Many predictor models already exist but information about them is proprietary. So model calibration in different development environments is difficult (Harrison, 2005). We need to build simple business-goal models that concisely articulate and prioritize business goals.

The way we select our projects is therefore significant. The way we understand client requirements on these projects is also very important. Given the agility of software development, there is a need to ensure that both parties agree on the same requirements and that any changes suggested along the development lifecycle are well articulated and communicated. However, such changes must be

well negotiated and assessed as to how they impact on the overall system. Also, existing software development models mostly adhere to the viewpoint of individual development projects and thus fail to sufficiently address common product development concerns such as multi-project or project portfolio management (Vähäniitty & Rautiainen, 2005).

The notion of different cycles (or levels) in managing software development may also help in choosing and combining practices from existing literature for process improvement purposes. While literature on software engineering (or managing new product development or strategic management, for that matter) is not lacking in methods, techniques and tools, practitioners tend to view these as disconnected from "real life" (for example, in the case of requirements prioritisation) (Lehtola *et al*., 2004). It is therefore argued that relating the economic models and tools to their practical context more closely could prove fruitful. This kind of mapping may also be helpful to readers looking to solve a particular problem in their own context.

3.5.2.3 Process improvement

Effective process improvement requires use of efficient ways of production such as the Extreme Programming (XP) approach. It also necessitates adoption of management-oriented models that effectively communicate stakeholder value on projects. Similarly, integrating formal decision making mechanisms as well as implementing proactive project coordination approaches is instrumental in ensuring a successful development process.

3.5.2.3.1 The Extreme Programming (XP) paradigm

Lippert *et al*. (2003) define XP as a disciplined approach to software development that emphasizes customer satisfaction and teamwork. XP principles and practices support iterative development, with the whole team revising and refining the solution as they increase their understanding of market needs (Mugridge, 2008). It is designed to deliver the software a customer needs when it is needed, thus increasing project commitment from the customer side. It is a methodology that requires that the process of software development be adapted to market needs so as to achieve business excellence and competitive advantage (Dyba, 2005) ;(French & Bell, 1999). Organisations that have perceived the need for customer satisfaction and teamwork are already starting to use this methodology.

In this methodology implementation is per feature. The economics surrounding this implementation therefore involves evaluating:

- the type of personnel that will undertake this implementation, whether experienced or not,
- how long this implementation will take,

University of Cape Town

- how much it will cost to implement this feature,
- how well this feature integrates or interrelates with other feature on the project, and
- whether the client is willing to pay for this implementation cost.

Any misunderstanding or disagreement on such crucial project economic properties poses risks for project revenue and profitability levels already from the onset, and must be averted. Negotiating on such project properties right from the onset would ensure that both parties would eventually agree on the scope, schedule and overall budget of the project - provided all system features go through the same cycle of analysis.

3.5.2.3.2 Management-oriented development models

Upper management commonly fear that a project will go on forever and prefer a model in which project budgets are approved and the project remains within the budget confines (Cohn & Ford, 2003). If the software engineering discipline is to evolve and reach maturity, we must have concepts, tools or models that guide us through the development process in order that we are better able to produce models that management can accept with ease.

Boehm *et al.* (2007) suggest that in the agile software development process, cost, schedule and quality are highly correlated factors. They basically form three sides of the same triangle. Beyond a certain point, it is difficult to increase the quality without increasing either cost or schedule or both for the software under development. As products and applications mature, users expect higher quality products. They want IT organizations to be responsible and accountable for the quality claims made by the product marketing teams.

Integrating new practices with existing processes and quality systems that govern software development processes requires further tailoring (Lindvall *et al*., 2004). Better software practices improve predictability of costs and schedules, reduce risk of cost and schedule overruns, provide early warning of problems, and support better management (McConnell, 2002). To fully benefit from agile practices, organizations must better define the interfaces between the agile team and its environment, thus avoiding the double work caused by the conflict between agile practices and traditional ones.

3.5.2.3.3 Formal decision making techniques

Heller (2000) observes that formal decision-making techniques could be used, not just for the economic and technological analyses they produce, but also for focusing attention and interaction on organizational issues. Complex adaptive systems can help especially in requirement prioritization and management. In a complex adaptive system, decentralized, independent components interact to create innovative, emergent results (Highsmith & Cockburn, 2001). Practices that support existing

University of Cape Town

processes might include prioritizing requirements so as to keep projects on schedule when new requirements emerge (Boehm & Turner, 2005).

Given that the field of software economics views software development as a value-generating activity, which must benefit the clients and developers, there is a need to analyze all phases of software development. As we address value creation challenges in the enterprise, given the agility of software development today, key project phases such as business case analysis, project evaluation, process selection, assessment of new technologies must be dealt with.

3.5.2.3.4. Proactive project coordination approaches

Software engineering project work is increasingly becoming a highly cooperative activity, and promises to continue in this way. The coordination of projects demands managerial control over time and resources. The more complex the project is the more expensive and time-consuming it becomes (Wateridge, 1999)**.** Project coordination and planning should consider the availability of good resources along with commitment to customers and the organization. A well-defined plan where all facts and practical aspects have been determined is the key to success.

Planning and executing are two different functions. Planning requires that consultations be held with those who are actually going to execute the plans, cross-checking facts and calculating risks using a good dose of practicality. If commitments to customers are broken, whatever the reason may be, it becomes one cause for the client to drift away from the project. Cost factors should be considered and balanced, and previous experience and best practices taken into account so as to refine the development approach.

Tracking tasks, compiling costs and expenses, and managing people involved in business projects are all made more manageable when using a project scheduling technique that views requirements as value-generating task with complex interdependencies. The way we schedule our resources on the project, mandated with various tasks, determines their productivity and overall performance and ultimately, the success of the project (Zhang *et al*., 2006)**.** The objective of the agile development process is to create software development and maintenance processes that maximize the agility of application development. Project failure directly impacts upon customers by lowering customer satisfaction, while giving competitive advantage to our competitors. A collective and systematic project management approach is required to substantially increase the project's success rate and help to minimize the loss of benefits.

Projects are often driven and defined by customers. Management need to review all existing projects in relation to strategy and if these projects are not contributing to strategy, they should be dropped. Benefits management which applies project profitability analysis as well as business

University of Cape Town

strategic evaluation is therefore important. Organizations need to make the business case for the benefits that each project can bring. Benefits analysis and evaluation need to be part of the metrics that senior management use to measure the business. Success of any project must be the objective for those individuals who are making decisions for the business.

3.5.2.3.4.1 Deriving value-oriented tasks on projects

The process of deriving value-oriented tasks on projects necessitates understanding the nature of requirements from which these tasks can be derived. It also mandates modeling these requirements conceptually with an aim of evaluating all possible solution scenarios. Similarly, the use of proactive project scheduling mechanisms that guarantee delivering a project on time and within budget is inevitable. On the other hand setting indicators for productivity measurement is crucial in ensuring that quality and value is sustained on these projects.

3.5.2.3.4.1.1 Understanding project requirements

Most researchers, in particular Nagappan (2005) and Kan (2002), agree that poor requirements are the biggest single source of defects in software projects. Poor requirements lead to weak estimates and over-ambitious project plans. There is enough evidence to suggest that during the early stages of the development process, most software projects are already in trouble, that project managers are overly optimistic in their perceptions, and that executives receive status reports very different from reality, depending on the risk level of the project and the amount of bias applied by the project manager (Snow & Keil, 2002). Key findings suggest that executives should be skeptical of favorable status reports. Rather, they should concentrate on decreasing bias if they are to improve the accuracy of project management and reporting (Snow & Keil, 2001).

The software business is surely about satisfying and sustaining our clients. We can only achieve this if we firstly understand project requirements and secondly improve our reporting techniques and mechanisms on each project. However, many questions still arise as we turn our endeavors to realizing this. If we understand the client requirements well, how can we selectively implement requirements that generate value to us as a business and the clients we serve on the project? How can we analytically verify those requirements that increase our clientele commitment to the project while accelerating milestone acceptance? How can our scheduling and resource allocation mechanism help us realize value on the project throughout the project life-cycle?

The approach used in this study is aimed at addressing these issues using both static and temporal probabilistic modeling and reasoning as a guiding factor in the project design space.

University of Cape Town

3.5.2.3.4.1.2 Use-case artifacts

Project tasks can be driven off the use-case list and every step in the life-cycle derives benefit from this list. If a project is approached from this perspective, use-case offers rewards for each group, including analysts, project managers, clients, testers, designers, estimators and programmers. The factors that drive all scheduling are the project requirements. Requirements play a vital role in enhancing value creation in the properties and value attributes of an entire project. The way we allocate time to these requirements must derive value in order to mitigate the unforeseen challenges on the project (Fidel *et al*., 2007). If a project is likely to operate within the break-even space, without necessarily making profit, it is imperative that experienced programmers be scheduled while the inexperienced ones undertake the testing especially if it is a small team where tasks cut across. If the project is likely to make a profit, it is essential that a combination of experienced programmers with juniors peer program so as to disseminate knowledge and build human capital development on the project.

3.5.2.3.4.1.3 Proactive project scheduling mechanisms

We need to firstly understand that there is complex interdependence between the main tasks of the project in this case the requirement features; which could be implicit or explicit. These dependences have significant bearing on the overall project value and profitability. The way we analyze our tasks must incorporate dependency matrices which result in either parallel task implementation or sequential scheduling. The matrices developed can then be implemented in a Critical Path Method (CPM) for further scrutiny and project first tracking.

Critical Path analysis is a mathematically based algorithm for scheduling a set of project activities. Over the years, it has been proven to be an important tool for effective project management (Wheelwright, 2002). Using the requirement dependency values, CPM would calculate the longest path of planned activities to the end of the project. The earliest and latest that each activity can start and finish without making the project longer will also be determined. This process determines which activities are "critical" and those which can be delayed without making the project longer. Thus, lengthening the duration of the project would accordingly increase its risks. If we preventively manage risk using such dependencies, we could allow for flexibilities in the development processes (Murthi, 2002).

As Stijn (2008) suggests, during execution, a project may be subject to considerable uncertainty, which may lead to numerous schedule disruptions. Activities can take shorter or longer than primarily expected, resource requirements or availability may vary, new activities might have to be inserted. Proactive project scheduling is therefore needed to build stable baseline schedules that are able to absorb most of the anticipated disruptions during project execution, hence managing the

University of Cape Town

intractable risks and subsequently deriving project value (Lambrechts *et al*., 2008).

The result of the CPM will allow managers to prioritize activities for the effective management of project completion, and to shorten the planned critical path of a project by pruning critical path activities, by performing more activities in parallel, and/or by shortening the durations of critical path activities by adding resources; hence preparing for scheduling and resource planning effectively. The identification of the critical chain of events would make it possible to mitigate potential negative effects since risk lists of the project can be generated as a result of sensitivity analysis. An understanding of these challenges would help evolve an appropriate project management model to address the market needs and challenges.

The idea is to identify potential steps in development and deployment that can delay the project or cause it to fail, and devise strategies that will mitigate the risk, should it materialize. Unlike prescriptive risk management, in which the team identifies risks only when they occur and attempts to mitigate them only after the fact, with CPM the team identifies and evaluates risks and devises mitigation strategies throughout development.

3.5.2.3.4.1.4 Setting productivity indicators

It is easier to quantify work done per functionality—which may occasionally be synonymous with requirements—than entire components at a time. Analyzing and achieving functionality in a bottom-up, micro value to macro system value approach, derives value for all stakeholders on the project and usually accelerates functional acceptability and project signoff (Morales & Barra, 1977). By using this approach you can gauge how productive the resources on the project are and how the project is progressing. Equally, milestones can be evaluated and attained easily, and issues that usually cause IT projects to fail, such as inability to deliver products on time, would be mitigated.

3.5.3 Appropriate techniques

Appropriate techniques of value determination and sustainability entail utilization of relatively accurate estimation methods. These methods are often geared towards handling any surprises or emergencies in this regard. More still, they involve using value driven models to predict and assess properties of a software project at each and every level in the development lifecycle.

3.5.3.1 Software Estimation

Gencel & Demirors, (2008) observe that software estimation in terms of size measurement, effort required and cost thereof continues to be a significant practical problem in software engineering. Estimation requires understanding the building constructs of software complexity including Functional Points (FP) assessment and overall interaction dependencies evaluation. Whereas FPs are derived from use-case artefacts, on the other hand dependencies are determined by the level of

University of Cape Town

interaction between FP calls during program execution time. Software engineering cost (and schedule) models and estimation techniques are used for a number of purposes including budgeting, tradeoffs and risk analysis, project planning and control as well as software improvement investment analysis(Boehm & Abts, 2000).

Various organisations engaged in the business of software development today struggle to accurately estimate the enormity of projects and the likely amount to be quoted for on such projects. To overcome this problem, various approaches have been developed including the metrics and methods on measuring the functionality attribute of a given project (Sureshchandar & Leisten, 2006). This approach was first proposed by Albrecht (1979). It has since been improved by several authors; specifically investigating how Use-Case Point methods or Functional Point Analysis (FPA) techniques can be utilised to estimate software projects accurately.

3.5.3.2 Use-Case modeling and FPA Models

Use-Case modelling is an accepted and widespread technique to capture the business processes and requirements of a software application (Clemmons, 2006). Use-cases are often used as input for estimating software development effort (Anda *et al.*, 2005). Actors are reflected as role-players in the envisaged system with various tasks executed as functions in the system. In systems modelling they are represented as a group of Objects containing various functions, interacting in a specific fashion to achieve various system goals. The functional point technique extended in this work provides the ability to estimate the man-hours a software project requires from its use-cases.

Garcia & Hirata, (2008) define a FP as a metric whose objective is to determine the software size by mapping the identifiable user functions onto software functions that allows the sizing estimative to be free of technology and technical factors. This heuristic is highly important in estimating software projects. As Gencel & Demirors,(2008) observe, we need to estimate how much software to build, just as we need to determine the weight and volume of an engineering product as part of the planning process. To the least detail this effort could be estimated in terms of software line of code Fadi *et al.*, (2005). With the objective of enhancing decision making, Li & Ruhe, (2007) maintain that the idea of offering decision support always arises when decisions have to be made in complex, uncertain and/or dynamic environments such as the one experienced during agile software development.

Boehm(2003) observes that a lot of work that has been done in this area seems to be concentrating on a neutral value setting, whereby every requirement, use case, object, and defect is treated as equally important. However such approaches have their own limitation in governing the optimization of the process of software development and overall project value limits. In addressing

University of Cape Town

the need for overall project value management, several models have been suggested in previous studies. They have however looked at project cost as a function of project complexity. Specifically, in COCOMO II, a tool widely used in industry today and initially developed by Chulani *et al*. (1998), it aggregates project complexity as a function of inputs, outputs, interfaces, files and queries required by a software system. All these inputs form the basis for determining transactions that have to be fulfilled by Actors in the domain model. Karner (1993) proposed the use-case Point method to analyse the use-case Actors, scenarios, and various technical and environmental factors and abstracts them into an equation. The functional point method implements this approach and is the most widely adopted method for performing the functional size measurement of software.

To complement this approach, several authors have suggested various techniques; in certain instances suggesting alternatives, while in others extending and supporting it. For example, Mohagheghi *et al*., (2005) propose an estimation method based on use-cases so as to assess the complexity of use-cases. Lavazza & Bianco, (2008) propose a problem frame-based method to FP Analysis in which requirements are represented by heterogeneous sets of documents and notations. Levesque *et al.,* (2008) present a COSMIC-FFP model which implements use-cases and Actor-Object sequence diagrams of a system application to determine the number of messages exchanged as a basis to estimate the number of function points in the system.

Although all these models are useful in assessing software estimation, they however do not address emergence management of predicted models. Requirement features have dependencies which could be explicit or implicit in nature. This dependency analogy requires refinement using adaptive models that manage these surprises in software project selection and overall management.

In this regard emergence is understood to mean the evolving nature of complex issues on software engineering projects. Usually such complexities arise from project requirements, specified as features and modelled as FPs represented by different Actors in the system. The dependencies between these requirements greatly determine the risk levels of the project and the implementation priorities thereof. Understanding and managing these dependencies, which often evolve iteratively, requires sophisticated techniques and tools to ensure that a project is well tracked throughout the development cycle and that it continuously generates value as anticipated.

3.5.3.3. Dependency analysis approaches

Dependency analysis in systems development is based on systems features. Over the years, it is a paradigm that has attracted a lot of attention in industry and in the research community specifically for product line planning (Clements & Northrop, 2002). This approach guides organizations towards

University of Cape Town

the development of products through the strategic reuse of product line assets such as architectures and software components.

Several authors investigating this area, for example Keck & Kuehn, (1998); Calder *et al*., (2003); Robinson *et al*., (2003), agree that the goal of system feature dependency analysis is to detect possible interactions between features and to provide techniques for resolving the interactions. Therefore, in order to minimize surprises in project estimation and to attain optimal project effort estimates requires that some effort is put into feature dependency analysis. These features could be public libraries or libraries already existent on the project database of a given organisation. Prior work shows that several techniques have been suggested to handle this emergence.

Suliu & Rule (2007) provide a decision support technique that assesses system feature implementation trade-offs in evolving systems, specifically using the bi-objective optimization approach. The heuristic proposed in this model is that product release planning can be well optimized if features dependency assessment is well attended to as a mechanism to minimize development effort. Schätz, (2007) provides an integrated model for both variability and domain-specific aspects by explicitly introducing variability into system descriptions so as to optimize feature dependency analysis. Freimut *et al*. (2005) provide a model for use in assessing the cost-effectiveness of inspection processes. While these models estimate costs incurred during the software development phase, other models consider issues related to post-development phases. For example, Ostrand *et al*. (2005) provide a model for use in estimating the fault proneness of systems. Wagner (2006) provides a model for use in assessing the costs and benefits of defect detection techniques.

Lee *et al*., (2004) propose a feature-based approach to product line production planning so as to identify core assets and develop a product line production plan. Akker *et al*., (2008) present a mathematical formalization of release planning in which integer linear programming is utilised. Their model assumes that an optimal set of requirements is the set with maximal projected revenue against available resources. Conejero & Hernández, (2008) propose a framework in which requirements are modelled to build a traceability matrix which represents the relations among features and non-functional requirements and the use cases.

To assess methodologies and predict their costs and benefits, effective economic models are needed. Successful software product development requires that the project use both a business decision model and a viable software development life-cycle model. All these models highlight the benefit for optimizing the production process by leveraging the existing assets. Although the approaches proposed are geared towards the production stage of software development, this

University of Cape Town

technique would equally be ideal and very valuable even during project selection; given that we may price new system features based on existing features.

The CASSE frame discussed in the next chapter utilizes such proposed techniques to derive a value-based Actor Object Model. It also extents these approaches by applying emergence handling in the model so as to predict realistic efforts required on a given project.

While these models offer great insight into the value-based paradigm, however, the way these models are applied is dependent on the business strategy as well as the investment involved in establishing and integrating them into the production process. Economic models such as these cannot be created and assessed in a single step. Rather, they must be refined and improved over time by incorporating new techniques and approaches that make them more extensible. This aspect therefore raises the need for integrated predictor models and techniques that give us insight into project complexities right from project inception so as to mitigate project risks and henceforth increase business value. A probable solution is the utilization of the Bayesian Network technology.

3.5.3.4. Bayesian Network technology utilization

The Bayesian Belief Network (BN) paradigm is derived from the Bayes theorem. Farmanim *et al*, (2009) define a BN as a probabilistic graphical model that represents a set of variables and their probabilistic relationships, which also captures historical information about these dependencies. Traditionally, a BN is perceived to be a directed acyclic graph (DAG) $G = (V,E)$ (where V is a finite set of nodes, and E is a set of edges, that is, $E \subseteq (G \times G)$ together with a joint probabilistic distribution P that satisfies the *Markov condition*, namely that each variable $x \in V$ is conditionally independent of the set of all its non-descendents given the set of all its parents (Neapolitan, 2004).

Over the last decade, this approach has gained popularity as a representation for encoding uncertain expert knowledge in many applications, including medical diagnosis system, gene regulatory networks, among others (Heckerman *et al*., 1995). Additionally, recent developments in this area provide new opportunities for developing better performing tools (Khodakarami *et al*, 2008). Such tools can be applied to various data mining tasks and also for shaping solid unifying theoretical frameworks which could be leveraged in the general area of knowledge discovery. This approach specifically provides decision-support for a wide range of problems involving uncertainty and probabilistic reasoning.Various real world examples illustrating the applicability of this approach are documented in Fenton *et al* (2002). However, for illustrative purposes, an example is given in the next section.

University of Cape Town

Although this technique is still evolving, it has proven significant for some data-analysis problems. Hence making it a sound technology capable of addressing various challenges encountered in software engineering today.

3.5.3.4.1 Benefit derived from this approach

The BN technology offers various capabilities particularly if well integrated in a given knowledge domain. In this study, this technology was specifically integrated because of its ability to:

- reinforce usage of expert's knowledge in domain modeling , thus solving both discriminative tasks (classification) and regression problems (configuration problems and prediction);
- learn about causal relationships in which observed knowledge is used to determine the validity of the acyclic graph that represents the Bayesian network.
- handle both complete and incomplete datasets;
- continuously update beliefs when new data item arrives or even when existing ones are eliminated; and
- mine patterns from datasets in which examination for the conditional distributions, dependencies and correlations found by the modelling process is made possible.

The benefit of the Bayesian network representation lies in the way such a structure can be used as a compact representation for many naturally occurring and complex problem domains; thereby prevailing over rule based decision trees.

3.5.3.4.2 Rationale for BN integration into CASSE

In systems analysis and design, class diagrams are developed as static structure diagrams particularly during the system analysis stage. These structures lay the foundation for understanding relationships between various system objects, their attributes and how resources get shared between these objects. This is a role largely undertaken by analysts as domain experts in the conceptual modelling of information system needs, translating them into possible software solutions. These relationships often change as new system understanding unfolds. New objects can be added or existing ones can be eliminated. Alternatively, new functionality can be added as object methods/functions or operations. These methods could be exposed as sharable resources with other objects in the model.

Literally, the static structure created is a graphical model capable of being transformed into a BN structure. With this domain modelling already prevailing in software engineering, the BN technology can squarely fit it in with ease. The idea is objects in the static structure can be defined as BN nodes. Object relationships on the other hand can be defined as arcs for the BN network. Conditional probabilities in turn can be defined on the attributes of the different nodes within the model, thus

University of Cape Town

forming acyclic graph which can be learned as a BN structure to determine any joint probabilistic distribution which satisfies the *Markov condition*. As a result, various inference queries can be made on the model to answer any uncertainties about the model and its overall management.

3.4.3.4.3. Where Bayesian Networks excel over dependability analysis

Research presents various areas in which BN technology excels over dependability analysis (Bobbioa *et al*, 2001). Kim,(2005) notes that the most popular technique of dependability analysis is the Fault Tree (FT) analysis technique. In addition to this, he observes that the Bayesian Network (*BN*) technology provides a robust probabilistic method of reasoning under uncertainty where FT analysis techniques fail immeasurably. This technology has been successfully applied in a variety of real-world tasks. Therefore, the suitability of this approach is now considered by several researchers. Bobbio *et al*, (1999), emphasizes that *FT* can be easily mapped into a *BN* and that basic inference techniques on the latter may be used to obtain classical parameters computed using the former (i.e. reliability of the Top Event or of any sub-system, criticality of components. More still by applying the BN technology additional power can be obtained, both at the modeling and at the analysis level (Hsua et al, 2009). In particular, dependency among components and noisy gates can be easily accommodated in the *BN* framework, together with the possibility of performing general diagnostic analysis (Ahmed, 2006). The representation and use of probability theory makes BNs suitable for combining domain knowledge and data, expressing causal relationships, avoiding over-fitting a model to training data, and learning from incomplete datasets (Needham, 2007).

FT techniques have been analyzed to fall short of these analysis capabilities and it on the basis of the above capabilities that the BN technique was used in the framework.

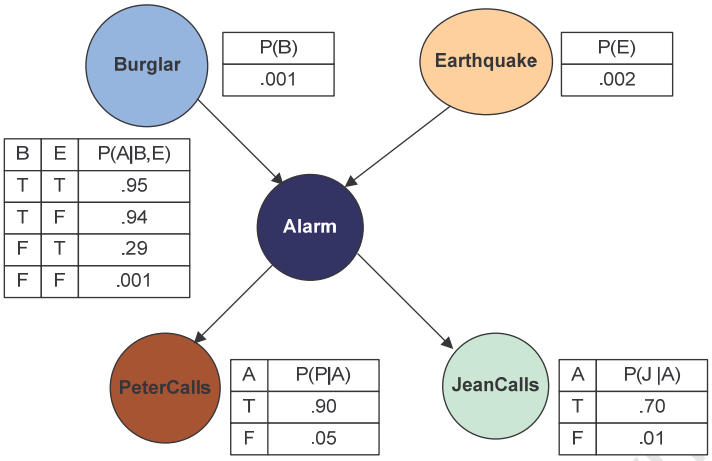
3.4.3.4.4. BN example

Assuming that we are aiming at answering the following question; "what is the probability that Peter will notify me but not Jean in the event where my house alarm goes off either due to a burglary or an earthquake?." Classical and Bayesian statisticians will answer this question from different perspectives. In classical statistics this cannot be answered before precise measurement of the physical properties of the alarm system is performed. The expected probability result however is fixed in the range of 0 and 1, regardless of any observations made.

In the Bayesian approach however, this question will be answered according to beliefs on these conditions as illustrated in the figure below.

University of Cape Town

Figure 7: NB Model example



The nodes in this model represent uncertain variables, which may or may not be observable. These variables include: *Burglary, Earthquake*, *Alarm*, *PeterCalls*, *JeanCalls*. This network topology reflects "causal" knowledge about either a burglar sets the alarm off or the earthquake and if the alarm can cause either Peter or Jean to call.

Each node has a set of states (e.g. 'due to burglary' and 'due to earthquake' for 'alarm' node). The arcs represent causal or influential relationships between variables. (e.g. 'burglary' and 'earthquake' may cause 'alarm going off'). There is a probability table for each node, providing the probabilities of each state of the variable. For variables without parents (called 'prior' nodes) the table just contains the marginal probabilities. This is also called 'prior distribution' that represents the prior belief (state of knowledge) about the variable. For each variable with parents, the probability table has conditional probabilities for each combination of the parent's states. This is also called 'likelihood function' that represents how likely is a state of a variable given a particular state of its parent.

The BN keeps updating beliefs on these nodes using the observed data; thereby ensuring that a joint probability (which satisfies a *Markov condition*) of getting a notification call ,when the alarm goes off, from either Peter or Jean being true could change.

From a graphical perspective, a BN is intuitively constructed by explicitly determining all the direct dependencies between the random variables of the problem domain. In this model, each node represents one of the observable features of the problem domain, and the arcs between the nodes represent the direct dependencies between the corresponding variables. In addition, each node has to be provided with a table of conditional probabilities, where the variable in question is conditioned by its immediate predecessors in the network. However, in instances where machine learning algorithms are applied, this graphical model can be constructed from empirical data automatically.

University of Cape Town

3.5.3.4.5 Applicability in software development

An important and novel aspect of this approach is its adaptiveness in using whichever indicators available. Handling risk and uncertainty is increasingly seen as a crucial component within the software development community, specifically in project management, planning and control. Integrating such technologies within the development process is still a major challenge. Available techniques of managing risk on software projects seem to be often event-oriented and try to model the impact of possible 'threats' on project performance. They tend to ignore the source of uncertainty and the causal relations between project parameters.

It is based upon this background that CASSE integrates this approach as process transformation competency in order to minimise risk on software projects while at the same time ensuring that economic value on these projects is maintained. In the next chapter, a detailed description of how this approach is integrated into the CASSE framework is given. The CASSE framework utilizes such proposed techniques to derive a value-based Actor Object Model. It also extents these approaches by applying emergence handling in the model so as to predict realistic efforts required on a given project.

While these models offer great insight into the value-based paradigm, however, the way these models are applied is dependent on the business strategy as well as the investment involved in establishing and integrating them into the production process. Economic models such as these cannot be created and assessed in a single step. Rather, they must be refined and improved over time by incorporating new techniques and approaches that make them more extensible. This aspect therefore raises the need for integrated predictor models and techniques that give us insight into project complexities right from project inception so as to mitigate project risks and henceforth increase business value.

3.6. Baseline strategy

3.6.1 Industrial trends

The software industry is growing rapidly and practitioners are struggling hard to cope with the ever-increasing complexity of software development. So, why is it that with all these technological advances, it appears that software development is becoming harder rather than easier? We have so many more tools and opportunities, yet the process of producing software seems to get ever more complex — even for relatively small applications. Having an optimized development process that is constantly revaluated can perhaps help.

University of Cape Town

3.6.2 The business need

Although software development life-cycle models and business decision models contribute to the control of product development in different ways (Wallin *et al.*, 2002), economic models such as these need to e established only through an iterative process involving analytical and empirical methods (Do & Rothermel, 2008). The use of sensitivity analysis in conjunction with complex adaptive theory has not been largely integrated in these models. Sensitivity analysis can help simplify those models for value creation; it can also identify key project factors that must be measured with care, leading to guidelines for better decision making.

Therefore, we need economic models, tools and methods to better understand how to design software system development projects for maximum value creation (Balikuddembe & Potgieter, 2006). Such an economic perspective has the potential to improve many aspects of software development, which is an increasingly important goal as the information revolution continues to accelerate. By applying this approach, an organization may estimate the variability of its expected profits related to a set of software projects. Such an approach is built upon an analogy that compares software projects to unhedged loans issued to unreliable borrowers. As loans may not be paid back, software projects may fail, leading their development organizations to sustain losses. There are many open questions regarding the integration of automated requirement management and the use of static analysis to assist project planning and value-based software engineering (Gotel & Finkelstein, 1994).

3.6.3 The research gap

While integrating automated requirement management and static analysis is a powerful heuristic, we lack adequate models of connection between this technical concept and value creation under given circumstances. Should one design for change if doing so take up additional time in an extremely competitive marketplace, where the speed with which a product reaches the market is a defining factor in its future success? How does the payoff from changing the system relate to the cost of enabling the change? What role does the timing of the change play? What if it is not likely to occur until far into the future? What if the change cannot be anticipated with certainty, but only with some degree of likelihood? What if the change is somewhat unlikely to be needed but in the case that it is needed, the payoff would be great? Value-optimal technical design for analytical software project portfolio management choices depend on many such factors. Our ability to answer these questions will largely determine whether an alignment of the stakeholder views on cost and benefits of tracing can be achieved, which in turn will determine the rate of adoption of tracing and software economics consideration in practice. Given that we live in and benefit from this value-determined situation, it is therefore in our own interests to increase our understanding of and our

University of Cape Town

ability to deal with the economic aspects of software, its development and use, so as to stay afloat in business.

CHAPTER FOUR

The CASSE Framework Engineering Approach

The size, complexity and criticality of software systems require innovative and economic approaches to development and for continuous business growth evaluation (Lehman, 1995); (Garlan & Shaw, 1993). The increasing pace of change in the Information Technology (IT) field makes feedback control essential for organizations to sense, evaluate, and adapt to changing value propositions in their competitive marketplace (Boehm & Huang, 2003). As noted by Atkinson *et al*., (2006) such feedback must highlight and communicate risks that may hinder project value propositions throughout the software development life-cycle.

This chapter introduces the CASSE framework developed in this research. CASSE is an integrated framework addressing compatible self-adaptiveness in managing the project selection and economic value complexity. It specifically uses the value-based requirements engineering techniques. It aims at increasing project feedback channels to decision makers engaged in the software development process. This approach complements the traditional cost, schedule, and product planning and control techniques of the value delivered to stakeholders as suggested by Holler, [2006]. It is driven by the need for profit maximization on software projects as a guiding principle underlying production by a firm. In particular, it is assumed that a firm undertakes an action and makes the decisions that increase profit. In this regard, profit is the difference between the total revenue received from selling output and the total cost of producing that output. The profit-maximization assumption is related to a production level that generates the greatest difference between total revenue and total cost. If a firm maximizes profit, then it is generating the highest possible reward for entrepreneurship resources.

The CASSE framework can be applied to the following situations.

- Business viability assessment of potential software projects;
- Bottom-lining functionality packages on evolving or even customable software products;
- Supporting the clarification and requirement negotiation process on software projects;
- Project measurement and evaluation; and
- Task performance measurement on software project schedules.

University of Cape Town

4.1 Engineering Approach

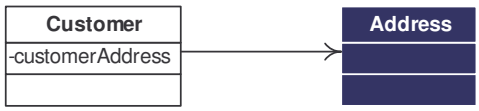
4.1.1 Background to deriving Actor Object dependencies

CASSE uses Actor Object Dependency (AOD) analysis techniques in predicting the probability distribution of profits and losses on a software project. This is also used in the tracking of the anticipated profitability, initially envisaged at project selection, throughout the development lifecycle. In defining the AODs values, this engineering approach is based on the four major areas of Actor Object (AO) relationships, that is, Association, Generalization, Aggregation and Composition relationships suggested in the Object Oriented System Analysis and Design Methodology using UML (OOSAD).

An AO represents a set of user actions as properties which get executed at the system interaction level in order to fulfil a system user's goal. The system user in this regard is the Actor, while the object representing him/her at system level is the Actor Object. These Actors have various roles in the system and relate differently when executing different tasks in the system. The static structure or conceptual model derived from system analysis is composed of these AO which are widely defined as system classes. In this approach they are referred to as Actor Objects in the system, having different attributes and varying causal relationships with other AO in the conceptual model. The static structural on the other hand is referred to as the Actor Object Graphical model. The four relationships governing dependency in the graphical model are described as follows.

Association: denotes a permanent relationship between two classifiers. For example, the Address object can use Customer object resources to fulfil its functionality as illustrated in Figure 8 below. In the absence of the Customer object, Address can exist alone and can be fully functional; hence resulting in ***a strong*** permanent association relationship but not very critical

Figure 8: Association relationship

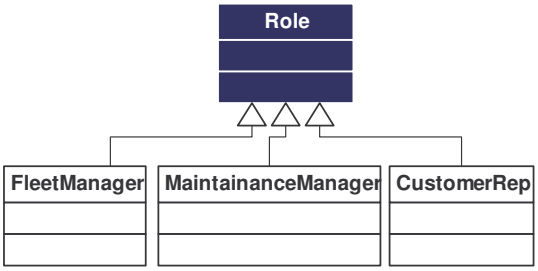



```
class Customer {
private Address customerAddress;
}
```

Generalization: captures a hierarchal relationship between classes of objects. For example, FleetManager object will inherit all properties and characteristics of the Role Object but the FleetManager object can even have more characteristics that are unique to itself; hence resulting in ***"Is A" strongest*** relationship as illustrated in Figure 9 below.

University of Cape Town

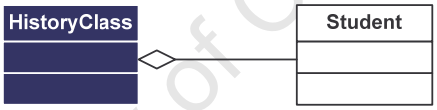
Figure 9: Generalization relationship




```
public class FleetManager extends Role {
        ...
}
```

Aggregation: is a special form of association that specifies a whole-part relationship between the aggregate (whole) and a component part. For example, as illustrated in Figure 10 below, the state of each Student object has an influence on the state of the History-Class object. If the History-Class object is destroyed, the Student objects may continue to exist, thus resulting in an **averagely *strong*** relationship.

Figure 10: Aggregation relationship

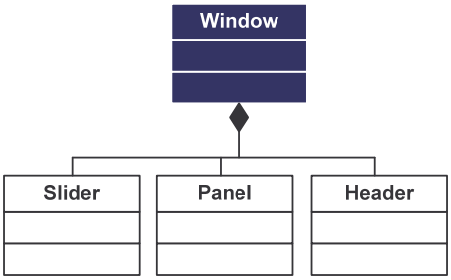



```
class HistoryClass {
private Student [] students;
}
```

In a **composition** relationship the part has the same lifetime as the whole. When the whole ceases to exist then the existence of the part makes little sense, thus resulting in an **extremely strong** relationship as illustrated in Figure 11 below.

University of Cape Town

Figure 11: Composition relationship

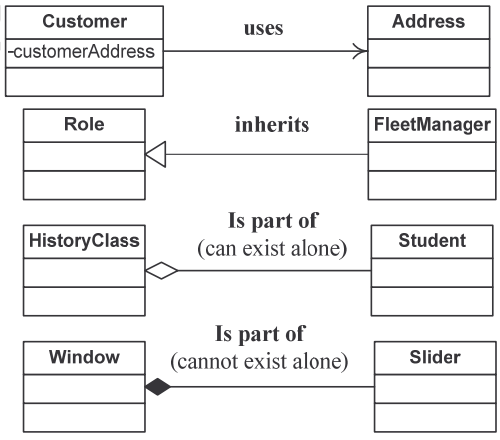



```
class Window
{
private final Header theHeader = new Header();
}
```

A detailed description of these relationships is given in Pilone & Pitman, (2005). These relationships types are widely used in system modeling and thus are reasonably understood by various practitioners in industry.

In order to analyze the value trade-offs of a given model, monitoring and observation of key AOs and their dependencies in relation to other AOs in the model is done. The argument advanced here is that if one is to adequately estimate the probability distribution of losses and profits on any project and at the same time sustain this profitability during the project lifetime, one therefore must have to understand and manage the complex dependencies that exist between the various AOs in the model. Figure 12 summaries these relationships as follows:

Figure 12: Actor Object relationships



- AO Customer ***uses*** AO Address to fulfil its functionality. In the absence of the Customer, Address can exist alone but can be fully functional; hence resulting in ***a strong*** permanent *association* relationship;

University of Cape Town

- AO FleetManager can ***inherit*** all properties and characteristics of AO Role but FleetManager can have even more characteristics that are unique to itself; hence resulting in ***"Is A" strongest*** *generalization* relationship;
- AO Student can be ***part of*** AO HistoryClass but Student can exist alone in the absence of HistoryClass; hence resulting in an **averagely *strong*** *aggregation* relationship; and
- AO Slider ***is part*** *of* AO Window. This means that Slider is part of Window and cannot exist without Window, thus resulting in an **extremely strong** *composition* relationship.

The above relationship definitions thus result in four types of weighting relationships that are used in this approach, i.e., ***strong, averagely strong, strongest and extremely strong***. These relationships are the basis for computing weight averages of each AO in the logical model as described later on in this chapter.

However, there exists hard to catch dependencies between requirements as well, for example AND, OR, ICOST, IVALUE, etc. Such relationships are ordinary visible during the requirement sweep-through process especially at the requirement clarity and negotiation stage. The implicit weighting relationships defined above, assist with illustrating such complexity as described later on in this chapter. These relationships therefore offer great capabilities in supporting and guiding the estimative function on any software project.

University of Cape Town

4.1.2 Realizing the estimative mechanism

Estimating software is a complex process requiring a lot of attention and good blend of technical skills. Requirements continuously change especially in instances when systems' understanding prevails. The overall aim here is to determine the appropriate implementation time estimates from the onset and the appropriate harmonization of the project scope with key stakeholder interests. Failure to align the two perspectives often escalates risk levels on projects as observed by Berger, (2007). This determination can come about only through negotiation and clarification of needs, between project planners/builders of systems, and stakeholders. After obtaining consensus around the required functionality only then can one set reflective software quotations, budgets, timelines and schedules.

At the requirement elicitation stage, clients often emphasize required features and their associated priority in achieving the desired system behaviour. From the system analysis perspective, these features are often modelled as actions fulfilled by Actors who have specialized roles in the system. These actions are translated into use-cases for high-level system understanding. At the design stage, these use-cases are translated into a single unifying Actor Object logical model (static structure) with

specialized causal dependencies. This logical model is composed of AOs having system operations and functions. It is upon these roles that modelling and gauging of the complexity of the required implementation against expected client priorities in relation to scope and schedule expectations is made.

These envisaged actions are considered vital inputs to the logical model upon which negotiation grounds on system requirements can be made in a trade-off analysis pattern, specifically by taking into consideration the complexity implications in the model. As Garcia & Hirata (2008) maintain, this process is understood to be a mechanism for mapping the identifiable user functions onto software functions that allow the sizing estimative to be free of technology and technical factors.

4.1.3 Modelling facets

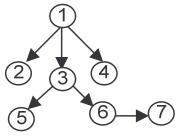
4.1.3.1 Explicit and implicit causal dependencies

In order to achieve a viable negotiation ground, concentration is made on defining and managing the dependencies between AOs in the system logical model as described above so as to analyze the trade-offs in the model. The conviction in here is that if one is to negotiate valuably and at the same time evaluate and manage technical system complexity on any project, one ought to understand and manage the complex dependencies that exist between the various AOs in the model.

With this kind of relationship mapping, complexity of system requirements can be well managed and translated for clarity purposes, thereby contributing valuably to the requirement elicitation workshops. AOs are crucial in this modeling since they provide functions for the definition of *actors*, as well as *actor* operations in the system defined by the client as described above. The AO causality relationships described above are instrumental in achieving this object.

During a requirement sweep-through process, features could be ranked differently depending on their business process dependency and priority. For example, as illustrated in Figure 13 below, node 1, as a *priori* node (where node represents a feature) is composed of nodes 2, 3 and 4 respectively as node subsets. As we go down the hierarchy, we hence realize that node 3 has child nodes 5 and 6, and a grandchild node marked 7, thus creating a likelihood function on these nodes.

Figure 13: Explicit node dependency



Often, this tree of interrelationships visualises the pictorial view of overall system features as “nice-to-haves” from a client’s perspective. It is on the basis of this view that implementation priorities are established, especially in instances where there are budget constraints or where incremental business process automation is envisaged. However, this dependency pattern is dependent on

University of Cape Town

current understanding of system requirements from a linkage perspective rather than from a system logical perspective. The resulting model is referred to as an explicit model with known dependencies.

From this, a client may choose to implement only a particular branch of the tree. At this stage, interest is vested in assessing the elimination or addition impact of AOs in the model. For instance, a client may opt to eliminate some functionality which may impact upon the tree relationship considerably. Even during the project life-cycle, a client may make suggestions in terms of new functionalities that need to be integrated into the tree as subsets. In instances where eliminations are imminent, measuring and mitigating the effect of AO silos unconnected in the model but affecting overall system functionality is vital.

However, to ensure that a realistic scope is being set for the project and that requirements are well understood, a complexity check on the explicit model resulting from feature relationships has to be undertaken. With this, one is more positioned to analyse the dependency magnitude of the various nodes in the model which will ultimately impact upon the project time, schedule and budget. With this kind of heuristic knowledge, demonstration can be made to the client about the nature of the system complexity ahead of implementation. Secondly, such manifested complexity can facilitate the quotation process valuable. The process is referred to as “mining the unknown” or “hidden dependencies in the explicit model”. The framework engine handles this by learning the system graphical model and as a result revealing any hidden dependency patterns that may arise. This mechanism is based on node weighting rules and system emergence engineering techniques described in the next section.

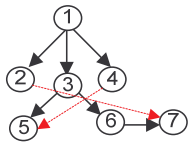
The graphical model is derived using the use-case realisation process in which functional points are counted to arrive at a suitable estimative level of the intended project. With this approach, visualisation can be made on how a given user action invokes a multitude of system functions interacting between different Actor objects so as to achieve the desired system behaviour. The criticality of this interaction is dependent on the number of resources required by another object from its dependants. This model is understood and handled as a complex adaptive system in which patterns of interrelationships emerge from local interrelationships between Actors and their artefacts. As additions, eliminations or even change dependency types between AOs in the explicit model are made, new patterns emerge on an ongoing basis. These emerging patterns could have a huge bearing on the overall project technical complexity and ultimately on the project scope. It is at this point that the BN technology becames useful in handing this emergence.

University of Cape Town

The value of each dependency link in the model is calculated from the four relationships described above. After computing the AO weights, the complexity management algorithm is then applied to the model so as to reprioritize nodes valuably for all stakeholders.

During the graphical model learning process, dynamic object monitors observe the dependence interaction behaviour of operations so as to mine the interrelationships between particular AOs in relation to other AOs in the model. When a query is passed to return an implicit relationship that exists in the model, a refined dependency model of relationships as shown in Figure 14 below could be generated.

Figure 14: Resulting implicit dependency



The dashed red lines show the implicit relationship that exists between AOs impacting on features in the explicit model. These two checks categorize and rank the AOs into three groupings, that is, from high to low. High would imply very critical AOs, which are essential for achieving overall system functionality. Medium would imply AOs that are relatively influential in achieving overall system functionality. Low would imply AOs that have the least impact on overall system functionality.

With this approach it is possible to identify key system integration points that will require more detailed attention and that will form the basis for overall customer acceptance criteria with regard to the envisaged system. In both instances, i.e. where project time is known and unknown, this option facilitates optimal scheduling and the prioritization of system features in view of time, cost and resource constraints.

For instance, if a client suggested that feature 3 be eliminated, it will be interesting to know what would happen to dependence relationships between features 5 and 6 respectively. A refined model would then be returned with probable relationship and their types. For example, a returned relationship could link feature 5 to feature 4 since they have an implicit relationship. Feature 6 would also be traced through feature 7 and 2 respectively. The beliefs generated for each node within the model are then utilized for analyzing implementation prioritization, impact assessment and change implication analysis.

To the ordinary user, all this modelling is abstracted. The technical aspect involved in mining the implicit relationship is handled within the framework. The UML graphical model is mapped directly to the features tree created during the requirement sweep-through stage. Any implicit relationships identified at the graphical level, are visualized with dotted links within the features tree, thereby

University of Cape Town

guiding trade-off analysis. The mechanism of translating this feedback is described later on in this chapter.

4.1.3.2. Operation structure dependency assessment

An operation structure presents the nature/structure of AO function which will be invoked during system runtime. At the system interaction level complexity is also modelled and managed. A functional point is translated into an operation which is executed at program interaction. To analyze the dependency between various Actor Objects Operation Structures (AOOS) in the model, matrix values are generated basing on causality, structure, and resource or task dependency relationship. Analysis is made of which kind of operation structure will be required by other objects in relation to the target AO. For example, let us consider AO 1 to be the target object with four operations, get (), update(), add() and remove(). All the four operation structures will be given a true value for the causality relationship, since this is a target object.

This implies that the target object needs all these operation structures to operate; thus implementation has to be done for all these operations for this object. Compare AO 2 operation structures with those of the target object, observation is made that AO 2 has no similar operation structures or it may have some similar ones. If there are any similar structures, the value would be true. If there is no similar structure, the value is false. For each object comparison pattern therefore, you expect either a (t, t), (t,f) or (f, f) combination. Table 1 below shows an example of an operation structure matrix used to derive dependency patterns as described above. The guiding principle is that properties used by another AO would determine its implementation priority.

	Actor Object 1	Actor 2	Actor 3	Actor n
step 1	Operation 1 (value = t)	t / f	t / f	t / f
	2 (value = t)	t / f	t / f	t / f
	..n (value = t)	t / f	t / f	t / f
step 2	1 (value = t)	t / f	t / f	t / f
	2 (value =t)	t / f	t / f	t / f
step n	…….	…….	……..	……

Table 1: Operation structure matrix

This analysis is handled by the XML generator in level A of the framework architecture. Extracted combinatorial values are then passed on to the learning engine in level B for analysis and interpretation. A detailed description of the CASSE architecture is given later on in this chapter.

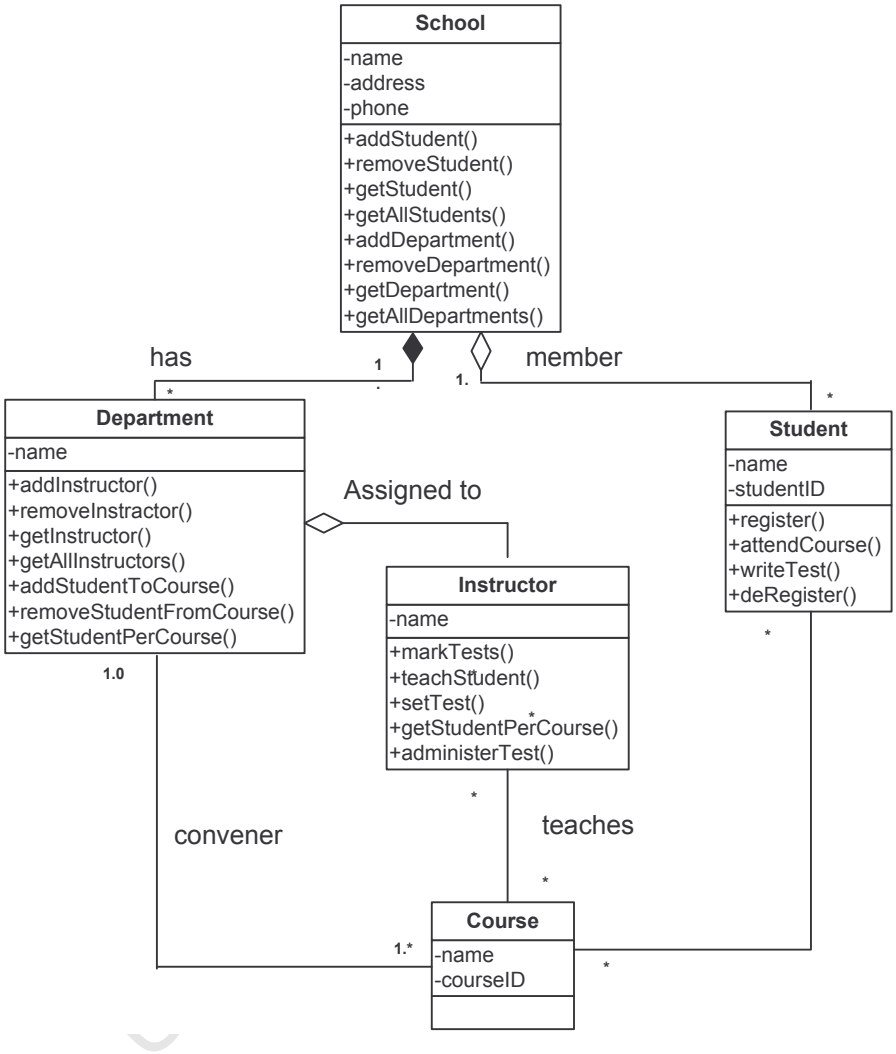
4.1.3.2.1 Implementation scenario

To illustrate this approach, let us look at a partial student administration system as shown in Figure 15 below. A student registers for a course with a school. Each course has an instructor. Each instructor belongs to a department within a school. An instructor from the department could be a

University of Cape Town

course convener. Given this background an AO logical/conceptual model is developed with attributes and operations.

Figure 15: Actor Object model



The model above shows the known AO dependencies (explicit relationships) in the system and how they interrelate. From this model values are generated for learning the hidden interrelationships.

Generating data for this scenario would require setting a target object first in relation to others in the model as shown in table 2 below. If we take school as the target object in this case, with a causality relationship dependency analysis of the operation structures, AO School would have a strong causality relationship with AO Department, Instructor and Course – thus the true value in the matrix. While there would not be a significant causality relationship between AO School and Student – thus a false value. This implies that operational structures and processes in AO School will have a direct consequence for AO Department, Instructor and Course operations but not necessarily for AO

Student. This comparison continues for all relationship types as generation of the values for the matrix are made so as to determine implementation implications in the system.

	Actor Object				
	School	**Department**	**Student**	**Instructor**	**Course**
School as a target Object	T	t	f	t	f
	T	t	f	f	f
	T	t	f	f	f
Operations	T	t	f	f	f
	T	f	f	f	f
Department as a target Object	T	t	f	t	f
Operations	T	t	f	f	f
	…	…	.	…	…

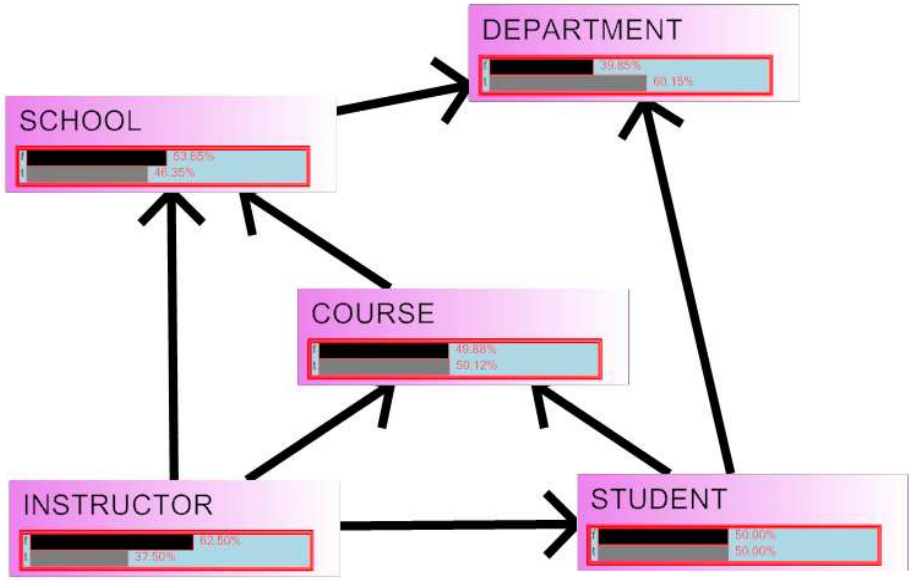
Table 2: Generated matrix example

For a client willing to implement such a system, he/she may opt for a module-per-module implementation basis. For instance, if a client wanted to start with a student registration module, we may want to analyze what the implications of this would be for the rest of the requirements in the model. However, the current model is dependent on our heuristic understanding of the current system requirements at this stage. With this kind of pattern, as a vendor, you can demonstrate to the client the nature of the system complexity ahead of implementation as a basis for contract and requirements negotiation. Since certain requirements cannot be implemented exclusively with others, we ought to constantly negotiate for those requirements that have a strong relationship but are to be implemented at a later stage. Although this seems like a strenuous exercise within a manual environment, it can easily be automated as matrix to ease usage and also cut down on the required time to perform this analysis; thereby saving on production costs.

University of Cape Town

4.1.3.2.2 Resulting model after the leaning process

Figure 16: Resulting model



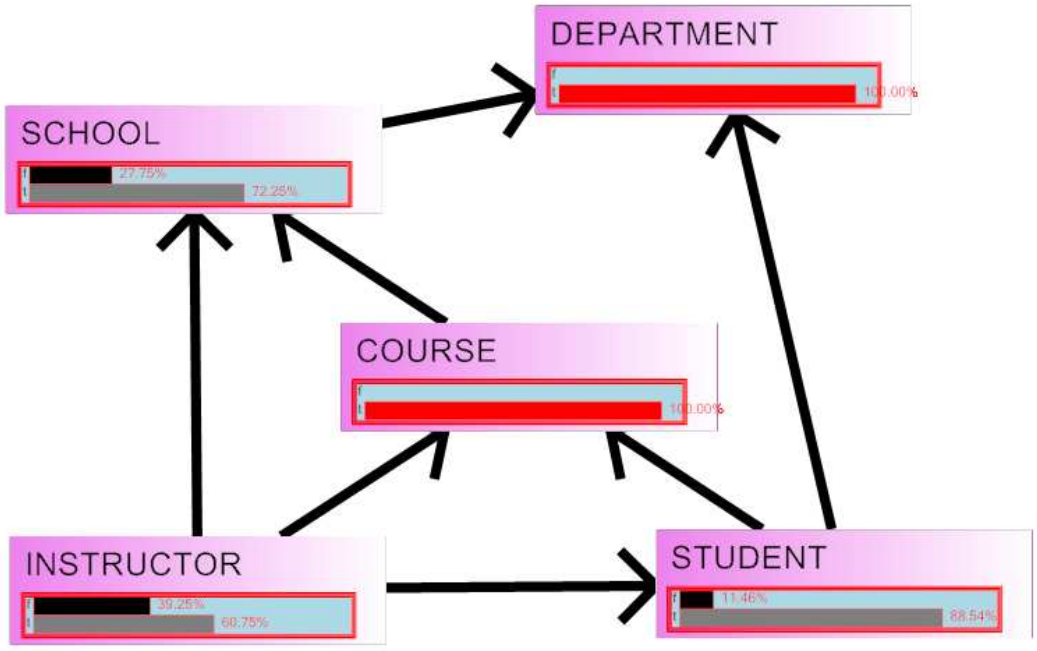
The model above (Figure 16) shows the hidden relationships between the AOs and their resulting beliefs. The arrows indicate that there is a strong relationship between each of the AOs. After generating data for the dependency matrix, this data is imported into the framework engine for hidden dependency analysis. The main objective of this step is to provide the optimal dependency pattern analysis which may not have been as explicit as possible in the initial stages of the analysis. With the engine, one can make inference queries to analyze requirement complexities and the change in dependency patterns between the AOs as shown in figure 17 below.

The framework engine at level B handles this dependency analysis process and gives us an opportunity to compare different scenarios in the model. The engine returns degrees of beliefs in percentage values (at a scale of 1 to 100) when evidence is set for any given AOs.

University of

4.1.3.2.3 Model analysis

Figure 17: Model beliefs when evidence is set



For example, in the model above, interest was vested in querying the dependency analysis within the objects, given that AO Course requires resources from AO Department. In this scenario, evidence was set to 100% for AO Course and Department as illustrated in Figure 16 above. When evidence is set on any AO in the model, for distinguishing purposes, it is coloured red. Other AOs in the model have other colours for pattern analysis purposes. The analysis engine of the framework revealed that there is an 88.5% chance that AO Student will require the same resources; a 75.25% chance for AO School and a 60.75% for AO Instructor. This implies that if implementation is to be made for object department and course respectively, the implementation time required on student object will be reduced by 88.5%, for the school by 75.25% and for the instructor by 60.75%. Prioritizing implementation of any of the objects in the model will result in the likelihood effect to the rest of the objects in relation to structure, resource, and task or causality relationships.

Therefore, if you go one step further into the operation structures of these objects to ascertain the required resources, we would implement them all into a single layer that will be invoked by all objects that need these resources. This would increase agility in development, and at the same time save resources, hence expanding our profitability curve.

University of

This modelling approach, although demonstrated with a sample partial small system, it is equally applicable in both large and small scale systems as illustrated in one of the case studies. The value of this approach is that clients will be able to understand the complexity of the problem right from the start of the development process. This enables them to make informed decisions, especially in instances where products have to be developed on a modular basis. From the software engineering context, we can already determine which functionality will have significant dependency in the model, as well as its criticality in the entire system, thus enabling us to negotiate better on IT projects. This approach would also assist us in planning our schedules better as well as improve our quotation abilities on projects. The consequences of this are that improved project budget negotiation, scheduling and implementation strategies, change control, project management and ultimately customer satisfaction and commitment are enhanced.

However, to achieve this result there are underlying facets that drive the framework that have to be understood by any adopter of this framework. These facets are described in the next section.

4.2 Key facets/aspects in the approach

To optimally score and analyze a given AO model for any project, consideration is made for the underlying aspects. These elements are pivotal in deriving the computations used to embrace the BN technology in this analysis. These include the following.

- The profitability object function to be achieved by the project,
- a mathematical problem of the objective function to be formulated and solved,
- constraints of the mathematical problem,
- properties of each individual AO,
- weight of each AO so as determine the implicit and explicit risks,
- earliest start and finish time required to complete a project, and
- a complexity check of the model.

Complexity check in the model uses emergence engineering approaches so as to map the results adequately on to the project value evaluator which is used for guiding the decision making process.

4.2.1 Project profitability function

The underlying assumption of this engineering approach is that profitability of any software project is dependent on how best you understand and scope the project features, how best you rank and schedule them, monitor their progress during implementation, how best you manage their evolving

University of Cape Town

nature and how best you manage your personnel on the project. Although this is largely a project management function, it can also be captured and tracked within the framework to ensure valuable outcomes. Thus, project value and success is shaped and is also perceived as a function of the optimized development process in which efficiency in analysing AO dependencies is required and also in which the project is scored against the business value threshold. This is summarised in equation 1 below.

$$f(x_1,x_2 \text{ ---}x_n) = a_x, + a_2x_2 + \text{------}a_nx_n + b) \quad (1)$$

Where *x* denotes a feature or functional point on a project which could be new or existing , *b* is a unit matrix of limited resources, unknown for example actual programming time and *a* is a constraint, that is, the uncertainty about a value-based coefficient that goes with a given functional point.

Optimizing profits on any project requires that a vendor understands well how much of the existing functionality will be priced against the new functionality. This is often difficult to determine especially during the quotation stage of the software project. Having a breakthrough in this aspect requires that viable tuning is made on the unit sale vale (*S*) of each type of functional points (FP) to be priced, which is also a very challenging commercial problem. This results in a mathematical problem of the objective function as given in equation 2 below. Where unit sale value for a new functional point(X_1) is donated as S_1 and new functional point(X_2) is donated as S_2. Existing functional points could be already existing libraries or components that a firm has on their project database or they could be open source libraries that can be interfaced.

$$\text{Max: } s_1 x_1 + s_2 x_2 \quad (2)$$

However, this is subjected to various constraints including the following.

1) Determining how many resources (z) will be involved in implementing the new functionality and how many are needed to fine tune the existing functionality to suit the new system need. This results in equation (i) below.

   i. $z_1 x_1 + z_2 x_2 \leq b1$

2) Determining total optimal time (t) needed to achieve the functionality of a given functional point, resulting in equation (ii) below.

   ii. $t_1 x_1 + t_2 x_2 \leq b2$

3) Determining total labour cost (c) to complete the functional point – resulting in equation (iii)

University of Cape Town

below.

$$\text{iii.} \quad c_1 x_1 + c_2 x_2 \leq b3$$

However, existing functional points can never be negative and new functional points can never be negative, thus resulting in equation (iv).

$$\text{iv.} \quad x_1 \geq 0, \; x_2 \geq 0$$

4.2.2 AO properties as inference attributes

Each OA has the following properties. ***Sale Value*** – as the amount quoted by the vendor to the client; an ***Implicit Cost*** – as actual cost incurred by the vendor to implement an AO; ***Nature*** of AO– whether it is an existing or new AO, ***Number of resources*** to implement that AO; ***number of*** FPs in that AO and ***time*** required to implement that AO. Each vendor decides his implicit cost and sale value respectively based on the costing model adopted by the vendor and on the prevailing market forces. This is often in the form of an hourly rate.

This approach assumes that a vendor has a qualified team to implement a given project or has a contracting pool he can draw from – thus balancing the human resource problems on a given project. The time attribute has a derived value. Total time (T_t) is a product of total number of FPs and the Productivity Factor (*PF*). The relative productivity factor per FP used in this framework is 7.2. This value has been recommended by Anda *et al.*, (2005) as a relative estimation value for a FP.

Functional counting methods are specifically used in this approach because of the advantages they offer including: their language representation which is meaningful to nonprogrammers and end users, measurement of a system primarily from logical perspective. Also, measurement of system development and maintenance independently of implementation technology and the capability of normalizing data across various projects are possible. This approach is also an industrial standard which is still evolving within both industry and academia. This approach is therefore considered appropriate as a profound basis for unit value benchmarking and also as a testing baseline for value mapping within this framework.

Although FPs have a major limitation of being labour-intensive, requiring significant training and experience to be proficient and also having functional complexity weights and degrees of influence determined by trial and debate, with adequate training this can be improved. Equally, the 7.2 productivity factor proposed in this framework can easily be adjusted to suite a given development environment. This adjustment is however dependent on the fact that appropriate measures are in place to capture estimated time and also actual time spent on a given task or FP. This would result in a more appropriate value which can be used for future estimations. The framework has been

University of Cape Town

development to embrace this as well. More still, in instances where functional counting is rendered ineffective, this method can easily be replaced with a more effectively proven method, provided it is industrially tested and can be used to benchmark unit value on any given project.

Although this engineering approach is perceived from a solution space in which solutions are devised and that most important requirements are caught at time of framework usage, the changing characteristic of requirements and the overall impingement this change has on the project scope is taken care of. New scope items can be added to the model as the project progresses. The framework would be in position to detect this change and ultimately map any surprises arising out of this change, either as an addition or elimination, to the project value evaluator so that informed decisions can be made.

4.2.3 Analyzing AO causal dependencies in the model

All FPs are contained in AOs which have specialized dependencies among them. These AOs interact in a specialized fashion using functional points to achieve a given system goal. Thus computing a unit AO`s weight, given the number of dependencies it has with other AOs in the model is imperative to determining its overall priority in the model. The main aim here is to establish how each object relates to each other in the model and what risk factor this weight will give it in relation to its neighbors. The computation used at this stage extends principles of information theory, specifically, weighted network principles proposed in various studies for example Xu *et al*., 2005; Park & Hastie, (2007).

The value of each dependency link in the model is calculated based on the four relationships described above, that is, ***strong, averagely strong, strongest and extremely strong***. The value obtained will be between 0 – 1 where 1 is extremely risky and 0 is Low. This results in 3 state categorisations for each AO in the model, that is, Low, Medium and High. It is upon these states that belief propagation is learnt in the network and upon which sensitivity analysis is made about the model. This computation results in the determination of the each AO risk level, in this regard it is the implicit risk. AO weight is computed as given in equation 3 below.

$$\varpi = \frac{\sum_{i=1}^{n} w_{ij}}{n} \qquad (3)$$

The implicit risk of an AO denoted by ϖ is computed as total average of all link dependencies denoted by w_{ij} into the AO in the model with *i* starting from 1 to *n* where *n* is the total number of all link dependencies going through an AO in the model.

University of Cape Town

4.2.4 Calculating the critical path

To calculate the critical path (CP) of the model, a forward pass and backward pass are used to determine the minimal project duration. The algorithm used has a starting object and works on computing 'flag value' for the objects in the model. The flag values represent the longest time for each AO. The AO in the model can be processed in any sequence so long as all preceding objects have flags at each step.

Early start time of the whole model is computed as:

$$ESK = MAX_{j:j\to K}\{EF(j)\} = MAX_{j:j\to k}\{ES(j)+d(j)\} \quad (4)$$
$$EF(k) = ES(k)+d(k)$$

Late start and finish is computed as:

$$LF(j) = MIN_{k:j\to K}\{LS(k)\} = MIN_{k:j\to k}\{LF(j)-d(k)\} \quad (5)$$
$$EF(j) = LF(j)-d(j)$$

An activity on a project, in this case an AO and its other properties is denoted as *k* and *j*. Duration of an activity is denoted as *d*. ***ES*** is the early start time, ***EF*** is the early finish time, ***LF*** is late finish time and ***LS*** is late start time.

4.2.5 Explicit risk of the model

The explicit risk is determined as time required for completing an AO implementation in the model. It is calculated as an average difference of implementation time and slack time as given in equation 6 below.

$$E_r = \frac{t - s_t}{t} \quad (6)$$

Where *t* is the time required to complete an AO implementation in the model and s_t is the slack time.

4.2.6 Prioritization

Prioritization of the AOs in the model is made for optimal scheduling purposes and identification of key AOs in the model. The CP method, the explicit risk algorithm and the complexity handling module, described in the next section, handles this emergence in this framework. These two modules categorize and rank the AOs into three groupings, that is, high to low. Where *High* would imply very critical AOs required in achieving the overall system functionality. *Medium* would imply

University of Cape Town

AOs that are relatively influential in achieving the overall system functionality. *Low* would imply AOs of least significance in impacting on overall system functionality.

With this approach it is possible to already identify key system integration points that will require much attention and often these will form the basis for overall customer acceptance criteria for the envisaged system. In both instances where project time is known and unknown, this option facilitates optimal scheduling and prioritization of system features against time, cost and resource constraints on the project.

4.2.7 Handling complexity in the model

Complexity handling in the model is attained with the use of the *Bayesian Belief Network* (BN). The Bayesian scoring metrics is applied in defining a scoring function for measuring model quality in terms of marginal likelihood or penalized likelihood. In this work specifically, the L1 Regularized Path algorithm proposed by Schmidt *et al.*, (2007) was adapted to handle the complexity check and management in the model as given in equation 7 below. This algorithm was integrated in the framework because of its ability to handle large data sets as proven in various studies (Kim *et al.*, 2007, Bangalore et al., 2007, Quattoni *et al*., 2009). This algorithm therefore renders the CASSE framework scalable. The goal of using this algorithm is to find the DAG G that equivalently minimizes the MDL (Minimum Description Length) cost. Assumption is made that the data is complete and is fully observed and can either be continuous or discrete.

$$MDL(G) = \sum_{j=1}^{d} NLL(j, \pi_j, \theta_j) + \frac{|\theta_j|}{2} \log n$$
$$NLL(j, \pi_j, \theta) = -\sum_{i=1}^{n} \log p(X_{ij} \mid X_{i\pi_j}, \theta) \quad (7)$$

Where *n* is the number of data cases, π_j are the parents of node j in G, and $NLL(j, \pi_j, \theta)$ is the negative log likelihood of node j with parents π_j and parameter $\theta . \theta_j$. $\text{argmin}_\Theta\ NLL(j, \pi_j, \theta)$ is the maximum likelihood estimate of j`s parameters.

This approach has been used in other studies, for example in Park & Hastie, (2007) and Wainwright *et al*.,(2006).

4.2.8 Profitability tracking on projects

The temporal measurement layer is added to the model to ensure adequate tracking of marginal revenue on the approved project. This sits in level B on the framework architecture as described in the next section. Modelling for temporal measurement is based on the Dynamic Bayesian Networks theory proposed in various works such as An *et al.*, (2006) and Fenga *et al*., (2009). This model is

University of Cape Town

specifically utilized to evaluate project performance over time by examining acceptance signoff of the various tasks on the project and how this acceptance pattern affects the overall profitability curve of the development process; thereby creating a structure and specification for the acceptance criteria on a project.

As observed by Osunmakinde & Potgieter (2008), variables of a time step usually referred to as frames, can have various impacts on the variables of the subsequent frames through temporality links across the frames. According to Russel, (2003), construction of the temporal model requires prior matrix, Pr (V_0); transition matrix, Pr ($V_t \mid V_{t-1}$); and sensor matrix, Pr ($E_t \mid V_t$) of state variables. The three matrices can be estimated during the intra- or inter-frame learning of the model using parameter learning algorithms such as maximum likelihood estimate (MLE). The intra-frame learning estimates the conditional probability distributions (CPDs) for every time step *t,* while the inter-frame learns the CPDs over time. These required matrices are described in equations 8 and 9 below.

The joint probability distribution for any frame of random variables V_1 to V_n at time *t* is given as:

$$\Pr(V_1,...,V_n) = \prod_{i=1}^{n} \Pr(V_i \mid \pi(V_i)) \quad (8)$$

The combined joint probability distribution for any temporal model to a finite time *t* is also described as:

$$\Pr(V_0, V_1,..., V_t, E_1,..., E_t) = \Pr(V_0) \prod_{i=1}^{t} \Pr(V_i \mid V_{i-1}) \Pr(E_i \mid V_i) \quad (9)$$

These two foundational steps result in Emergent Situation Awareness (ESA) techniques adapted in the CASSE architecture.

4.3 Project measurement and evaluation

As inference and fine-tuning on the value matrices on the project is made, new patterns are continuously emerging. These emergencies need to be managed and monitored at a scope level. This is achieved through identifying relationships between scope artefacts. In this regard, concentration is on the Actor Object Graphical model composed of Actor Objects with specialized relationships between them and also having functional points upon which the estimative effort is inclined.

The overall aim is to manage complexity at the scope level so that viable project schedule, budget and scope estimates are achieved. This is handled by the dynamic project monitoring and updating observers within the framework. Its main role is to observe any changes in the model and continually map the project status to the project evaluator, as illustrated in Figure 17 below.

University of Cape Town

This observation and updating capability is triggered by the estimative effort upstream. At the estimation level, analysts determine and recommend project scope, budget and schedule, during project selection. However, as the project starts, resources allocated to this project log their time in a timesheet database. Furthermore, added scope items could be introduced during the project life-cycle, which can exert a considerable impact on the graphical model in the form of model changes, as either additions or eliminations. These scope changes, as well as the implementation time logged, update both the budget and the observers downstream so that schedule changes are visible. These observers in turn rank the project appropriately on the project evaluator according to its performance. In instances where the project scope is updated, the observers evolve the project status according to changes introduced and update the project evaluator and the schedule appropriately. This is an ongoing process. The model keeps track of these patterns for pattern analysis and future predictions, thereby aiding adaptive knowledge discovery on project trends. With this, improvements can be made in instances where divergent project trends are detected so that appropriate realignment with the business strategy is ensured. Equally, time patterns tracked on projects and applications can ultimately be used to improve the estimative effort for future projects.

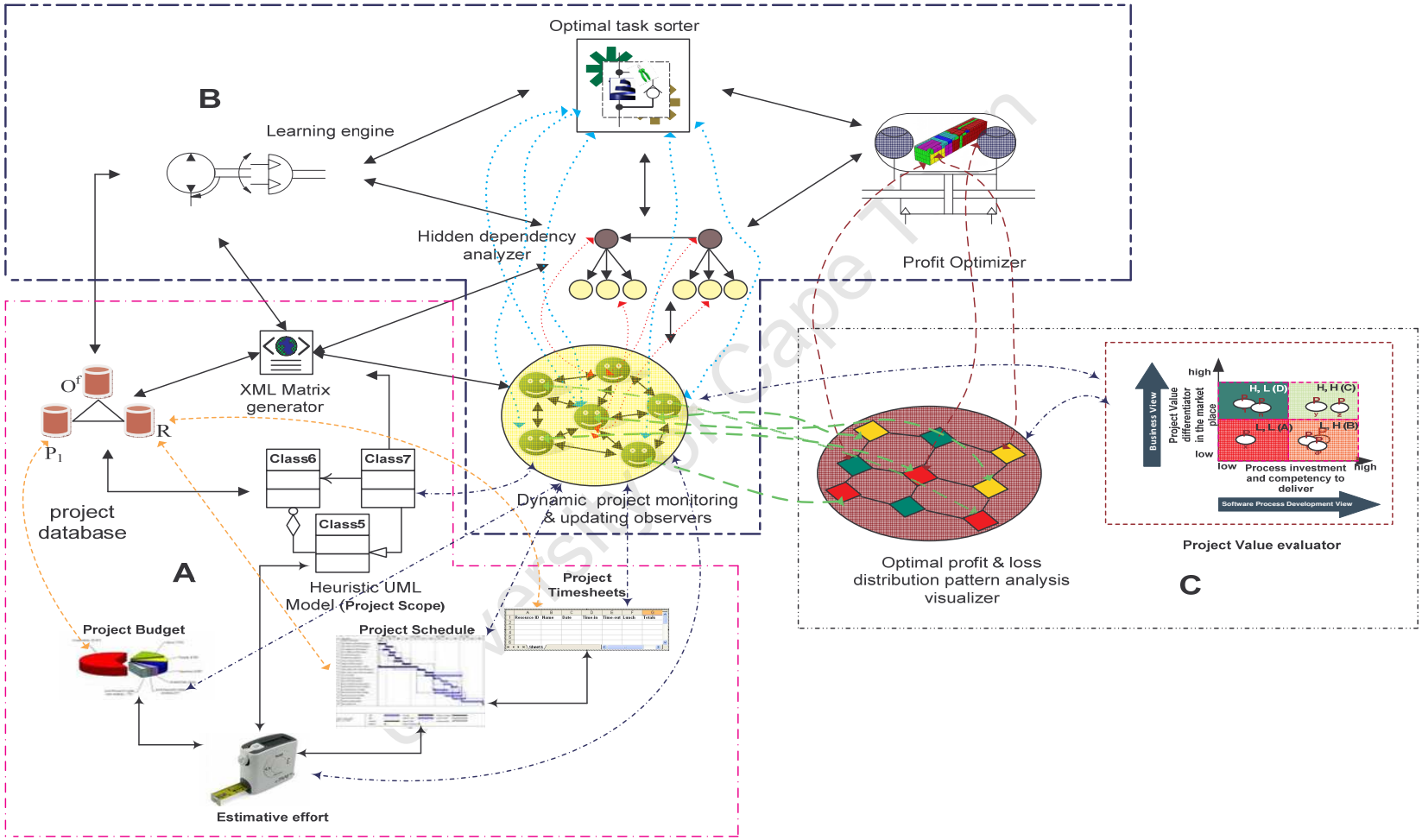
4.4 CASSE Framework Architecture

The foundation of the CASSE framework is based on understanding and modelling the entire project as a complex adaptive system in which patterns of interrelationships emerge from local interrelationships between requirements and their artefacts. As we tune the value matrices on the project, new patterns emerge on an ongoing basis. These emerging patterns could have a huge bearing on the profit-loss and distribution curve for the project. This is referred to as emergence or surprises for that matter.

In order to embrace this contingency in project modelling and analysis, the design of the framework architecture was levelled-out into three levels, that is, A, B and C each with specialized roles and functions. These three layers are described below.

University of Cape Town

Figure 18: Framework mode



- **Level A** denotes the initial stages of project development in which the adaptive management of requirements is vital. At this point a lot of effort is concentrated on capturing all known project variable including requirements and their artefacts. This information is logged in the project database with P_1 – Individual Project Characteristics, R – Project requirements and artifacts and O^f – Other factors. From the static graphical/ logical AO model developed using the UML methodology, a dependency matrix is derived and extracted using the XML matrix generator.

  The XML Matrix generator classifies the model into structure, resource, task and causality. Structure would imply that two requirements are similar; resource would imply that two requirements depend on the same resource; task would imply that one requirement describes a task required for another and causality would imply that one requirement describes a consequence of another. The resulting matrix values are used by the learning engine in level B to determine hidden relations and how these hidden relationships can impact upon the model and overall project attributes. This dependency matrix defines values that are salient to risk analysis, profit maximization and project scope. At this level, requirement change control is seen as an influential externality that affects project management which must be managed with continuous model updates. In this regard, externally is understood to mean costs or benefits arising from any change in requirements which in turn affects the initial project scope and where this change is not reflected fully in the initial project cost given.

- **In level B,** the learning engine helps with the dependency analysis by handling the complexity checks as well as their overall management. This process presents an opportunity of comparing different scenarios in the model. For example, you can get several answers to questions like, "what would happen to the profitability curve if the client is not willing to pay for requirement X which was agreed upon in the initial model given that Y which has to be implemented uses resources from requirement X in the model? Or even what would happen to the profitability curve if the client introduces a new requirement Z which requires resources from requirement X?"

  The hidden dependency visualizer displays the object relationships. The resulting structure is mapped to the dynamic object monitors that constantly observes and update the structure with any changes in beliefs and in the database. They monitor dynamic patterns of evolving project risk traceability during the software development life-cycle.

University of Cape Town

These monitors determine the profitability curve which is visualized in level C.

- **Level C**: presents the project value evaluator upon which a given project is mapped and tracked. At this level the optimal profit-loss distribution pattern is in a moderated fashion. Here, the decision maker can decide which point is favorable for deriving the value of the project; so as to go to either the aggressive risk-taking route or the cost downside protection development route. In this architectural design, green would imply that the project will yield profit; orange would imply that the project will breakeven while red would indicate that the project is likely to go into losses. Otherwise, on the project value evaluator, the project will be mapped directly to one of the four quadrants that fit its profile.

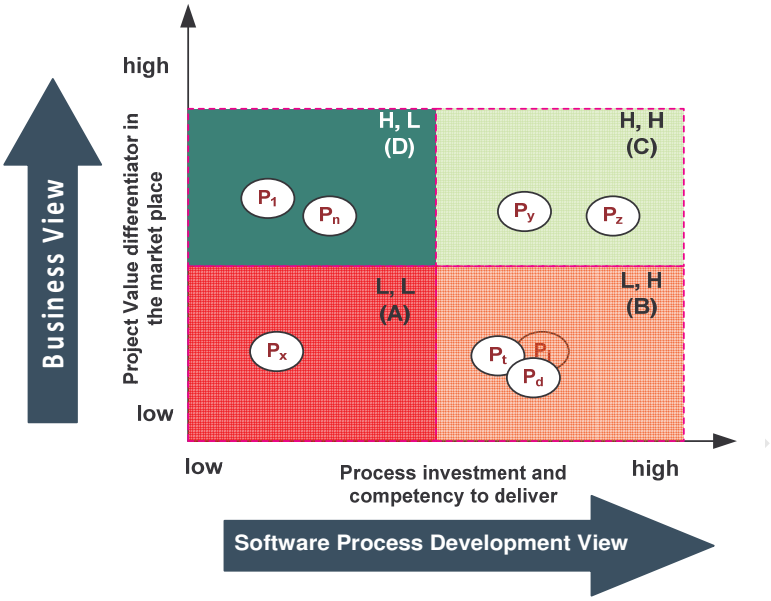
4.5 Project value evaluator

Out of industrial experience, observation is made about the prevailing need for informed feedback on projects especially when it comes to selling ideas to management or even convincing management about certain decisions that are strategic to business. Decisions makers often don't have a lot of time to read long project reports prepared by the technical people. Rather, they only require brief information which at the same time is informed enough to give them a wider view of what is being presented.

With this in mind, a project value evaluator was developed within the framework as a feedback channel for decision makers. The project value evaluator is the key feedback control translator and communicator used to broadly asses projects and their value within a given project portfolio. It compares project benefit against investment in the project as illustrated in Figure 8. This is based on the process and business view dimensions illustrated in Figure 2 within chapter 2 of this thesis.

University of Cape Town

Figure 19: Project value evaluator



The X-axis represents the software process development view in which measurement is done on the process investment and competency required to deliver a project within time, scope and budget. The Y-axis represents the business view in which the project value differentiator in the market is evaluated.

- **Quadrant A** indicates that a project has a low value and low process investment to deliver. This implies that any project falling in this quadrant will yield low return on investment and may not necessarily be valuable to the business, both in the short and long run. Therefore, projects that demonstrate such symptoms of low return yields and have limited promise should be exited and potential resources that would have been invested on such projects be channelled to other investment options
- **Quadrant B** indicates that projects falling in this quadrant have high process investment to deliver yet with low business value. This implies that the development process will be constrained with high investment required to implement and complete the project gracefully. This could be in terms of time or cost. The value axis in this quadrant indicates that there will be low value for that project in both the short term and in the long term. Therefore, projects in this quadrant are highly susceptible to project failure with high losses. However, they have significant promise and can easily be fixed.
- **Quadrant C** indicates that projects falling within this quadrant will have high development process investment as well as high business value. This could be for both

University of Cape

the short and long run time frames. This quadrant characterises new technology development projects which requires time to mature but have high value in the long run. Thus, these projects must be managed and exploited for the fact that cash flow from these projects can be used to fund more attractive investments.

- **Quadrant D** indicates that projects falling in this quadrant would have high value and low process investment. This is the most viable quadrant that every business would want to operate in. It reflects maturity in the software development process. It may imply that you have the right competency to deliver and that most of the features proposed already exist in your project database and/or software platform. It is very favorable for customizable software products. Whence, such projects exhibiting significant opportunity for growth must be given first priority as such projects hold the greatest potential for value creation.

As a profitability objective function, the framework requires that at project selection, AO dependencies are analyzed and the entire project is scored against the business value threshold for any project. The resultant effect of this dependency impact assessment as well as implementation time prediction would map directly to the project value evaluator. It is upon the decision makers in this regard to decide which route to take. Should the model results show that the overall project score falls within quadrants that have low value and low investment as well as high investment and low value; this would imply that such projects do not necessarily fall within the business line for this specific company. They are high risk projects that are capable of compromising the company progress business-wise. A company operating in these two zones (A & B) for more than a short period is likely to be edged out of the market quickly.

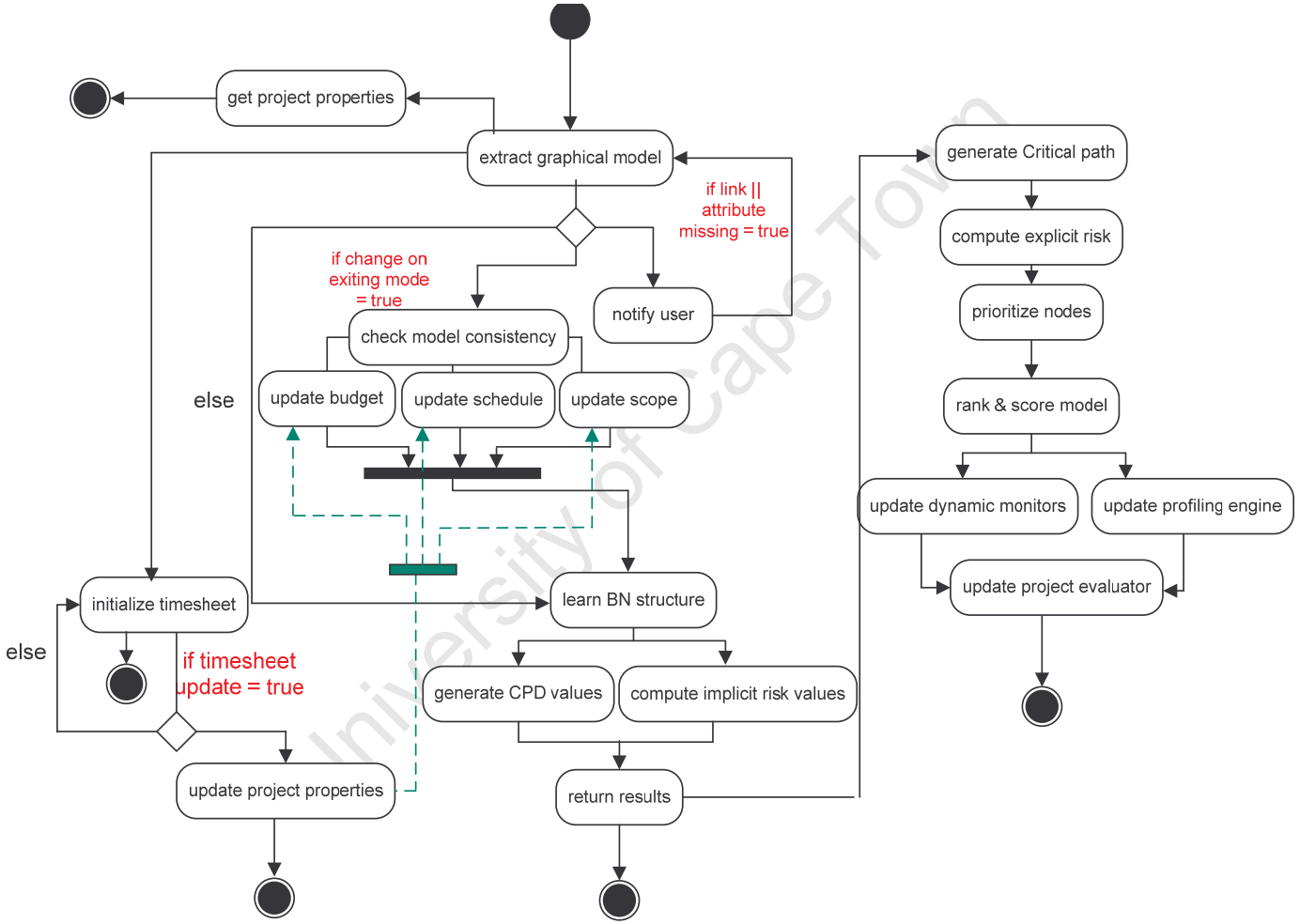
A company selecting projects falling within quadrant C will yield benefit in the long run but it must have enough investment to survive throughout the development process. Equally, it must also have a strategic business direction otherwise it is more likely to shift to quadrant B operation area if the business case for this project was not well assessed. In the event that the business case was well assessed and the business vision of this project is well articulated and maintained during the development process, it is likely to shift to quadrant D in the long run.

4.6 Algorithm flow

Figure 20 below illustrates the high-level algorithm flow within the framework.

University of Cape Town

Figure 20: Algorithm flow in the framework



University of Cape Town

As the model gets extracted from the xml file, the getProjectProperties algorithm is executed. This extracts all project properties such as resources, budget size, start data, finish date, project time among others. These properties can later on be reset at the model analysis stage.

Secondly, it also invokes the initializeTimesheet algorithm to initialize a new timesheet template for the extracted model and setting the FPs in the model nodes as part of the timesheet inputs.

More still, this extraction algorithm initiates the check model consistency algorithm. Here, the framework checks that dependency links exist between nodes and if there are any missing links or attributes. In the event this condition is true, the framework user gets notified. Otherwise, if there are no missing links, the updateBudget , updateSchedule and updateScope algorithms get executed. This process involves populating the budget properties. Functional points extracted from the model are then populated into the project schedule using the known dependencies.

Once successfully updated, the learnBNStructure algorithm is executed. This in turn executes the generateCPDValue and computeImplicitRiskValue algorithms that must return results used by the generateCritialPath algorithm. Once all values are returned, computing for the explicit risks gets done. Results from this computation drive the node prioritization in the model. Once this process is successfully executed, the updateDynamicMonitors as well as the updateProfilingEngine algorithms get implemented which in turn initiates the updateProjectEvaluator algorithm to display the project status accordingly.

However, in the event that a changed is being made to the existing model (this could be in terms of removing nodes, adding new nodes or functional points, changing dependency types or even updating the project schedule via the project timesheet database), the updateProjectProperties algorithm gets executed to update the scope, budget and schedule. This in turn invokes the learnBNStructure algorithm to relearn the model structure and update the downstream components accordingly. This kind of cycle is prevalent when development on projects start and this is where evolving project patterns start to emerge.

4.8. Scalability of the CASSE Framework

Scalability is a central problem in software engineering research. However, although the framework has not been evaluated in the operation environment, the L1 Regularized Path Algorithm adapted in this work is scalable enough to handle this

University of Cape Town

problem. Scalability issues pertaining to the optimization problem can be handled fairly efficiently via conjugate gradients of this algorithm (Rahul *et al,* 2009; Abernethy et *al*, 2009). This algorithm has also been tested in other scalable projects such as the one investigated by Koh *et al* (2007).

4.9 CASSE Framework Scope

By design, this approach provides an alternative approach in predicting project benefits from the onset using the initial AOs of the envisaged system and their properties.

Predictors: AOs and their dependencies are specifically analyzed to predict and understand project complexities and the economics surrounding system development, specifically looking at the project selection process. In system design, AOs dependencies represent the logical system model and how it translates into the implementation model. These AOs are driven off system use-case artefacts (Actors and their roles in the system) and their properties. Properties in this regard are the roles in the system being modelled as functional points which a system must fulfil.

An analysis is carried out of use-case artefact properties and into how a change in any of properties can impact on the project monetary value and probable project management complexities during the development process, thereby enhancing implementation prioritization and schedule optimization.

Applicability: The framework is only aimed for commercial software development rather than open source software development settings. For the purpose of this study it was based on the project selection process and tracking capabilities. However, the model is sufficiently generic to address other areas of the software development lifecycle such as change management implications on the overall system value as well as project value review evaluation at project completion. Once well adapted, the framework can be extended to address adaptive project management challenges, not only in software engineering but also in other disciplines.

However, it has not be adapted to allows for analysis of the trade-offs between writing software which is well documented and, therefore, more supportable but takes longer and costs more as opposed to lower levels of documentation and, therefore lower costs but more difficult to support.

Human resource problems on projects: This approach, if adopted and integrated in the development process, does not take into consideration resource issues that frequently arise on software projects. Key personnel on a given project can die or resign their positions.

University of Cape Town

There could be no leadership on the project, or if it exists, it could be very intrusive. At times there may be inadequate planning during the project life-cycle. As a competitive strategy, the approach assumes that an organization keeps a small core 'permanent' workforce, and a larger contracted workforce, so that they can contract for the right person. Otherwise, project managers constantly need to review project approaches based on available human capital resources and should ensure that there is clear leadership.

In the next chapter, presentation of the case study scenarios in which the framework was applied to determine value propositions and criticality of software projects is described.

University of Cape Town

CHAPTER FIVE

Case Study Presentation, Analysis and Interpretation

5.1 Overview

This chapter presents the various case studies illustrating the different dimensions within which CASSE is applicable. This framework has been tested on the following areas.

- Business viability assessment of potential software projects, which constitutes case study one;
- Bottom-lining functionality packaging on evolving or even customable software products, which constitutes case study two;
- Supporting the clarification and requirement negotiation process on software projects, which constitutes case study three;
- Project measurement and evaluation, which constitutes case four; and
- Task performance measurement on software project schedules, which is presented in the next chapter.

Empirical data from industry and results are presented in this section. Analysis of each case is made and case-based conclusions are made. It is upon this cross-case examination that the research questions for the study are answered.

5.2 CASE 1: Business viability assessment of potential software projects

5.2.1 Case study objective

This analysis aimed at examining how CASSE can be valuable in aiding the decision making process; in particular when facing uncertainties in business viability assessment for potential software projects to be added to the project portfolio. Inference mechanism provided for in the framework would thus help in assessing firstly, a favourable operating point on such projects. Secondly, for an interested software company, it would provide a capability for assessing if the business viability status obtained is worth the investment effort or not.

5.2.2 Case Study Description

This specific case study entailed evaluating system needs as presented in the RFP document for the development of an on-campus crime management system for Joliet Junior College, Illinois. In this RFP call, it was described as a police records management system. With the rate of crime and muggings on the rise at various areas on the college premises, the college authorities were focusing on implementing a comprehensive computerized policing records

University of Cape Town

management system. This system would fulfil the College's need for documenting and managing of crime records as well as overall crime control on the premises. By using a formal RFP process, Joliet Junior College management thus requested for proposals from any interested software company that would provide a useful software solution to this problem. A copy of this RFP is available in the appendix section of this thesis.

5.2.3 Project characteristics

This project had the following characteristics. The RFP was issued on March 7, 2008 and the deadline for interested company's questions was set for March 14, 2008. This particular scenario meant that the client expected any company to investigate proposal needs and make the required quotation within 7 days after issuing the RFP. The anticipated announcement of award was April 9, 2008. Any successful company for this job (based upon the lowest cost and the criteria set forth under the Evaluation section of the RFP) was expected to install the system by June 30, 2008. As a result, this would allow for only a maximum of 50 production days (about 400 man hours, excluding overtime) to analyse, design, implement, test and install the system. Joliet Junior College as a client was strict on the final product delivery deadline. This therefore rendered the project schedule inflexible requiring resource optimisation in order to beat the deadline.

This project thus assumed that any interested company would have the necessary capacity to deliver such a system within such a short period of time (production with agility while ensuring that all requirements are documented and understood). Vendors whose product-line fell within this ambit, therefore, would only need to customise existing features to suit the desired functionality. New players in the market, if successful, would have to develop everything from scratch or even to adapt existing open-source libraries to suit the desired functionality.

5.2.4 Case study procedure

As a systems analyst employed in this case company which had interest in responding to this RFP call, I was required by management to submit a quotation for this project proposal. This quotation had to take into account the high-level requirements given in the RFP document as well as other budgetary concerns. This case company was initially using heuristic ways of quoting for projects without any major metric or tool used for estimation but rather basing on project characterises and experience.

University of Cape Town

Therefore, the framework came in handy at this stage. The original estimate using traditional means was a round figure of R195, 000. The CASSE approach predictive figure as a new suggested approach however, achieved different results not far off from the estimated one.

The quoting process for this project was based on the last scenario describe above (new players) since this company had never developed any such solution of this kind before. However, this company believed that it had the right competency to deliver this solution within the specified schedule.

The results obtained by this approach were not final; they were only suggested as guidelines for informed decision making to management. These results were forwarded to management for the final recommendation on the viability of the project. This would eliminate bias in analysis and it would also give management an opportunity of comparing the suggested approach with the traditional means so as to ascertain if such an approach is worth integrating in the development process.

5.2.4.1 Quotation preparation procedure using CASSE analytics

Given this background and using CASSE analytics (a tool born out of the framework), I analysed the viability of the project based on the preliminary requirements in the RFP document. The idea for using this tool was that results obtained would be benchmarked with the heuristic figure obtained using the traditional quoting methods adopted in this company. This would provide a favourable situation to obtaining a reconcilable quotation for the envisaged project based on the two approaches.

The huge threat that prevailed in the analysis was that the requirements were incomplete and were lacking in clarity at this stage, thus any modelling and subsequent estimation would only be indicative of the likely project status. There were higher chances that the project requirements would change as the client got to understand the system needs better. This would ultimately have cascading effects on the overall project budget, timelines and scope.

Consequently, CASSE analytics was specifically applied to illustrate how one can fine-tune the project constraints to ensure that a viable project is selected and to predict the viable implementation time, to rank the project value and to prioritise the features in order to optimize a given production process.

University of Cape Town

5.2.4.1.1 Analysis parameters

In this analysis, parameters of interest included the following.

- Actor Object Graphical model resulting from the high-level requirements provided;
- The overall anticipated budget that would be obtained as a sum of expected development costs and other expenses on the project;
- Expected income as the overall project figure derived;
- Expected net profit after budgetary expenses;
- Overall time span as provided for in the RFP document;
- Predicted time from CASSE analytics; and
- Expected slack/overran time as a difference between available and predicted time.

The highest investment value of R700 (as the hourly rate for this project) per functional point was proposed. This value represents the actual costs incurred in analysing, designing, developing and testing this functional point on an hourly basis. On the other hand, the proposed highest return on investment was R2, 500 per functional point. This meant that anticipated profit per FP amounting to R1800 would be realized as a mark-up difference between high return on investment and high investment.

Making any adjustments on any of these parameters would results in assessing a most suitable operating point. This point would be used in making investment decisions on this project.

5.2.4.1.2 Obtaining the Actor Object Graphical model

A use-case realization technique was applied in deriving the Actor Object Graphical model. Specifically, on top of conceptual modelling, this process involved counting the number of functional points what would be required in this system, given the available requirements in RFP document. A computer-aided software engineering tool available in this company was used to manage this process.

The Actor Object Graphical model generated from this tool was then imported into CASSE analytics for result analysis. This model was extracted and exported as an XML file. This entire process was completed within 6 to 7 man hours. Model quality was checked by another system analyst within the company in order to ensure completeness, and also to eliminate bias in analysis.

The discovery at this point was that once the user is very conversant with the use-case realization process, functional counting would eventually become easier to understand. Secondly, any analyst well vast with Object-Oriented system analysis and design

University of Cape Town

methodology would not find it difficult to use the framework, as it only requires extraction of the graphical model which is passed as an XML file to the analytics tool. Evaluation of the project results also did not require much training in using the framework as it provides clear guidelines, provided analysis parameters are well understood.

The biggest challenge however is ingrained in interpreting the meaning for the different project status indicators mapped to the quadrants on the project value evaluator and how they relate to the business strategy. Secondly, being able to understand the required cumulative time pattern implications of the different AO in the model is essential. This would set the perspective for establishing viable acceptance criteria on the project. This is a potential area that requires training when using this framework.

5.2.5 Case study analysis results

The following results were obtained, as shown in Table 3 (the figures have been adjusted for confidentiality purposes but are relatively reflective of the project status).

Table 3: Project viability status

Item	Quantity
Total project costs (Budget)	R125,000
Expected development costs	R50,760
Other project expenses	74240
Expected project income	R211,500
Expected gross profit	R160,740
Net profit expected	86500
Time span	400hrs
Predicted time	194hrs
Expected slack/overrun	206hrs

With a budget estimate of R125, 000 (as a sum of development costs of R50, 760 and other budget expenses of R74,240) the model predicted 194 hours of project time out of the expected 400 hours derived from the RFP document. This left 206 hours of slack time. Expected income in this regard (as the overall quoted figure) was R211, 500. The difference between expected income and expected development costs resulted in R160, 740 as profit before deduction of other budgetary expenses. Net income would thus be R86, 500 derived as a difference between expected project income and the anticipated total project budget. This was the expected profit on this project.

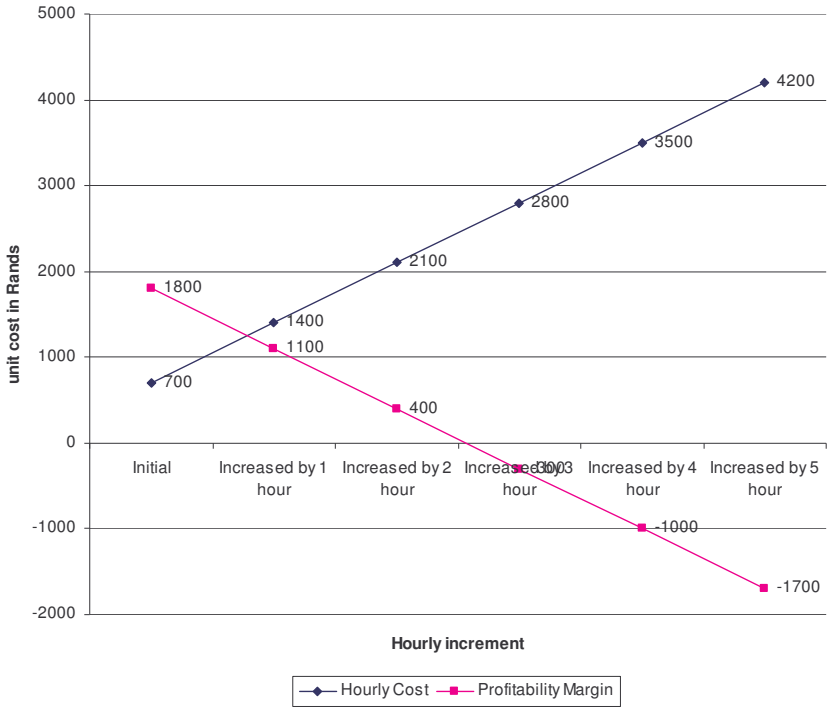
5.2.5.1 Added time implications

The model results showed that for this anticipated profitability to be sustained throughout the software development lifecycle, development time had to be kept within 194. Any added time over the predicted project time would mean that the project became

University of Cape Town

unprofitable. As illustrated in Figure 21, this added time would mean a reduction in the mark-up value anticipated per functional point.

Figure 21: Added time implications



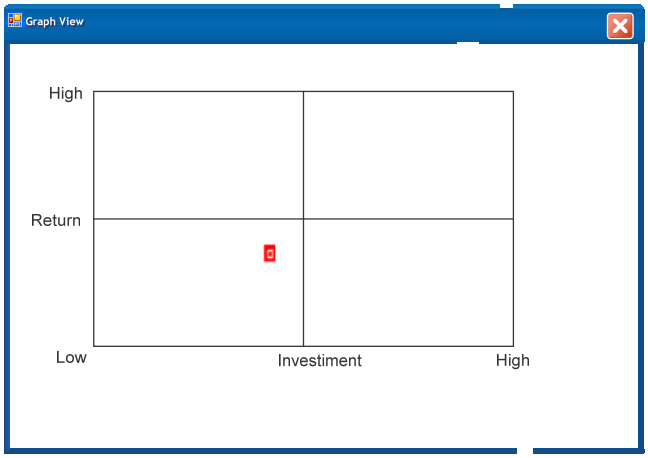
The graph above shows that as development costs rise, profitability per FP decreases. Adding an extra hour per functional point would mean that the highest investment value envisaged doubles to R1400. R700 of R2500 results in a profit margin of 72%. Doubling this value to R1400 would result in a 28% profit margin reduction on a given FP profitability. If this process continues beyond the breakeven point (where development cost equate to the highest return on investment per FP), losses on this FP will be inevitable. If this scenario cuts across the entire FPs in the model, the project would thus be in losses. Alternatively, in the event that the budget allows for provision of more project funds, the reverse to this scenario would be true; provided no new requirements are introduced.

5.2.5.2 The quadrant analysis

Mapping this project to the project value evaluator revealed that this project fell into Block A, the risky quadrant of low investments and low returns, as illustrated in Figure 22.

University of

Figure 22: RFP project mapping



In the figure above, the X-axis represents the investment status on the project while the Y-axis represents the business status on the project. The red dot on the graph shows the position of the project on the value graph. Fine-tuning such values with different value combinations would return different mappings on this graph. For example, adjusting the minimum investment and maximum return would change the project status significantly depending on the adjustment percentages made. However, as a business there is a set threshold for minimum warranted investment levels that must be guaranteed on any project. Therefore, for a project that has to be profitable, the adjustment tradeoffs must guarantee the initial investment envisaged.

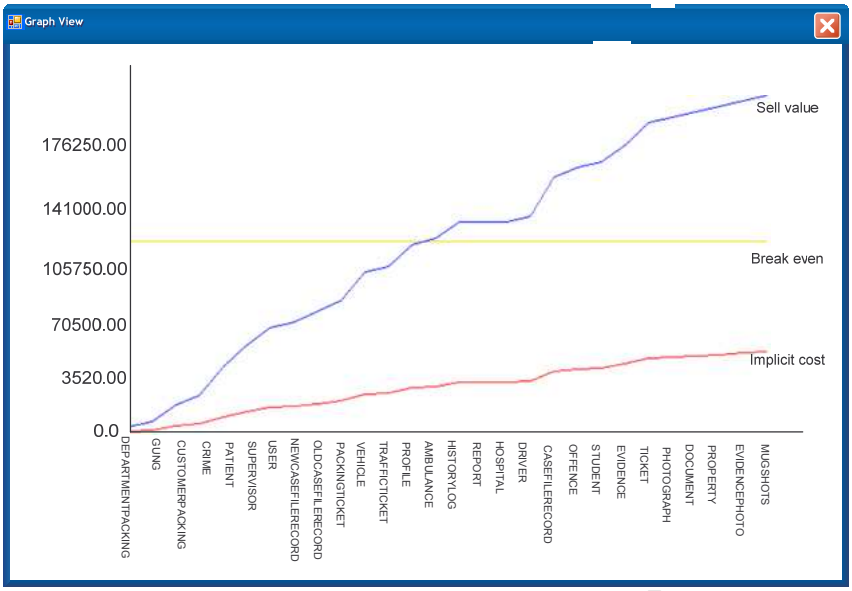
This implies that this given project was susceptible to being risky in that it attracts fewer profits in comparison to expected profitability threshold for any project taken on by this company. As illustrated above, given the current budget, this project would yield only 51.81% profitability levels on the overall (R86, 500 of R160, 740) yet targeted profit on any project for this company must be over and above 70%. Although this can be considered a very high level profit margin commercially, there was however, a high likelihood that these projected profits would be tattered by change management bottlenecks, evolving project needs among others. Overcoming such fixed price quotation issues would entail having good project management skill (which was lacking in this regard) that would limit changes or charge for them separately.

5.2.5.3 Profitability/AO graph analysis

Viewing the same project on a profitability/AO graph would give a visual indication of the likely break-even point. The break-evening point on this project was around the project budget amount as illustrated in Figure 23.

University of Cape Town

Figure 23: Profitability/AO graph



In figure 23 above, the y-axis represents the project costs while the x-axis represents the AOs. The red curve represents the implicit cost of each AO and its associated dependants against the overall implementation cost required to complete them. The blue curve represents the sell value of each AO and its associated dependants in the model against total implementation cost. The intersection point of the yellow line and the blue line represents the break-even point. This graph shows that, for this project, profitability decreases as the total implicit costs rise towards the sell value curve. This was likely to occur during the development lifecycle because of volatile requirements on this project and other project management bottlenecks. The reverse is true for the profitability increment of the given AOs and their associated dependencies. Therefore taking on such a project requires that, at the implementation stage, project costs must be kept within the anticipated boundary if the project is to stay profitable.

5.2.6 Case study implications

According to the project data published on the college website, the tender for this project was awarded to the successful vendor for about R400, 000 and was completed on time.

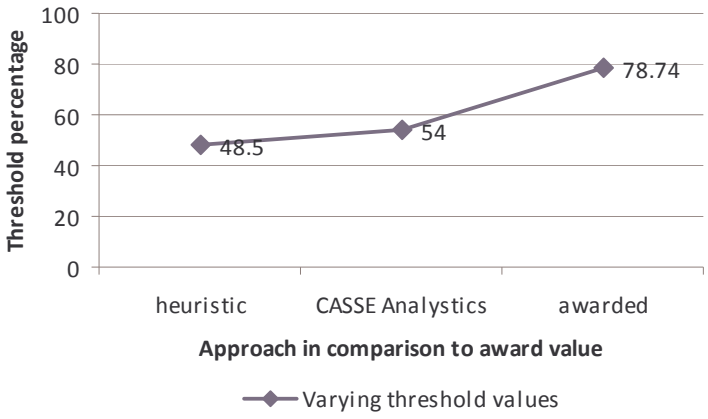
Therefore, comparing analysis results against the awarded amount resulted in the following observations.

This case company has a 70% profitability margin threshold for any projects to be added to the project portfolio, in case such projects don't have long term strategic benefits.

University of Cape

When comparison was made between CASSE Analytics and the heuristic amount initially projected against the amount for which the tender was awarded, the following results were obtained as illustrated in Figure 24.

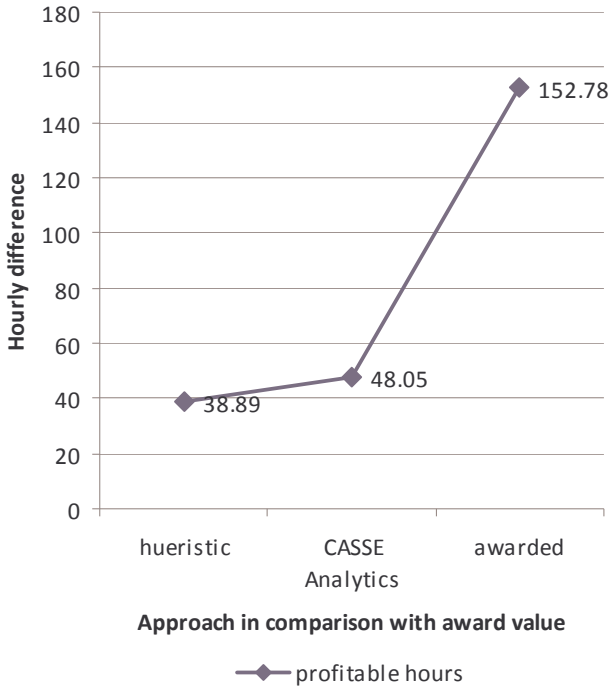
Figure 24: Threshold implications



Scoring the heuristic value (of R195, 000) against the profitability threshold revealed that this project was likely to fetch only a profitability margin threshold of 48.5%. This therefore resulted in a 21.5% difference away from the expected profitability margin threshold. CASSE Analytics on the other hand projected a 54% profitability margin threshold; thereby rendering 16% difference from the expected profit margin threshold. This analysis therefore showed that by using the CASSE framework, this company would improve its quotation process in terms of profitability margin assessment by 5.5%. This percentage difference conversion can easily be interpreted by the profitability hour difference analysis illustrated in Figure 25.

University of Cape

Figure 25: Hourly project difference



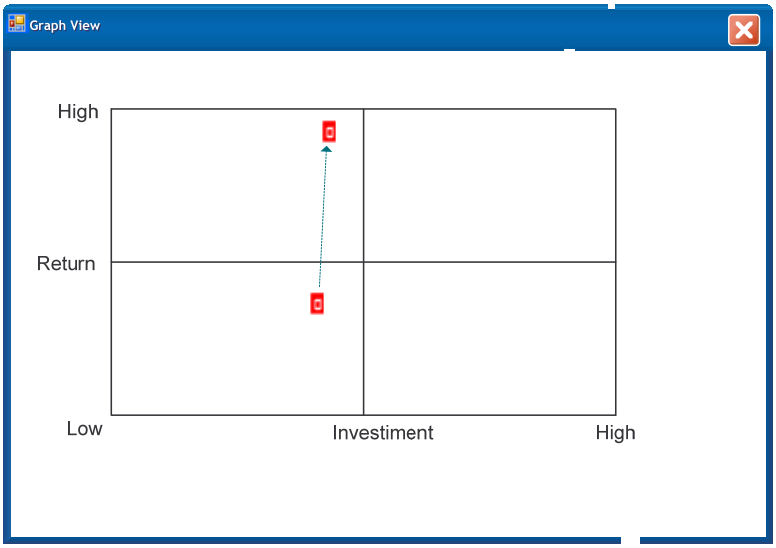
While analysing the hourly differences between the heuristic method and the predictions from CASSE Analytics, the model revealed the following results. It was observed that based on the awarded amount and after subtracting the total project costs, available profitability difference per functional point on an hourly basis would be 152.58 hours. This is derived as award amount less total project cost divided by the hourly profit margin of R1800.

The heuristic method assumed that there would be 38.89 hour profit maximisation per FP. CASSE Analytics on the other hand predicted 48.05 hours profit maximization per FP. This thus provided a difference of 9.16 hour difference between what CASSE Analytics provided and what was assumed by the heuristic method. Ultimately, when this hourly difference is translated into monetary value, considerable return on investment will have been increased substantially by R16, 488 (9.16 multiplied by R1800). This would in turn improve and resultantly shift the profitability curve upwards.

When anticipated profitability was adjusted in accordance with the award amount, the project status shifted into a different quadrant of the project value evaluator as illustrated in Figure 26.

University of Ca

Figure 26: Quadrant shifting



If this R400000 was the original amount quoted on this project, this project would be displaying in quadrant D which as a favourable quadrant for investment.

These results therefore demonstrate that various investment questions can be answered with guiding tools such as CASSE Analytics. Business viability assessment of potential software projects requires such tools if informed decisions have to be made. Information provided by such tools not only aids business case assessment and evaluation of the project but also identifies risky projects before investment commitment is made on such projects.

This process revealed that tool estimate results were not far from what management expected. However, the difference was that by adapting to tool usage, process improvement would have been ensured, as project metrics that impinge on the business strategy would be defined for future projects. Secondly, personnel involved in the development process would have been presented with the opportunity of being more involved with the software estimation process. This in turn would ensure that business and process needs are aligned in the long run.

5.3 Case two: Bottom-lining functionality packaging on customable software products

5.3.1 Case study objective

The primary objective for this analysis was evaluating how the AOD technique within the CASSE framework can aid functionality packaging on customizable products. The biggest challenge presented by customizable software products is pricing. For example, it is always a

University of Cape Town

challenge to gauge the quotation baseline on a customization request as new functionality against existing functionality on such products.

Given this background, CASSE analytics was applied to assess system dependencies using AOD so as to ascertain the following.

- Determining the viable customization base-line of system modules and features;
- Determining the relationships and strengths between different AOs in the model - with an overall aim of identifying which AOs were critical in providing overall system functionality given others; and
- Extracting an AO-based value matrix which could be used as a basis for quoting on future customizations on the system and system maintenance impact assessments. As an additional benefit, the resultant matrix would highlight module dependencies which in return would highlight strategic system integration points.

5.3.2 Case study description

This case involved assessing product feature dependency analysis of a customizable mobile patient monitoring system. This system was developed by a leading Non Government Organization (NGO) in South Africa. It specializes in developing mobile health management systems with a client base in both the national and in the international market. The Intelligent Dispensing of Antiretroviral Treatment system (iDART) is one of the products in this NGO`s product line. iDART is a software solution designed to assist ART patient monitoring as well as the supply chain management of Antiretroviral drugs in the public health sector in South Africa. Developed in the Java technology, it is capable of providing real-time assessment of an ARV programme through the generation of a various reports.

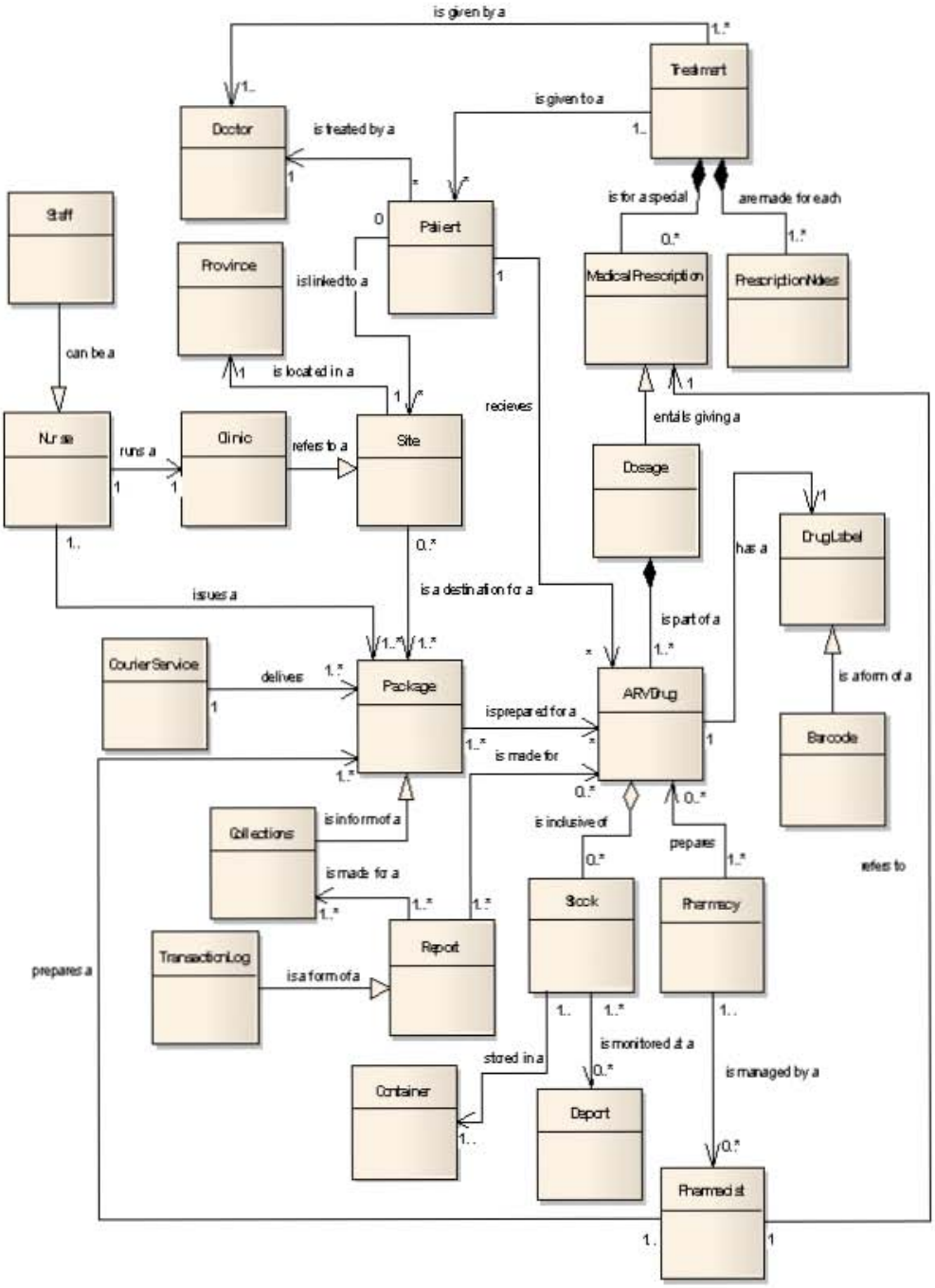
5.3.3 iDART product characteristics

The iDART system comprises of two applications for different functions within the supply chain. The first application (*iDART Pharmacy*) is housed at a central supply pharmacy, and the second application (*iDART Clinic*) is installed at a remote clinic where patients collect their medication from a nurse. These two applications can operate in geographically different locations. Some of its features include: Patient management, prescription management, dispensing drug management, label printing, report generation, stock management, remote clinic management as well as data export functionalities. This project was developed and released under the General Public License (GNU) as Antiretroviral dispensing software in the public health sector in South Africa. Some of the components of this system utilized existing libraries; thus not all AOs on this project were new; rather a

University of Cape Town

combination of existing and new components. A high-level AO graphical model of the iDART system is given in Figure 27.

Figure 27: iDART Logical Model

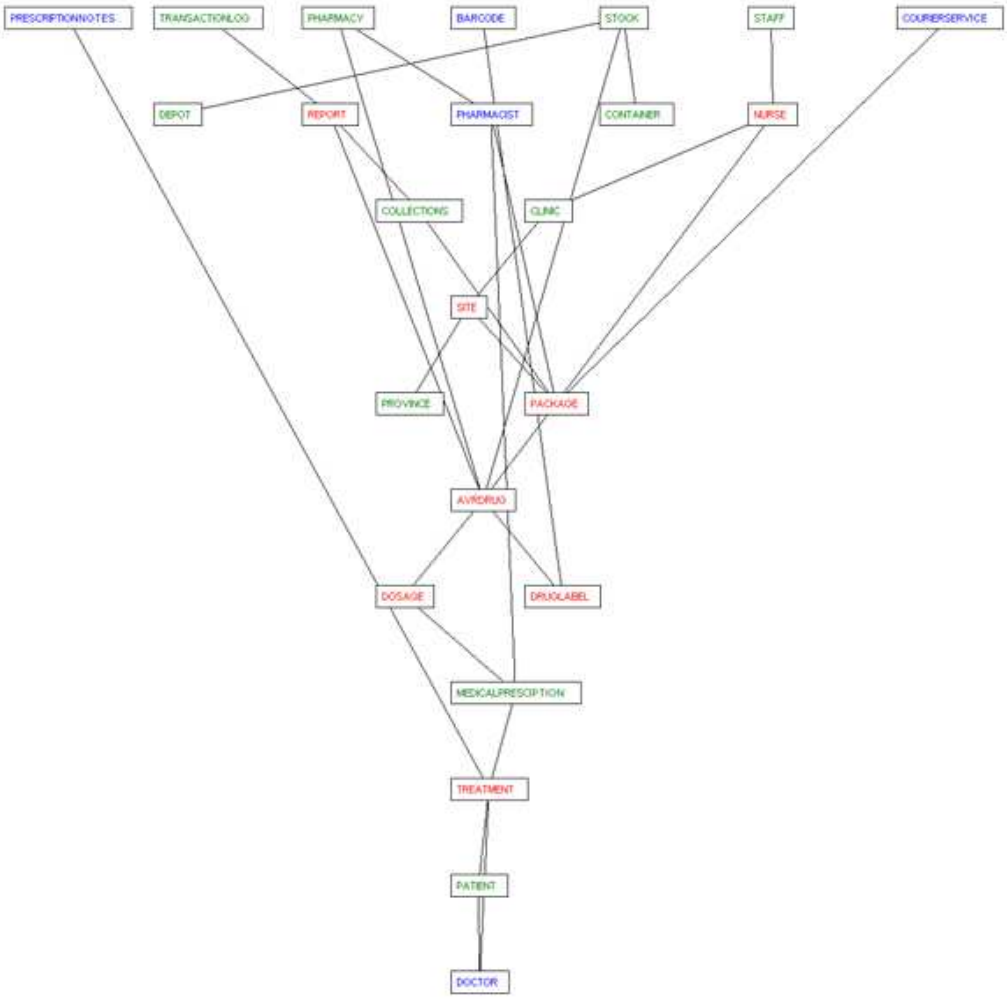


Time and again various clients approach this organization requiring customized features of this system. Clients interested in such features are required to pay customization costs.

5.3.3 Case study procedure

To assess the AO dependencies, the logical system model developed in UML as given in Figure 26 above was used. This logical model was imported into CASSE analytics tool for assessment. Based on the nature of interrelationships, the dependency assessment classified AOs into 3 categories into; High (Marked red), Medium (marked Green) and Low (marked blue). This is illustrated in Figure 28.

Figure 28:iDARD AOD model



The AOD graph above illustrates the complex interdependency of all the AOs in the iDART system. High risk AOs take the highest risk in the model. AOs in this category have a high impact on their dependants. Operation structures in these high risk AOs have a high task and causality impact. A high task dependency in this regard implies that certain AOs` tasks are described by these high risk AOs. A high causality dependency in this regard implies that these high risk AOs describe a consequence of their dependants in a medium risk or lower

level. Therefore, overall functional performance in the model would rely heavily on these AOs if desired system behavior is to be achieved.

Medium risk AOs in the model have an averagely significant task and causality impact. However, they do have a high resource dependency with their parent. Resource in this regard implies that these AOs depend on their parent for the same resource. Any changes in their property parameters would not influence the model immensely but would have significant effect on the overall model outlook.

Low risk AOs have a less task, causality and resource dependency. Their overall status in the model does not directly impact on overall system behavior. However, any change in any properties of these AOs would change their category classification status thereby changing their dependency impact and ultimately desired behavior in the iDART system.

Using the complexity check module of the CASSE framework, this emergence pattern is bound to be propagated through the entire model each time a change is made on the model property. This is also true when a new AO is introduced or an existing AO is deleted. This kind of dependency checks therefore, support impact assessment analysis, customization baseline, and functional sizing measurement as well as change management implications.

5.3.4 Case study analysis results

5.3.4.1 AO-based matrix used to determine functionality packages

An experimental analysis was done to determine AO dependencies on this system as illustrated in the Table 4.

University of Cape Town

	Risk Level	No. of Strong dependants	Dependants
ARVDrug	H	12	Clinic, Collections, CourierService, Nurse, Package, Pharmacist, Pharmacy, Report, Site, Staff, Stock, TransactionLog
Barcode	L	0	
Clinic	M	2	Staff, nurse
Collections	M	2	TransactionLog, Report
Container	M	1	Stock
CourierService	L	0	
Depot	M	1	Stock
Doctor	M	18	ARVDrug, Clinic, Collections, CourierService, Dosage, MedicalPrescription, Nurse, Package, Patient, Pharmacist ,Pharmacy, Report, Site, Staff, Stock, TransactionLog, Treatment, TreatmentNotes
Dosage	H	13	ARVDrug, Clinic, Collections, CourierService, Nurse, Package,Pharcist,Pharmacy,Report,Site,Staff,Stock,TransactionLog
DrugLabel	H	14	ARVDrug,Barcode,Clinic,Collections,CourierService,Nurse,Package,Pharcist,Pharmacy,Report,Site,Staff,Stock,TransactionLog
MedicalPrescription	M	14	ARVDrug, Clinic, Collections, CourierService, Dosage, Nurse, Package, Pharmacist, Pharmacy, Report, Site, Staff, Stock, TransactionLog
Nurse	H	1	Stuff
Package	H	10	Clinic,Collections,CourierService,Nurse,Pharcist,Pharmacy,Report,Site,Staff,TransactionLog
Patient	M	17	ARVDrug, Clinic, Collections, CourierService, Dosage, Nurse, Package,Pharcist,Pharmacy,Report,Site,Staff,Stock,TransactionLog,Treatment,TreatmentNotes, MedicalPrescription,
Pharmacist	L	1	Pharmacy
Pharmacy	M	0	
Province	M	4	Site, Staff, Nurse, Clinic
Report	H	1	TransactionLog
Site	H	3	Nurse, staff, clinic
Staff	M	0	
Stock	M	0	
TransactionLog	M	0	
Treatment	H	16	ARVDrug, Clinic, Collections, CourierService, Dosage, MedicalPrescription, Nurse, Package, Pharmacist, Pharmacy, Report, Site, Staff, Stock
TreatmentNotes	L	0	

Table 4: AO value matrix in the iDART system

Table 4 above illustrates the classification of each AO in the model (M: Medium, H: High and L: Low). The number and names of dependant AOs that strongly depend to this AO is given.

University of Cape Town

The framework revealed that certain AO have strong relationships with others in the model yet others do not necessarily have strong relationships with others.

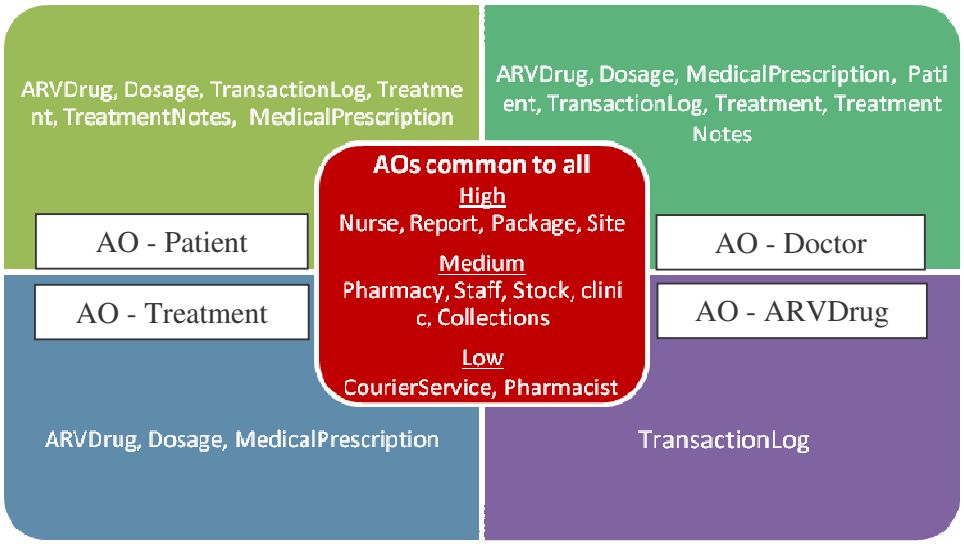
The results depicted in Table 4 reveal that Each AO with its associated AOs therefore form a system package within the system. This kind of packaging would form an aggregated estimation of the number of functional points impacted if a customization request is made. Profitable quotation on either system package therefore, holds a maximum macro value of total number of functional points contained within each AO multiplied with the productivity factor of 7.2. Any customization made for less than the maximum hours required per AO therefore would increase project revenue. In other words, if the customization of a given function point takes less hours than initially scheduled, monetary value in terms of time saved on this FP will have been realised. If this trend cuts across various FPs customized for a given customization request, this will eventually ensure that development time is optimized. Secondly, accumulating such valuable time will enhance the viability and profitability of such requests. Revenue generated in turn will ensure business continuity on developing and extending the capabilities of these evolving systems.

The reverse would be un-profitability of any customization request.

5.3.4.1 Functionality package example

As illustrated in Figure 29, customizations that require adjustments in functionality pertaining to patient responses to ARV drug treatment given by a qualified doctor will impact on the following AOs.

Figure 29: Common AOD scenario



iversity of Cape Town

According to the model results, 11 AOs cut across the four packages, out of which 4 are of high risk (Nurse, Report, Package and site as depicted in Figure 28), 5 are of medium risk (Pharmacy, Staff, Stock, Clinic and Collections) and 2 are of low risk (CourierService and Pharmacist). This implies that changes made on any of these four AOs in one system package in the model will have to be propagated to the rest of the other packages specifically addressing resources or tasks from the 11 common AOs. This impact therefore has to be taken into consideration if any customization request is to be made in terms of hours required to achieve overall system behavior and functionality. For a client willing to pay for limited functionality, dependent AO costs also need to be built into the quotation since a change in any of the properties of the model will propagate through other system packages.

5.3.5. Case study implications

This illustration shows that the complexity associated with hidden dependencies between project requirements, if underestimated, poses serious project risks. The case study results presented in this section demonstrate that based on AOD analysis and their properties, it is possible to estimate the probability distribution of losses and earnings that can be incurred by a software development organization according to its software project portfolio. This analysis also shows that viable quotation of customization products is attainable given the dependency of the AOs in the product logical model.

This further shows that successful project accomplishment depends on addressing technical complexity, market and financial uncertainties and competent manpower availability throughout the development process. Therefore, in order to improve our project selection processes and the entire software development processes effectively, it is important to reduce the level of uncertainty at all levels. The key strategy therefore is to continuously assess the impact of each AO in the model so as to derive different dimensions of system implications. Profitability on evolving systems needs to be well tracked and analyzed firstly if initial investment made on such products has to be realized and secondly if the product has to remain profitable in the long run. As we make inroads into certain market niches, we therefore ought to ensure that as our products mature that we are always operating in the profitable quadrant of the value conundrum for competitive advantage purposes.

Therefore, by adopting new requirement engineering techniques, it is hoped that evaluation of project viability will be enhanced and that feedback control mechanisms on projects will also be improved. As given in the next section, this approach is not limited to software

University of Cape Town

development per say, it can also be applied in other project management scenarios in other disciplines.

5.4 Case three: Supporting the requirements clarification and negotiation process

5.4.1 Case study objective

In this analysis, investigation aimed at evaluating how the CASSE framework can support the requirement clarification and negotiation process on software projects. This was driven by the view that requirements negotiation on software projects necessitates use of value-driven approaches that guarantee a win-win situation for all stakeholders, thereby permitting agreement on desired functionality and stakeholder commitment on projects.

The purpose of such an approach is to reach stakeholder agreement on what the real acceptance criteria would be for a designated phase or release of a software product given the technical complexity reality and perhaps technology constraints, thereby establishing a shared vision of the real requirements for the project. In this way, requirements negotiation and project technical complexity clarification are increasingly enhanced so as to support viable budget, scope and schedule estimation.

5.4.2 Case study description

During this research period, I was contracted by a Government Department over a 7 months period to assist with a RFP preparation process. This was for the automation of the birth and death registration system as a service delivery improvement initiative. This Department was in the process of automating its business processes as part of a Government mandate to lay grounds for integrating technology into the daily management of its activities.

The Department is responsible for managing and issuing birth and death registration certificates to both nationals and foreigners in the country. Output data collected is then used by other Government Departments (such as the national bureau of statistics) for planning purposes. However, the current system is manual and prone to a great deal of errors; moreover, various loopholes still exist, most of which impact on the Department's image and overall service delivery expectations.

Through the initial needs assessment exercise commissioned by the Department which i executed, the current Departmental processes were evaluated. In the resulting report, initial technology integration areas were identified. This report served as a basis for evaluating the initial requirements. The major hope was that with the introduction of automated business process management, productivity would be enhanced in this sector and redundancy

University of Cape Town

eliminated. Ultimately, this initiative would fit the Department into the overall e-Governance initiative commissioned by the central Government.

5.4.3 Client interest

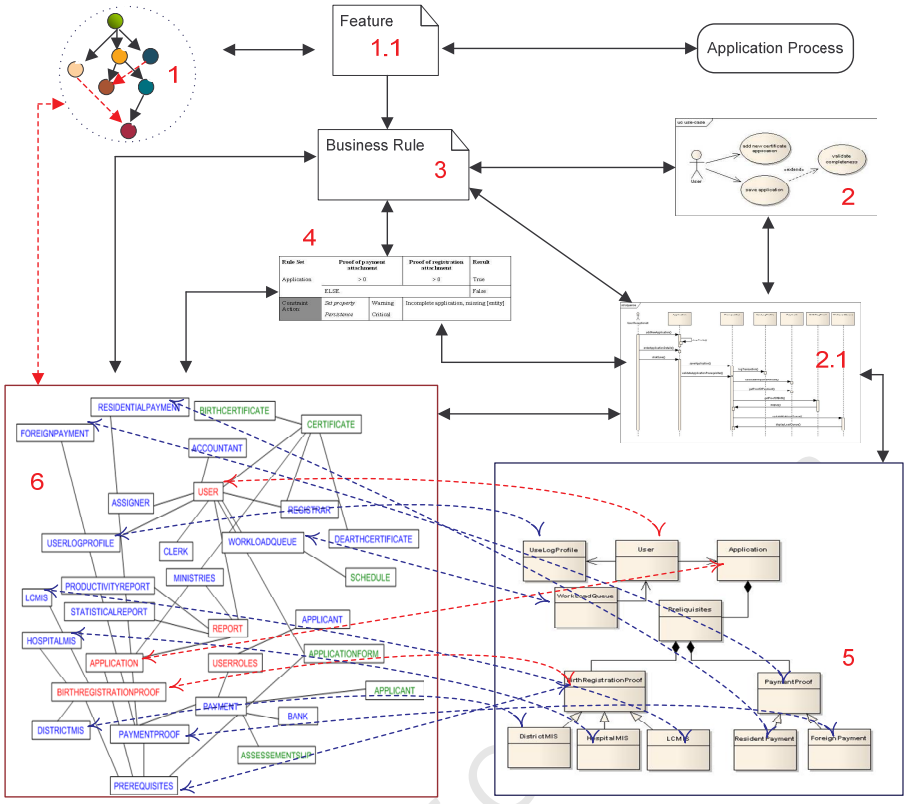
The client was interested in using this report to call for an RFP from interested vendors. However, he had some financial and technical concerns. He wanted to implement this system in an incremental approach starting with certain modules that mark the beginning of the business process, specifically the front-desk processes where major loopholes existed. With this strategy, employee resistance to change would be mitigated. At the same time, the amount of time and resources spent on training users on the new system would be scaled down considerably, since modules would be added as users became accustomed to working with the new system. To accurately gauge a fair scope and budget, the client wanted guarantees that the identified areas in the report would embrace those that would be solicited in the RFP. He was also interested in ensuring that a credible vendor, who would charge a fair price given the current need, be identified.

5.4.4. Case study procedure and analysis

In addressing the challenge, I applied my technique to measure this complexity within the given features so that a prioritization baseline could be established. This prioritization base-line would indicate which feature would be attainable in a given module. I conducted an analysis of the pressing features that the client wanted. In this analysis I exemplify my approach with a partial system only, due to confidentiality issues. I used a 6-stage use-case realisation dimension to achieve these results, as illustrated in Figure 30.

University of Cape Town

Figure 30: Bird's-eye view of analysis decomposition levels



From this list, I executed a feature sweep-through to decide which features should be implemented and those potentially for elimination, determine features to be implemented first, to assess system architectural implications, to identify high-priority features that should have technical performance measures and to manage the requirements creep huddles while at the same time ensuring that the system generates value to the client. I was only able to achieve this by assessing the downstream effect through the realization process starting with stages 1 through to 6 as illustrated in the figure above. With the client indicating that he wanted the system implemented in an incremental approach starting with front-desk processes, there was a need to balance client expectations against available resources. I achieved this by identifying key features that would form an architectural foundation, defining and agreeing upon trade-off decisions so that realistic schedules could be produced. My major emphasis was in using the framework to identify key AOs in the overall logical model since different features have different values to the business including tactical usefulness and long-term strategic value.

To analyze the technical risk associated with this project, I evaluated the number of functional points that had to be passed between different AOs at execution time. With this analysis, I could firstly be in a position to have a rough estimate of the likely functional points

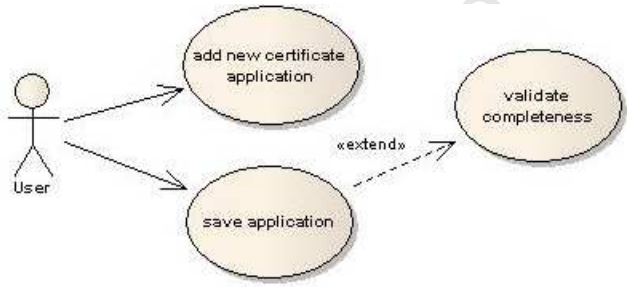
University of Cape Town

required to achieve a given feature. Secondly, evaluating which AO in the model is predominantly referenced by other AOs in a model, given its risk category, would help me in understanding the architectural needs of the system.

For instance, the client wanted to have a mechanism which would electronically validate that submitted applications have all the requisite accompanying proof documents as required by law. To illustrate the complexity associated with this feature, I evaluated the number of Actor Objects that will be impacted by this feature at the interaction level.

At stage 1, I identified the feature as the *electronic validation of application prerequisites* in the priority tree and the business process where it is executed. This feature is managed and handled at the application processing stage within the business process of issuing a client certificate. This feature results in the following use-case illustrated in Figure 31 (stage 2) and business rule (stage 3) respectively.

Figure 31: Use-case diagram



The user actions captured at this level, that is, stage 2, include: adding a new certificate application and saving the application, which *extends* validating that the submitted application is complete with the required proof document. This translates into the following business rule:

Business Rule: APV01: Each application must be fully paid for; must be accompanied by payment proof; must have a birth registration proof from the point of birth.

The system rule resulting from this business rule is ValidateApplicationPrerequisites expressed in the Table 5.

University of Cape Town

Rule Set	Payment attachment	Registration attachment	Result
Application	> 0	> 0	True
	ELSE		False

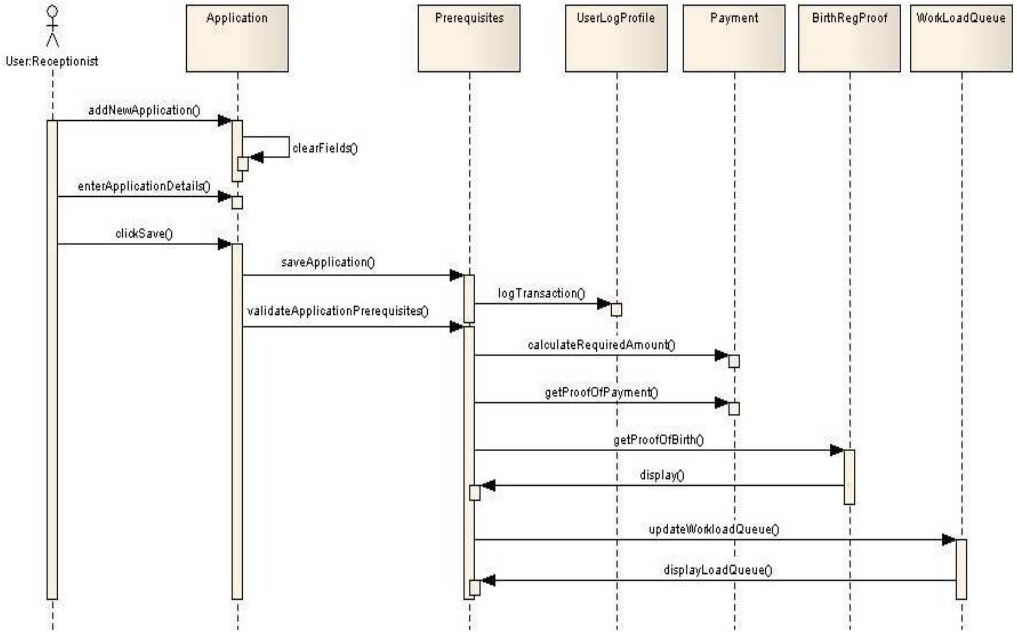
Constraint Action:			
	Set property	Warning	Incomplete application, missing [entity]
	Persistence	Critical	

Table 5: System rule

This is expressed at stage 4 of the analysis phase. However, in order to count the number of interaction calls between different Actor objects in the system that are triggered by this feature, I used the sequence diagram at stage 2.1. With this, I was more positioned to evaluate the number of functional points that would be impacted by this feature, thereby obtaining an estimative implementation time for the module. It is this time that I use for schedule prediction purposes in the framework. To attain a viable schedule negotiation zone on the project, I measure predicted time from the framework against estimated time. Estimated time is expressed as a product of a number of Functional Points and the productivity factor of 7.2. I infer that for successful on-time project completion, the client must manage the project between the range of predicted time and calculated time. The sequence diagram is given in Figure 32.

University of Cape Town

Figure 32: Object interaction calls of user actions

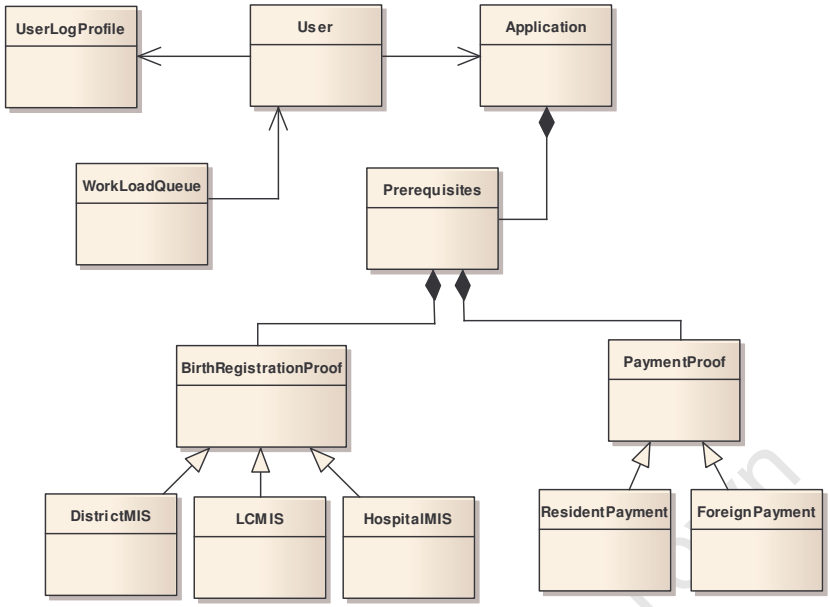


As an example, in the figure above the user carries out the addNewApplication action which triggers the application object to invoke the clearFields method. The user proceeds to the enterApplicationDetails action, at the end of which he clicks the save action. The save application action requires that the Application object invokes the saveApplication method. This in turn invokes the Prerequisite object to instruct the UserLogProfile object to execute the logTransaction method. The concurrent execution of the validateApplicationPrerequisites method at the saving stage mandates the Prerequisites object to invoke the calculateRequiredAmount method in the Payment object. Equally, this also requires that the getProofOfPayment method in Payment object and getProofOfBirth method in the BirthRegProof be executed. This in return results in BirthRegProof object to execute the display method. At the execution of the display method, the updateWorkLoadQueue and displayLoadQueue method from WorkLoadQueue object are executed. The successful execution of these steps means that the required application has the requisite proof documents, all details have been entered and it has been saved onto the system; and the next person in queue is notified of the current status for action.

After obtaining this information, I developed an Actor Object Logical Model at stage 5 as given in Figure 33. This logical model was imported into the framework for dependency assessment purposes.

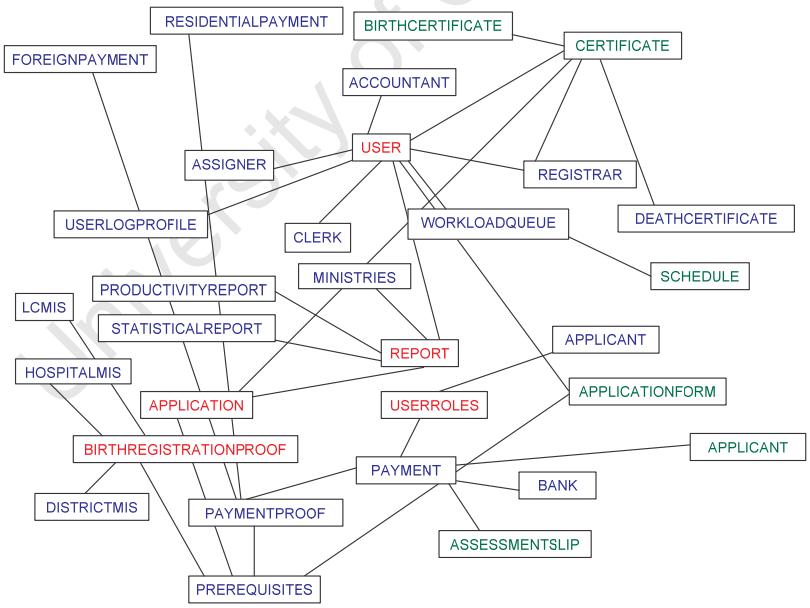
University of Cape

Figure 33: Actor Object Logical Model



Using the weighting relationships between the different actors, the complexity analysis classified each AO in the model as High, Medium or Low as given in Figure 34.

Figure 34: Implicit model



Those marked red are of high risk, green of medium risk and those marked blue of low risk. For example, AO *User* is strategic with 2 high risk dependants, and 5 low risk dependants. *Application* has 2 high risk dependants and 1 low risk one. *Report* has 2 direct high risk ones having various low risk AOs. In this situation, if the system is to be developed in modules,

University of Cape Town

Report is important (because of indirect relationships), and *User* is important because of many strong dependencies as summarised in Table 6.

Module	*Direct*	*Indirect*	*AOs Totals*	*Priority*
User workload management	*9*	*8*	*17*	*4*
Reports	*5*	*15*	*20*	*2*
Application processing	*5*	*20*	*25*	*1*
Certificate issuing	*3*	*16*	*19*	*3*

Table 6: AO dependencies

Validating application prerequisites will impact on all the modules, including workload management which has priority 4, reports which has priority 2, application processing which has priority 1 and certificate issuing having priority 3. However, for module acceptance criteria determination, realizing full functionality of application validation will require implementation of some functionality on workload management since the application has to be processed and passed on to the next stage within the business process. Therefore, a reconcilable situation has to be negotiated whereby both the client and the vendor benefit when this module is completed and delivered.

5.4.5 Case-study implications

These results showed that any potential vendor interested in undertaking this project would be concerned about the architectural complexities associated with achieving envisaged functionality and customer satisfaction. Determining a fair project scope therefore would imply that high risk AO generated in the model would require more attention than others. This is due to the fact that they are more likely to change or even produce changes in other requirements. If the client is not willing to pay for their implementation within the initial release, consensus has to be reached since such functionality will be required to derive acceptability of deliverables. By demonstrating this complexity to the client, at least a win-win situation can be guaranteed. Either, the client can agree to pay for testing costs, or agree to adjust the implementation timeframe for the deliverable or even pay for the extra development required to ensure that overall functionality is attained. From a client perspective, especially in instances where the project is cancelled, money that would have been misspent satisfying low-risk requirements would have been saved.

Therefore, in order to improve the requirement negotiation process effectiveness, it is important to reduce the level of uncertainty. Equally, attaining consensus among the success-critical stakeholders is crucial for the success of any project. If this complexity is not demonstrated to the client at this stage and only a face value of the system feature is

University of Cape Town

assumed, project risks, especially scope creep problems are likely to occur. Therefore, with this approach I was able to assist the client plan for staged deliveries, and make the necessary trade-off decisions during project negotiation with the selected vendor. This analysis showed that this process is effective in situations where a client is struggling to size functionality to suit the available budget and at the same time identify a vendor that will request a realist price for the project.

This analysis further reinforces my belief that the requirement shaping is an ad-hoc and iterative process in which filtering, negotiating and tuning of different matrices about expected functionality must be sufficiently developed and all key project stakeholders must be involved.

5.5 Case four: Project performance evaluation

5.5.1 Case Study objective

This analysis aimed at evaluating how the CASSE Framework can assist with project measurement and evaluation. The software business is continually faced with the challenges of tracking multiple-project progress and establishing a methodology for evaluating, selecting, and prioritizing these projects. Equally, identifying potential projects and determining their value in supporting organizational strategy is still a daunting task.

5.5.2 Case Study description

In this case study, I obtained industrial data for three software projects applications under one portfolio from an industrial partner within the rapid prototyping and simulations business environment. Initial estimate times at the inception of these projects as well as the actual time and money spent on these projects were provided. For the purpose of testing my approach, I examined and compared the different quadrants within which each project fell. Specifically, I was interested in ascertaining whether a given project was falling within risky quadrants or within the viable quadrants. Secondly, I compared the status of these projects at completion stage using actual time and money spent on the project, as illustrated in Figures 37 and 38. This would give me an opportunity to visualize how a given project shifted within the project evaluator quadrants and how this shift impacted on the budget portions.

5.5.3 Data characteristics

The case study company uses the object systems analysis and design methodology. I obtained the actor object graphical models for the three applications under evaluation. This model contained the functional points to be implemented in each actor object as well as its

University of Cape Town

relationship with its neighbours in the graphical model. I was provided with the productivity factor used in the company's estimative effort. I integrated this productivity factor into the model so as to obtain reflective results unique to this company. This productivity factor is not far off from the standard 7.2 value.

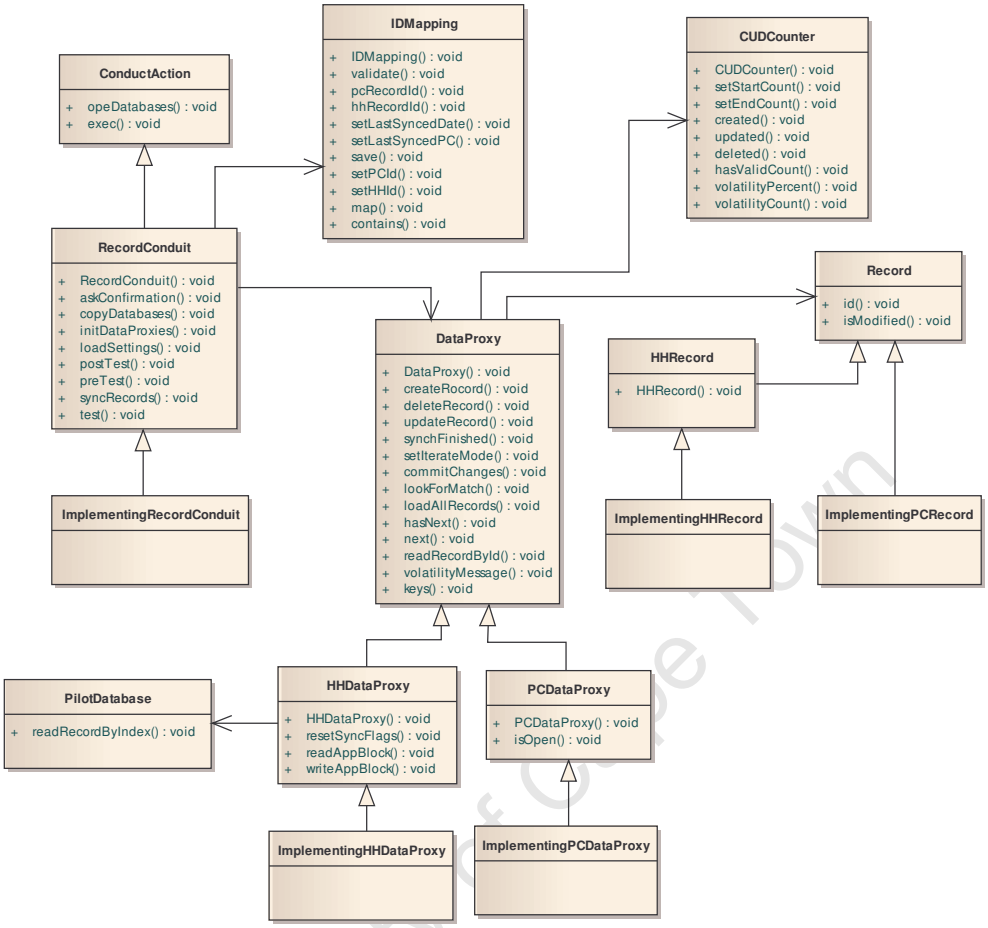
Timesheet data included actual hours, both projected and actual, for the overall project, resources names on these applications as well actual time spent on each functional point. Ideally, I wanted to ascertain how long each functional point took to be implemented; taking into account analysis, development and testing time.

Overall portfolio data included the project names, application names as well the budgets allotted to each application and the overall portfolio. Hourly rates for each resource were provided so as to ascertain the variances in costs incurred per each resource. For confidentiality purposes, these amounts and figures are not included in this analysis. I provide only the project performance analysis across the quadrants using inception and completion data. I also present how the project budget looked at the signoff of this portfolio. All applications in this portfolio were started and executed in the same month.

The Actor Object graphical models used in this analysis are illustrated in Figure 35, 36 and 37.

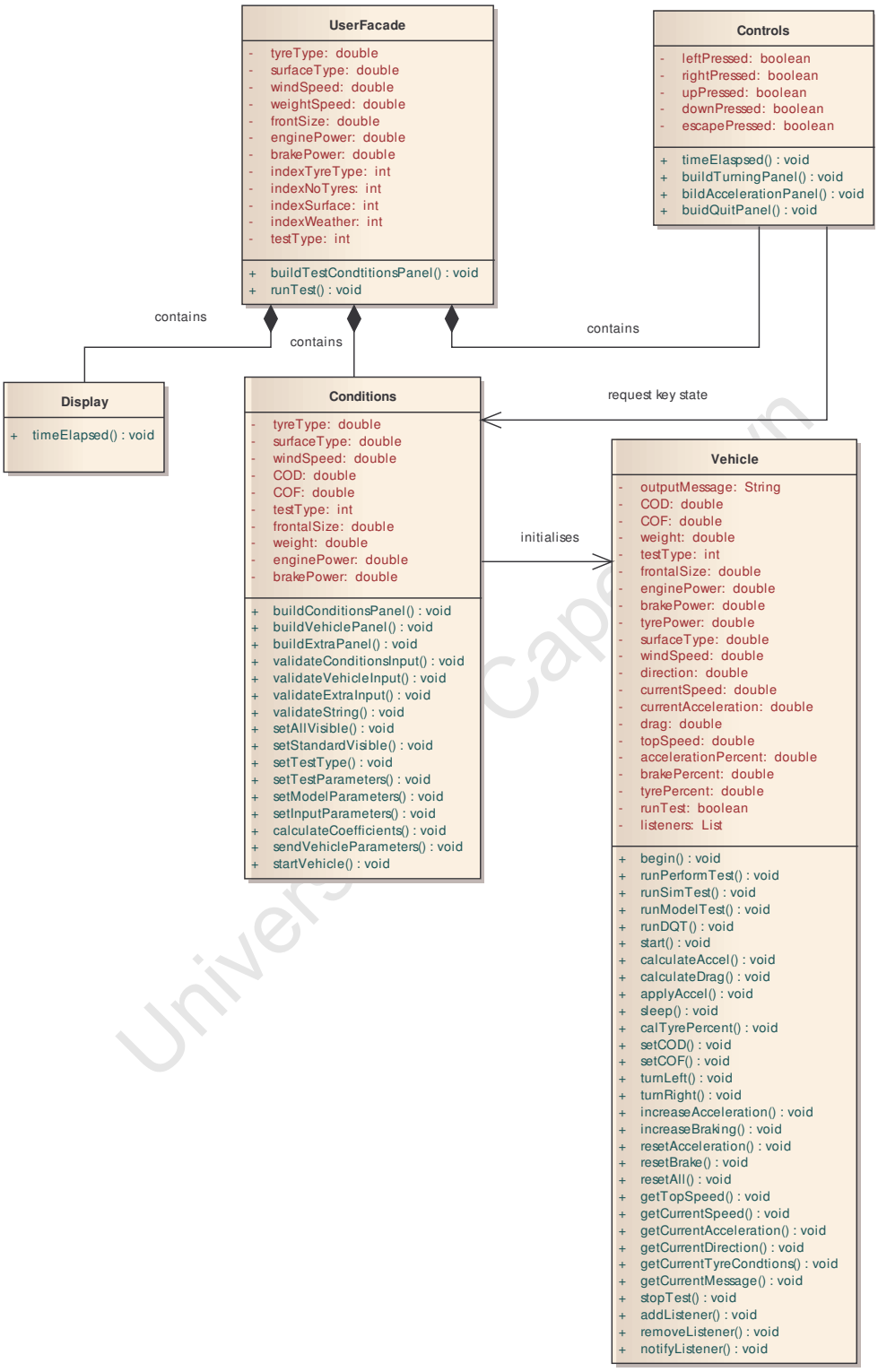
University of Cape Town

Figure 35: Data Proxy Actor Object graphical model



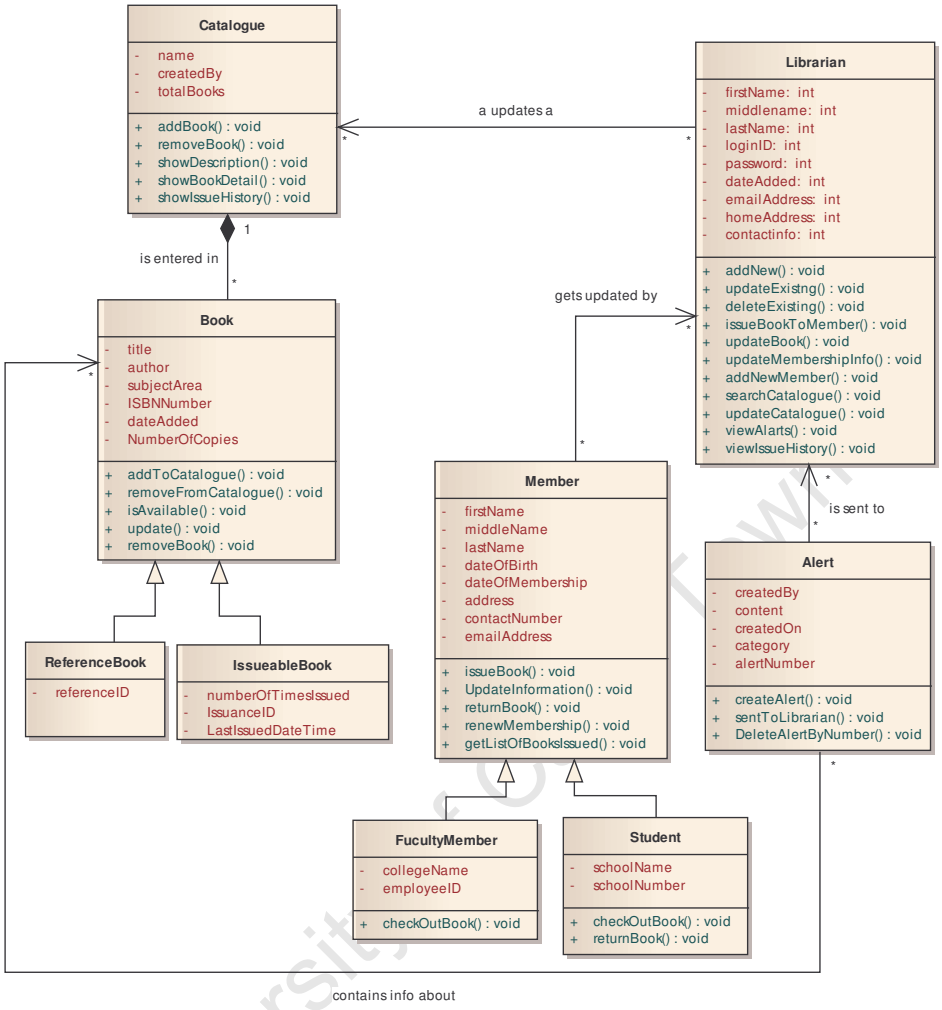
University of Cape Town

Figure 36:A1 Application Actor Object graphical model



University of Cape Town

Figure 37: Library Application Actor Object graphical model

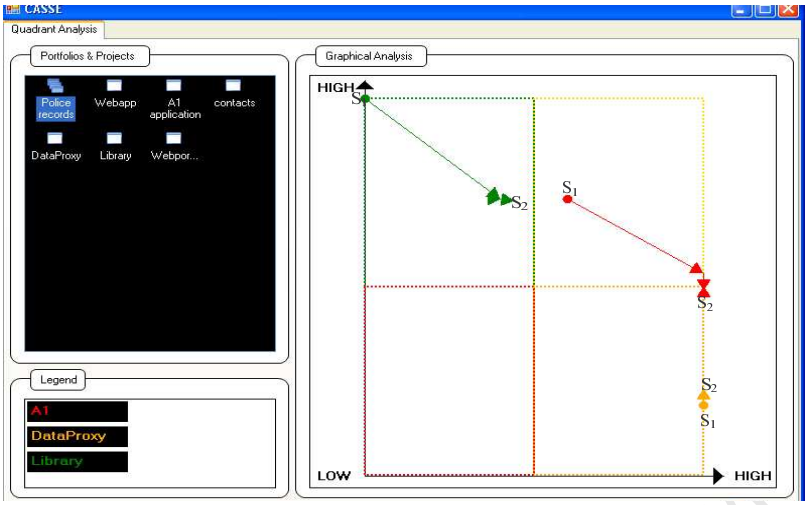


5.5.4 Application performance analysis

For the performance analysis, I considered one portfolio which had three applications, namely: A1 application, DataProxy and Library, as illustrated in Figure 38. Other applications belonged to another portfolio which was outside the scope of this analysis. The figure below illustrates how these applications performed within the police records portfolio.

University of Cape Town

Figure 38: Project evaluator - application shifting patterns



S_1 represents the initial position where a given application was scored at project inception. S_2 represents the end point where the application was positioned at the signoff stage. As illustrated in Figure 38, the Library application, indicated in green under the legend panel, scored highly in quadrant D on the value evaluator at inception. Its initial investment was very low, yet its business value was relatively high. However, as the project progressed to completion, this value seemed to have decreased significantly. Although it still featured in quadrant D, the process investment and competency to deliver increased in terms of process investment, thus decreasing the business value anticipated. However, it still maintained significant profitability levels given its score.

For the A1 application, represented in red on the legend, it was scored and ranked in C quadrant of the project value evaluator at project inception. However, at the release stage of this application, it had shifted to block B on the value evaluator. Essentially, its value dropped due to the increase in resource usage with regard to the process investment and competency to deliver this application at signoff.

As for the DataProxy application, it started off in block B on the project evaluator. At the completion stage of this application, however, it had shifted slightly upwards. This implies that its business value increased slightly in the market-place, while the development costs were maintained within the anticipated boundaries or perhaps even some development time was saved during the development process.

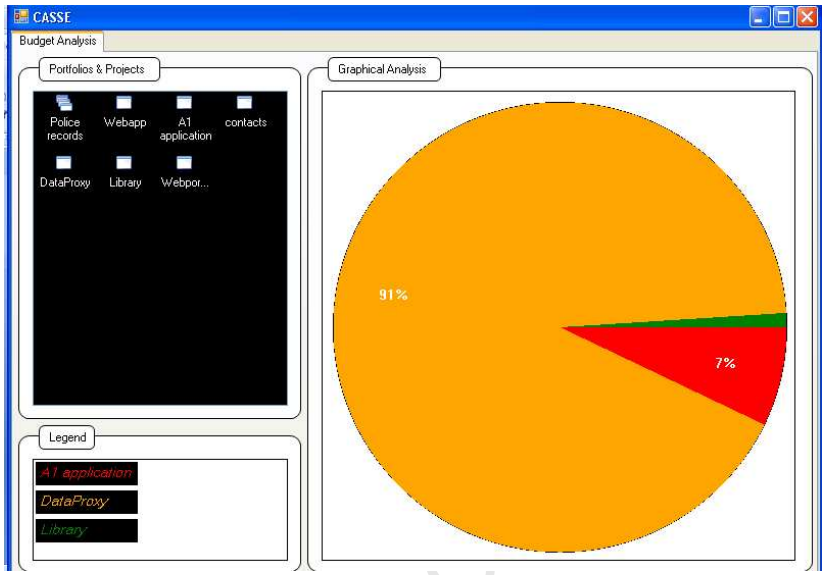
5.5.5 Budget analysis

The budget analysis was made only for all applications in the police records portfolio at the completion stage, and the following results were obtained, as illustrated in Figure 39. This

University of Cape Town

analysis revealed that the DataProxy application consumed 91% of the overall Police Records portfolio budget at the completion stage. This showed it to be the most expensive application in the portfolio. 7% was spent on the A1 Application. However, the Library application shared only 2% of the portfolio's budget.

Figure 39: Budget evaluation



According to project records, the Library application was a customisation job which implied that existing components existed on the company database, thus requiring less process investment. The shift within the quadrant seems to have been triggered by the new features that were introduced during the customisation process. However, these added changes nearly pushed the project to quadrant C. The DataProxy and A1 Applications were both new applications within the portfolio.

5.5.6. Case study implications

These results show that this post-project review assessment provides an opportunity for capturing explicit knowledge which can be leveraged in future projects, in bid to reduce overall cost. It can also help with transferring tacit knowledge which is even more extensive and potentially valuable. For instance, understanding shifting patterns of each application in the various quadrants would be pivotal in enhancing the development process improvement, as well as the estimative effort on projects.

These observations further demonstrated that projects are undertaken to produce value for the organization. Thus, the proper ranking metric utilized is fundamentally important in communicating this value to the stakeholder as project unit costs change during the development life-cycle. With this kind of feedback, especially if handled in real time as

University of C

development time is captured and other projects events unfold, such as scope changes, the portfolio office is empowered to decide which projects to initiate, which on-going projects to continue to fund, and which projects to terminate. This will ensure that organizational resources are well-utilized and that the project "pipe-line" which aligns with the organizational strategy is well-maintained.

Therefore, with this approach, the following can be achieved:

- improved resource utilization and planning as well as appropriate decision- making about adding new projects and continuing current ones;
- learning to establish a methodology to evaluate, select, and prioritize projects;
- developing skills to monitor resource utilization, cost, and project progress across the portfolio, and
- selecting the right projects and ensuring their success by establishing a portfolio that is aligned with corporate strategies and return-on-investment goals.

Fundamentally, the objective of the PPM process is to determine the optimal mix and sequencing of proposed projects to best achieve the organization's overall goals. Typically, this is expressed in terms of hard economic measures, business strategy goals, or technical strategy goals – while honoring constraints imposed by management or external real-world factors. However, without value-based metrics or feedback channels that link these processes, people and tools within the portfolio, this process becomes extremely difficult. Indeed, a project might be successful from a schedule, budget, or scope perspective, but if it fails to meet business objectives, it fails overall. Accordingly, enforcing real-time risk reduction on projects is a principal task for portfolio managers mandated with overseeing groups of projects. With this, a continuous feedback looping process will be created by which management absorbs and prioritizes plans and allocates financial and human resources to the investment initiatives, manages the governance-orientated collaboration with the business stakeholders, delivers expected results from the investment, and provides reporting to stakeholders for decision-making and the communication of investment status.

University of Cape Town

CHAPTER SIX

Task Performance Analysis and Measurement on Project Schedules

6.1. Introduction

The endeavour to determine, operationalize, and quantify the elements contributing to project complexity is motivated by the fact that project management processes and techniques are influenced by the level of "complexity" in a project. Project management activities such as planning, coordination, control, goals determination, organizational form, project resources evaluation, personnel management, and project cost and time are all affected by the level of complexity involved in a project. This endeavour therefore sets a need for adaptive models having the capability of managing project complexity as well as overall enhancement of valuable project monitoring, control and evaluation.

This chapter aims at demonstrating the applicability of the CASSE framework in managing uncertainty and the anticipated profitability on project schedules during the project lifetime. This analysis also illustrates that this approach is applicable within the general project management discipline. Specifically, profitability level prediction and management as well as tracking tasks progress during the project lifetime.

6.1.1 Analysis objective

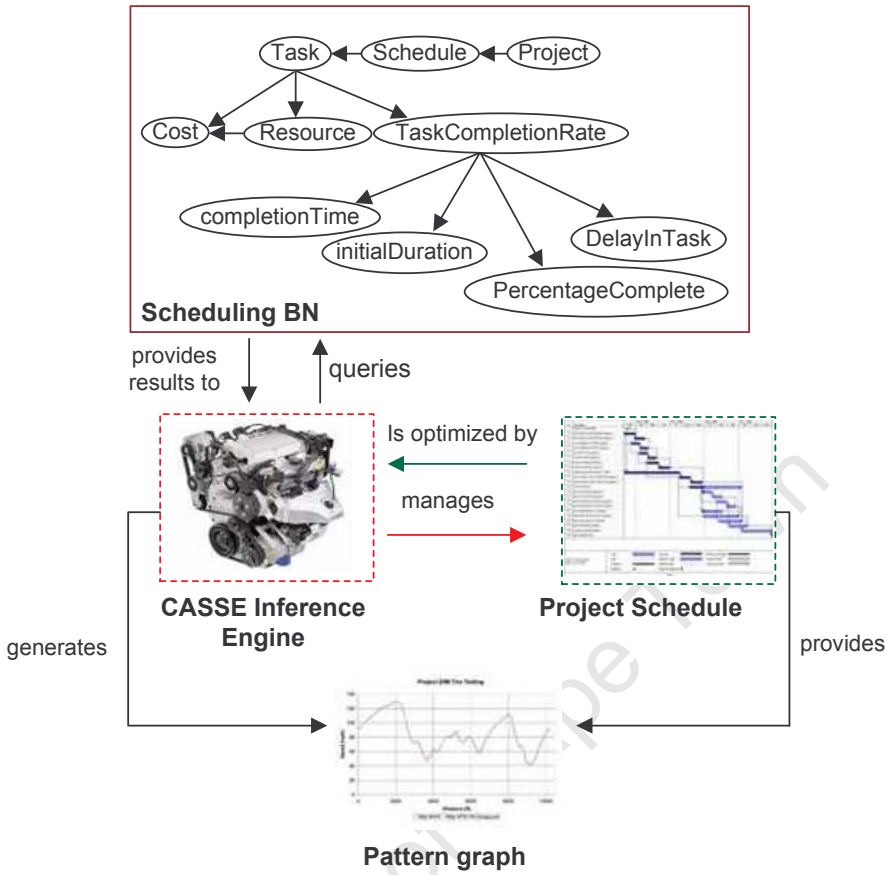
This analysis was undertaken to test and also demonstrate how task performance measurement on software project schedules can be achieved. This is achieved through viewing and understanding a project as complex adaptive system in which path dependence on tasks and sensitivity to initial conditions is applied and also where change and learning from experience is integrated.

6.1.2 How it all fits together

Figure 40 illustrates the graphical interrelationships of this mechanism and how results can be generated for effective decision-making.

University of Cape Town

Figure 40: Pattern generation on project schedules



A Bayesian Network (BN) model is defined for an observed project. Depending on the analysis results required, conditions are defined on the BN model nodes. These can be for example on schedule, project, task, cost for the task, resources to execute this task, completion rates of the task that includes completion time, initial duration projects, percentage completion on this task delay in task, among others. The CASSE framework learns this model structure and outputs results that are used by the inference engine. This inference mechanism can be configured to give real-time task pattern results graphically. Equally, the changing conditions and patterns in the schedule tasks are displayed on the project schedules. These can include optimal paths that can be taken in instances where divergent patterns are detected from the initially anticipated critical path.

Upon such analysis therefore, predictions can be made for optimal project time. More still anticipated profitability levels can be measured against project tasks at the initial stages of the project or constantly tracked as the project progresses.

University of Pretoria

6.1.3 Basis for empirical analysis

In this chapter, an illustrative example is given on how task interdependences can be used to achieve this objective. This is complemented by results from an empirical study conducted on two commercial software projects in industry. This empirical study particularly looked at temporal task acceptance patterns, task completion rates and project profitability levels at different time steps during the project lifetime. This was compelled by the need to ascertain business performance of these projects over time during the development lifecycle.

6.2. General Project Management

In general, project management entails understanding and evaluating the effect of task characteristics and coordination strategies on coding-task outcomes. These project characteristics must therefore be well understood and project management policies, techniques, and procedures must directly be mapped upon project complexity levels within the project. Measuring the significant effect of task interdependence on project success and profitability tracking would thus revolutionize the organisation's ability in managing project dynamics especially productivity. Contingently

Central in ensuring project success is the capturing of structural, dynamic and interactive aspects of elements that make a project complex and hard to manage. These include project task time duration (anticipated and actual) measurements as well as completion and successful acceptance rates of these tasks, among others.

Although project complexity has been recognized recently as a major issue in project management, there is still a great deal of confusion on how it should be defined and articulated. The importance of determining project complexity factors and their levels in a project is crucial as these factors have a significant impact on project planning, execution, control, and management, and ultimately project success or even failure. Although the discipline of project management lies in trying to reduce project variations as far as possible, however, the current trends indicate the fail to account for variations in context between projects, ranging from managerial complexities to technical complexities is still prevalent.

Project managers therefore need to understand the complexity that exit between project tasks so that adequate project schedules are given. This would greatly enhance viable project selection, measurement and management. Imperatively, as project needs evolve, the need to propagate and reflect this change to the overall anticipated project status for managerial

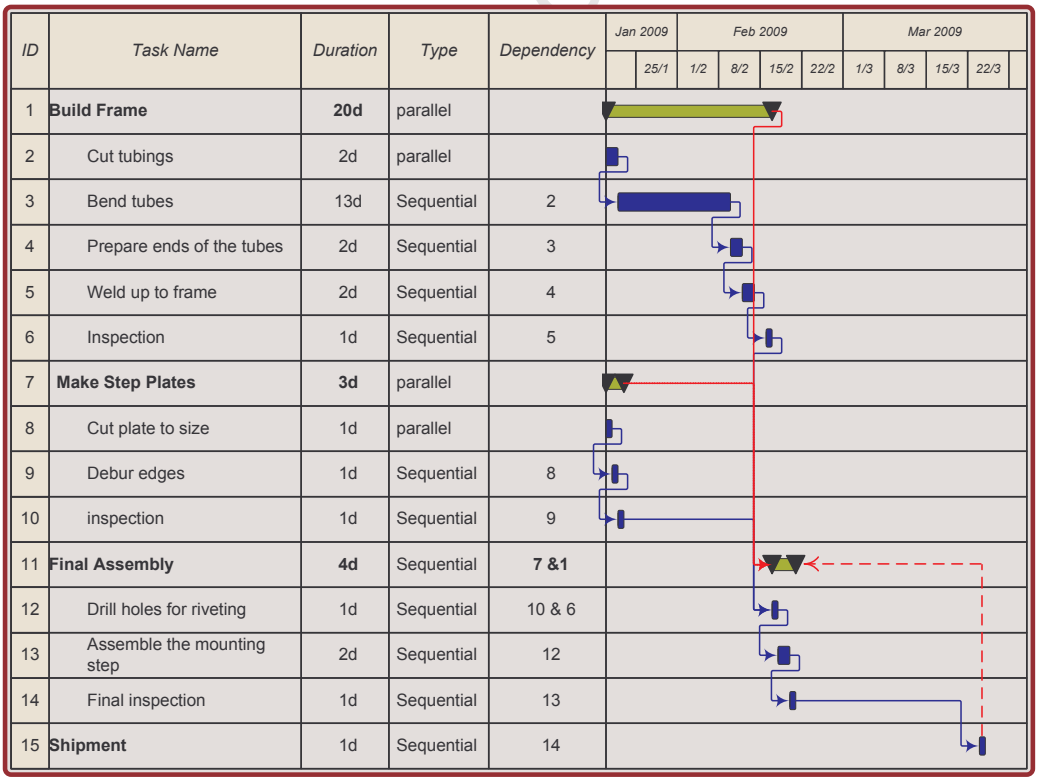
feedback and control purposes must be matched. This adaptiveness would thus enhance dynamic complexity management and productivity optimization. In a bid to reduce uncertainty in project management, this illustration aims at the technical aspects of project task modeling so as to portray an overall predicted and managed informative status of the project in terms of expected project duration, cost and resultantly profitability assessment. This assumes that that an agreeable set of requirements for implementation are given. Although much of the stimulus for change comes from customers' requests, generated by their own increasing knowledge, this change can still be integrated within the model for adaptiveness.

6.3 Illustrative example

Each project has its unique characteristics. In this regard, consideration is put on tasks and their properties as well as how they relate to each other in terms of dependency. In order to successfully schedule and manage any project, project tasks have to be outlined, their properties defined as well as their dependence mapped as illustrated in the Gantt chat diagram example below.

Cape Town

Figure 41 Software Development Project example



The digraph above shows a typical Gantt chat of any project schedule with tasks defined and their relationships mapped and duration periods elaborated. This particular example is for

project tasks on a product engineering project. However, this kind of mapping does not give us the overall profitability levels of the project in a nutshell; given the task relationships and the duration estimates. Secondly, it is hard to ascertain if there is any implicit relationship between tasks, and if so, how this impacts on the overall project schedule and time estimates as well as overall project profitability levels.

What would be optimal is to go one step deeper to analyze the implicit interdependence among these tasks so as to evaluate the quadrant in which the examined project falls on the value conundrum. It would secondly be important to know what profitability levels need to be managed as tasks get completed along the project execution path. Thus, building a process history from which future estimation can be based. This would be the ideal project status that most managers would want to see on any project. However, in order to get to that level, various steps have to be followed. The assumption at this stage is that projects that have reached the scheduling status have gone through the vetting or approval process, irrespective of budget implications or strategic business objectives. Therefore, mapping them to the value conundrum may be irrelevant. Rather, what is required here is to ensure that the project is kept within budget and is delivered on time as motivated in the business case.

6.3.1 Step 1: Task relationship definition

As a starting step, task relationships have to be defined. Each task is modelled as an Actor Object with set properties and relationships. A task can be linked with a sequential or parallel dependence relationship. There could be a situation where tasks are grouped based on certain characteristics or based on aggregated result or outputs anticipated. On the other hand, there could be tasks that may neither be sequential nor parallel but are linked somewhat. These are some of the dependence mappings often used on project Gantt chat schedules to depict project sequencing. However, adding a layer of complexity reveals that this basis can add value on the way project complexities are revealed and understood as given in the table below.

University of Cape Town

Explicit Dependence relationship	Relationship Description	CASSE UML Interpretation relationship
Sequential	Is part of the predecessor task. However, it cannot start before the predecessor task is complete	Composition
Parallel	Is part of predecessor task but can start even if the predecessor task is not complete	Aggregation
Neither sequential or parallel but linked (Simultaneous)	Uses predecessor task's inputs for its full completion	Association
Grouped tasks	Inherits predecessor task's properties for its completion	Generalization

Table 7: Task dependency relationships

This kind of analysis results in four potential task relationships, that is, **composition, aggregation, generalization and association**; each of which represents a given pattern of dependence.

- Some task(s) can be part of predecessor task(s) and cannot start before predecessor tasks are complete – resulting in a composition relationship.
- Some task(s) could be part of predecessor task(s) but can start even if the predecessor task is not complete – resulting in an aggregation relationship.
- Some task(s) require predecessor task inputs in order for them to fully be complete – resulting in an association relationship.
- Some tasks inherit predecessor task properties in cases where tasks are grouped under one theme or category – resulting in a generalization relationship.

With these defined relationships, a graphical model can be generated. Building on the previous Gantt chat dependences defined above, the explicit graphical model would result in this kind of relationship as given in figure 42 below.

University of Cape Town

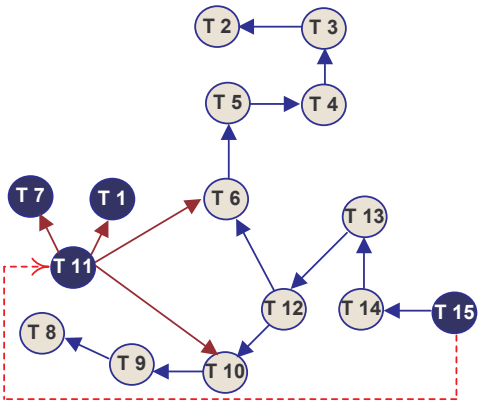


Figure 42: Explicit dependency model

In the diagram above, T3 (where T refers to task) strongly dependents on T2, T4 strongly depends on T3, T5 strongly depends on T4 and T6 strongly depends on T5 respectively. T12 has two dependants, that is, T6 and T10, but T13 strongly depends on T12, T14 strongly depends on T13 and T15 strongly dependent on T14. However, T15 also has sort of relationship with T11 as a stage. T10 strongly depends on T9 while T9 strongly depends on T8. This web of relationship demonstrates the explicit complexity that can externally be analyzed to assess project risks.

Such explicit risk indication however, does not emphasize the critical tasks which impacts on on-time project completion. More still, it also does not show the implicit strengths that exist in the sub nodes. For example, it could possibly be extremely difficult to quantifiably measure the strength between T2 and T10 or T13 and T8.

6.3.2 Step 2: UML Modelling

Step 2 requires modelling the project task in object oriented fashion using UML. At this level each task is modelled as an Actor Object in the graphical model with set properties that get populates with values. For example, a task is implemented by a resource or a number of resources in a given time. This resource is remunerated on an hourly basis. For each task, there is an implicit cost value equated to the unit cost for each resource per hour as well as a sell value as a mark-up on each unit cost of each resource. A task has a *nature* type, that is, it could be an existing task that runs on every project (therefore requires less effort and investigation time implying that in the past, well designed experience has been built around such a task. For example, in terms of technology or development platform used) or a new task which has never been implemented before (therefore requires ample time for investigation and execute). This thus results in the following properties; time, resource, implicit cost, sell-value and nature as illustrated in figure 43 below.

University of Cape Town

Figure 43 Task attributes

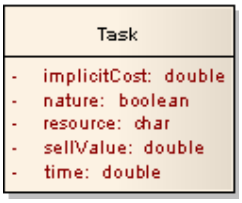
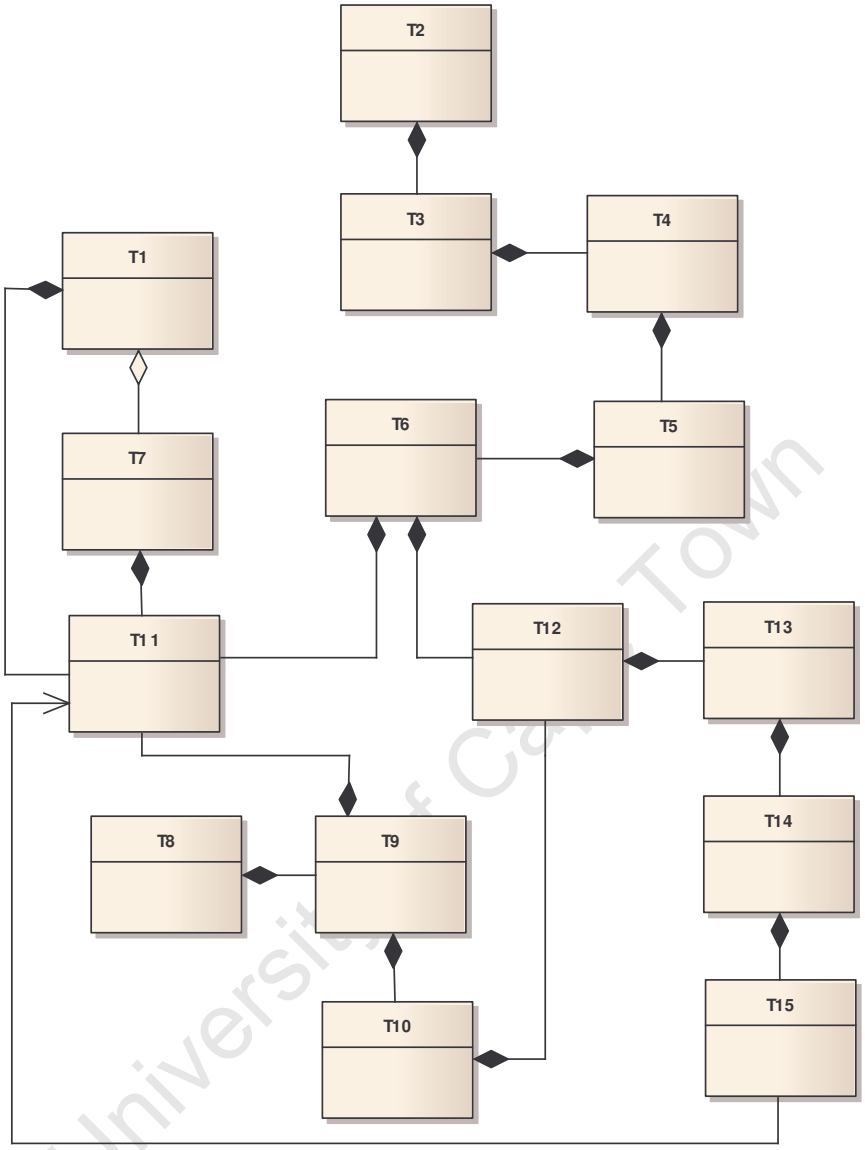


Figure 18 above illustrates an example of an AO with set properties upon which task characteristics and strengths in the model can be analysed.

The Gantt chat is then transformed into a UML graphical model as given in figure 44 below.

University of Cape Town

Figure 44: Task explicit model



University of Cape Town

Using this model structure, the framework then runs a sensitivity analysis on the model. This is done to:

- predict the optimal project completion time ranges,
- map the project on the value conundrum,
- relate tasks to cost dependency, and
- highlight the implicit node strengths in the model.

Nodes are categorized into low risk tasks, medium risk tasks and high risk tasks. This complexity modelling thus unveils how implicit task analysis can be formulated and then later executed in the framework.

6.3.3 Set 3: Analyze the model in the Framework

This step presents different dimensions of analyzing project status results. Continuing with the previous examples, assuming that:

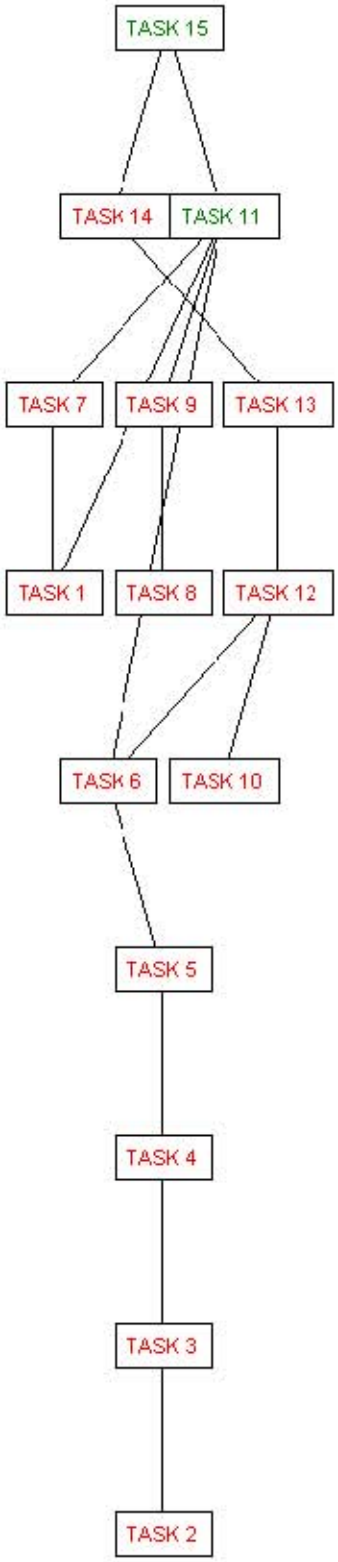
- this project has a running budget of R400,000,
- it has 1 product engineer, a welder, assembler and an inspector, and
- the product has to be shipped by the end of week 8,
- the hourly rate range of the resources on the project is between R160 to R250 as minimum unit investment and that the unit maximum return on investment per hour is R600.

Combining all the time given for all tasks on the project, results in 28 working days. This translates into 224 man-hours excluding overtime for 4 resources.

Assessing the task strength on the project reveals that T15 and T11 are of medium risk to the project while the rest are of high risk as illustrated in figure 30 below. T15 as a medium risk task has a strong relationship with T14 of high risk.

University of Cape Town

Figure 45: Task strengths in the model



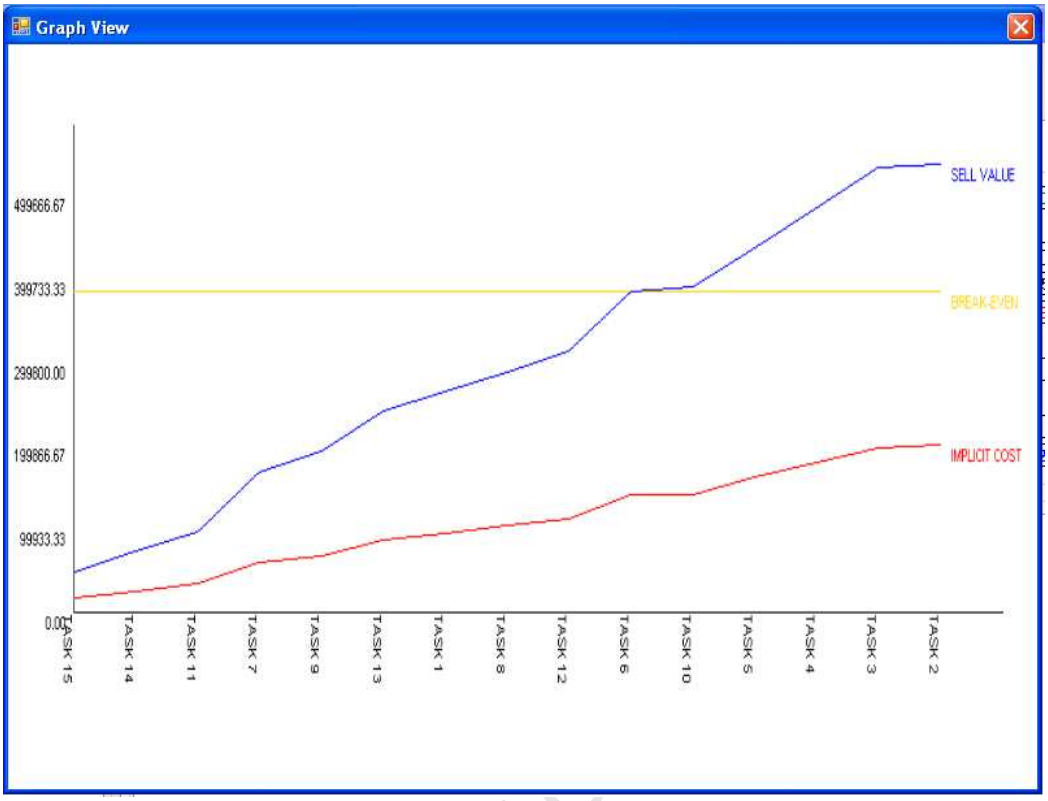
Taking Task 5 (welding up to frame) as one of the high risk tasks on the project, the model shows that it will strongly influence T6 (Inspection), which in turn influences T11 (final assembly) and T12 (drill holes for riveting) respectively. This implies that if this task is not completed on time and according to acceptable levels, other tasks dependent on it will be impacted significantly. Therefore what a project manager needs to ensure is that he sets an early delivery date for this task so that its deliverables are checked and validated within the scheduled time. Otherwise, any extra time incurred in fine tuning this task will increase project implicit costs and resultantly, lead to profitability decrement on this task. This kind of analysis is comparable to the critical path analysis.

Therefore the critical path for this project would be having T2 implement first, followed by T3, T4, T5 and T6 respectively. However, T6 and T10 will have to be implemented in parallel. This would be followed by T1, T8 and T12 also implemented in parallel, T7, T9 and T13 would have to be implemented in parallel thereafter. Lastly, it would be T14, T11 and T15 to be implemented last with T14 and T11 being implemented in parallel.

Assessing the tasks against the project value (as illustrated in figure 46 below) reveals that the implicit cost of all the tasks, if completed within budget and time is below the project breakeven point.

University of Cape Town

Figure 46: Task project budget assessment



From the graph above, the X-Axis represents the tasks on the project while the Y-Axis represent the project budget levels given task dependencies. The margin between sell value and the implicit cost curve shows that implicit cost will rise as extra time is incurred on completion of tasks. This difference also represents marginal revenue of all tasks on the project. More still, if the implicit cost curve equates with the sell value curve during the project life time, this project will have reached a breakeven point, hence yielding no profitability. Should the implicit cost curve rise above the sell value curve, this project will incur losses.

6.4 Analysis implications

With this kind of analysis therefore, as a key project stakeholder, especially at project inception, you have the opportunity to see the big picture of your profitability levels on the project conundrum. As a project manager you have the ability to re-prioritize your schedule and optimize your resources accordingly. However, as the project goes into the execution stage, there is a need to track dynamically the productivity indicators in terms of how long tasks are taking to complete and whether they are being completed within the required time - resulting in the task complementation rate indicator measurement. Important to note also

is whether the completed task(s) meets the stakeholder requirements and expectations or more time has to be dedicated on the task for modifications – resulting in measurement of the acceptance rate of the task. If this is aligned back to the project budget, a clearer project status will be maintained.

This kind of tracking thus ensures that the marginal revenue anticipated on the project is sustained throughout the development lifecycle. With this kind of pattern, you can easily track your productivity levels on the project and at the same time continuously map the project on the value conundrum to assess its viability as it progresses. This continuous assessment increases risk detection on projects and also enhances contingency planning so that graceful project completion is attained always or where possible early warning bells are sound before the project goes into trouble. This would in turn minimize levels of project failure due to budget overruns and inadequate project time management.

It would also be interesting to compare the status of both the implicit cost and sell value curves at the end of the project with the initial graph. This would give an indication of what went wrong (in the event that the implicit cost curve rose towards the sell value curve) or what was done best (in the event that the implicit cost curve moved away from the sell value curve – implying less cost and more revenue) on the project. This would mean that certain tasks were completed on time and therefore rewarding the team as a motivational strategy would not be a bad idea.

In testing this approach, an industrial case study was undertaken to evaluate temporal task completion rate assessments and acceptability level measurements on two commercial projects. This case study is detailed in the next section.

6.5. Case study: Tracking project profitability levels

6.5.1: Case Study Description

As a mechanism to illustrate the temporal evaluation of task completion rates on projects, given the actual and anticipated profitability levels on projects, a case study was carried out with one of the software development houses in South Africa. In this analysis, two commercial projects were selected. Project 1 started in November 2007 and was scheduled to end in March 2008, while Project 2 was scheduled to start in February 2008 and end in April 2008. Specific interest was in wanting to understand project progress complexities and how they impacted on the overall company revenue. In this empirical study, profitability analysis implied evaluation of the time taken to complete a given task. This implementation time determines the overall cost for that project since resource inputs are valued and paid

University of Cape Town

for on an hourly basis. For example, if a task was scheduled to take 40 hours, and it overlaps to 60 hours, this implies that the company is spending more time to complete this task than anticipated, thus eroding project profits or even setting the project into loss levels. If this task implementation time has overran and worse still it has not reached acceptance standards for the client, more time would be added to satisfy client acceptance standards. More time added would imply more cost to the company and less on profit. If this pattern applies to various tasks on the project and there are mechanisms of detecting such performances behavior, one can already detect such complexities on even ongoing projects.

Given this background therefore, I examined one project which was complete and another which was ongoing. For the completed project, my interest was in understanding how tasks were quickly completed and how best they met client requirements to be acceptable for signoff. For the ongoing project, I wanted to analyze the possible risks that were slowing the project either in terms of task completion or acceptance bottlenecks. This was for the purpose of evaluating whether there was a need to realign the development process or optimize on resource allocation. The temporal model of the CASSE Framework would help capture these evolving properties of the project.

Both projects selected were utilizing requirements of the same product-line, that is, utilizing core components of the development house. They only differed in visualization requirements and data-handling types. Both projects had inelastic budgets. Project schedule requirements differed depending on a number of externalities affecting the projects. Both projects were developed in agility using eXtrem Programming and had to be executed in parallel from the point of project period intersection. All projects went through the requirements analysis phase where key project requirements and milestones were agreed upon. There was a 60% allocation rate of the resources assigned to execute tasks on both projects.

I therefore, looked at a 3-dimensional variable analysis of the results. For both projects, the key variables of interest examined entailed: requirement or task acceptance patterns, task completion patterns throughout the project duration and profitability analysis.

6.5.2. Task acceptance patterns

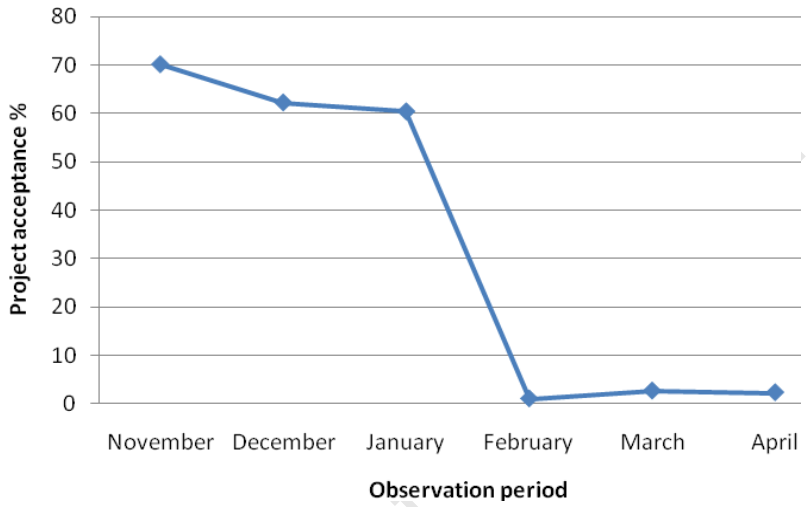
6.5.2.1 Project 1 acceptance analysis

Using the framework, an analysis was made on how each project milestone was accepted by the client throughout the project duration. In order to increase client participation and commitment to the project, an incremental functional delivery and signoff development

University of Cape Town

approach was utilized, thus presenting stage payments strategies as linkages with project milestones. In measuring the task acceptance pattern, examination was made on how each project performed on a monthly basis. Figure 47 below shows the results obtained on Project 1 in this dimension of acceptance criteria evaluation and validation pattern analysis. The X-Axis measures observation period while the Y-Axis measures the project acceptance percentages.

Figure 47: Project acceptance pattern



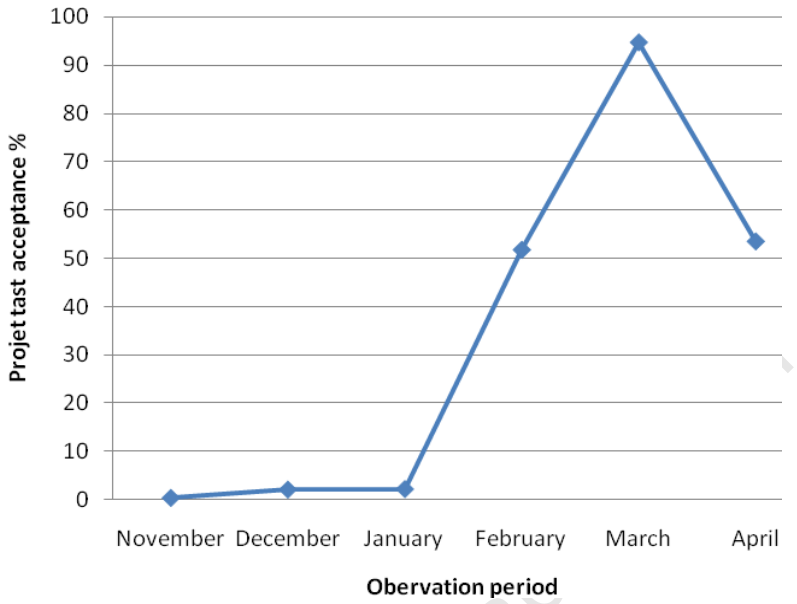
The graph above shows that for Project 1 in November, there was a 70.72% likelihood that task acceptance would not suffice in the first frame of the project. The 29.28% difference implied that earlier milestones in the project were accepted, such as project inception deliverables which were derived in the earlier part of the month. Given that requirement analysis, documentation and validation had to precede implementation, functional acceptance would therefore have a higher completion failure in that time frame. As the project progressed in December, the validation failure decreased by 8.59% of the overall tasks in that time frame. This implies that as development started, project delivery maximization was being delayed at a rate of 8.59%. An optimal validation rate would be targeted to being higher than 37.87% since functionality required was mostly from the existing core components. In January, the delivery rate was not very significant either; it only increased by 1.67% from the previous time frame. However, as the February time frame set in, the validation rate increased to 98.99%. This shows that a great deal of effort was expended to ensure that the March deadline would be reached while all tasks were completed and signed off. The remaining 1.01% would only be for bug fixes while more attention was directed to Project 2.

University of

6.5.2.2 Project 2 acceptance analysis

Figure 48 below shows the acceptance pattern evaluation of Project 2 in the study.

Figure 48: Project 2 acceptance



At the start of Project 2 in February, the acceptance targets were improved to 52%. Perhaps lessons learned on Project 1 led to this improvement, or completed components development in Project 1 which were required by Project 2 led to this accelerated increase in performance. The March time frame reveals that the acceptance rate was improved by 42.94%, leading to a 94.69% likelihood of overall project acceptance in that time frame. However, in the April time frame, this performance seems to have dropped by 41.21%, leading to a 53.48% likelihood of project acceptance in this time frame.

These findings show that there is a strong need to manage the acceptance rate over and above 60% in each project time frame especially if a given project is to be signed-off gracefully at completion, if all requirements will be accepted at signoff and if resources need to take up other roles on other projects that may begin before the present project is completed. Maintaining the project acceptance rates at the rate of 40% in a given time frame would increase the likelihood of a project being delayed due to unaccepted tasks. This result therefore showed that project managers need to evaluate project time frames with keen interest so as to mitigate project delays that may arise due to relative acceptance rate targets.

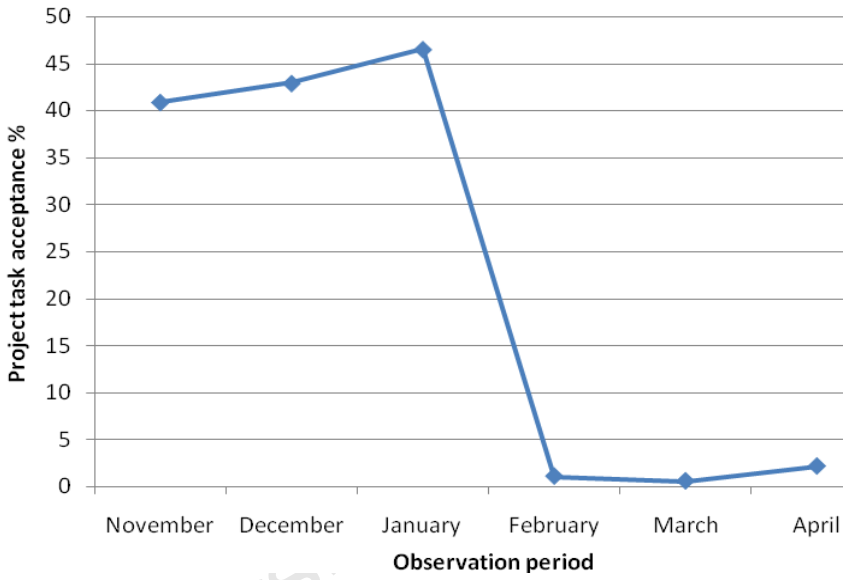
University of Cape

6.5.3 Task completion patterns

6.5.3.1 Project 1 completion analysis

In this measurement, an analysis of the task completion rate of each project in a given time frame was made. As shown in Figure 49 below, for Project 1 there was 40.88% likelihood that tasks would be completed over and above 70% at project inception in November. In December the completion rate likelihood seems to have decreased by 2.02%, with tasks only being completed between 31 – 45% on average.

Figure 49: Completion rate analysis for project 1



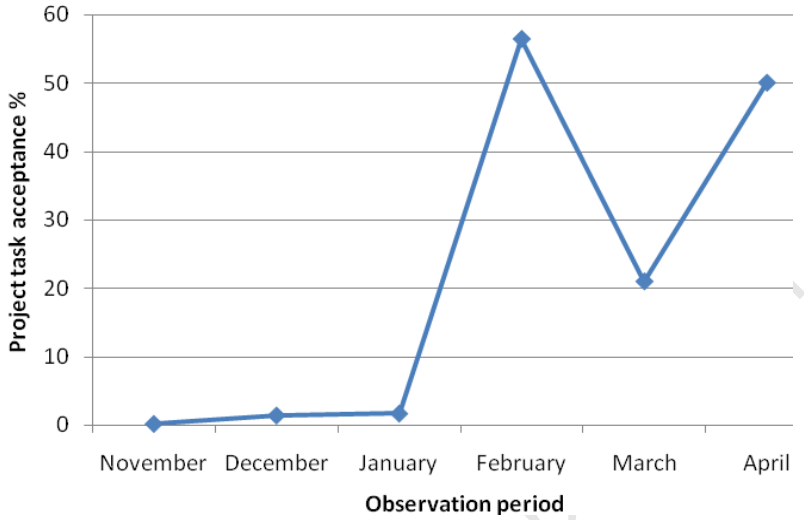
In January the completion rate dropped by 3.61%, leading to a 46.51% likelihood of tasks being completed with a progress level of 31% on average, perhaps due to the festive and holiday season around the previous time frame. As the project tended towards completion in February, the task completion rate improved significantly at a rate of 1.1% per task, leading to 98.9% increment in task completion with a task progress average of 69%. In March the completion rate dropped to 38%. This was the time for bug fixes and resources were executing other tasks on the next project in parallel.

The findings indicate that as the project tends towards completion, the task completion rate reduces considerably. This could be due to other project externalities on the project, such as delayed customer feedback on the deliverables. Although the acceptance rates rise in the last time frames, other project externalities affect the completion rate on final tasks preceding project signoff, hence impacting on the average profitability margins on the project.

6.5.3.2 Project 2 completion analysis

Figure 50 below shows the results of the completion rate analysis done on Project 2 in this study.

Figure 50: Completion analysis for project 2



Although Project 2 started in March, the framework revealed that should the project have started in November, there would have been an average of 1.12% likelihood of task completion through till January. This can be attributed to the fact that since the same resources were working on both projects, they would not achieve higher completion margins on competing projects; hence completing one at a higher acceptance rate before embarking on the next one. At project inception in February, the project completion rate increased by 55.27% leading to a 69% completion progress rate on average. This can be explained perhaps with the fact that both projects required similar components. Given that Project 1's acceptance rate had already reached 97.37% around this time frame and that the functional baseline had already been realized, the acceleration of project completion for Project 2 tasks was invariably enhanced. In March, however, the completion rate dropped by almost 78.98% leading to an average task completion rate of 38%. This is perhaps due to the fact that resources were divided between both projects; as they fixed bugs on the other, they would also try to work on tasks on Project 2. The pattern seems to have changed in April after Project 1 was completed. The completion rate rose to 50%; implying that resources were now redirected to Project 2.

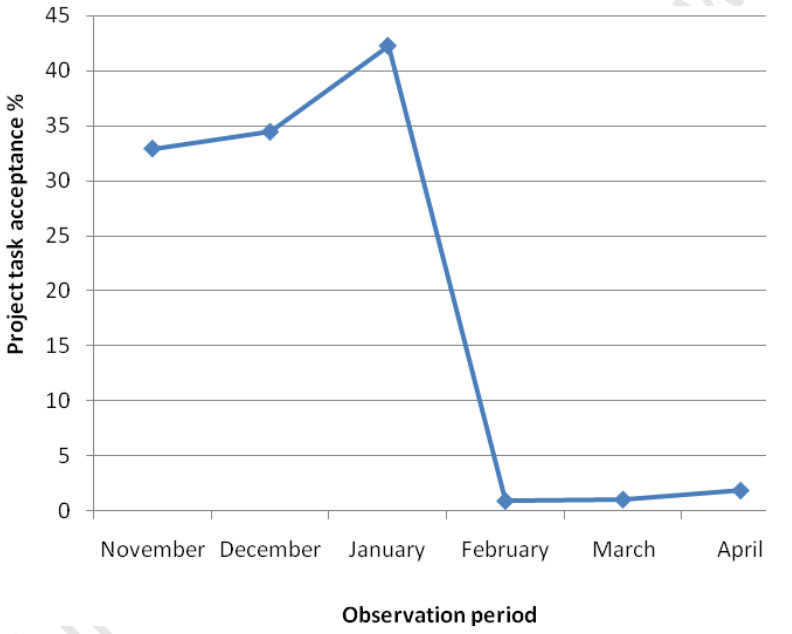
University of Cape

6.5.4 Profitability analysis patterns

6.5.4.1 Project 1 analysis

Examination was made on how project profitability fluctuated over time in the various time frames. Figure 51below shows the results obtained on Project 1 under this evaluation. In the November time frame, there was a 67.04% chance that profits would be realized in that time frame given the number of tasks and resources available. The 32.96% difference implies that profits on the tasks would be realized at such a threshold. In December, however, the profitability likelihood increased by 1.52%, leading to a 34.48% chance that maximum profit of R1600 would be realized on each task.

Figure 51: Profitability on project 1



In January, however, the likelihood of making some profit on project tasks increased by 7.83% from the previous time frame. This led to an overall 42.31% profit realization likelihood. From February until the end of the project in March, the likelihood that profit margins would increase dropped significantly. In February for instance, there was a 99.1% chance that profits would be realized on tasks executed in that time frame but zero profits were attained. In March, the profitability likelihood dropped to 98.97%, implying that there was only a 1.03% chance of making R1 600 on each task in this time frame.

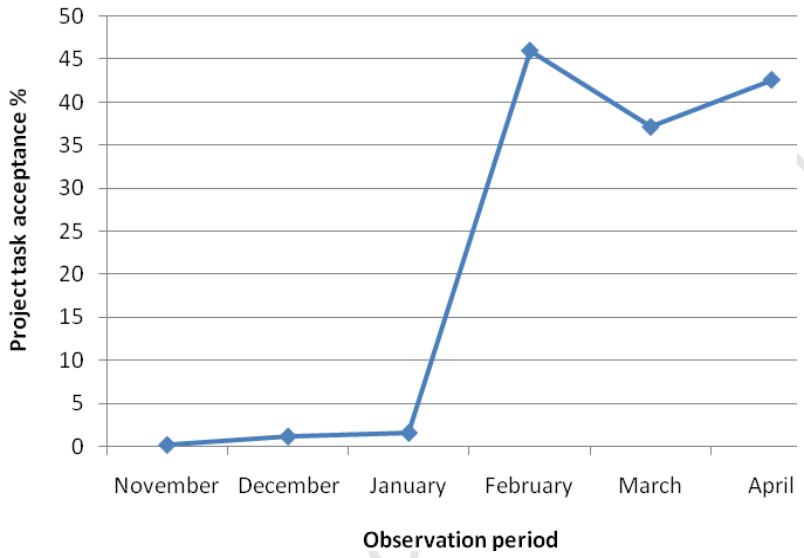
Although the tasks were being accepted as the project was progressing in the last three months, the profitability pattern seems to have been dropping considerably. This implies

that more time was invested in completing the tasks to acceptance standards while compromising profit on this project.

6.5.4.2 Project 2 analysis

For Project 2, the profitability pattern seems to have been different, as shown in Figure 52 below.

Figure 52: Project 2 profitability analysis



As the project started in February, there was a 46% chance that some profits would be realized in that time frame. However, the threshold was too low to realize any profit. In March, the likelihood of realizing R1600 profit per task executed in this time frame decreased by 8.86% to 37.13%. In April the pattern seems to have improved significantly, by 5.47% from the previous time frame, implying that there was a 42.61% chance that a profit margin of R1600 would be realized.

As the task acceptance rates increased in February and March, the profitability margins seem to have fluctuated considerably, with a 3.39% drop rate on the average and R1600 task profitability projection. In this analysis however, the correlation between project completion stage and anticipated profitability was not measure.

6.6. Case study implications

As we increase the acceptance rates in each time frame therefore, we ought to take our profit margins into consideration. The likelihood that we can win customer trust in the first two time frames after project inception, by delivering validated tasks, implies that we are likely to decrease our average profitability rates by 3.39% in the subsequent time frames.

This analysis only considered review of strategic project performance at different periods within the development lifecycle. However, it would have been interesting to measure what was predicted against what the final project cost was given the pattern of task completion and customer acceptance rates. This would have given the overall viability picture of these projects for management review and assessment and how this would impact on future projects with similar characteristics. However, given company policy on external studies, this was not achieved.

The area of post project review as a key project management competence is still in its infancy requiring rigorous techniques of analysis. It is quite instrumental in capturing explicit knowledge which can be leveraged in future projects, reducing overall cost. It can also help with transferring tacit knowledge which is even more extensive and potentially valuable. This chapter has demonstrated that with the help of economic concepts, models and tools such as the CASSE Framework, our understanding and improvement of the development of software and process that produce them can be advanced. Secondly, it has also demonstrated that extensibility of this framework is applicable in general project management, not tied specifically to software development but also other disciplines. As a way forward therefore, it is in best interest of business to ensure that such value-based methods are integrated into project planning and control for strategic business valuation. This will guarantee that software projects are successful and that are profitable. It will also certify that productivity is measured constantly and benchmarked against project objectives. Such feedback arising out of project evaluation thus can significantly assist in re-aligning our development process so as to produce profitability in our engineering processes.

University of Cape Town

CHAPTER SEVEN

Aligning the software project selection process with the business strategy: Analysis of the process view investigation

7.1 Introduction

7.1.1 Purpose

The purpose of this investigation was to exam whether project selection and requirements engineering processes used in industry today align appropriately with organizational business strategies in general. Secondly, it aimed at examining the way projects are selected in industry with or without use of analytical tools such as the CASSE Analytics, and also if there are any requirement engineering techniques used in assessing risk on software projects. This would highlight potential technical and financial gaps where this framework could be applicable. This in return would show that academia complements industry in addressing business challenges arising often in software development.

7.1.2 Investigation procedure

In order to achieve the above objectives, a survey was conducted over a four months period. This was carried out after a successful presentation of the CASSE framework engineering approach to the SPIN members in Cape Town. A total of 70 questionnaires were emailed to various software companies through this network. Out of the 70 questionnaires emailed, only 50 questionnaires were returned, thus representing a 71% response rate from the targeted sample. Given that this was an initial survey, this sample was relatively representative enough in assisting with determination of process characteristics and trends in this area.

All raw data were entered into an Excel spreadsheet. Data was analyzed using SPSS Statistics version 17.0 (a computer-based statistical software package).

7.1.3 Target sample

The targeted sample included software Project Managers, Systems Analysts, Business Analysts and other roles involved in the software development. This specific sample, specifically the analysts, were targeted due to their strategic role in guiding management on project scoping and viability implications especially during the software project selection stage and throughout the development lifecycle.

University of Cape Town

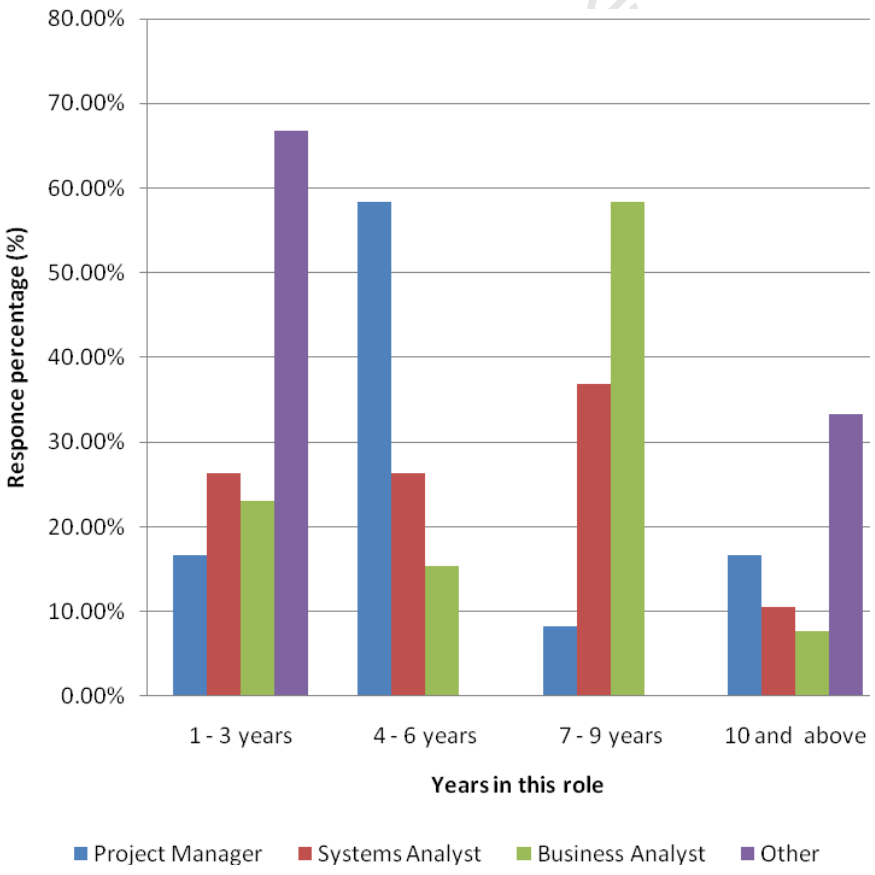
7.1.4 Chapter overview

The rest of the chapter details how the analysis was done, in terms of tools and procedure as well as interpretation implications to the software industry. This investigation stems for the process view investigation dimension illustrated in Figure 2. The underlying argument for this investigation is presented in Figure 3. A summary of these findings is given in the *results summary* section of this chapter.

7.2 Sample Characteristics and Frequencies

Frequencies were run on the respondents` role in the company in relation to their experience in this area of specialty as illustrated in figure 53 below.

Figure 53: Targeted Sample Characteristics



The graph above shows that out of the 50 respondents, 24% were Project Managers, 38% Systems Analyst, 26% were Business Analysts while other respondents constituted 12%. Of the 12 Project Managers interviewed, 17% had 1 to 3 years experience in this role, 58.3 %

had 4 to 6 years experience, 8% had 7 to 9 years experience and 17% had over 10 years experience in this role respectively. For the Systems Analysts role, 26% had 1 to 3 years experience, 26% had 4 to 6 years experience, 37% had 7 to 9 years experience and 10.5% had over 10 years experience respectively.

23% of the Business Analysts interviewed had 1 to 3 years experience, while 15% had 4 to 6 years experience, 58% had 7 to 9 years experience and 8% had over 10 years experience respectively. For those in the others category, 67% had 1 to 3 years experience and 33% for those with over 10 years experience.

The results show an equal distribution of respondents for the 1 to 3 years experience category and 4 to 6 years experience category each with a 28% score. The 7 to 9 years experience category had 30% score while the respondents in the 10 and above years of experience category had only 14% score. Thus most respondents had sound experience generally of over 7 years in the respective fields of specialty. With such experience therefore, they were more positioned to understand business and process complexities or even challenges associated with software development. More still they would be more positioned to understand the assertions presented in the study.

7.3 Study assertions

In order to determine the variable relationships, various assertions were given to respondents to ascertain their agreement levels on factors pertaining to project selection and risk detection in project requirements. All these assertions were coded and the following results were obtained.

7.3.1 Risk assessment in requirement process

Figure 39 below illustrates response parentages of respondents in relation to the assertions described in table 7 below which were posed in the area of risk detection in requirements engineering on software projects.

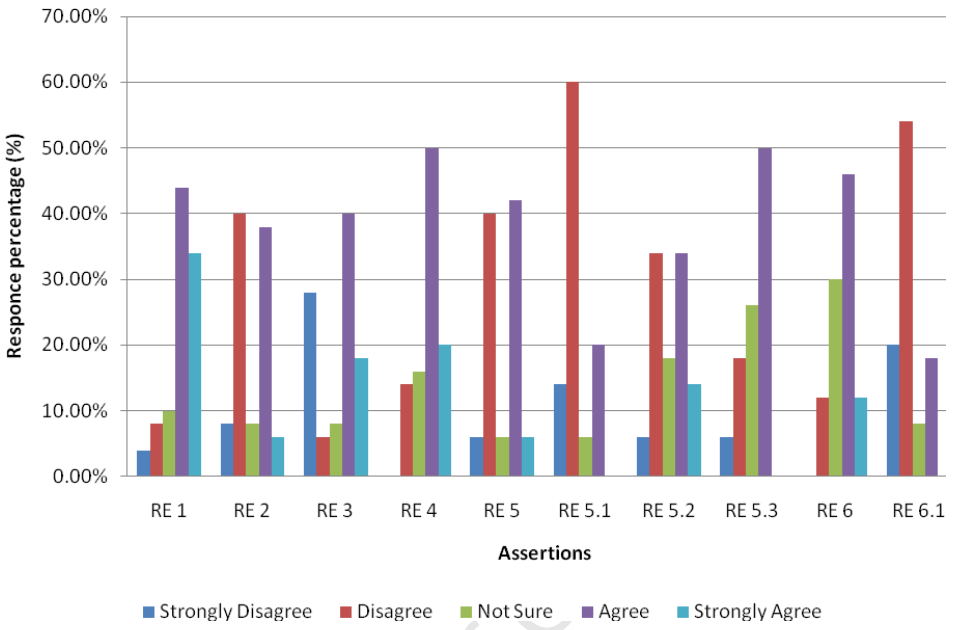
Requirements engineering assertions
RE 1: We usually first understand client requirements before project implementation
RE 2: We implement the requirements in parallel with requirement elicitation and validation
RE 3 We undertake systems modeling using OOSA using UML, etc.
RE 4: We use a standardized process in collecting, documenting and validating our requirements
RE 5: We usually develop a traceability matrix for project requirements
RE 5.1: The traceability matrix is generated using a matrix generation tool
RE 5.2: The traceability matrix is generated manually
RE 5.3: We don't consider the traceability formulation important because of lack of time

University of Cape Town

RE 6: It is part of our development procedure to undertake a requirement dependency assessment for risk assessment
RE 6.1: There is generally less or no requirement dependency assessment undertaken

Table 8: Assertions in requirements engineering

Figure 54: Risk detection in requirements engineering



For RE.1 (We usually first understand client requirements before project implementation), 44% and 34 % of the respondents "Agreed" and "Strongly Agreed" respectively to this assertion. 10% were "Not sure" while 8% and 4% "Disagreed" and "Strongly Disagreed" respectively. This implies that although 22% of the respondents disagreed with the statement, the 78% as majority of the respondents consider requirements understanding as an important step in systems development if a successful project has to be implemented. This implies that for these companies the analysis phase is reasonably critical if successful project completion is to be obtained and consensus has to be achieved on desired functionality –thus increasing client commitment on the project.

For RE 2(We implement the requirements in parallel with requirement elicitation and validation), 6% "Strongly Agreed" to the statement while 30% "Agreed". 8% were "Not Sure", 40% "Disagreed" and 8% "Strongly Disagreed" respectively. From this result, therefore the agile practice of rapid prototyping where requirements are implemented in parallel with elicitation and validation is not widely used, thus constituting 48% response rate. Only 36% of the respondents practice this approach. The 8% in this regard implies that there is a likelihood of practicing either of the practices maybe for some projects or perhaps the software development process utilized in these firms is not well defined.

University of

For RE.3 (We undertake systems modeling using OOSA using UML) 40% "Agreed" and 18% "Strongly Agreed" to this assertion, thus constituting an aggregate percentage of 58% of the total respondents. 8% were "Not Sure", 6% "Disagreed" and 28% "Strongly Agreed". This implies that UML is being widely embraced in this area although there are some companies that do not consider it as the best option for systems modelling. The other aspect would be the possibility of lack of expertise in this area of modelling – thus losing out on the benefit of functional decomposition, system transparency modelling and consequently adequate project estimating using this approach.

For RE.4 (We use a standardized process in collecting, documenting and validating our requirements), 50% "Agreed" and 20% "Strongly Agreed" with the statement. 16% were "Not Sure", yet 14% "Disagreed". Thus 70% of the respondents have some sort of standardized process in requirement collection, documentation and validation. However, a few of the companies do not. This poses a big risk to successful project completion and traceability complexities and challenges are bound to happen.

For RE.5 (We usually develop a traceability matrix for project requirements), 6 % "Strongly Agreed" and 42% "Agreed" respectively. 6% were "Not Sure", 40% "Disagreed" and 6% "Disagreed". This implies that only 48% of the respondents consider traceability matrix generation as an important step in requirements risk assessment. The other 46 % however, seem to put little emphasis on traceability matrix generation. This makes it difficult to quantitatively measure and track tasks on the project. Specifically, if you cannot track what happened where, who did it and how it was done, what is left of it and how it maps to the overall project picture.

For RE 5.1 (The traceability matrix is generated using a matrix generation tool), only 20% agreed with the assertion. 6% were "Not Sure", 60% "Disagreed" yet 14% "Strongly Disagreed" respectively. This implies that although certain firms generate traceability matrices for project requirements, few or no tools are utilized, thus creating a need to leverage available tool options so that the software development process is enhanced. At the same time, lack of tool usage in matrix generation is error prone and can result in misguidance in project choice and overall management.

For RE 5.2 (The traceability matrix is generated manually) as a control statement, 14% "Strongly Agreed", 34% "Agreed" while 18% were "Not Sure" about this statement. 34% "Disagreed" and 6% "Strongly Disagreed". Thus most of the companies that generate

University of Cape Town

requirements matrices for project requirements do it manually. This manual generation is error prone and can be time consuming, thus affecting implementation time ultimately.

For RE 5.3 (We don't consider the traceability formulation important because of lack of time), 50% "Agreed" to the statement. 26% were not sure, 18% "Disagreed" and 6% "Strongly Agreed" respectively. This implies that due to time constraints common for software projects, half of the respondents cut out the traceability generation process. This is attributed to the fact that it is time wasting especially if it has to be done manually given that there are perhaps no readily available tools to achieve this.

For RE 6 (It is part of our development procedure to undertake a requirement dependency assessment for risk assessment), 12% "Strongly Agreed" and 46% "Agreed" with the statement. 30% were "Not Sure", yet 12% "Disagreed" with the statement. Thus, 58% of the respondents acknowledged the fact that requirement dependency assessment was important if project risks were to be identified earlier on in the development process given that they practice this notion in their development process.

For 6.1(There is generally less or no requirement dependency assessment undertaken) as a control statement, 18% "Agreed" with the statement. 8% were "Not Sure", 54% "Disagreed" and 20% "Strongly Disagreed" with the statement. This result implies that while some firms undertake dependency analysis on the requirements, some others do not. This raises concerns on how risk detection is executed and managed on projects taken on by these companies.

7.3.2 Project selection process

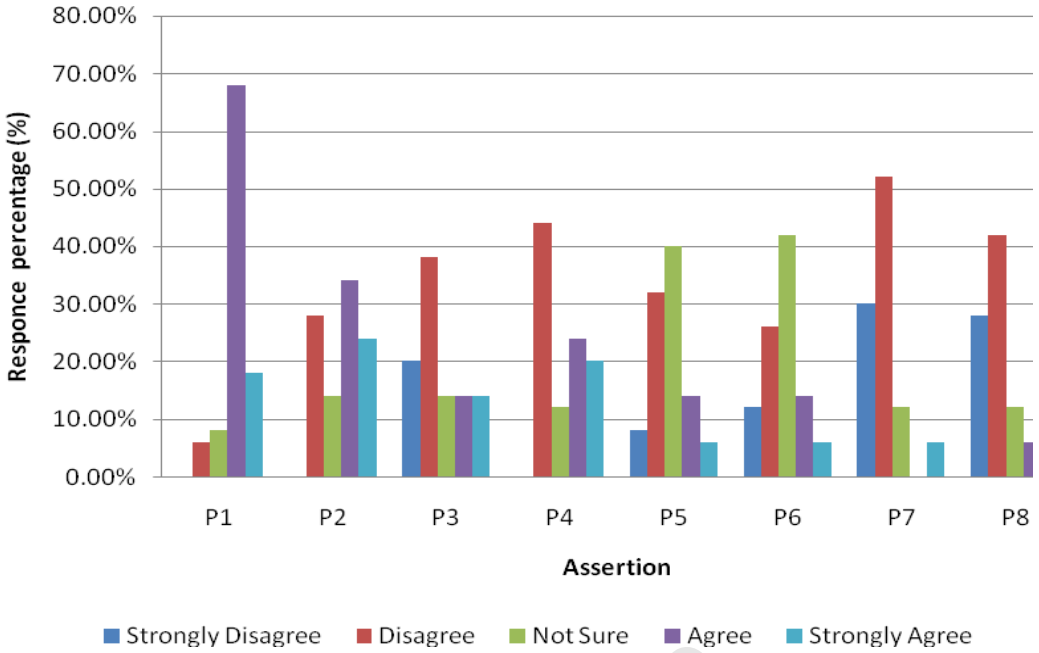
In the area of project selection, the following results were obtained as illustrated in figure 55 below. The following assertions described in table 8 bellow were used in achieving these results.

Project selection
PS 1: Our projects are selected based on strategic business objectives
PS 2: We choose our projects based on profitability level anticipated
PS 3: Our projects are in-house and my organization is the direct recipient
PS 4: We depend on feasibility assessment reports to determine project implementation
PS 5: We use analytical tools that guide us in project selection
PS 6: Our project selection is purely heuristic
PS 7: We just take on any project we get hold of
PS 8: Our projects are for research and development

Table 9: Assertions in project selection

University of Cape Town

Figure 55: Project selection basis



The graph above illustrates that for PS1 (Our projects are selected based on strategic business objectives), 6% "Disagreed" to this statement while 8% were "Not Sure". However, 68% "Agreed" to the statement and 18% "Strongly Agreed". This pattern of results shows that a majority of firms do base their project selection process on strategic business objectives. However, a small percentage seems not to or perhaps the business objectives are not clearly defined in some of these companies and therefore there is no strategic realignment of the process with business.

For PS 2 (We choose our projects based on profitability level anticipated), 28% of the respondents "Disagreed" to this assertion while 14% were not sure. However, 34% "Agreed" and 24 % "Strongly Agreed" respectively. Therefore, profitability seems to be a biggest driving factor in choosing projects for the majority of firms interviewed. However, a few of them do not necessarily rely on profitability per se, perhaps on how strategic a given project is aligned with the business objectives or perhaps their projects are for research and development.

For PS 3(Our projects are in-house and my organization is the direct recipient), 20% "Strongly Disagreed", 38% "Disagreed" and 14% were "Not sure". However, 14% "Agreed" and "Strongly Agreed" respectively. This seems to suggest that majority of the respondents work on external projects for clients while a negligible number work on in-house projects. This raises a need to have effective risk assessment approaches directed towards

University of Cape

understanding of project requirements and ensuring that a misinterpretation or misunderstanding of these requirements doesn't reflect negatively on the projects.

For PS 4 (We depend on feasibility assessment reports to determine projects implementation), 44% "Disagreed" with the assertion while 12% were "Not sure". While 24% "Agreed" and 20% "Strongly Agreed" to the assertion respectively. 44% of the firms do not depend on feasibility assessment to determine project implementation but on other factors. However, a few of them do.

For PS 5 (We use analytical tools that guide us in project selection), 8% "Strongly Disagreed" to the assertion while 32% "Disagreed" and 40% were "Not sure". However, 14% and 6% "Agreed" and "Strongly Agreed" respectively. This shows that most firms do not use any analytical tools to guide them through the project selection process. However, a few of them do and more still a biggest percentage are unsure of whether any tools are utilized in the selection process. This implies that to most of the respondents, although they may have an idea of the selection process, but there is uncertainty of whether any tools are used.

For PS 6 (Our project selection is purely heuristic), 12% "Strongly Disagreed", 26% "Disagreed" and 42% were "Not sure" about the assertion. However, 14% "Agreed" and 6% "Strongly Agreed" respectively. This implies that the biggest percentage of respondents do not necessarily rely on heuristic methods of project selection but follow some of metrics to determine project viability; a factor that applies to PS 7 assertion as well.

For PS 7 (We just take on any project we get hold of), 30% "Strongly Disagreed", 52% Disagreed and 12% were "Not sure". However, only 6% "Strongly Agreed" to the assertion.

For PS 8 (Our projects are for research and development), 28% "Strongly Disagreed", 42% "Disagreed" and 12% were "Not sure" about the assertion. However, 6% and 12% "Agreed" and "Strongly Agreed" to the assertion. This shows that majority of the companies interviewed focus on commercial software development rather than research and development projects which may not have strategic business direction in the short term.

7.4 Cross tabulations results of study assertions

7.4.1 Technique description

Given the above interpretations, a cross-tabulation analysis was undertaken to determine the significant relationship between the different assertions in the study as given in this subsection. Cross tabulation is a statistical technique that establishes an interdependent relationship between two tables of values, but does not identify a causal relationship

University of Cape Town

between the values. In this analysis it was specifically utilized because it offers the flexibility of describing the distribution of two or more variables simultaneously. In this distribution, each cell shows the number of respondents who gave a specific combination of responses. Secondly, this approach is easy to understand, appeals to people who do not want to use more sophisticated measures, it can be used with any level of data either nominal, ordinal, interval, or ratio - treating all data as if it were nominal, it provides greater insight than single statistics, it solves the problem of empty or sparse cells and also is very simple to conduct.

On the hand, the p-value was used to answer the statistical hypothesis of the investigation. A p-value is understood as the probability of obtaining a result at least as extreme as the one that was actually observed, assuming that the null hypothesis is true. In this regard, the lower the p-value, the *less* likely the result assuming that the hypothesis is null and so the *more* "significant" the result, in the sense of statistical significance. Statistically, this ranges from 0.05 to 0.01

The dependent variable investigated was the project selection basis for software projects. The independent variable was the requirement engineering approach used to determine risk in software projects. Cross tabulations between different assertions posed in this study were done to ascertain any significance among the variables and how this impacts on the entire value-based software engineering paradigm. Obtained results from the sample are illustrated using bar graphs and tables as detailed in rest of the chapter sections.

7.4.2 Anticipated project profitability & dependency assessment for risk

Table 10 below illustrates the results obtained in cross-tabbing the assertion “We choose our projects based on profitability level anticipated and it is part of our development procedure to undertake a requirement dependency assessment for risk assessment”

University of Cape Town

Table 10: Project choice & development procedure

			It is part of our development procedure to undertake a requirement dependency assessment for risk assessment					
			Disagree	Not Sure	Agree	Strongly Agree	Total	P Value
We choose our projects based on profitability level anticipated	Disagree	Count	3	4	7	0	14	0.14
		% within We choose our projects based on profitability level anticipated	21.4%	28.6%	50.0%	.0%	100.0%	
	Not Sure	Count	0	4	0	3	7	
		% within We choose our projects based on profitability level anticipated	.0%	57.1%	.0%	42.9%	100.0%	
	Agree	Count	0	4	10	3	17	
		% within We choose our projects based on profitability level anticipated	.0%	23.5%	58.8%	17.6%	100.0%	
	Strongly Agree	Count	3	3	6	0	12	
		% within We choose our projects based on profitability level anticipated	25.0%	25.0%	50.0%	.0%	100.0%	
Total		Count	6	15	23	6	50	
		% within We choose our projects based on profitability level anticipated	12.0%	30.0%	46.0%	12.0%	100.0%	

University of Cape Town

In testing whether projects are selected based on profitability level anticipated and whether it is part of the company's development procedure to undertake a requirement dependency assessment for risk, 58.8% of the respondents "Agreed" while 50% "Strongly Agreed to both assertions. However, 57.1% were "Not Sure" about both assertions. 42.9% were "Not Sure about project choice based on anticipated profitability while at the same time they "Strongly Agreed" that risk assessment is part of their development procedure. 21.4% "Disagreed to both of these assertions.

These results show that that the biggest percentage of companies that participated in this study undertake dependency analysis for risk on software projects given that most of their projects are selected based on profitability levels anticipated.

However, the p-value of 0.14 obtained between these two study variables illustrates that there is not significant relationship between the two variables. Therefore no conclusions can be drawn on this analysis.

7.4.3 No dependency assessment done & business objectives driving selection

Table 11 below illustrates the results obtained from a cross-tabulation analysis between the assertion “There is generally less or no requirement dependency assessment undertaken” and “Our projects are selected based on strategic business objectives “

University of Cape Town

Table 11: No dependency assessment undertaken & selection based on strategic business objectives

			Our projects are selected based on strategic business objectives					
			Disagree	Not Sure	Agree	Strongly Agree	Total	P Value
There is generally less or no requirement dependency assessment undertaken	Strongly Disagree	Count	0	0	4	6	10	
		% within There is generally less or no requirement dependency assessment undertaken	.0%	.0%	**40.0%**	**60.0%**	100.0%	
	Disagree	Count	0	4	23	0	27	
		% within There is generally less or no requirement dependency assessment undertaken	.0%	14.8%	**85.2%**	.0%	100.0%	**0.000**
	Not Sure	Count	0	0	4	0	4	
		% within There is generally less or no requirement dependency assessment undertaken	.0%	.0%	**100.0%**	.0%	100.0%	
	Agree	Count	3	0	3	3	9	
		% within There is generally less or no requirement dependency assessment undertaken	33.3%	.0%	33.3%	33.3%	100.0%	
Total		Count	3	4	34	9	50	
		% within There is generally less or no requirement dependency assessment undertaken	6.0%	8.0%	68.0%	18.0%	100.0%	

University of Cape Town

The results show that all the respondents who "Strongly Disagreed" that there is generally less or no requirement dependency assessment undertaken also generally "Agreed" to the assertion that "our projects are selected based on strategic business objectives". 85.2 % of the respondents who "Disagreed" with the assertion that there is generally less or no requirement dependency assessment undertaken again "Agreed" that their projects are selected based on strategic business objectives. However, all respondents that were "Not

Sure" about having no dependency assessment undertaken on projects, "Agreed" that their projects are selected based on strategic business objectives.

This shows that 74% respondents who disagreed (either strongly or just disagreed with the assertion "there is generally less or no requirement dependency assessment undertaken), are the same respondents that constituted to the highest percentage of respondents (86%) that acknowledged that their projects are selected based on strategic business objectives. This implies that generally in these firms, there is requirement assessment taken and projects are selected based on strategic business objectives.

The p value of less than 0.04(.000) also demonstrates the significance difference between the groups of respondents, that is, respondents who disagree with risk assessment being undertaken and agreed that projects are selected based on strategic business objectives as a two- majority and the second group of respondents that might have agreed with the former and disagreed with the later or were totally not sure.

7.4.4 No dependency undertaken & reliance on feasibility reports

In testing the relationship between the assertion "there is generally less or no requirement dependency assessment undertaken as RE 6.1 and the assertion 'we depend on feasibility assessment reports to determine projects implementation as PS 4", the following results were obtained as given in table 8 below.

University of Cape Town

Table 12: No dependency assessment undertaken & reliance on feasibility assessment reports

			We depend on feasibility assessment reports to determine projects implementation					
			Disagree	Not Sure	Agree	Strongly Agree	Total	P Value
There is generally less or no requirement dependency assessment undertaken	Strongly Disagree	Count	4	0	3	3	10	0.48
		% within There is generally less or no requirement dependency assessment undertaken	40.0%	.0%	30.0%	30.0%	100.0%	
	Disagree	Count	8	6	9	4	27	
		% within There is generally less or no requirement dependency assessment undertaken	29.6%	22.2%	33.3%	14.8%	100.0%	
	Not Sure	Count	4	0	0	0	4	
		% within There is generally less or no requirement dependency assessment undertaken	100.0%	.0%	.0%	.0%	100.0%	
	Agree	Count	6	0	0	3	9	
		% within There is generally less or no requirement dependency assessment undertaken	66.7%	.0%	.0%	33.3%	100.0%	
Total		Count	22	6	12	10	50	
		% within There is generally less or no requirement dependency assessment undertaken	44.0%	12.0%	24.0%	20.0%	100.0%	

University of Cape Town

40% of the respondents that "Strongly Disagreed" with RE 6.1 also "Disagreed" with the PS 4 assertion. 30% of the respondents that "Strongly Disagreed" with the RE 6.1 assertion, "Agreed" and "Strongly Agreed" with the PS 4 assertion respectively; thus contributing a 60% majority agreement with the PS 4 assertion.

29.6% of the respondents that "Disagreed" with RE 6.1 also "Disagreed" with the PS 4 assertion. 22.2% of the respondents that still "Disagreed" with the RE 6.1 assertion were

"Not Sure" about the PS 4 assertion, while 33.3% and 14.8% "Agreed" and "Strongly Agreed" to the PS 4 assertion respectively – thus contributing a 48.1% agreement to PS 4 majority over the 29.6% that disagreed.

However, all the respondents who were "Not Sure" about the RE 6.1 assertion "Disagreed" with the PS 4 assertion. 66.7% of the respondents that "Agreed" with the RE 6.1 assertion "Disagreed" with the PS 4 assertion, only 33.3% of them "Agreed" with the PS 4 assertion.

However, the p-value of 0.48 obtained between these two study variables illustrates that there is not significant relationship between the two variables. Therefore no conclusions can be drawn on this analysis.

7.4.5 Standardized process utilization & analytical tool usage

Table 9 below illustrates the results obtained from cross-tabbing the assertion "we use a standardized process in collecting, documenting and validating our requirements as RE 4" and the assertion "we use analytical tools that guide us in project selection as PS 5".

University of Cape Town

Table 13: Standardized process utilization & analytical tools usage

			We use analytical tools that guide us in project selection						
			Strongly Disagree	Disagree	Not Sure	Agree	Strongly Agree	Total	P value
We use a standardized process in collecting, documenting and validating our requirements	Disagree	Count	4	3	0	0	0	7	0.000
		% within We use a standardized process in collecting, documenting and validating our requirements	**57.1%**	**42.9%**	.0%	.0%	.0%	100.0 %	
	Not Sure	Count	0	0	8	0	0	8	
		% within We use a standardized process in collecting, documenting and validating our requirements	.0%	.0%	**100.0%**	.0%	.0%	100.0 %	
	Agree	Count	0	6	12	7	0	25	
		% within We use a standardized process in collecting, documenting and validating our requirements	.0%	**24.0%**	**48.0 %**	**28.0** %	.0%	100.0 %	
	Strongly Agree	Count	0	7	0	0	3	10	
		% within We use a standardized process in collecting, documenting and validating our requirements	.0%	**70.0%**	.0%	.0%	**30.0%**	100.0 %	
Total		Count	4	16	20	7	3	50	
		% within We use a standardized process in collecting, documenting and validating our requirements	8.0%	32.0%	40.0 %	14.0 %	6.0%	100.0 %	

The results show that 57.1% of the respondents that "Disagreed" to RE 4 "Strongly Disagreed" with PS 5 assertion. While 42.9% who "Disagreed" to the RE 4 assertion also "Disagreed" with the PS 5 assertion. All the respondents that answered both these

University of Cape Town

assertions were “Not Sure” about RE 4 and PS 5. 24% of the respondents that “Agreed” with RE 4 “Disagreed” with PS 5, 48% that “Agreed” to RE 4 were “Not Sure” of the PS 5 assertion and the 28% that still “Agreed” with RE 4 also “Agreed” with the PS 5 assertion. 70% of the respondents that “Strongly Agreed” with the RE 4 assertion interestingly “Disagreed” with the PS 5 assertion. Only 30% of the respondents “Strongly Agreed” to both assertions.

The p-value of 0.000 obtained shows that there is a significant relationship between use of standardized process in collecting, documenting and validating project requirement and the use of analytical tools in guiding the project selection process.

The 70% response score of respondents having a standardized process for collecting and validating requirements while lacking analytical tools for project selection purposes, suggests that there exists a gap between the process and business strategy somewhat. Standards may be followed to the very level best. However, if no tools that present all possible scenarios of a project are used (specifically looking at risks associated with the project requirements), however accurate validations may be, it could still be a wrong business choice to the company. Therefore, there is a need to align the two perspectives so that management decides appropriately on project endorsement aligned with business strategy.

7.4.6 System modelling & projects being in-house

Table 10 below illustrates the results obtained from cross-tabbing the assertion “we undertake systems modeling using OOSA using UML etc as RE 3” and the assertion “ our projects are in-house and my organization is the direct recipient as PS 3”

University of Cape Town

Table 14: Systems modeling & projects being in-house

			Our projects are in-house and my organization is the direct recipient						
			Strongly Disagree	Disagree	Not Sure	Agree	Strongly Agree	Total	P Value
We undertake systems modeling using OOSA using UML etc	Strongly Disagree	Count	7	7	0	0	0	14	0.000
		% within We undertake systems modeling using OOSA using UML etc	50.0%	50.0%	.0%	.0%	.0%	100.0%	
	Disagree	Count	0	0	3	0	0	3	
		% within We undertake systems modeling using OOSA using UML etc	.0%	.0%	100.0%	.0%	.0%	100.0%	
	Not Sure	Count	0	0	0	0	4	4	
		% within We undertake systems modeling using OOSA using UML etc	.0%	.0%	.0%	.0%	100.0%	100.0%	
	Agree	Count	0	9	4	4	3	20	
		% within We undertake systems modeling using OOSA using UML etc	.0%	45.0%	20.0%	20.0%	15.0%	100.0%	
	Strongly Agree	Count	3	3	0	3	0	9	
		% within We undertake systems modeling using OOSA using UML etc	33.3%	33.3%	.0%	33.3%	.0%	100.0%	
Total		Count	10	19	7	7	7	50	
		% within We undertake systems modeling using OOSA using UML etc	20.0%	38.0%	14.0%	14.0%	14.0%	100.0%	

University of Cape Town

The table above shows that 50% of the respondents that "Strongly Disagreed" to the assertion "RE 3 also "Strongly Disagreed" and "Disagreed" respectively with PS 3 assertion. All the respondents however that "Disagreed" with RE 3 were "Not Sure" of PS 3 assertion. Interesting, all the respondents that were "Not Sure" of RE 3 "Strongly Agreed" with the PS 3 assertion. 45% of the respondents that "Agreed" with RE 3 "Disagreed" with the PS 3 assertion. 20% of the respondents that "Agreed" with RE 3 also "Agreed" with the PS 3 assertion. Thus, 35% of the respondents that agreed to RE 3 also agreed with the PS 3 assertion. However, 33.3% of the respondents that "Strongly Agreed" with RE 3 also "Disagreed" with PS 3. More still 33.3% of the respondents that "Strongly Agreed" with RE 3

also “Disagreed” with PS 3 thus resulting in 66.6% disagreement on this assertion in relation to RE 3.

P-value of 0.000 obtained in this cross tabulation indicates that there is a significant relationship between undertaking systems modelling using OOSA using UML and projects being in-house based, where the organization is the direct recipient. This implies that there seem to be more time available in these companies to effectively analyze requirements and that these companies strive to use modern techniques of design and analysis. Furthermore, client pressure does not seem to be immensely impacting on the process of these companies. Otherwise production time pressures would arise. Use of modelling approaches such as UML therefore portrays system criticalities from all possible system vintage points and impacted areas within the system. This leads to production process optimization especially if product marketing targets have to be met within limited time.

7.4.7 R&D projects & less consideration of traceability formulation

Table 11 below illustrates the results obtaining from a cross tabulation between the assertion “our projects are for research and development as P S 8” and the assertion “we don’t consider the traceability formulation important because of lack of time as RE 5.1”.

University of Cape Town

Table 15: R&D Projects & traceability formulation importance

			We don't consider the traceability formulation important because of lack of time					
			Strongly Disagree	Disagree	Not Sure	Agree	Total	P Value
Our projects are for research and development	Strongly Disagree	Count	0	6	0	8	14	0.000
		% within Our projects are for research and development	.0%	42.9%	.0%	57.1%	100.0%	
	Disagree	Count	0	0	7	14	21	
		% within Our projects are for research and development	.0%	.0%	33.3%	66.7%	100.0%	
	Not Sure	Count	3	3	0	0	6	
		% within Our projects are for research and development	50.0%	50.0%	.0%	.0%	100.0%	
	Agree	Count	0	0	3	0	3	
		% within Our projects are for research and development	.0%	.0%	100.0%	.0%	100.0%	
	Strongly Agree	Count	0	0	3	3	6	
		% within Our projects are for research and development	.0%	.0%	50.0%	50.0%	100.0%	
Total		Count	3	9	13	25	50	
		% within Our projects are for research and development	6.0%	18.0%	26.0%	50.0%	100.0%	

University of Cape Town

57.1% of the respondents that "Strongly Disagreed" with PS 8 assertion, "Agreed" with RE 5.3 assertion. However, 49.2 % of the respondents that "Strongly Disagreed" with the PS 8 assertion also "Disagreed" with RE 5.3 assertion. 66.7% of the respondents that "Agreed" with RE 5.3 "Disagreed" with PS 8. The 33% of the respondents that were "Not Sure" about RE 5.3 also "Disagreed" with PS 8. 50% of the respondents that were "Not Sure" about PS 8 also "Disagreed" with RE 5.3. More still 50% of the respondents that were "Not Sure" about PS 8 also "Disagreed" with RE 5.3. All the respondents that agreed with PS 8 also were not sure about RE 5.3. 50% of the respondents that "Strongly Agreed" with PS 8 also "Agreed" with RE 5.3.

The p-value obtained of 0.000 shows that there is a significant relationship between projects selected being for research and development and the fact that there is less importance attached to requirement traceability formulation because of lack of time.

These results also seem to suggest that the “code now and design later” syndrome seems to be high in some of these companies. This sets a negative precedence on the production process. For instance, how can we capture all the design steps or how can we learn from mistakes done in the past if we do not assess the magnitude of the work before production time is incurred? Although projects may be for research and development, traceability mapping would help in identifying potential loopholes in design as well as the technical debts on the product under development. It is always important to track who did what, what was done, how it was done and what is left of it. All this is answered through traceability. Therefore, ignoring traceability formulation because of lack of time has huge consequences for product quality and therefore its acceptability and performance in the market.

7.4.8. Standard process & UML modelling

Table 12 below illustrates results from cross-tabbing the assertion “we use a standardized process in collecting, documenting and validating our requirements as RE 4” and the assertion “We undertake systems modeling using OOSA using UML etc as RE 3

University of Cape Town

Table 16: Standardized process & UML modeling

			We undertake systems modeling using OOSA using UML etc						
			Strongly Disagree	Disagree	Not Sure	Agree	Strongly Agree	Total	P Value
We use a standardized process in collecting, documenting and validating our requirements	Disagree	Count	4	3	0	0	0	7	.000
		% within We use a standardized process in collecting, documenting and validating our requirements	57.1%	42.9%	.0%	.0%	.0%	100.0%	
	Not Sure	Count	0	0	0	8	0	8	
		% within We use a standardized process in collecting, documenting and validating our requirements	.0%	.0%	.0%	100.0%	.0%	100.0%	
	Agree	Count	6	0	4	12	3	25	
		% within We use a standardized process in collecting, documenting and validating our requirements	24.0%	.0%	16.0%	48.0%	12.0%	100.0%	
	Strongly Agree	Count	4	0	0	0	6	10	
		% within We use a standardized process in collecting, documenting and validating our requirements	40.0%	.0%	.0%	.0%	60.0%	100.0%	
Total		Count	14	3	4	20	9	50	
		% within We use a standardized process in collecting, documenting and validating our requirements	28.0%	6.0%	8.0%	40.0%	18.0%	100.0%	

University of Cape Town

The results show that 57.1% of the respondents that "Disagreed" with the RE 4 "Strongly Disagreed" with RE 3. However, all the respondents who were "Not Sure" about RE 4 "Agreed" with RE 3. However, 24% of the respondents that "Agreed" with RE 4 "Disagreed" with RE 3 and 16% who still "Agreed" to RE 4 were "Not Sure" about RE 3. 48% and the 12% who "Agreed" and "Strongly Agreed" to RE 3 also "Agreed" with RE 4, thus resulting in a total of 60% response percentage for both assertions. 40% of the respondents that 'Strongly

Agreed" with RE 4 however, also "Strongly Disagreed" with RE 3.

The p-value of 0.000 shows that there is a significant relationship between use of standardized processes in collecting, documenting and validating of requirements and undertaking of systems modeling using *OOSA* using UML. This implies that a formalized process that looks at functional decomposition and abstraction for effective project execution and understanding is utilized.

With a sound requirement collection and validation process coupled with system transparency modelling using UML, a vendor is therefore guaranteed of analyzing possible risk areas within the system. This is often represented in a graphical or logical model of the system. An organization that doesn't leverage such aspects therefore stands to lose certain aspects of the detail. This may later on in the process translate into increased project time required to address the inadequately identified detail. This implies that more implicit costs on the project will be incurred, thus resulting in decrement of project profitability.

7.4.9. Standard procedure & selection based on profitability levels

In testing the relationship between the assertions "we use a standardized process in collecting, documenting and validating our requirements as RE4" and "we choose our projects based on profitability level anticipated as PS2" the following results were obtained as illustrated in table 13 below.

University of Cape Town

Table 17: Standardized procedure &selection based on profitability levels

			We choose our projects based on profitability level anticipated					
			Disagree	Not Sure	Agree	Strongly Agree	Total	P value
We use a standardized process in collecting, documenting and validating our requirements	Disagree	Count	7	0	0	0	7	0.000
		% within We use a standardized process in collecting, documenting and validating our requirements	100.0%	.0%	.0%	.0%	100.0%	
	Not Sure	Count	0	4	4	0	8	
		% within We use a standardized process in collecting, documenting and validating our requirements	.0%	50.0%	50.0%	.0%	100.0%	
	Agree	Count	3	3	10	9	25	
		% within We use a standardized process in collecting, documenting and validating our requirements	12.0%	12.0%	40.0%	36.0%	100.0%	
	Strongly Agree	Count	4	0	3	3	10	
		% within We use a standardized process in collecting, documenting and validating our requirements	40.0%	.0%	30.0%	30.0%	100.0%	
Total		Count	14	7	17	12	50	
		% within We use a standardized process in collecting, documenting and validating our requirements	28.0%	14.0%	34.0%	24.0%	100.0%	

University of Cape Town

The results show that all respondents that answered these assertions “Disagreed” to both of them. While 50% who were “Not Sure” about both assertions. However, 50% of the respondents who were “Not Sure” about RE 4 “Disagreed” with the PS 2 assertion. 40% and 36% of the respondents, “Agreed” and “Strongly Agreed” to both assertions respectively. However, 40% of the respondents that “Strongly Agreed” to RE 4 “Disagreed” to PS 2. More still 12% of the respondents that “Agreed” to RE 4 “Disagreed” to PS 2.

The p-value obtained (0.000) reveals that there is a significant relationship between use of

standardized processes in collecting, documenting and validating of requirements and selection of projects based on profitability level anticipated. Consequently, supporting the commercial world view which emphasizes that business is driven by profitability generation. Therefore, every vendor in business of software development ought to consider this as a business strategy.

7.4.10. Assessment for risk & analytical tool usage in project selection

In testing whether there is a significant relationship between the assertions "it is part of our development procedure to undertake a requirement dependency assessment for risk assessment as RE 6" and the assertion "we use analytical tools that guide us in project selection as PS 5", the following results were obtained as given in the table 14 below.

University of Cape Town

Table 18: Assessment for risk & analytical tool usage

			We use analytical tools that guide us in project selection						
			Strongly Disagree	Disagree	Not Sure	Agree	Strongly Agree	Total	P value
It is part of our development procedure to undertake a requirement dependency assessment for risk assessment	Disagree	Count	0	6	0	0	0	6	0.002
		% within It is part of our development procedure to undertake a requirement dependency assessment for risk assessment	.0%	100.0%	.0%	.0%	.0%	100.0%	
	Not Sure	Count	0	4	8	3	0	15	
		% within It is part of our development procedure to undertake a requirement dependency assessment for risk assessment	.0%	26.7%	53.3%	20.0%	.0%	100.0%	
	Agree	Count	4	6	6	4	3	23	
		% within It is part of our development procedure to undertake a requirement dependency assessment for risk assessment	17.4%	26.1%	26.1%	17.4%	13.0%	100.0%	
	Strongly Agree	Count	0	0	6	0	0	6	
		% within It is part of our development procedure to undertake a requirement dependency assessment for risk assessment	.0%	.0%	100.0%	.0%	.0%	100.0%	
Total		Count	4	16	20	7	3	50	
		% within It is part of our development procedure to undertake a requirement dependency assessment for risk assessment	8.0%	32.0%	40.0%	14.0%	6.0%	100.0%	

University of Cape Town

All the responds that responded to these assertions “Disagreed” with both of them. While 53% were “Not Sure” about both of these assertions, but 17.4% however, “Agreed” to both

assertions. However all the 6 respondents that 'Strongly Agreed" that it was part of their development procedure to undertake a requirement dependency assessment for risk were "Not Sure" if any analytical tools are used in guiding the project selection process in their development process. More still the 17.4% who "Agreed" to RE 6, "Disagreed" that they use analytical tools that guide them in project selection.

The p-value obtained of 0.002 reveals that there was a significant relationship between tool usage in project selection and requirement dependency assessment for risk. The results show that some level of risk assessment is done at the level of requirements collection and validation. However, tool usage is not well defined or readily available to some of these organizations.

7.5 Results summary

The above results are summarised as follows. Specific trends of interest were ranked according to a scale designed for overall results interpretation. This is illustrated in Table 19.

Table 19: Key for trends` scale

Prevailing trend	Scale
Not practiced at all	1
Require serious improvement	2
Occurs rarely	3
Satisfactory	4
Well practiced	5

It is upon this scale that conclusions were drown specifically looking at overall performance of a parameter of interest across the sample, majority responses to a parameter in comparison to the overall sample and a few or minority responses to a given parameter across the sample. The key parameters of interest are given in Table 20.

University of Cape Town

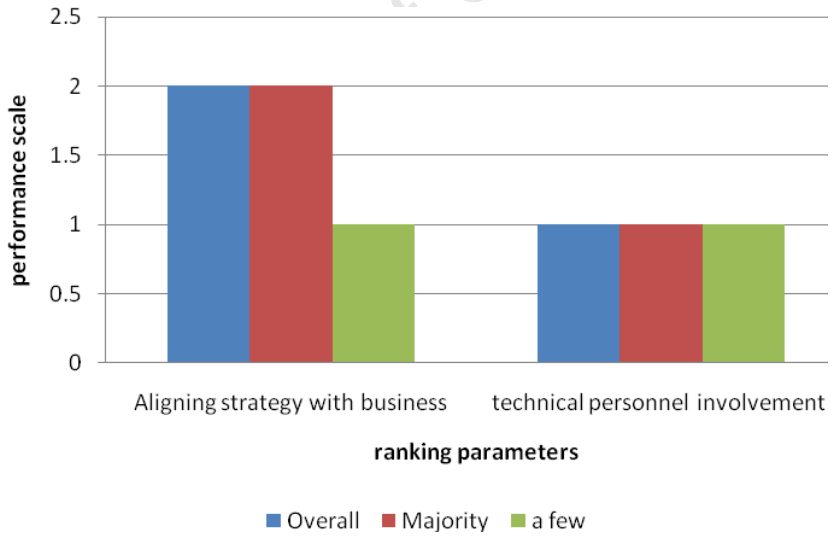
Table 20: Key parameters of interest

	Ranking parameters	Performance scale		
		Overall	Majority	a few
Overall business strategy formation	Aligning strategy with business	2	2	1
	technical personnel involvement	1	1	1
Requirement engineering	UML modelling	2	3	5
	tool usage in project selection	1	1	2
	tool usage matrix generation	1	1	1
	requirements traceability	1	1	2
	dependency analysis	1	1	2
Project selection	risk analysis on projects	2	3	4
	based on profitability & business goals	4	4	2
	Heuristic tendencies	3	1	3
	based on feasibility reports	5	4	3

As illustrated in the table above, for each parameter of interest, a scale ranking was attached arising from the statistical results obtained from the data.

7.5.1 Overall strategy formation

Figure 56: Business strategy formation

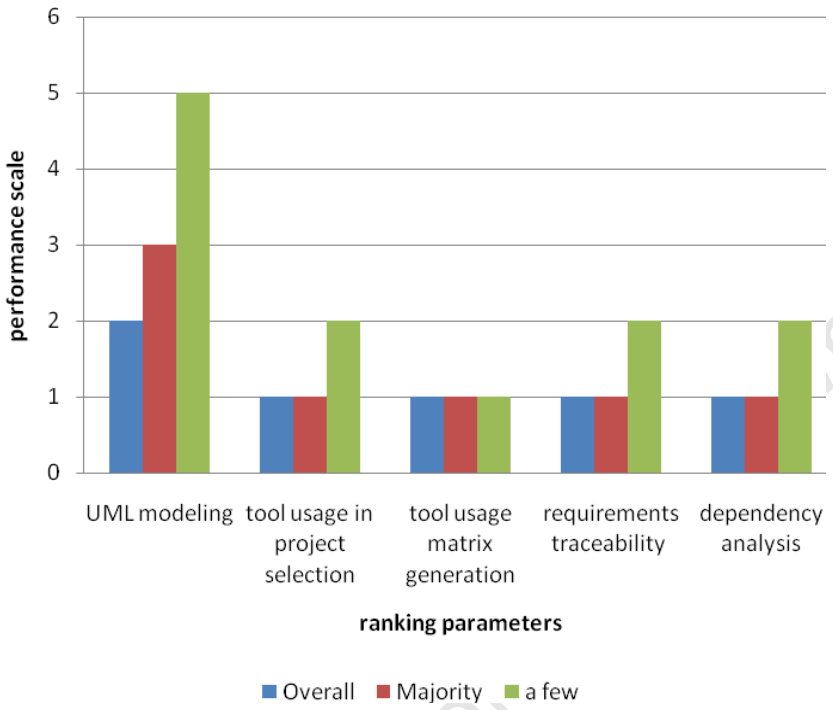


As illustrated from the figure above, it was observed that generally, aligning the selection and requirements engineering process with the business strategy requires serious improvement within the participating companies. The results also show that involvement of technical personnel in the formation of the business strategy is lacking across the board.

University of Cape Town

7.5.2 Requirements engineering

Figure 57: Strategic requirement engineering

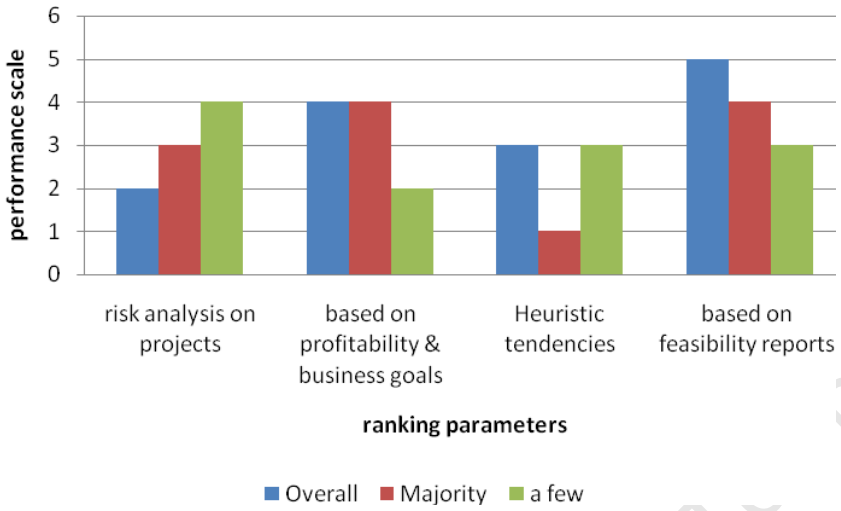


As illustrated in the figure above, UML modelling is well practised in only a few of the participating companies. In some companies however, it is used rarely, thus suggesting overall improvement in this area. On the other hand, tool usage in project selection and requirements matrix generation is not used at all. More still, requirements traceability as well as dependency analysis on software project requirements still scored poorly within this sample.

University of

7.5.3 Project selection

Figure 58: Project selection parameters



These results showed that, across this sample, projects are selected based on feasibility reports as well as on anticipated profitability and established business goals. However, risk analysis on these projects is only undertaken by a few of the respondents within this sample; majority of the respondents do not necessarily undertake this analysis.

7.6 Study Implications

The results from this study suggest that the need to realign processes with business objectives is still evident. If business objectives are not well translated into the processes that produce the software, then certain business elements may be sacrificed like, productivity measurement, quality and risk detection in business selection and management. This aspect is further emphasized by observations made by some of the respondents.

> *"…..in my experience, developer's get away with murder and should be better trained in business operations and held accountable for what they produce and the time taken to complete a piece of work. I agree that models, tools and techniques to procure better results are vital is success. To engender any 'faith' in such methodologies means changing the mind set of management – improvement can only come from the top downwards….."*

> *".....if Project Managers could ensure that sufficient diligence was exercised in requirements gathering and the client must sign-off on the quoted work, calculated by analysts & SME's, using adequate tools, these losses and earnings can be estimated to within 10%. The biggest flaw to getting this correct in any business implementation today, is the prohibitive cost estimates are purposefully and / or carelessly under-quoted in order to get the project 'approved' / accepted by the client. In the past, these were un-chartered waters and costs were genuinely unfathomable due to the nature of software implementations and lack of tools and expertise. There should be no such excuses in the 21st century……".*

University of Cape Town

"……the usability of these tools has been developed well enough for the people to receive and generate the relevant and required information out of them. Now is the time to work on the people that use them. In the end, it is the interaction between them that determines the success or failure of projects." These tools need to be developed to be as intuitive as possible. Currently, they have an abstract nature….."

"….. I do believe that economic concepts or models and tools should have a value and that development houses and business should use and fine tune such tools & follow proven methodologies but this will not happen until management change their mind-set about how a development team functions….."

"…..I feel that more discipline should be introduced into software development. This should include weekly code reviews according to a schedule, per developer. The format should include code & functionality rating as well as developer sign-off on each predetermined 'mini-module', complete with unit test results and documentation. Performance should include a 'par' [under or over] of man-days taken to complete. My current experience is that the development team is a "black box" and progress is a developer's opinion and not factual. The role of an analyst on a project needs to be increased in order to take responsibility for results of the design and drive the developers and report on progress accurately as the concept of "complete" is not uniform across the board.

"……..tools and techniques that place a microscope on development and measure: the accumulative 'par' in man-days for each developer's performance, a performance rating of each functioning 'mini-module' by code weekly reviews, the cost of delays, any SLA violated, the savings from expected services [under development] that will benefit the company; as development progresses, should be standard practice. The management of scope creep is underestimated and dealt with on an ad hoc basis without being analyzed……."

"……..business has no faith in any cost or time estimate of software development. This has resulted over +/- 3 decades of disillusion. It will take many decades to train developers to use any estimating tool accurately. The concept of accepting a time constraint on their development from a tool administered by 'others' [using such a tool] will not be accepted easily. Developers are such a variable factor in every aspect of software development, even though this should not be the case……"

"……..the general approach to software development is more cavalier than scientific. Insufficient time is allowed in the feasibility & quotation stage as it is expected that the cost will be many times more that estimated. The mind-set of development managers is more in line with 'get going' and the news of escalating costs will be dealt with as and when. Development managers usually hail from the development pool and have little on no respect for administration or finance. They fiercely defend their 'team' when time estimates are violated. Developers are notoriously inaccurate in estimating their work–load, as they are not properly trained to do estimates or to assess a development holistically. Extraneous limiting factors are not taken into account as this is often not their responsibility or the solution is currently unknown……."

From all the sentiments highlighted above, it is still evident that if processes are not well adapted to take care of technical personnel`s divergent perceptions that could be anti business change but pro ad hoc process usage which may not be economically valuable,

University of Cape Town

however much tools are provided for in the selection process, resistance may still likely to occur. This may in the long run fail the overall aim of process realignment as a market differentiation strategy.

As the study's statistical results show, a given percentage of the participating companies undertake risk analysis but the extent to which it is done cannot be quantified. Tooling is not widely used in project selection which implies that heuristic project selection approaches are usually applied. As observed by some respondents, if well adopted and integrated, it will bring much relief in terms of operations.

> *"……..management needs to understand and mandate the use of any improvement tools and / or techniques that focus on software development and not rely on the development manager so much. The strategic implementation of any tool or technique that forces transparency on software development will be a welcome relief to operations. A common concept adopted by management is that the cost of implementing any 'change' will result [initially] in higher development costs anyway without any guarantee of success. Management would rather reserve these expenses to off-set against development over-runs, which is the norm. This paradox relieves management of the responsibility of expensive over-runs in development costs. There is no time to sharpen the saw – just keep cutting down the trees!!……..."*

The other factor is that requirement traceability is sacrificed due to time constraints, yet if well integrated it can aid decision making reasonably. In the absence of such measure, it become difficult to track and interpret complexities embedded within project requirements. From the observations made by some respondents, this is quite evident.

> *"…a clear understanding of risks involved in requirement changes would greatly help project managers in determining what kind of requirement changes is viable, with minimal impact on the project. Furthermore, software models can make a more visual representation of the project for project managers that are not very good at manually updating and checking project time lines. Hence, timeline warnings or pop-ups can focus a project manager's attention to pending requirements and their impact on the entire project….. "*
>
> *"…… the biggest challenge facing the software development industry is the lack of consistency and readability of requirements documents. Every project manager seems to have a different approach to compiling requirements documents, which generally tends to lead to confusion or misunderstanding when it comes to development time. I believe a software model that can introduce a level of consistency and clarity would greatly aid in project management……"*

University of Cape Town

UML as a modeling technique has not been adequately embraced; on the whole, little or no requirement dependency assessment is undertaken and some companies often depend on feasibility assessment reports to determine projects implementation which has its pros and cons anyway. All these are key business issues that need to be addressed.

Therefore, the process characteristics revealed by the results of this survey are an indication of the business trends for all companies irrespective of size and business direction. As businesses, if we do not address such business needs, it makes it extremely difficult to survive in this fluid market. Development processes thus need to evolve as technology evolves and more still as the market trends change.

> *"……it is hard to stick to "a" particular development process because it all depends on the competency level of your team members. You will find that when you follow an approach to a particular development process it will be customized for the project or nature of the business. Some businesses follow a waterfall approach to software delivery but will adopt some of the good agile software development characteristics to make their development process work……"*
>
> *".....one word really, SCRUM. If this is followed with good development practices then you are 90% there……"*
>
> *"……..planning and forecasting are vital to business continuity and sustainability. Businesses need to learn from the past, apply sound operating practice and stop re-inventing the wheel……."*

Although projects are selected based on business objectives and anticipated profitability, these business objectives must be well explained to all stakeholders including the technical personnel who execute the tasks on these projects. Furthermore, the profitability anticipated must be mapped to the process so that it is well managed right through the development process especially if sustainable business continuity and project success must be achieved at project completion.

The only differentiating factor in the market place is the way projects are chosen, the techniques used to select these projects and the processes followed in executing and managing these projects. If analytical tools are available and adequately utilized in project selection processes, but well articulated development processes are lacking to manage these projects risks and profitability levels in the projects, then chances are that anticipated profits will be eroded during the project lifecycle, thus rendering the projects taken on unprofitable. If this kind of situation persists without proper risk mitigation and contingency planning or even with short of feedback control mechanisms to management, it may lead to business closure.

University of Cape Town

Although the results from this survey results cannot be generalized across the entire community of software developers, they are indicative of what is happening in a significant proportion of the these companies. The results still show that there is a significant relationship between requirement risk assessment approaches and viable project selection processes – consequently confirming the relationship between the dependent and independent variables investigated in this study. However, this relationship seems to be well managed, defined and understood in most of the companies that participated in this study. To others however, this entire process is not clear, as demonstrated in the high percentages of "Not Sure" responses to most of the assertions.

Thus, using frameworks such as the CASSE framework proposed in this study, the entire process can be managed congruently so as to ensure that the anticipated profitability levels on projects which are in line with business strategy are well tracked and managed. This can be achieved firstly by, identifying risks in requirements due to hidden and explicit relationships so that project viability is assessed and highlighted, secondly, by managing volatile requirement that change iteratively during the project lifecycle and thirdly by managing the relationships in project tasks so that anticipated profitability at project inception is well tracked as demonstrated in the previous chapter.

University of Cape Town

CHAPTER EIGHT

Conclusion and Future Work

8.1 Discussion

Effective and efficient selection of projects in strategy focused environments is vital for maintaining business sustainability. Software development is a complex process driven by factors that relate to problems as well as solutions. In evaluating investment options, every software project can be considered as representing an option that can be initiated, modified, cancelled or adopted. This opportunity cost decision-making process therefore, presents itself as a complex scenario involving multiple stages. Usually such scenarios that cover-up as bottlenecks to the success of projects because breaking down complexities is inconceivable without a proper understanding of its content.

Ideally, project management is concerned primarily with the detection, correction and prevention of risks, many of which originate in the conceptual stages of project development. With the increasing volatile trend of the software market today, a systematic evaluation of investment options is needed, especially if it is competing for the same capital resource. Therefore, the increased need for identifying where these risks are likely to arise is imperative, especially if a sustainable balanced project profitability level is to be attained even before commitment is made on these projects. The development processes and techniques utilized thus must evolve as the market forces evolve. While traditional evaluation methods incorporating risk analysis techniques require the input of relative frequencies, such frequencies are not easily available in the software engineering discipline.

Evaluating the probability distribution of losses and profits of a software project is a rigorous exercise which requires careful thinking and innovative approaches of evaluation before commitment is made on a project. Generally, evaluating economics surrounding software development is still an open research problem. Improvement of any project portfolio process can be achieved by ensuring effectiveness and efficiency in terms of each single project goal. There is still a great need therefore of effectively refining the software discipline towards maturity. Industry and academia must collaborate more closely in all aspects of systems and software engineering—designing, building, and managing complex "systems of systems.

University of Cape Town

It is my conviction that a significant portion of most strategies can be made operational by investing in and developing the process capabilities that can lead to market differentiation and competitive advantage. Strategic project selection therefore requires reliable data to align improvements with strategy, and to understand precisely the processes with which the business needs to develop.

As a business fact, value proposition of technology has always been about reducing cost, increasing revenue, ease of use, opening new markets and capabilities. In instances where it is inappropriately managed, it causes a lot of business frustration to management. Business managers therefore need to be aware of the root causes of business frustrations often which arise out inadequate techniques used in project selection, management and business choice. Such frustrations are traceable to broken yet fixable methods of creating and using business case assessments integrated in the entire project value management cycle.

8.2 Addressing validity threats in the study results

8.2.1 Business view investigation dimension

The biggest validity threats in the business view investigation dimension were the estimation weaknesses as well as confidence in the framework results.

- **Confidence in framework results**: This was partially due to the fact that this was a new approach that hadn't been tested anywhere in industry. Secondly, there was no other tool available which offered the same capability in order to draw meaningful comparisons of the results. Ideally, it seemed a very challenging task to balance improvement to the software development process versus the management of economic value. The research may have benefited by rebalancing, perhaps in favour of development process. Like any other decision support system, the final decision about the appropriate action to be taken has to be made by the domain expert. Therefore, the evaluation results were provided as a manifestation of the project status. This threat did not affect the study results since evaluation of the performances and comparison of the tool with other approaches is yet to be done in future work.

- **Estimation weaknesses**: It offered an alternative since the available approach was purely heuristic. The company preferred to use the framework estimates in the quotation process since it offered a relatively higher revenue projection value than the existing approach. The fact that the tender wasn't won, didn't not affect the study results in any way. This award amount was only used for evaluation purposes

University of Cape Town

of the project score for future analysis and besides the award criteria and decision was in the hands of the client. Therefore, this was an uncontrollable variable.

8.2.2 Process view investigation dimension

In the process view investigation dimension, social desirability bias of questions used as well as varying perception of the true meaning of the term business strategy and questions used were identified as the major threats to the study results.

- **Social desirability bias of Questions:** This validity threat in a way affected Questions PS1, PS6 and PS7 of the survey instrument used. At the study design level, it was anticipated that respondents were likely to agree that they select projects that further business objectives and not randomly. In order to control this threat, a cross tabulation technique was utilized to ascertain the significance relationship between these questions and assertions raised in the requirement engineering section. This resulted in establishing whether there was any correlation between these questions. This in part minimized the validity threat of desirability bias since project selection wasn't treated mutually exclusive from requirements engineering technique in drawing conclusions of the study.
- **Misinterpretation of the term business strategy and questions used:** The main validity threat here was that just by asking respondents if project selection is based on strategic business objectives would not necessarily answer if project selection process is aligned with the business strategy. Secondly, no control questions in the questionnaire were integrated to measure this aspect. Misinterpretation of this term or different perception of it would therefore arise. However, in ensuring that the results are not affected significantly, I presented the framework in one of the SPIN meetings in Cape Town. Explanation of the framework fundamentals was highlighted in which business strategy formation and alignment in relation to project selection and management was emphasized. Besides, in this meeting participants were given a chance to ask various questions in relation to this approach and its applicability in industry. Of the total respondents in this study, only about 70% were in this SPIN meeting. Taking this percentage and given the fact that explanation was given on what business strategy is in relation to this approach, this threat was minimized. On the whole given that the majority of the respondents were in the SPIN meeting, the remaining 30% who were not part of the meeting couldn't not alter the study results significantly. Based on the returned questionnaires, no mention of question clarity issues was highlighted. This reinforced my inclination that the questions were fairly clear to the respondents.

University of Cape Town

8.3 Answering the research questions

Question 1: ***Based on the analysis of Actor Object Dependencies (AODs) and their properties, can we estimate the probability distribution of losses and earnings that can be incurred by a software vendor on any project given his software project portfolio***?

The AO dependencies utilized in this approach offers the foundation for understanding complexity at the technical level and how it affects the overall product performance over time. This would demonstrate that profitability as well as viability bottom-line establishment can be achieved with adaptive tools that keep track of what is happening in the process at every level within the development lifecycle.

Findings from case 1, 2, 3 and 4 of the business view investigation dimension demonstrate that it is possible to estimate the probability distribution of losses and earning that can be incurred by a vendor on any project given their project portfolio while using this approach. This is on the assumption that the available project requirements are well structured and engineered in a value creating pattern. Secondly, there is sufficient expertise in the analysis and modeling of project requirements within the organization. Thirdly, that the business strategy, on which business operations are based, is adequately diffused to the low levels of the organization in order for it to be successful. However, without addressing these fundamental value-driving factors within the production process, even if various tools are introduced, estimating and managing the distribution of earning will be difficult to achieve.

Therefore, the CASSE framework is valuable in examining the profitability distribution of losses and earnings on a project portfolio with a significant degree of certainty. The benefit in this is that it offers the capability for increasing the predictability and evaluation of a viable risk-balanced favorable operating point on any project before this project is added to a portfolio. Thus, frameworks such is, if well adapted and integrated in the operational environment, would minimize the level of failing projects as well as those non-profitable ones.

Question 2*: Can economic concepts, models and tools help to advance our understanding and improvement of the development of software and of the processes that produce them?*

Case study results from case 5 in business view investigation dimension affirm the notion that economic concepts, models and tools can help to advance our understanding and improvement of the development of software and of the processes that produce them. By using the CASSE framework to evaluate the various project performances over time, a lot of valuable patterns emerge. Coupled with findings adduced in case 1, 2, 3 and 4, these value patterns are very essential in shifting the entire software development philosophy of a given

University of Cape Town

software producing organization to a value-oriented paradigm. Project variants including requirement or task acceptance, task completion throughout the project duration and profitability analysis patterns impact considerably, on development processes in the first place, and on resource allocation and management capabilities in the second. The order in which selected tasks are prioritized, executed, completed and validated is very important.

The study finds further illustrate that if we use such modeling techniques and concepts (while being governed by value-oriented tools), for ongoing project review processes, we can enhance project portfolio management in the following ways: by determining proper project estimation, by delivering projects on-time and on-budget, and by properly identifying key project requirements and risks. Thus, managers will be able to prioritize activities for the effective management of project completion, and shorten the planned critical path of projects by pruning critical path activities, by performing more activities in parallel, and/or by shortening the durations of critical path activities through adding resources, hence preparing for scheduling and resource planning effectively. Therefore, alternative engineering approaches such as these can enhance how we select software projects and how we plan project schedules optimally so as to increase business value derived on projects.

Question 3: ***Do the processes of project selection and requirements engineering employed in industry today align appropriately with organisational business strategies for supporting value creation on software projects?***

The findings from the survey undertaken in process view investigation dimension reveal that on the whole, project selection and requirements engineering approaches employed in industry today do not necessarily align fully with organisational business strategies for supporting value creation on software projects. While these results cannot be generalized across the entire spectrum, they are indicative of what is happening in most of the software-producing companies, irrespective of size and business direction. This is therefore a key business challenge that requires redress. Business strategies therefore have to be diffused into the organizational operational streams and must be well understood at all levels. By enhancing our requirement engineering approaches to address value creation on software projects, substantial revenue will be generated on the projects undertaken. Equally, by aligning the project selection process with a well diffused business strategy across the organizational boundaries, employee commitment to value creation on software products will be ensured.

University of Cape Town

Therefore, a significant portion of most value-based engineering strategies available can be made operational by investing in and developing the process capabilities that can lead to increased competitiveness in the software business.

8.4 Conclusion

The case studies presented in this work signify that the integration of qualitative risk assessment and quantitative risk evaluations on various project options right at the project selection process is paramount in making wise strategic decisions. Equally, the complexity checks made on the project using emergence engineering approaches while embracing probability theory can further enhance decision quality. The results of this study show that this approach works effectively for making strategic business decisions within a micro project value environment. Its application can serve a wider range of business scenarios for small, medium and large organizations in the business of software with a view of business growth management and sustainability in the agile market.

With such a technique, a vendor can harness market based principals to maximize competition, reduce cost and guide investments and at the same time facilitate consumer-driven decisions that respond to traditional market forces. Integrating such approaches therefore into the software development process thus ensures sustainable competitive advantage in all production time runs. Sustainable competitive advantage takes place when businesses are able to achieve operational excellence in aspects of the business that are inherently linked to the strategy by differentiating a business through what it does as much as how it does it.

This approach thus can help vendors in appropriate opportunity cost analysis to identify the most significant gaps between "baseline" and "targeted" returns on investment given investment levels on a project using AODs; thus directing production towards process capabilities that are critical to meeting the new business targets.

The CASSE Framework therefore was used to demonstrate that:

- Companies engaged in software development business often need to assess and evaluate their development processes and how these processes link to the overall business strategy. As illustrated in case study four, within chapter 5 of this thesis and also from the results obtained from the process view investigation, failure to have such mechanism in place will always lead to process immaturity which may ultimately lead to being out-competed in the market place. This may ultimately lead to business closure. Management often needs to get this constant feedback from its

University of Cape Town

specialists mandated to execute the process assessment role so as to make viable decisions. Without such feedback and assessment mechanisms, risky decisions that sink business could be made.

- Various aspects of the project, including profitability analysis can be examined. As illustrated in chapter 6, project variants including requirement or task acceptance, task completion throughout the project duration and profitability analysis patterns impact considerably, on development processes in the first place, and on resource allocation and management capabilities in the second. The order in which selected tasks are prioritized, executed, completed and validated is very important.

  If we use such modelling techniques for ongoing project review processes, we can enhance project portfolio management in the following ways: by determining proper project estimation, by delivering projects on-time and on-budget, and by properly identifying key project requirements and risks.

- It is possible to determine project value based on AOD analysis techniques, specifically by predicting and evaluating viable risk-balanced favourable operating point for any software vendor; thus guiding business decisions making especially in the times of uncertainty. This is illustrated in case study one of chapter 5. Even in its preliminary state, this framework is sufficiently expressive to be useful in explaining and characterizing project evaluation techniques.

- Measurement of overall project success can be improved, thereby enhancing a company's ability to handle project portfolio significantly, as it compels the company to consider the magnitude or complexity of software projects taken on. This is illustrated in case study 3 within chapter 5.

In addition, results obtained from the survey can be used to establishing baseline data for future comparisons and problem analysis, and secondly for identifying trends, issues, and concerns unique to software development process management and challenges.

This work is not only aimed at improvement of software engineering management as a discipline, but also the improvement in management of projects in order to derive value in development of IT products. As shown in chapter seven, one of the benefit it offers is that managers will be able to prioritize activities for the effective management of project completion, and shorten the planned critical path of projects by pruning critical path activities, by performing more activities in parallel, and/or by shortening the durations of critical path activities through adding resources, hence preparing for scheduling and

University of Cape Town

resource planning effectively. This identification of the critical chain of events would make it possible to mitigate their negative effects since risk lists of projects can be generated as a result of sensitivity analysis.

Therefore, by adopting optimal approaches supported by economic concepts, models, and tools, we can clearly estimate the probability distribution of losses and earnings that can be incurred by a software development organization according to its software project portfolio. Such tools would be instrumental in enhancing decision analysis on projects and improving software development processes.

This approach however, is an alternative solution that has been tested in the commercial environment on real life projects specifically embracing key business project values such as project profitability. This technique is developed in such a way that it is extensive and pluggable, thus it can be generalized to other areas of Software Engineering and other disciplines. Consequently, if this approach is extended to address changing tasks and requirements on a project, possibly some of the problems that arise under multi-project environments such as resource allocation, coordination or communications can be overcome.

Although the CASSE approach is expressive at project selection, it would equally be important to assess the project status at project completion to see how the project scored. If it started out in one block of the value conundrum, did it shift to another block or it remained stagnant? If it shifted to a different block, did it make it to a more value-oriented block or a more risky block?

University of Cape Town

8.5 Limitations of the CASSE framework

- **CASSE Framework Scalability:** Although conflicting dependencies can be detected in the current state of the framework, they are not as optimal as they should be. In future, a multi-agent probabilistic network technique suggested by Wong & Butz`s (2009) will be integrated in order to detect all inconsistent information and also remove all redundant information automatically. This technique will in turn be compared with the belief logic programming technique suggested by Wan,(2009) to compare results and decide on the optimal option worth integrating in the framework for optimal performance.

- **Intangible benefit evaluation**: The additional considerations of value (from the business value perspective) which are not measured with this framework are the intangible benefits and additional future benefits established by a project.

Intangible benefits may include improved customer satisfaction, better image. Additional future benefits established by project benefits may include deploying a new CRM system that lays the groundwork for a human resources system within the same structure.

- **Technology limitation**: The area of comparison between BN techniques against dependability analysis techniques has already been extensively studied in the previous work (Maglogiannis, 2006; Karlsson, 2008). Substantial results have been presented in recent studies too (Montania & Boudalib, 2008; Distefano & Puliafito, 2009). I therefore excluded this aspect. However, it would be interesting to see the various results obtained from both techniques and how each of these results compares. Invariably, these results will determine future direction of which technique is optimal given various scenarios and operational environment when applied to the CASSE Framework.

- **Interpretation:** The biggest challenge of the framework is related to interpretation. While the tool attempts to take subjectivity and bias out of the equation, it in fact it may introduce lots of additional subjectivity. Thus adequate training needs to be taken into consideration when utilizing its analysis results.

- **Learning aspect of CASSE:** This was not well developed in the research because to do this well, a longitudinal study would be required, thus rendering the learning aspect weaker.

- **Testing**: It was not possible to validate the framework in an operational environment or testing components of it in managing the entire software development lifecycle due to time limitations. However, this has been suggested as ground for future research direction of this work. One of the probable approaches could concentrate on testing the effectiveness and efficiency of the framework in practice. This could take a properly designed and executed experiment where CASSE can be one treatment and a competing method could be the other. This could measure effectiveness and efficiency of the framework in a meaningful comparison. An alternative could be a proper case study in industry testing parts of it.

University of Cape Town

8.6 Future work

8.6.1 Deployment in an operation environment

There are various areas within which this work can be expanded by the research community. This work has only been tested at the project selection stage as well as at the post implementation review stage from a task analysis perspective. It has not been deployed in an operational environment to demonstrate its capabilities in real time. Perhaps the deployment of this tool in an operational environment would provide different results. However, while using this framework, it would be interesting:

- To see how any given project scored against the business threshold at project selection stage is tracked through the project lifetime. This would provide trends in analysing lessons learnt on a project and whether the anticipated profitability was well sustained or not. The detected patterns would provide these answers and probably, these answers would provide a mechanism for improving both the estimation process on projects as well as the quoting process on these projects over time.
- To investigate how emerging project patterns captured by this framework improve decision-making in real time and how this can be compared and quantified in productivity terms as a mechanism for improving the software development process.
- To investigate how different applications or projects under a given portfolio or portfolios react to any detected pattern(s) during the project execution stage. Analysing such a trend would provide a mechanism for flagging different trends that occur regularly when certain conditions become true on these projects. This in turn would minimize repeating the same mistakes across a multitude of projects within a given portfolio, that is, a pattern detected on one project could have positive or negative results on another project within the same portfolio or across the entire portfolio.

8.6.1 Evaluation with other frameworks

The studies reviewed during this investigation seem to be concentrating on developing value-based frameworks in a single setting such as requirements engineering**,** architecting, design and development, verification and validation, planning and control, risk management, quality management and people management. As a result, holistically evaluating the CASSE

University of Cape Town

approach with any exiting integrated framework became extremely difficult, specifically when using common parameter of analysis interest. The CASSE approach is provided as the initial step in integrating these frameworks. Another possible area of investigation can be extended to:

- measure the performance of CASSE in relation to other frameworks. Specifically, this can be done in assessing where a given ranking or optimising framework can provide more efficient results in predicting and managing value on software projects.
- Investigate how the available frameworks can be integrated into the CASSE framework to provide an integral overall management of software projects so as to provide greater benefit for the industry. Understanding how these frameworks can work together and how results obtained in one can complement decisions made in the other, would offer a layered approach in assessing and managing value across all value generating activities on a software project.

8.6.3 Varying software document patterns

The CASSE framework has not be adapted to allows for analysis of the trade-offs between writing software which is well documented and, therefore, more supportable but takes longer and costs more as opposed to lower levels of documentation and, therefore lower costs but more difficult to support. Therefore, assessing such a mechanism in future would provide valuable research grounds.

8.6.4 Applicability in general project management

The CASSE framework was only developed for commercial software development rather than open source software development settings. However, enabling mechanisms have been integrated in the framework to provide generic capabilities so as to adapt to non-commercial software development and the general project management spectrum. It is an interesting view as well to assess how this framework performs in such environments since similar analysis patterns seem to cut across this area.

8.6.5 Other project value differentiator

This study has not explored how other project value differentiators can be plotted on Y-Axis o the project value evaluator. In future, it would be interesting to see how this can be done and how it can align the framework metrics.

University of Cape Town

8.6.6 Other possible areas

Other possible future areas of investigation require extending the framework to handle multi-project comparative value analysis and benchmarking, change management profitability assessment and adaptability as well as dynamic profitability indication throughout the development lifecycle. Certain strategic issues within the business and the process need to be examined. For instance, given the changing client requirements and in certain cases where budgets and schedules are inelastic, how can we adaptively update our project plans without impacting significantly on our profitability levels or budget and schedule requirements? If we are successful in tracking this pattern, how can we strategically maintain our projects in the profitable quadrants without compromising quality of our products?

Thus, it is a significant business security to select proper project selection techniques that increase the rate of successful yet profitable projects. This has to be done while realizing the benefits equilibrium between stakeholders and considering the different interests requirement from different interests groups. The approach presented in this thesis therefore aims at simultaneously satisfying the two-step requirements at pre-evaluation phase and during the development lifecycle; hence assuring the decision to be not only scientific but also practical in the selection and management levels of software projects.

University of Cape Town

References

1. Abercrombie, N., Hill, S., and Turner, B. S. Dictionary of Sociology. 3rd ed. Penguin Books, 1994.
2. Abernethy, J., Bach, F., Evgeniou, T. and Vert, J. A new approach to collaborative filtering: operator estimation with spectral regularization. J. of Machine Learning Res., 10(2009), 803-826.
3. Ahmed,Y. Design guidelines for the noise optimization of a 0.18 µm CMOS low-noise amplifier. Analog Integrated Circuits and Signal Processing, 46, 3(2006), 193-201
4. Akker, M. V.D, Diepen, S.B.G , Versendaal, J. Software product release planning through optimization and what-if analysis. Info. & Soft. Tech.50, 1-2 (2008), 101-111.
5. Alba, E., and Chicano, J.F. Software project management with GAs. Info. Scie. 177(2007), 2380—2401.
6. Albrecht, A. J. Measuring application development productivity. In Proceedings of the IBM Applications Development Symposium (October 14–17, Monterey, California). IBM, 1979, pp. 83–92.
7. Al-Hajri, M.A, Ghani, A.Z.A., Sulaiman, M.N., and Selamat, M.H. Modification of standard Function Point complexity weights. J. of Sys & Soft. 74, 2(2005), 195-206.
8. Ambler, S.W. Agile Software Development at Scale. LNCS, 5082/2008(2008).
9. An, X., Jutla, D., and Cercone, N. Privacy intrusion detection using dynamic Bayesian networks. In Proceedings of the 8th international conference on Electronic commerce (Fredericton, New Brunswick, Canada). ACM, New York, 2006. pp. 208 – 215.
10. Anda,B., Benestad, H.C., and Hove, S.E. A multiple-case study of effort estimation based on use case points. In Proceedings of the 2005 International Symposium on Empirical Software Engineering. IEEE CS Press, 2005, pp.10.
11. Andersen, P. B., Emmeche, C., Finnemann, N. O., and Christiansen P. V. Downward Causation. Minds, Bodies and Matter. °Aarhus University Press, 2000.
12. Armour, P. Project portfolios: organizational management of risk. Comm. of the ACM, 48, 3(2005), 17.
13. Atkinson, R., Crawford, L., and Ward, S**.** Fundamental uncertainties in projects and the scope of project management. Int. J. Proj. Man.24, 8(2006), 687-698.
14. Babu, A.J.G, "Modelling and optimizing software quality", Int. J Qua. & Reliability Man.13, 3(1996)95-103.
15. Baccarini, D., Salm, G, and Love, P. E.D. Management of risks in information technology projects. Ind. Man.& Data Sys, 104, 4(2004), 286 – 295.
16. Balakirsky, S. and Lacaze, A., Value-driven behavior generation for an autonomous mobile ground robot. In Proceedings of SPIE Aerosense Conference Vol. 4715 (Orlando, Florida). IEEE CS Press, 2002, pp.1-5.
17. Balikuddembe, J.K. and Bagula, A. Business viability assessment of potential software projects: an empirical study with the CASSE Framework. Paper accepted for the 2009 International Conference on Advanced Software Engineering & Its Applications (ASEA 2009) , December 10 ~ 12, Jeju, Jeju Island, Korea.
18. Balikuddembe, J.K. and Bagula, A. Requirements negotiation on software projects; a value-based Actor Object Dependency Analysis Technique. In Proceedings of the 4th IFIP TC2 Central and Eastern European Conference on Software Engineering Techniques, (Krakow, Poland)2009.
19. Balikuddembe, J.K. and. Potgieter, A.E. Predicting value-creation in software project complexity. In Proceeding of the 10th IASTED International Conference on Software Engineering and Applications (Dallas). Acta Press, 2006pp. 514-049.
20. Balikuddembe, J.K. Bagula, A Osunmakinde, I.O. Software Project Profitability Analysis Using Temporal Probabilistic Reasoning; an Empirical Study with the CASSE Framework.

University of Cape Town

Advances in Security Technology – Comm. in Comp.& Info. Springer Berlin Heidelberg, 2009, pp.138-150

21. Balikuddembe, J.K. Bagula, A., Rivets, U. Complexity management of customization requests on software products; A value-based Approach. In Proceedings of the 2009 International Conference on Software Technology and Engineering (ICSTE 2009), Chennai, India. 2009.
22. Balikuddembe, J.K., and Bagula A. Requirements negotiation on software projects; an empirical study. In. Proc. Of the IFIP Central and Eastern European Conference on Software Engineering Techniques, (Krakow, Poland). 2009.
23. Bannerman, P.L. Risk and risk management in software projects: A reassessment. J. of Sys. & Soft., 81, 12(2008),2118-2133.
24. Barclay, C. Towards an integrated measurement of IS project performance: The project performance scorecard. Inf Syst Front, 10(2008), 331–345.
25. Barney, S., Aurum, A., and Wohlin, C. A product management challenge: creating software product value through requirements selection, J. Sys Arch. 54, 6 (2008), 576–593.
26. Belmiro, T.R., Gardiner,P.D., Simmons, J.E.L., and Rentes, A.F. Are BPR practitioners really addressing business processes? Int. J. Oper. & Prod. Man. 20, 10(2000), 1183 – 1203.
27. Benjaoran, V. A cost control system development: a collaborative approach for small and medium-sized contractors. Int. J. Proj. Man. 27, 3(2009),270-277.
28. Bobbio, A., Portinale, L., Minichino, M and Ciancamerla, E. Comparing fault trees and bayesian networks for dependability analysis. LCNC, 1698/1999, (1999),689.
29. Bobbioa, A., Portinalea, L., Minichinob, M. and Ciancamer, E. Improving the analysis of dependable systems by mapping fault trees into Bayesian networks. Reliability Eng. & Sys. Safety, 71, 3(2001),249-260.
30. Boehm B. and Sullivan, K.J. Software economics: a roadmap. In Proceedings of the Future of Software Engineering, 22nd International Conference on Software Engineering. ACM, New York, 2000, pp.319-344.
31. Boehm, B. and Chris Abts, C. Software Development Cost Estimation Approaches – A Survey. Annals of Soft Eng.10, 1-4(2000), 177-205.
32. Boehm, B. and Huang, L**.G**, Value-based software engineering: a case study**,** Computer 36**,** 3(2003), 33 – 41.
33. Boehm, B. and Huang, L.G. Value-based Software Engineering: a Case Study. Computer 36, 3(2003), 33 – 41.
34. Boehm, B. and Turner, T. Management challenges to implementing agile Processes in Traditional Development Organizations. IEEE Software 22, 5(2005), 30 – 39.
35. Boehm, B. Value-based software engineering, ACM SIGSOFT Soft. Eng. Notes 28, 2(2003), 4.
36. Boehm, B. Value-based software engineering: reinventing. *ACM SIGSOFT Soft. Eng. Notes* 28, 2(2003), 1-12.
37. Boehm, B., Chulani, S., Verner, J., Wong, B. Fifth workshop on software quality. In Proceedings of the 29th International Conference on Software Engineering (ICSE'07 Companion). IEEE CS Press, 2007, pp.131 – 132.
38. Burge, J. E, Carroll,J.M., McCall, R and Mistrik, I. decision-making in software engineering, rationale-based software engineering, Springer Berlin Heidelberg, 2008.
39. Calder, M., Kolberg, M., Magill, E. H., and Reiff-Marganiec, S. Feature interactions: a critical review and considered forecast. Computer Networks, 41, 1(2003), 115-141.
40. Cao, L. and Ramesh, B. Agile requirements engineering practices: an empirical study, I E E E S o f t wa r e 25, 1 (2008), 60-67.
41. Cao, L. and Ramesh. B. Agile software development: Ad Hoc practices or sound

University of Cape Town

principles? IT Pro. 9, 2(2007), 41-47.
42. Charan, R. The game-changer: how you can drive revenue and profit growth with Innovation. Crown Business, New York, 2008.
43. Chulani, S., Clark B., and Boehm, B. Calibration Results of COCOMOII.1997. In Proceedings of the International Conference on Software Engineering. 1998.
44. Claude, B. and Hobbs, B. The perceived value and potential contribution of project management practices to project success. Proj. Man. J. 2006,
45. Clements, P., Northrop, L. Software Product Lines: Practices and Patterns. Addison-Wesley, NJ, 2002.
46. Clemmons, R.K. Project estimation with use case points. The J. of Defence Soft. Eng, (2006), 18 – 22.
47. Cockburn, A. Agile Software Development. Addison-Wesley, 2002.
48. Cohn, M. and Ford, D. Introducing an agile process to an organization. Computer 36, 6(2003), 74-78.
49. Coleman, G., and O'Connor, R.V. An investigation into software development process formation in software start-ups. J of Enter. Info.Man. , 21, 6(2008),633 – 648.
50. Conejero, J.M., and Hernández, J. Analysis of crosscutting features in software product lines. In Proceedings of the 13th international workshop on Software architectures and mobility (Leipzig, Germany). ACM, New York, 2008, pp. 3-10.
51. Cooke, D., Gelman, L., Peterson,W.J. ERP Trends. The Conference Board Survey Report, 2001.
52. Coopera K, Daia L, Dengb Y. Performance modelling and analysis of software architectures: An aspect-oriented UML based approach. Sci. of Comp. Prog.57, 1 (2005), 89–10.
53. Costa, H.R., Barros, M.O. and Travassos, G.H. A risk based economical approach for evaluating software project portfolios. In Proceedings of the seventh International Workshop on Economics-driven Software Engineering Research (St. Louis, Missouri). ACM, New York, 2005, pp.1-5.
54. Costa, R.H., O. Barros, M.O., and Travassos, G.H. Evaluating software project portfolio risks , J. Sys and Soft. 80, 1(2007), 16-31.
55. Croll P., Croll, J. Investigating risk exposure in e-health systems. Int. J. Med. Info. 76, 5-6(2007), 460 – 465.
56. Currie, W.L and. Parikh, M. A. Value creation in web services: An integrative model. The J. Strategic Info. Sys.15, 2(2006), 153-174.
57. Curriea, W.L. and Mihir A.ParikhbValue creation in web services: An integrative model. The J.of Strat.Info.Sys.15, 2(2006), 153-174.
58. Czuchry, A.J., and Yasin, M.M. Managing the project management process. Ind. Man. & Data Sys.103, 1 (2003), 39 - 46.
59. De Jong, J.P.J., and Hartog, D.N.D. How leaders influence employees' innovative behaviour. European J. of Inn. Man. 10, 1(2007), 41 – 64.
60. De Reyck, B., Grushka-Cockayne, Y., Lockett, M., Calderini, S.R., Moura l.M and Sloper, A. The impact of project portfolio management on information technology projects. Int. J. of Project Man. 23, 7(2005), 524-537.
61. De Souza, C.R.B., and Redmiles, D.F. An empirical study of software developers' management of dependencies and changes. In Proceedings of the 30th international conference on Software engineering(Leipzig, Germany). ACM, New York, 2008, pp.241-250.
62. Deng J, Seifert T, Vogel, S. Towards a Product Model of Open Source Software in a Commercial Environment. In Proceedings of the International Conference on Software Engineering (Portland ICSE'03, Oregon May 3-11). 2003.
63. Dey, P.K. Integrated project evaluation and selection using multiple-attribute decision-

University of Cape Town

making technique. Int. J of Prod. Econ.103, 1(2006), 90-103.
64. Diaz, J., Garbajosa, J and Calvo-Manzano , J.A. Mapping CMMI level 2 to Scrum practices: an experience report. Comm. in Comp. & Info. Scie., 42(2009),93-1-4.
65. Dingsøyr,T., Hanssen, G.K., Dybå, T., Anker, G. and Nygaard, J.O. Developing Software with Scrum in a Small Cross-Organizational Project. LNCS, 4257/2006 (2006), 5-15.
66. Distefano,S. and Puliafito, A. Dependability evaluation with dynamic reliability block diagrams and dynamic fault trees. IEEE Trans. on Dependable and Secure Computing, 6,1(2009), 4-17.
67. Do, H and Rothermel, G. Using Sensitivity Analysis to Create Simplified Economic Models for Regression Testing. In Proceedings of the 2008 international symposium on Software testing and analysis (Seattle, WA, USA)**.** ACM New York, 2008, pp. 51-62.
68. Dong, F., Li, M., Zhao, Y., Li, J., and Yang, Y. Software Multi-project Resource Scheduling: A Comparative Analysis. LNCS. 5007/2008 (2008), 63-75.
69. Dossani, R. Software engineering: globalization and its implications. In Proceedings of the Workshop on the Offshoring of Engineering: Facts, Myths, Unknowns and Implications (October 24-25, Washington DC). 2006.
70. Dyba, T. An empirical investigation of the key factors for success in software process improvement. IEEE Trans. on Soft. Eng.31, 5(2005). 410 – 424.
71. Eisenhardt, K. M. and Brown, S. L. Time pacing: competing in markets that won't stand still", Harvard Business Review, 1998, 3-4 (1998), 59-69.
72. Emam, K.E., and Koru, A.G. A replicated survey of IT software project failures. IEEE Soft.25, 5(2008), 84-90.
73. Erdogmus, H, The economic impact of learning and flexibility on process decisions, IEEE Software 22, 6(2005), 76 – 83.
74. Erdogmus, H., Boehm, B.W., Harrison, W., Reifer, D.J., and Sullivan, K. J. Software engineering economics: background, current practices, and future directions. In Proceedings of the 24th International Conference on Software Engineering (Orlando, Florida). ACM New York, 2002, pp.683 – 684.
75. Erdogmus, H., Favaro, J. and Strigel, W. Guest Editors' Introduction: Return on investment. IEEE Software 21, 3(2004), 18-21.
76. Fadi P. D., McHugh, J.A. M., and Eljabiri, O.M. Strategic Software Engineering: An Interdisciplinary Approach, Auerbach Publications, USA, 2005.
77. Favaro, J. Managing requirements for business value. IEEE Software19, 2(2002), 15-17.
78. Fenga, L.,Wang, W., Zhua, L.and Zhanga, Y. Predicting intrusion goal using dynamic Bayesian network with transfer probability estimation. J.of Network & Comp.App.32, 3(2009), 721-732.
79. Fenton, N.E, Krause P and Neil, M. Software measurement: uncertainty and causal modelling. *IEEE Software 10*, 4 (2002), pp.116-122.
80. Fidel, R., Scholl, H. J., Liu, S., and Unsworth,K. Mobile government fieldwork: a preliminary study of technological, organizational, and social challenges**.** In Proceedings of the *8th* AICPS: dg.o '07. Digital Government Society of North America**,** 2007, pp. 131-139.
81. Freimut B., Briand, L. C, and Vollei F. Determining inspection cost-effectiveness by combining project data and expert opinion. IEEE Trans. Soft. Eng.31. 12(2005), 1074–1092.
82. French, W.L. and. Bell C.H. Organization Development: Behavioral Science Interventions for Organization Improvement, sixth ed. Prentice-Hall, 1999.
83. Garcia, C.A.L and Hirata, C.M. Integrating functional metrics, COCOMO II and earned value analysis for software projects using Pmbok. In Proceedings of the 2008 ACM symposium on Applied computing(Fortaleza, Ceara, Brazil). ACM, New York**,** 2008, pp. 820-825.

University of Cape Town

84. Garlan, D. and Shaw, M. An introduction to software architecture. In V. Ambriola and G. Tortora, editors. Adv.in Soft. Eng.& Know.Eng.1,(1993), 1-40.
85. Gencel, C. and Demirors, O. Functional size measurement revisited. ACM Trans. Softw. Eng. Methodol. 7, 3 (2008), 36.
86. Gero, J. Research methods for design science research: computational and cognitive approaches. In Proceedings of ANZAScA, (Sydney), 1999.
87. Ghislain Levesque, G., Bevo, V., and Cao, D.T. Estimating software size with UML models. In Proceedings of the 2008 C^3S^2E conference (Montreal, Quebec, Canada). ACM, New York, 2008, pp.81 -87.
88. Gotel, O., and Finkelstein, A. An analysis of the requirements traceability problem. In Proceedings of International Conference on Requirements Engineering 1994, IEEE CS Press, 1994, pp.94-101.
89. Grace, P., Blair, G.S., Cortes, C.F., Bencomo, N. Engineering complex adaptations in highly heterogeneous distributed systems. In Proceedings of the 2nd International Conference on Autonomic Computing and Communication Systems (Turin, Italy). ICST, 2008.
90. Gruschke, T. Empirical studies of software cost estimation: training of effort estimation uncertainty assessment skills. In Proceedings of the 11th IEEE International Software Metrics Symposium. IEEE Computer Society Washington, 2005, pp. 48.
91. Guan H. Lim, Khai S. Leeb and Soo J. Tanc. Gray marketing as an alternative market penetration strategy for entrepreneurs: Conceptual model and case evidence. J. of Bus. Venturing, 16, 4(2001), 405-427.
92. Guan, J.C.,. Yam, R.C.M, Mok, C.K., and Ma, N. A study of the relationship between competitiveness and technological innovation capability based on DEA models. European J. Oper.Res. 170, 3(2006), pp. 971-986.
93. Harrison, W. Risk and the economic value of the software producer, Value-Based Soft. Eng. (2006), 91-105.
94. Harrison, W. What do software developers need to know about business? IEEE Software 22, 5(2005), 5- 7.
95. Heckerman, D, Mamdani, A and Wellman, M. Real-world applications of Bayesian networks. *Comm ACM, 38, 3(1995),* pp.25-26.
96. Heller, T. If only we'd kknown sooner: developing knowledge of organizational changes earlier in the product development process. IEEE Trans.On Eng. Man.47, 3(2000), 335 – 344.
97. Highsmith, J. Agile Software Development Ecosystems**.** Addison Wesley, 2002.
98. HighSmith, J.A. Adaptive Software Development: A Collaborative Approach to Managing Complex Systems. Dorset House, New York, 2000.
99. Hosbond, J.H. and Nielsen, P.A. Misfit or misuse? Lessons from implementation of Scrum in radical product innovation. Lecture Notes in Bus. Info. Processing, 9(2008), 21-31.
100. Hougham, M. London ambulance service computer-aided despatch system. Int. J.l of Project Man., 14,2(1996),103-110.
101. Hsua,C., Shihb, M., Huangc, B., Lina, B., and Lin, C. Predicting tourism loyalty using an integrated Bayesian network mechanism. Expert Sys. with App. 36, 9(2009), 11760-11763.
102. Huang, L. and Boehm, B. How Much Software Quality Investment is Enough: A Value-Based Approach. Technical Report, University of Southern California, 2005.
103. Hunt A. and Thomas, D. The trip-packing dilemma. IEEE Software 20, 3(2003), 106-107.
104. ITRD. Workshop on new visions for software design and productivity: research and applications Report. Nashville, Tennessee, 2001

University of Cape Town

105. Jackson M. Why software writing is difficult and will remain so. Info. Proc. Letters 88, 1-2(2003), 13–25.
106. Jamieson, K., and Hyland, P. Good intuition or fear and uncertainty: the effects of bias on information systems selection decisions. Informing Sci. J. , 9(2006), 49-69.
107. Jiamthubthugsin, W. and Sutivong, D. Resource decisions in software development using risk assessment model. In Proceedings of the 39th Hawaii International Conference on System Sciences – (HICSS'06). IEEE CS Press, 2006, pp.1-8.
108. Kan, S. H. Metrics and Models in Software Quality Engineering - 2nd Ed., Longman, Boston, 2002.
109. Karandikar, H. and Nidamarthi, S. Implementing a platform strategy for a systems business via standardization, J.of Manufacturing Techn. Man., 18, 3(2007), 267 – 280.
110. Karlsson, A., Johansson, R and. Andler, Sten. F. Imprecise probability as an approach to improved dependability in high-level information fusion. Advances in Soft Computing, 46/2008(2008), 70-84.
111. Karner, G. Resource estimation for objectory projects. Objective Systems SF AB, 1993.
112. Keck, D. O. and Kuehn, P. J. The feature and service interaction problem in telecommunications systems: a survey. IEEE Trans. on Soft. Eng.24, 10 (1998), 779-796.
113. Ken, P., Rothenberger, T.T., Marcus, A. and Samir, C. A design science research methodology for information systems research. J. of Man. Info. Sys., 24, 3(2008), 45 - 77.
114. Kettunen, P. Adopting key lessons from agile manufacturing to agile software product development—A comparative study. Technovation, 29, 6-7(2009), 408-422.
115. Khodakarami V, Fenton, N, and Neil, M. Project Scheduling: Improved approach to incorporate uncertainty using Bayesian Networks. *Proj. Manag. J* 39, 1(2008), pp.99 – 100.
116. Killen, C.P.,.Hunt, P.A., Kleinschemidt, EJ. The project portfolio management for product innovation. Int. J. of Quality & Reliability man., 25, 1(2008),24-38.
117. Kim, K., Yoo, J and Cho, S. Robust inference of bayesian networks using speciated evolution and ensemble. LNCS, 3488/2005, (2005), 92-101.
118. Kiper, J.D., and Feather, M.S . A risk-based approach to strategic decision-making for software development. In Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05) - Track 9. ACM New York, 2005, pp313c.
119. Koh, K., Kim, S. and Boyd, S. A method for large-scale _1-regularized logistic regression. In Proceedings of the 22nd national conference on Artificial intelligence (Vancouver, British Columbia). AAAI Press. 2007, pp.565-571.
120. Kutsch, E. The effect of intervening conditions on the management of project risk. Int. J. of Man. Proj. in Bus. 1, 4(2008), 602—610.
121. Kvale, S. The social construction of validity. In Interviews. Thousand Oaks, CA, 1996.
122. Lambrechts, O. Demeulemeester, E and Herroelen, W. Proactive and reactive strategies for resource-constrained project scheduling with uncertain resource availabilities. J. Sched.11, 2(2008), 121-136.
123. Lavazza, L., and Bianco, V.D. Functional size measurement based on problem frames: a case study. In Proceedings of the 3rd international workshop on Applications and advances of problem frames (Leipzig, Germany). ACM, New York, 2008, pp. 44 – 47.
124. Lawrie, T and Gace C. Issues of Dependability in Open Source Software Development. ACM SIGSOFT Soft. Eng. Notes. 27, 3 (2002), 34.
125. Lee, J., Kang, K.C, and Kim, S. Feature-based approach to product line production planning. LNCS, 3154 (2004), 183-196.
126. Lehman, M.M., FEAST - Feedback, evolution and software technology. Soft. Process Newsletter: SPN, 3. IEEE CS Press, 1995.

University of Cape Town

127. Lehtola, L., Kauppinen, M., and Kujala, S. Requirements prioritization challenges in practice, LNCS.3009/2004, (2004), 497-508.
128. Lhuillery, S. and Pfister, E. R&D cooperation and failures in innovation projects: Empirical evidence from French CIS data. Research Policy, 38,1(2009), 45-57.
129. Li, J., and Ruhe, G. Decision Support Analysis for Software Effort Estimation by Analogy. In Proceedings of the Third International Workshop on Predictor Models in Software Engineering. IEEE CS Press, 2007, pp.6.
130. Liao. Y. Business strategy and performance: the role of human resource management control. Personnel Review, 34, 3(2005), 294 – 309.
131. Lin, G.T.R. and. Tang, J.Y. H. Appraising intangible assets from the viewpoint of value drivers. J. of Bus. Ethics, 88,4(2008),pp.679-689.
132. Lindvall, *M. et al.* Agile software development in large organizations, **Computer** 37, 12 (2004), 26 - 34.
133. Linenberg Y, Stadlker Z, Arbuthnot S. Optimising organisational performance by managing project benefits. In Proceedings of the PMI Global Congress (Euorope). 2003
134. Linetsky, B.L. The project management paradox: achieving more by doing less. Ivey Business Journal. Available from http://Www.Iveybusinessjournal.Com/Article.Asp?Intarticle_ID=747; accessed on 10 June 2008.
135. Linton, J.D., Walsh, S.T., Kirchhoff, B.A., Morabito, J. and Merges, M. Selection of R&D projects in a portfolio. In Proceedings of the 2000 IEEE Engineering Management Society. IEEE CS Press, 2000, pp.506 – 511.
136. Lippert, M. *et al.* Developing complex projects using XP with extensions. Computer 36, 6 (2003), 67 – 73.
137. Liu, G., Liston-Heyes, C and Ko, W. Employee participation in cause-related marketing strategies: a study of management perceptions from British consumer service industries. J of Business Ethics. Springer Netherlands, 2009
138. Madachy, R. Simulation for business value and software process/product tradeoff decisions. In Proceedings of the 2006 international workshop on Economics driven software engineering research (Shanghai, China). ACM New York, 2006, pp. 25 - 30.
139. Maglogiannis, I., Zafiropoulosa, E., Platisa, A. and Lambrinoudakisa, C. Risk analysis of a patient monitoring system using Bayesian Network modeling. J. of Biomedical Informatics. 39,6(2006), 637-647.
140. Mantyla, M.V. Developing new approaches for software design quality improvement based on subjective evaluations. In Proceedings of the 26th International Conference on Software Engineering. IEEE CS Press, 2004, pp.48 – 50.
141. Marchenko, A., Abrahamsson,P. Scrum in a multiproject environment: an ethnographically-inspired case study on the adoption challenges. Agile,(2008),15-26.
142. Martin, N.L., Pearson, J.M., and Furumo, K.A. IS project management: size, complexity, practices and the project management office. In Proceedings of the 38th Annual Hawaii International Conference on System Sciences. IEEE CS Press, 2005, pp.234b - 234b.
143. Martin, S. Software project economics: a roadmap**,** In Proceedings of Future of Software Engineering. IEEE CS Press, 2007, pp.304 – 315.
144. Mazumder, R., Hastie, T., Tibshirani,R. Spectral Regularization Algorithms for Learning Large Incomplete Matrices.
145. McConnell, S. The business of software improvement. IEEE Software19, 4 (2002), 5- 7.
146. McGregor, J. D. Strategic Software engineering Journal of object technology. Vol. 8, No. 3, May-June 2009.
147. Mohagheghi, P., Anda, B. and Conradi, R. Effort estimation of use cases for incremental large-scale software development. In Proceedings of the 27th international

University of Cape Town

conference on Software engineering (St. Louis, MO, USA). ACM, New York, 2005, pp.303 - 311.

148. Montania, S., and Boudalib, H. Special session on Bayesian networks in dependability. Reliability Eng. & Syst. Safety., 93, 7(2008),921.

149. Morales, A., Barra, L. System development techniques for small and medium size installations. In Proceedings of the 15th SIGCPR. ACM, New York, 1977, pp. 241-247.

150. Mugridge, R. Managing agile project requirements with story test-driven development. IEEE Software 25, 1(2008), 68 – 75.

151. Murthi, S. Preventive risk management for software projects, IT Prof. 4, 5(2002), 9 – 15.

152. Nagappan, N., Ball, T. Use of relative code churn measures to predict system defect density. In Proceedings of the 27th ICSE`05. ACM, New York, 2005, pp. 284 –292.

153. Neapolitan, R. *Learning Bayesian Networks*. Prentice Hall : New Jersey, 2003.

154. Needham, C.J, Bradford, J.R., Bulpitt, A.J. and Westhead, D.R. *A primer on learning in bayesian networks for computational biology.* PLOSS Comp.l Bio., 3, 8(2007), 1553-7358

155. Nicolas, L and Marie-Laurence, C. Strategic scanning project failure and abandonment factors: lessons learned. Euro. J.of Info. Sys., 17, 4(2008),371-386.

156. Nigel, T., Penix, J. and Visser, W C. 2nd International Workshop on Automated Program Analysis, Testing, and Verification (WAPATV). Available from http://www.csr.uvic.ca/icse2001; accessed on 14th May 2007.

157. Nordqvista,S., Hovmark, S., and Zika-Viktorssonb, A. Perceived time pressure and social processes in project teams. Int. J. Project Man., 22 (2004), pp.463-468.

158. Ooms, M. Trends in Applied Econometrics Software Development 1985-2008. Palgrave MacMillan, U.K., 2009.

159. Ostrand, T., Weyuker, E. and Bell, R. Predicting the location and number of faults in large software systems. IEEE Trans. Softw. Eng. 31, 4(2005), 340–355.

160. Osunmakinde, I.O and Potgieter, A., Emergent Situation Awareness Using Temporal Probabilistic Reasoning, J. of Pattern Recognition, Article in Press.

161. Palanisamy, R. Strategic information systems planning model for building flexibility and success. Ind. Man. & Data Sys. 105, 1(2005), 63 – 81.

162. Park, M. Y. and Hastie, T. L1-regularization path algorithm for generalized linear models. J. Royal Stat. Society. Series B Stat. Meth. 69, 4(2007), 659-677.

163. Parnell, J.A., Carraher, S and Holt, K. Participative management's influence on effective strategic diffusion. J. of Business Strategies (2002).

164. Patton, J. Ambiguous business value harms software products, IEEE Software 25, 1(2008), 50 - 51.

165. Peppard, J. The conundrum of IT management. European J. Info. Sys, 16(2007), 336–345.

166. Pilone, D., Pitman, N. UML 2.0 In a Nutshell. O`Reilly Media. 2005

167. Polack, F. and Stepney, S. Emergent properties do not refine. In Proceedings of the REFINE 2005, BCS-FACS Refinement Workshop, ENTCS. 137, 2(2005), 163-181.

168. Raybould, B. Performance support engineering: an emerging development methodology for enabling organizational learning. Perf. Impro. Quart.8, 1(1995), pp.7-22.

169. Robinson, W. N., Pawlowski, S. D., and Volkov, V. Requirements interaction management. ACM Computing Surveys (CSUR), 35, 2 (2003), 132-190.

170. Ross, J.G. Risk and uncertainty in portfolio characterisation. J. Petr. Sci. & Eng. 44, 1-2 (2004), 41– 53.

171. Russell, S. and Norvig, P. Artificial Intelligence, A Modern Approach, 2nd Ed. Prentice Hall, New Jersey, 2003.

University of Cape Town

172. Saliu, M.O., Ruhe, G. Bi-Objective Release Planning for Evolving Software Systems. In Proceedings of the 6th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering. ACM Press, New York, 2007, pp.105-114.

173. Sanchez, R.J., Truxillo, D.M., and Bauer, T.N. Development and examination of an expectancy-based measure of test-taking motivation. J Appl Psychol. 85, 5(2000), 739-50.

174. Schätz, B. Combining product lines and model-based development. Elec. Notes in Theo. Comp. Sci. 82, 29 (2007), 171-186.

175. Schmidt, M., Niculescu-Mizil, A., and Murphy, K. Learning graphical model structure using L1- Regularization paths. In Proceedings of the 22nd Conference on Artificial Intelligence (Vancouver, British Columbia). MIT Press, Cambridge, 2007, pp. 1278-1283.

176. Schwaber, K. and Beedle, M. Agile software development with Scrum (Series in Agile Software Development). Prentice Hall, USA, 2001.

177. Shaban-Nejad, A., and Haarslev, V. Towards a framework for requirement change management in healthcare software applications. In Proceedings of the Conference on Object Oriented Programming Systems Languages and Applications (Montreal, Quebec, Canada). ACM, New York, 2007, pp. 807 – 808.

178. Shaw, M. Software Engineering for the 21st Century: A basis for Rethinking the Curriculum, Institute for Software Research International, School of Computer Science. Carnegie Mellon University, Pittsburgh, PA, 2005.

179. Sillitti, A., Ceschi, M., Russo,B., Succi, G. Managing uncertainty in requirements: a survey in documentation-driven and agile companies. In Proceedings of the 11th IEEE International Software Metrics Symposium (METRICS'05). IEEE Press, 2005, pp.17.

180. Silverman, D. Doing Qualitative Research: A Practical Handbook. Sage Publications, London 2005.

181. Silvius, A. J. G. Exploring differences in the perception of business & IT alignment. Comm. of the IIMA, 7, 2(2007b).

182. Singh, M. U-SCRUM: an agile methodology for promoting usability. Agile, (2008), 555-560.

183. Slater, S.F., and Olson, E.M. Marketing's contribution to the implementation of business strategy: an empirical analysis. Strategic Man. J. 22, 11(2001), 1055 – 1067.

184. Smaczny, T. IS an alignment between business and IT the appropriate paradigm to manage IT in today's organisation? Man. Dec.39, 10(2001).

185. Snow, A.P., Keil, M. The challenge of accurate software project status reporting: a two-stage model incorporating status errors and reporting bias**.** IEEE Trans on Eng. Manag. 49, 4(2002), 491—504.

186. Solomon, P.J. Integrating Systems Engineering with Earned Value Management. *Defense AT&L,* 2004.

187. Spradley, J.P. The ethnographic interview. Holt, Rinehart, and Winston, New York, 1979.

188. Stamelos, I. and Angelis, L. Managing uncertainty in project portfolio cost estimation. Info & Soft. Tech., 43, 13(2001), 759-768.

189. Standing, C., Guilfoyle,A., Lin, C., Love, P.E.D. The attribution of success and failure in IT projects. Ind. Man. & Data Sys., 106, 8(2006), 1148 – 1165.

190. Stefan B. Risk Management with enhanced tracing of requirements rationale in highly distributed projects. In Proceedings of the 2006 international workshop on Global software development for the practitioner (Shanghai, China). ACM New York, 2006, pp. 20 – 26.

191. Sterling, L. and Juan, T. The software engineering of agent-based intelligent adaptive systems. In Proceedings of the 27th International conference on Software Engineering (St.

University of Cape Town

Louis, MO). ACM, New York, 2005, pp.704 – 705.

192. Stijn V.D.V., Demeulemeester, E., and Herroelen, W. Proactive heuristic procedures for robust project scheduling: An experimental analysis. Euro. J. of Oper. Res. 189, 3(2008), 723–733.

193. Sullivan K. 7th International Workshop on Economics-Driven Software Engineering Research**.** In Proceedings of the seventh international workshop on Economics-driven software engineering research. ACM New York, 2005, pp.65

194. Sullivan K. First workshop on the economics of software and, computation. In Proceedings of the 29th International Conference on Software Engineering**.** IEEE CS, Washington. 2007, pp.125-126.

195. Sureshchandar, G.S., and Leisten R., A framework for evaluating the criticality of software metrics: an analytic hierarchy process (AHP) approach. Measuring Business Excellence 10, 4(2006), 22 – 33.

196. Thomas, G. and Fernándeza, W. Success in IT projects: A matter of definition? Int. J. of Project Man. , 26, 7(2008),733-742.

197. Tichy,F.W and Padberg, F . Empirical methods in software engineering research. In Proceedings of the 29th International Conference on Software Engineering (ICSE'07 Companion). ACM, New York, 2007, pp.163-164.

198. Vähäniitty, J. A tentative framework for connecting long-term business and product planning with iterative & incremental software product development. In Proceedings of the seventh international workshop on Economics-driven software engineering research (St. Louis, Missouri). ACM New York, 2005, pp. 1 – 4.

199. Vähäniitty, J. and Rautiainen, K. "Towards an approach for development portfolio management in small product-oriented software companies. In Proceedings of the 38th Hawaii International Conference on System Sciences (HICSS-38). IEEE CS Press, 2005, pp. 314c - 314c.

200. Verner, J., Cox, K.,Bleistein, S., Cerpa, N. Requirements engineering and software project success : an industrial survey in Australia and the U.S. AJIS. 13, 1(2005).pp.225-238.

201. Wainwright, M. J., Ravikumar, P. and Lafferty, J.D. High-dimensional graphical model selection using L1-regularized logistic regression. In Proceedings of Twentieth Advances in Neural Information Processing Systems (Vancouver). MIT Press, Cambridge, 2006, pp.1465-1472.

202. Wallin, C., Ekdahl, F., and Larsson, S. Integrating Business and Software Development Models. IEEE Software 19, 6(2002), 28 – 33.

203. Walter W. C. Chung, Yam, A.Y. K. and Chan M.F. S**.** Networked enterprise: a new business model for global sourcing. Int. J Prod. Eco.87, 3(2004), 267-280.

204. Wan, H. Belief logic programming with cyclic dependencies. LNCS, 5837/2009(2009), 150-165.

205. Ward, S., and Chapman, C. Transforming project risk management into project uncertainty management. Int. J. of Project Man. 21, 2(2003), 97-105.

206. Wateridge**,** J. The role of configuration management in the development and management of Information Systems/Technology (IS/IT) projects. Int J. Proj Manag. 17, 4(1999), 237—241.

207. Wheelwright, R. Critical path analysis in head and neck cancer: a management technique for surgical oncology. Euro. J Onco. Nursing 6, 3 (2002), 148 – 154.

208. Wickramatillake, C.D, Koh, S.C. L., Gunasekaran, A., and Arunachalam, S. Measuring performance within the supply chain of a large scale project Supply Chain Management. Supply Chain Man.; An Int. J. 12, 1(2007), 52 – 59.

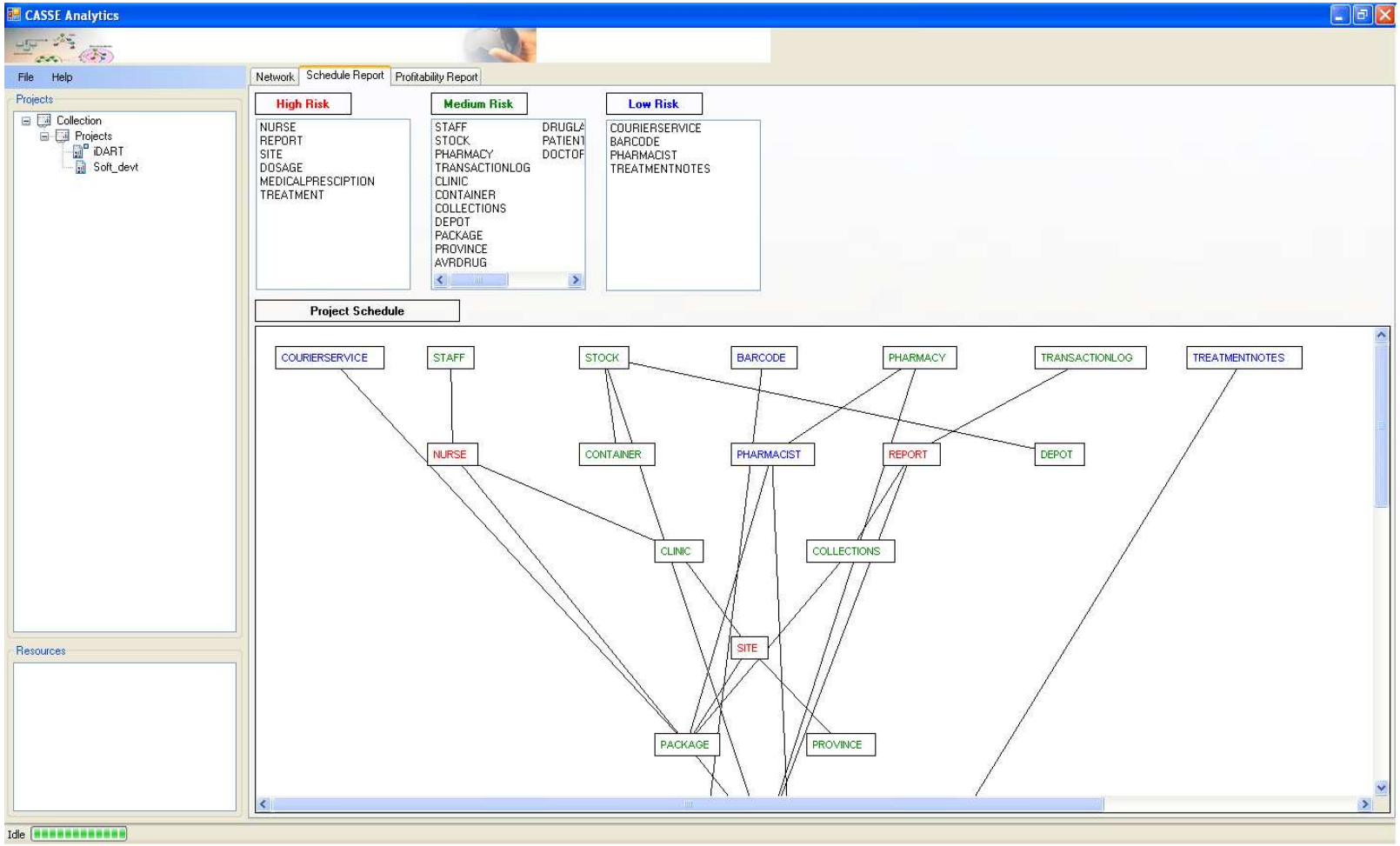
University of Cape Town

209. Wieczorek, I. Improved Software Cost Estimation – A Robust and interpretable modelling method and a comprehensive empirical investigation. Emp. Soft. Eng., 7(2002),177-180.
210. Wilton, D. A tale of two projects: why IT projects fail (and why they sometimes actually succeed). AJIS, 12, 2 (2005), 134-146.
211. Woit, D.M. Requirements interaction management in an eXtreme programming environment: a case study. In Proceedings of the 27th international conference on Software engineering (St. Louis, MO). ACM Press, New York, 2005, pp.489 – 494.
212. Wong, S. K. M. and Butz C. J. Constructing the dependency structure of a multiagent probabilistic network. IEEE Trans. on Knowledge & Data Eng., 13, 3(2009), 395-415.
213. Wortman, B. Value-changing behavior: getting the habit. J. Bus. Strat. 26, 4(2005), 38 – 45.
214. Xia, W., and Lee, G .Grasping the complexity of IS development projects. Comm. of the ACM. 47 , 5 (2004),68 – 74.
215. Xu, K., Zhang, Z. and Bhattacharrya, S. Profiling Internet Backbone Traffic: Behavior Models and Applications. In Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications (Philadelphia, PA). ACM, New York,2005, pp.169 - 180
216. Zafra-Cabeza, A., Ridao M. A., Camacho, E. F. Using a risk-based approach to project scheduling: a case illustration from semiconductor manufacturing. EJOR. 190(2008), 708—723.
217. Zhang, H., Li, H., Tam, C. Particle swarm optimization for resource-constrained project Scheduling. Int J Proj Manag. 24, 1(2006), 83—92.

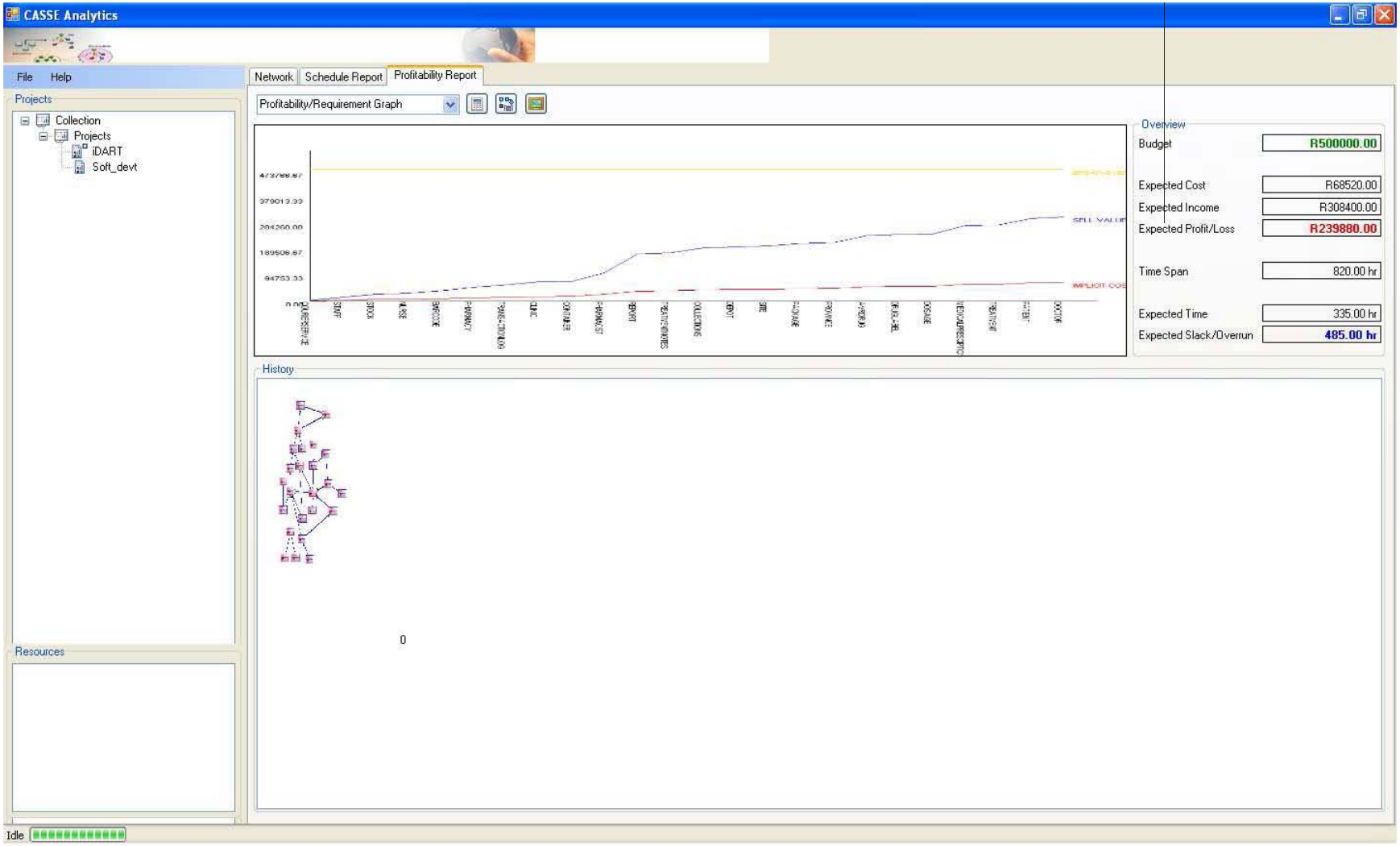
University of Cape Town

Appendixes

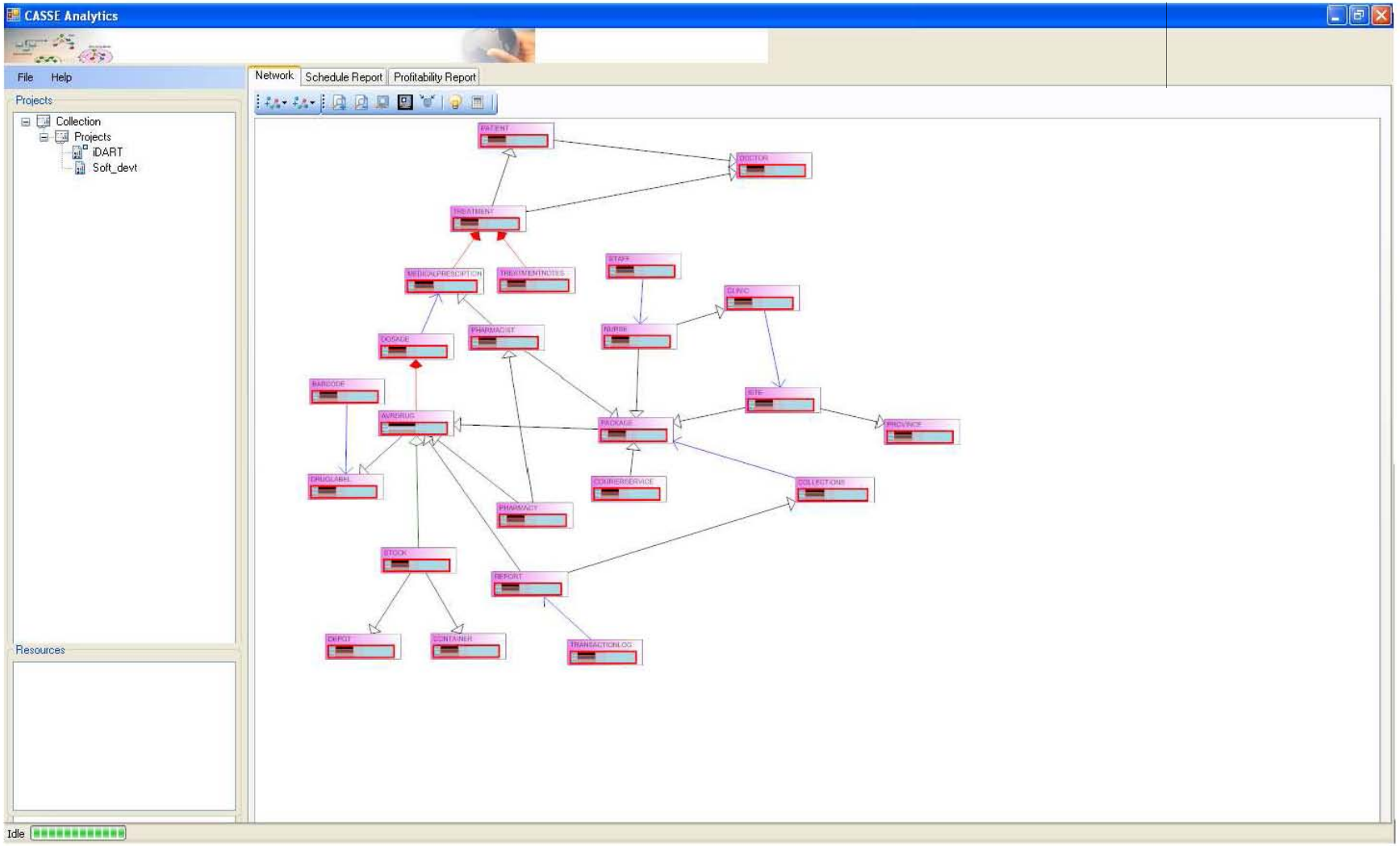
Appendix 1: Sample Screen Shots of CASSE



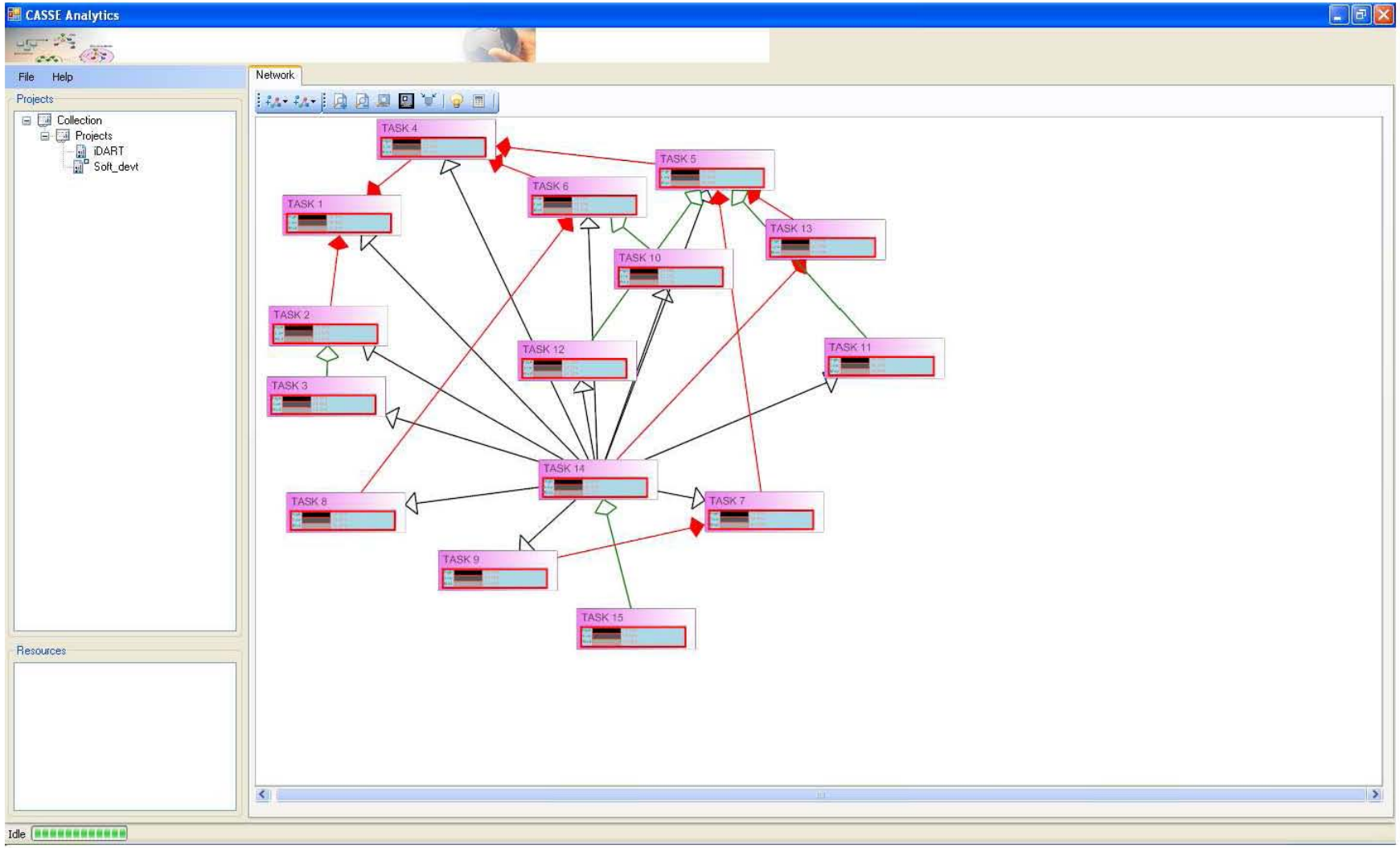
Resulting Actor Object Model in the CASSE Framework where each AO is ranked depending on the risk level in the model.



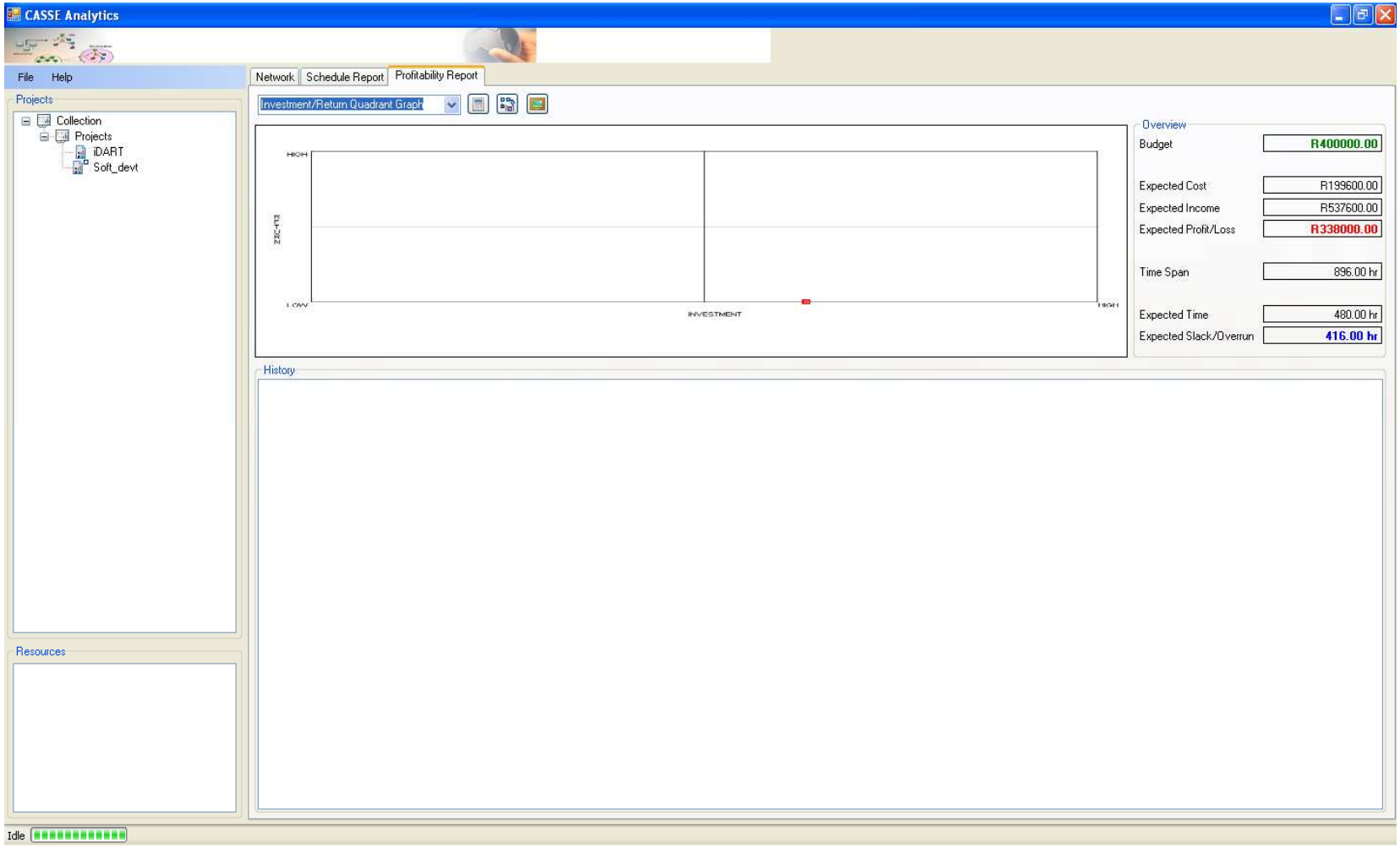
Each AO is scaled on the graph to show cumulative AO implementation time against cost



Imported UML model in the framework before dependency analysis is made



Imported UML model in the framework before dependency analysis is made for project tasks



Display of the project status on the value conundrum

Appendix 2: Request for Proposal Document

REQUEST FOR PROPOSAL

FOR

FURNISHING, DELIVERY, AND INSTALLATION

OF

POLICE RECORDS MANAGEMENT SYSTEM

FOR

JOLIET JUNIOR COLLEGE

OPENING DATE: MARCH 25, 2008

TIME: 2:00 PM, CDT

University of Cape Town

Dated: 3/07/08

INTRODUCTION

A. History of the College

Joliet Junior College is a comprehensive community college. The college offers pre-baccalaureate programs for students planning to transfer to a four-year university, occupational education leading directly to employment, adult education and literacy programs, work force and workplace development services, and support services to help students succeed.

JJC, America's oldest public community college, began in 1901 as an experimental postgraduate high school program. It was the "brain child" of J. Stanley Brown, Superintendent of Joliet Township High School, and William Rainey Harper, President of the University of Chicago. The college's initial enrollment was six students. Today, JJC serves more than 13,000 students in credit classes and 17,000 students in non-credit courses. JJC has approximately 1,000 full-time and part-time staff.

B. Calendar

The College expects to adhere to the following schedule in undertaking the selection process; however, the selection of the successful Vendor is subject to the approval of the College Board of Trustees, and this schedule is subject to modification at the sole and absolute discretion of the College:

Request for Proposal Issued:	March 7, 2008	
Deadline for Proposers' Questions:	March 14, 2008	10:00 a.m.
Distribution of RFP Amendments:	March 17, 2008	
Proposal Opening Time:	March 25, 2008	2:00 P.M., CDT
Anticipated Announcement of Award:	April 9, 2008	

Inquires and Proposer questions concerning the RFP must be submitted in writing or via email to:

Ms. Judy Mitchell, Director, Business & Auxiliary Services Joliet Junior College
1215 Houbolt Road
Joliet, Illinois 60431
Telephone: 815-280-6640 Fax:
815-280-6631 Email:
jmitchel@jjc.edu

Oral communication with any College employee or representative is not binding on the College and in no way shall modify the RFP or the obligation of the College or the Proposer.

University of Cape Town

INFORMATION FOR RESPONDENTS AND GENERAL CONDITIONS

1. Proposals:

   a. All proposals MUST BE SUBMITTED with one (1) original and two (2) copies, and must be enclosed in a sealed envelope plainly marked "Proposal for Furnishing, Delivery, and Installation of Police Records Management System, and addressed, mailed and delivered to Judy Mitchell, Director of Business & Auxiliary Services, Joliet Junior College, 1215 Houbolt Road, Joliet, IL 60431. A mailing label is enclosed for your convenience. Respondents shall be responsible for actual delivery of proposals during business hours at the above address, and it shall not be sufficient to show that a proposal was mailed in time to be received before scheduled closing time for receipt of proposals.

   b. All respondents shall include prices in their proposal and shall assume and pay all taxes and contributions, State, Federal and Municipal which are payable by virtue of the furnishing and delivery of item(s) specified herein. Materials and services furnished the College are not subject to either the Federal Excise Tax or Illinois Sales Tax.

   c. The College reserves the right to waive informalities in proposals and to reject any and all proposals which may be in the best interests of the College.

2. Modification or Withdrawal of Proposals:

   Any respondent may withdraw their proposal at any time prior to the scheduled closing time for the receipt of proposals but no respondent may withdraw their proposal for a period of ninety (90) days after the scheduled closing time for the receipt of proposals. Only written requests for the modification or corrections of a previously submitted proposal which are addressed in the same manner as proposals and are received by the College prior to the scheduled closing time for the receipt of proposals will be corrected in accordance with such written requests; provided that any such written request is to be contained in a sealed envelope which is plainly marked "Modifications of Proposal on Police Records Management System." Oral or telephonic modifications or corrections of proposals cannot be submitted after the scheduled closing time for the receipt of proposals.

3. Statement of Respondent's Qualifications:

   a. Each respondent submitting a proposal shall present evidence of experience, qualifications, financial responsibility and ability to carry out the terms of the contract by completing and submitting with proposal the schedule of information set forth in the form furnished with the proposal form.

   b. Such statement, single copy required, will be treated as confidential information by the College.

4. Indemnification:

   The College shall not accept any contract provisions which require the College to indemnify another party. Any indemnity language in proposed terms and conditions will be modified to conform to language that the College is able to accept.

5. Award of Contract:

   In awarding the contract, the College may take into consideration relevant factors, including but not limited to the skill, facilities, capacity, cost of and compatibility with existing systems and

University of Cape Town

operations when major changes of practice would be required, experience, ability, responsibility, previous work, the financial standing of the bidder(s); the amount of other work being carried on by the bidder; the quality, efficiency, and construction of equipment proposed to be furnished; the period of time within which the equipment or service is to be provided, and necessity of prompt delivery of the items herein described. The inability of any bidder to meet the requirement mentioned above may be cause for rejection of their proposal.

6. Execution and Form of Contract:

The successful respondent shall, within ten (10) days after the receipt of formal notice of award of the contract, enter into a written contract, prepared by the College.

7. Assignment of Contract:

The contract to be awarded and any amount to be paid there under shall not be transferred, sublet or assigned without the prior approval of the College.

8. Patent and Copyright:

a. The Contractor and its Surety shall pay for all royalties, license fees and patent or invention rights or copyrights and defend all suits or claims for infringements of any patent or invention right or copyrights involved in the items furnished hereunder.

b. The Contractor and its Surety shall hold and save the College and its officers, agents, servants and employees harmless from liability of any nature or kind, including cost and expenses, for, or on account of, any patented invention, process, article or appliance furnished in the performance of the contract, including its use by the Owner, unless otherwise specifically stipulated.

c. Copyrights for any item(s) developed for the College shall be the property of the College and inure to its benefit, and Contractor shall execute such documents as College may require for the perfection thereof.

9. Confidentiality of Information:

All data made available to the Contractor by the College are and remain the property of the College and must be treated as confidential information. All listings which may reveal names or identification numbers of individuals or employees, etc., if not returned to the College, must be properly destroyed so as to keep such information confidential.

All records received from a Contractor will be deemed public records and presumed to be open. If the contractor submits with the proposal any information claimed to be exempt under the Illinois Statutes, this information must be placed in a separate envelope and marked with:

> "This data shall not be disclosed outside the College or be duplicated, used, or disclosed in whole or in part for any purpose other than to evaluate the proposal; however, if a contract is awarded to this Contractor as a result of or in connection with the submission of such information, the College shall have the right to duplicate, use, or disclose this information to the extent provided in the contract. This restriction does not limit the College's right to use information contained herein if it is obtained from another source."

12. Vendor Audit:

University of Cape Town

In the event that time and materials are a portion of this proposal, the College reserves the right to audit vendor's records concerning this proposal.

13. Right of College to Terminate Contract:

In the event any provisions of contract are violated by Contractor, the College may serve written notice upon Contractor and Surety setting forth the violations and demanding compliance with the contract. Unless within ten (10) days after serving such notice, such violations shall cease and satisfactory arrangements for correction be made, the College may terminate the contract by serving written notice upon the Contractor; but the liability of Contractor and Surety for such violation; and for any and all damages resulting there from, as well as from such termination, shall not be affected by any such termination.

14. Delivery:

Delivery shall be accomplished in the manner and to the place and at the time required of the Detailed Specifications.

15. Payment:

Payment will be made at the time and in the manner as specified in the Detailed Specifications.

The College may withhold payment or make such deductions as may be necessary to protect the College from loss or damage on account of defective work, claims, or damages, or to pay for repair or correction of existing equipment..

16. Any agreement which results from the RFP will be governed and construed accordingly to the laws of the State of Illinois. This agreement is performable in Will County. Any disputes arising from the agreement shall be adjudicated in the Circuit court of the 12th Judicial Circuit, Will County, Illinois.

17. Detailed Specifications and Special Conditions:

Detailed Specifications and Special Conditions should be consulted for additional provisions and amendments to the General Conditions.

18. Proposal Documents:

The Proposal Documents shall include Introduction, Advertisement for Proposals, Specifications and Addenda, Exhibits, Response Time and Proposal.

University of Cape Town

DETAILED SPECIFICATIONS AND SPECIAL CONDITIONS

1. GENERAL

Joliet Junior College proposes to contract, hereinafter referred to as "College" with the successful vendor (referred to as "Contractor") to provide a Police Records Management System as described herein.

2. OBJECTIVE

The Campus Police department of Joliet Junior College seeks proposals to acquire and implement a comprehensive computerized police records management system to fulfill the College's need for documentation and management of its documents. Systems proposed by Respondents shall be the latest version, state of the art design and have a proven record of usage in other law enforcement agencies.

4. CONTRACT PERIOD

The contract shall be awarded as a one-time purchase of the software, installation and training. Additionally, the contract shall be automatically renewable for additional one-year periods to cover the annual maintenance, support and license fees, unless canceled by either party in writing with 60 days notice. All original terms and conditions shall apply for the renewal periods.

5. AWARD OF CONTRACT

Proposals will be awarded based upon the lowest cost and the criteria set forth under the Evaluation section of the proposal.

Joliet Junior College may award on the basis of the proposals initially submitted, without discussion, clarification or modification, or on the basis of negotiation with any of the Proposers or, at Joliet Junior College's sole option and discretion.

Joliet Junior College may discuss or negotiate all elements of the proposal with selected Proposers that represent a competitive range of proposals. For purposes of negotiation, Joliet Junior College may establish a competitive range of acceptable or potentially acceptable proposals comprising the highest rated proposals. After the submission of a proposal, but before making an award, Joliet Junior College may permit a Proposer to revise its proposal in order to obtain the Proposer's best final offer. In conducting such discussions, Joliet Junior College shall, to the extent permitted by law, not disclose any information derived from the proposals submitted by competing Proposers. If Joliet Junior College elects to accept a proposal, it will award the Proposer whose proposal provides the best value to the College, as determined by Joliet Junior College, based upon a thorough evaluation of all factors set forth in this RFP. The College Board of Trustees retains the right to modify the terms and conditions in the Proposal for

the records management system or reject terms and conditions proposed by the successful Proposer prior to the execution of the contact if it, in its sole discretion, deems necessary to ensure a plan satisfactory to it.

6. RESPONDENT'S PROPOSAL

In order to be considered for selection, respondents must submit a complete response to this

University of Cape Town

Request for Proposal that complies with all the mandatory requirements as listed in sections 10 thru 16.

7. DELIVERY

All deliveries shall be FOB: Destination with all freight charges thereto included and fully prepaid. The seller bears and pays the freight costs.

Any package shipped under any order resulting from this RFP shall clearly reference the purchase order number and the equipment item name of which the package is a component. Delivery time shall be stated in the proposal response and may be taken into consideration in the award. Please state an accurate and realistic delivery and installation time frame from receipt of an order as requested.

8. PAYMENT

All proposals to be submitted on the basis of payment by College check, terms Net 30. Payment shall be made within thirty- (30) days after receipt of properly certified vendor's invoices, rendered in duplicate, as follows. The College may withhold payment or make deductions as may be necessary to protect the College from loss or damage on account of defective work, claims, damages, or to pay for repair or correction of products/services furnished hereunder. **Any different payment terms desired by the respondent must be clearly stated and may or may not be accepted by the College**. A payment schedule may be negotiated with the successful respondent.

9. PROPOSAL EVALUATION

From the total information requested below, determination shall be made of the prospective contractor's ability to furnish, deliver and install Police Records Management System to the College. All information requested below must be included as a part of the prospective contractor's proposal.

9.1 Evaluation of each proposal shall be made based on the following criteria:

Cost (40 points):
Based upon initial and long-range costs of the system to include purchase price, installation, expansion, operation, training and maintenance costs.
Contractor Experience and Qualifications (10 points)
Implementation Plan (10 points)
Training (5 points)
Warranty/Support (5 points)
Required Specifications (20 points)
Optional Requirements (10 points)

9.2 Since respondent must comply with ALL mandatory requirements, proposals WHICH DO NOT MEET THE MANDATORY REQUIREMENTS, WILL BE ELIMINATED.

9.3 The College reserves the right to reject any and all proposals submitted.

9.4 Proposals shall remain open and subject to acceptance for 90 days from the date of proposal opening. During this period, respondents may not make material modifications, corrections, or changes (including pricing) to their proposal.

University of Cape Town

PROPOSAL REQUIREMENTS

10. MANDATORY REQUIREMENTS

Police Records Management System for the Joliet Junior College Police Department must be installed by June 30, 2008
System must be web based and operate on a Windows platform. Vendor must provide required hardware specs.
Vendor must be able to convert data from existing, in house, Microsoft Records System
System must have a CAD system as an optional feature.
System must be capable of accepting computer generated case reports.
System must be capable of customizing fields, drop boxes and reports to meet customer's needs.
System must be capable of storing scanned documents in electronic case files. System must have secure modules and sub modules.
System must have time stamps and user ID to show by whom and when report was edited
System must have spell check
System must be able to minimally input and retrieve data in the following fields.

CASE REPORTS

o Name and full demographic data on persons o Addresses
o Mug shots & evidence photo electronic uploads & attachment to report files
o Nicknames
o Identifying marks o Gang information
o SID & other identifying numbers o Emergency contacts
o Cautions
o Vehicle information including make, model, registration information
o Court Dispositions
o Customizing of data boxes must be included
o Must be able to add supplemental reports and edit reports
o Must have ability to establish secure user and supervisor approval groups
o Must have a Property & Evidence module

TICKETS / TRAFFIC

- o All fields on traffic ticket including issuing officer, type of offense, location, vehicle information, disposition
- o Same as above for parking tickets and also generate automated late notices/letters.
- o Must also be able to support department custom parking tickets.
- o Automated Stop Card data entry and upload to IDOT

REPORTS / SEARCHES

Must be able to produce automated reports in the following areas o Customized Reports as needed at no additional cost
o Types of Crime
o Monthly crime / incident reports by area or location o Person or vehicular descriptions
o Stop Card data (Racial profiling) o Tickets
o Student Code Referrals o Warrants
o Recovered/ Found Property o Juvenile arrests
o State UCR reports

University of Cape Town

o Clery Act annual reports
o Will County transfer listing report o Daily Log

11. PRICING

- List the base price of your system
- Provide a detailed summary of the capabilities and features / modules included in the base price of the system.
- List all optional features / modules that are available with your system.
  - o Include costs for each option. o Include cost expiration
  - o Include how cost increase would be calculated in the future

12. WARRANTY

- List duration of warranty and coverage
- List any cost or fees during warranty period

13. LICENSING

- How many licenses are included in basic cost?
- How many workstations/access points permitted per license?
- Are separate licenses needed for Mobile Data Terminals?
- Are separate licenses needed for remote campuses?
- What is the cost for additional licenses not included in the basic cost?
- Are license fees included in the annual maintenance contract cost?

14. ASSURANCE OF SERVICE

- Annual Maintenance / Service Contract

Contract must include version updates, new releases enhancements; repair of system problems, customization of reports, drop boxes and creation of new reports.

- List cost of Annual Maintenance / Service Contract
- How many years will this price be valid for?
- What is the cost escalation process after that?
- Detail what is included in contract.
- List any repairs not covered in the service contract.
- List labor costs not included in the service contract.
- Onsite repairs or trouble shooting. Regular Business Hours vs. After Hours & Weekends
- Remote repairs or trouble shooting Regular Business Hours vs. After Hours & Weekends
- Training on new updates Regular Business Hours vs. After Hours & Weekends
- What is your service response time?

16. INSTALLATION

- Installation of the system must be completed by June 30, 2008
- List any travel and/or housing costs for installers or trainers

Labor Costs

Installation: $ per hour_____x estimated # of hours_____ = total cost

University of Cape Town

Training: $ per hour_____x estimated # of hours_____ = total cost

Data Conversion $ per hour_____x estimated # of hours_____ = total cost

16. TROUBLE SHOOTING/IMPLEMENTATION

- Will vendor provide personnel on site during roll out and implementation period?
- Cost per hour / per day for this service
- Number of staff assigned

CERTIFICATION OF CONTRACT/BIDDER

The below signed contractor/bidder hereby certifies that it is not barred from bidding on this or any other contact due to any violation of either Section 33E-3 or 33E-4 of Article 33E, Public Contracts of the Illinois Criminal Code of 1961, as amended. This certification is required by Public Act 85-1295. This Act relates to interference with public contracting, bid rigging and rotating, kickbacks and bribery.

_____
Name of Contractor/Bidder

_____
Title

_____
Date

University of Cape Town

THIS FORM **MUST** BE RETURNED WITH YOUR PROPOSAL TO:

Joliet Junior College
Illinois Community College District #525
Director of Business & Auxiliary Services, H-1018
1215 Houbolt Road
Joliet, IL

Appendix 3: Questionnaire

Section 1: Role and Professional experience

1. Which category of staff do you belong?
   - Project Manager ☐
   - Systems Analyst ☐
   - Business Analyst ☐
   - Other ☐ ____
2. How long have you been in this position?
   - 1-3 years ☐
   - 4-6 years ☐
   - 7-9 years ☐
   - 10 and above ☐
3. In this section, please note the extent of your agreement with the following statements/questions on a scale provided by ticking the appropriate box.

Section 2: Product lifecycle optimization and management

2.1 Project selection

Ref	Assertion	Response				
		Strongly Disagree	Disagree	Not sure	Agree	Strongly Agree
		1	2	3	4	5
PS.1	Our projects are selected based on strategic business objectives					
PS.2	We choose our projects based on profitability level anticipated					
PS.3	Our projects are in-house and my organization is the direct recipient					
PS.4	We depend on feasibility assessment reports to determine projects implementation					
PS.5	We use analytical tools that guide us in project selection					
PS.6	Our project selection is purely heuristic					
PS.7	We just take on any project we get hold of					
PS.8	Our projects are for research and development					

University of Cape Town

2.2 Requirement Engineering

Ref	Assertion	Response				
		Strongly Disagree	Disagree	Not sure	Agree	Strongly Agree
		1	2	3	4	5
RE.1	We usually first understand client requirements before project implementation					
RE.2	We implement the requirements in parallel with requirement elicitation and validation					
RE.3	We undertake systems modeling using OOSA using UML etc					
RE.4	We use a standardized process in collecting, documenting and validating our requirements					
RE.5	We usually develop a traceability matrix for project requirements					
RE.5.1	The traceability matrix is generated using a matrix generation tool					
RE.5.2	The traceability matrix is generated manually					
RE.5.3	We don't consider the traceability formulation important because of lack of time					
RE.6	It is part of our development procedure to undertake a requirement dependency assessment for risk assessment					
RE.6.1	There is generally less or no requirement dependency assessment undertaken					

University of Cape Town

2.3 Project implementation and monitoring

Ref	Assertion	Response				
		Strongly Disagree	Disagree	Not sure	Agree	Strongly Agree
		1	2	3	4	5
PMI.1	A strict change management process always guides us in our project implementation.					
PMI.2	We often use tools that help us in understanding the change control process.					
PMI.3	We often evaluate our profit and risk levels (including systems overhauls) at any point during the implementation cycle.					
PMI.4	We have various techniques of analysis risk factors on the project during implementation especially when it comes to					

	requirements and their interrelationships.					
PMI.5	I am confident that our change control process results in good project management and scope creep bottleneck elimination.					
PMI.6	We document all our processes and changes during the development lifecycle					

2.4 Project completion and signoff

Ref	Assertion	Response				
		Strongly Disagree	**Disagree**	**Not sure**	**Agree**	**Strongly Agree**
		1	**2**	**3**	**4**	**5**
PCS.1	We always hold project review meetings so as to optimize our development process.					
PCS.2	We conduct post-project reviews internally for all projects and often with our clients to ensure that successful approaches and solutions are documented and shared.					
PCS.3	Any changes provied after product roleout are often fed back into the original model to determine maintainance scope.					

4. Comments and Recommendations on how software development processes can be improved using economic model, tools and other techniques.

……
……
……
……
……
……
……
……
……

Thank you for taking time to complete this questionnaire.

University of Cape Town