Exploring and Measuring Software Development success/failure rates in South Africa from two perspectives.

By

Name	:	Jeanine van der Gugten
Student Number	:	3377-080-8

This dissertation is submitted in partial fulfilment of the requirements of the Masters degree of Business Leadership

At

UNISA School for Business Leadership.

November 2004

ACKNOWLEDGEMENTS

First, I would like to praise God Almighty, my Father in Heaven: "I can do all things through Christ who strengthens me" (Philippians 4:13).

I wish to thank the following people in appreciation for their contribution to the successful completion of my MBL:

- My loving partner, Monte Hague, who stood by my side and supported me through this long journey.
- The software development companies and their clients for their participation.
- My study group, PTAO6O2A members, for their support and encouragement.

My sincere thank you goes to Ansia Daniel for her encouragement towards my enrolment for this degree, and her support and being my study-partner for the first two years of the MBL Course.

Finally, I wish to extend my gratitude to my academic leader, Prof. P.D. Rwelamila, whose suggestions, guidance and encouragement were invaluable to me.

The clients and development perspective also matches and correlate, which means that they do not differ and therefore hypothesis three is rejected, which states that *"Development teams and clients perspectives differs when it comes to defining success of software applications".*

The Clients survey analysis do emphasise the following areas that software development companies need to investigate and improve upon: Business fit, application usage, changes and value provided. The Development teams' survey analysis indicates that the following areas internally to the software development companiesshould focus and improve on: Applicability, usability, functionality and support service. There are also certain factors that the software development companies report as lacking on the clients side, such as: Articulation of their requirements, ability to constrain changes, their understanding of the technologies used, their trust and respect towards the software development staff. These aspects could be addressed by the software development companies by applying better project management principles and communicating, educating and training the clients on their processes, technologies and change management procedures.

The sample size was focused on small software development companies, which means that this result might not necessarily be applicable to larger software companies and it would be interesting to do the same research on larger software companies.

3.1 Life-Cycle Phases 41

3.1.1 Introduction 41

3.1.2 Engineering and production stages 42

3.2 Artefacts of the Process 43

3.3 Model-Based Software Architectures 45

3.3.1 Introduction 45

3.3.2 Architecture: A management perspective 45

3.4 Workflows of the Process 46

3.4.1 Software process workflows 47

3.5 Checkpoints of the process 47

3.6 Conclusion 48

4 SOFTWARE MANAGEMENT DISCIPLINES 49

4.1 Introduction 49

4.1.1 Work breakdown structures (WBS) 49

4.1.1.1 Conventional WBS Issues 49

4.1.1.2 Evolutionary WBS 50

4.2 Project Organisations and Responsiblities 51

4.3 Process Automation 52

4.4 Project Control and Process Instrumentation 52

4.5 Tailoring the Process 53

4.6 Conclusion 53

5 RESEARCH DESIGN STRATEGY 54

5.1 Sample Selection 54

5.2 Data Collection 54

6 DATA ANALYSIS AND INTERPRETATION 57

6.1 General Assumptions and Limitations 57

6.2 Hypothesis One – Clients Perspective 58

TABLE OF FIGURES

Figure 1.1 : Agency Theory Model (Mahaney & Lederer, 2003) 14

Figure 2.1 : The Waterfall model (Royce, 1998) 31

Figure 2.2 : The top five principles of a modern process (Royce, 1998) 39

Figure 3.1 : Details of the Artefact sets (Royce, 1998) 44

Figure 3.2 : Cycle focus on the artefact sets (Royce, 1998) 44

Figure 6.1 : Client Survey Hierarchical Structure and Percentage Weightings Allocation 60

Figure 6.2 : Development Survey Hierarchical Structure and Percentage Weightings Allocation 73

Figure 6.3 : Client and Development Mapping Diagram 87

Figure 6.4 : Simplified mapping with new weighting allocations 89

Figure 7.1 : Client Rating Summary 114

Figure 7.2 : Development Rating Summary 116

Figure 7.3 : Client and Development comparison 118

Figure 7.4 : Client view mapped to literature 120

Figure 7.5 : Development view mapped to literature 121

Figure 7.6 : Client and Development view mapped 122

Table 6.20 : Open-ended responses of development team of Company 4 110

Table 7.1 : Factors of each company that are below the total client score 115

Table 7.2 : Factors of each company that are below the total development score 117

Table 7.3 : Data table of the client and development comparison 118

By thoroughly understanding the motivations of stakeholders and adopting a customer-focused approach, a project manager can begin to effectively manage expectations and reduce overall stakeholder dissatisfaction (Boyd, 2001).

1.1.2 Reasons for software development projects failing

According to the literature reviews there are various reasons why software development projects fail, for example lack pf clear goals and objectives, lack of user involvement, insufficient top management support, lack of proper project management to control scope creep, just to mention a few.

The most recent figures available on the success rates of large IT projects published by the UK Standish Group, indicates that only 28 percent are successful, the remainder being either abandoned, delivered late, over budget, or with lower functionality than was initially envisaged.

NZIM (2003) have two interesting findings:

- **IT projects are risky because software is inflexible and fragile.** Software's inflexibility means that even slight modifications in function may require major design modifications. Being fragile means that the more times computer programs are modified, the more likely they are to fail and need further modification.
- **Most of the problems associated with project failure aren't technical.** Instead they concern the softer issues about how the project is set up and managed. Technical features may make IT projects difficult but it is inappropriate management that makes them fail. Too often projects are undertaken without sufficient thought about their aims or because senior managers are seduced by consultants into adopting cutting-edge technology that may be totally inappropriate.

Agency theory is another potential explanation for the high failure rate of software development projects. An agency relationship governs all employment settings. A manager is a principal and an employee is an agent. A problem—referred to as the agency problem—arises when the agent does not work entirely on the principal's behalf. As a result, the work of the agent may conclude less successfully than it otherwise would (Mahaney and Lederer, 2003).

- **Organizations fail to manage goal conflict, shirking, and privately held information well enough.** Incentives systems do not ameliorate these problems, and the problems produce delays and overruns.
- **Organizations do not effectively employ task programmability techniques.** Perhaps contemporary techniques are ineffective, or perhaps organizations simply do not use them well enough.

It has often been argued that systems' failure may result from **inadequacies in the treatment of organizational issues**, but there is little consensus on what is meant by the term inadequate treatment and how it influences systems' success. The results of a statistical analysis indicate that higher levels of systems' success are associated with treating a wide range of organizational issues throughout the development process and ensuring that members of the user community are actively involved (Doherty, King & Al-Mushayt, 2003).

Project failures may come from **lack of resources, inadequate funding, poor performance measures, or buggy software**. In many cases, technology is the easy part. The tough part is how you apply it to your business, optimize your processes, find new levels of collaboration, reduce risk, become more competitive, please your customers more, and constantly seek innovation (Stahl, 2003).

In the Engler survey of information systems executives, the reasons cited for project failure was:

- **Lack of user involvement** – primary reason.
- **Lack of top executives support**
- **Lack of clear statement of business objectives.**

Yoon, Guimaraes, and O'Neal investigated eight factors in expert systems success, and also concluded that user involvement, top management support, and skill development were all important in project success (Olson, 2004:12).

Miller (2001) reported that the British government had wasted 'more than £1 billion in high profile government IT failures' since the mid-1990's. From the delayed and scaled back Denver International Airport baggage handling system to the failed automation of the London Ambulance Service system, the

- Continued involvement
- Support from top

3. User Involvement
    - Client consultation
4. Project Manager
    - Competence
    - On-site
5. Other Factors
    - Project team
    - Manpower
    - Accurate estimates
    - Control consultants
    - Testing
    - Training
    - Problem recognition

Glass (1998) concludes his study of 16 case examples by further citing the KPMG research to reveal that companies are looking to mitigate project risk by:

- Improving project management;
- Conducting feasibility studies;
- Get users more involved;
- Seeking external advice.

Three factors consistently appeared as success factors in studies (Olson, 2004:12):

- **Client involvement**
- **Top management support**
- **Clear statement of project objectives**

According to Darlene Fichter (2003), president of Northern Lights Internet Solutions, the ten keys to success are:

1. Clearly defined objectives
2. Good planning and control methods
3. Excellent project management
4. Management support
5. Adequate time and resources
6. Commitment by all
7. High user involvement
8. Good communications
9. Good project organisation and structure
10. Being able to stop a project when needed

In the mid-1990s, at least three important analyses of the state of the software engineering industry were performed. The results were presented in *Patterns of Software Systems Failure and Success* (Jones, 1996), in "Chaos" (Standish group, 1995), and in *Report of the Defence Science Board Task Force on Acquiring Defence Software Commercially* (Defence Science Board, 1994). All three analyses reached the same general conclusion: The success rate for software projects is very low.

In 1995, the Standish Group interviewed IT Managers and asked what aspect makes a project most successful. Table 1.1 indicates the top factors in a project's success. The top factor, user involvement, garnered only 15.9 percent of the responses, so it is evident that no single factor correlates to success. Subsequent studies have indicated that perhaps the most critical factor in a successful project implementation is **cultural acceptance**.

Success Factors	**% of Responses**
User Involvement	15.9
Executive Management Support	13.9
Clear Statement of Requirements	13.0
Planning	9.6

Table 1.1 : Top factors for project success (Jones, 1996)

1.1.4 Software project management

Software projects are very much like mountain climbing. The project manager and the lead climber gather a team, select tools, and use measurements to plan, and execute the project and the climb. Both the software project and the mountain climb begin with an explanation of the schedule, process, tools, and measurements to the team. As the project or climb progresses, both teams adjust the schedule and process based on measurements and team input on status, progress, and risks. The software team deploys their product, just as the climbers "assault" the summit. Once a project or climb is over, both teams review and evaluate their schedule, route, tools and measurements to improve future projects and climbs (Joel, 2004).

According to Joel the four basic building blocks of software project management are:

- People
- Process
- Tools
- Measurements

Improvements in the economics of software development have been not only difficult to achieve but also difficult to measure and substantiate. The key to substantial improvement is a balanced attack across several interrelated dimensions (Royce, 1998):

1. Reducing the size or complexity of what needs to be developed.
2. Improving the development process.
3. Using more-skilled personnel and better teams.
4. Using better environments (tools to automate the process).
5. Trading off or backing off on quality thresholds.

Software Project management as defined by Royce (1998):

- Projects can under plan and they can over plan. Once again, balance is paramount in the level of planning detail and the buy-in among stakeholders.
- Organisational structures form the architecture of the teams. Organisations engaged in a software line of business need to support projects with the infrastructure necessary to use a common process.
- Process automation, and change management in particular, are critical to an iterative process.

1.2 Problem Statement

Information Technology (IT) presents many challenges. Constant change, e-business, massive amounts of information, global operations, and building new alliances and capabilities are just some of the transformations being driven by the use of IT in business today (Mckeen and Smith, 2003).

More so than any other organisational function, IT had to face pressures for continual change and challenge. For example, in the mid-1990s we wrote: "It is evident that more sophisticated mechanism for delivering information systems (IS) to the organisation is now required. Like the process of retooling an outdated factory to turn out products faster and more efficiently, the IS function must undergo a change that is no less comprehensive if it is to fulfil its organisational mandate.... the risk not doing so is increasing inadequacy and eventual obsolescence (Mckeen and Smith, 1996). Five years later we read, "IT needs to transform itself, the way it operated, the way it does business. Those who are not successful will disappear" (Marron, 2000). This kind of pressure stems from two sources:

- The changing business environment
- The changing technology landscape

The software industry is therefore also experiencing a renaissance. Many ingrained software engineering principles are going by the wayside, obsolesced by new technology or replaced by better techniques or advanced levels of automation.

No matter what the discipline, it is important to understand the current state before attempting to transition to a new one. Before considering a software management framework for the future, it is necessary to understand where the industry is today and how it got here. Typical goals of software process improvement efforts are to achieve a 2X, 3X or 10X increase in productivity, quality, time to market, or some combination of all three, where X corresponds to how well the company does now. The funny thing is that many of these organisations have no idea what X is, in objective terms (Royce, 1998).

The best thing about software is its flexibility: It can be programmed to do almost anything. The worst thing about software is also its flexibility: The "almost anything" characteristic has made it difficult to plan, monitor, and control software development. This unpredictability is the basis of what has been referred to for the past 30 years as the "software crisis".

1.3 Proposition/Hypothesis

The research report intends to test:

Hypothesis 1 (HYP1):

"Clients think there is a high degree of software application failure".

Hypothesis 2 (HYP2):

"Developers think there is a low degree of software application failure".

Hypothesis 3 (HYP3):

"Development teams and clients perspective differs when it comes to defining success of software applications."

1.4 Objectives

This Research paper's main objective is to determine what the success rate of software development projects in South Africa is according to the development team and client perspectives and how do they correlate with the literature's suggested success factors.

The following objectives must be achieved in order to accomplish the overall research objective:

- To synthesise a complete set of success factors for successful project delivery as defined in the literature.
- To measure software development projects success/failure rate, in the South African industry, against the set of factors determined by literature, from two perspectives – the client and the development team perspectives.
- To determine if the two perspectives differ in any way.
- To determine if the clients and development team have a different set of success factors than defined by the literature.

1.6 Document Structure

Table 1.4 illustrates the various steps taken in this research to achieve the stated objective. These steps are classified into four main phases and they are organised into 7 chapters:

Chapter 2 – The changing role of software development

This chapter introduces the state of the practice in the software industry. The software industry is experiencing a renaissance. Many-ingrained software engineering principles are going by the wayside obsolesced by new technology or replaced by better techniques or advanced levels of automation.
Before considering a software management framework, it is necessary to understand where the industry is today and how it got here.

Furthermore it looks at how management control of the key software economics leverage points with respect to traditional software management is obtained. These are (1) reducing the amount of software you need to build, (2) reducing rework via improved processes and teamwork, and (3) reducing the labour-intensiveness of the remaining work via automation.

Chapter 3 – Software management process framework

This chapter takes a comprehensive look at the specifics of a new organisation of the software life cycle. It combines the flexibility of the spiral model with the discipline of risk management and a set of major life-cycle phases and milestones. These milestones are focused on major management commitment to life-cycle courses of action.

Chapter 4 – Software management disciplines

This chapter addresses how to ensure that the software project's management artefacts are concurrently evolving and coalescing. These include the project's plans and associated cost and schedule estimates, the project's organisation and team-building activities, and the project's metrics, instrumentation, and control processes. It not only emphasises that the management solutions are situation-dependent, it also provides guidelines for tailoring them to the project's scale, team culture, process maturity, architectural risk and domain experience.

2 THE CHANGING ROLE OF SOFTWARE DEVELOPMENT

This chapter introduces the state of the practice in the software industry. The software industry is experiencing a renaissance. Many-ingrained software engineering principles are going by the wayside obsolesced by new technology or replaced by better techniques or advanced levels of automation.

Before considering a software management framework, it is necessary to understand where the industry is today and how it got here.

Furthermore it looks at how management control of the key software economics leverage points with respect to traditional software management is obtained. These are (1) reducing the amount of software you need to build, (2) reducing rework via improved processes and teamwork, and (3) reducing the labour-intensiveness of the remaining work via automation.

2.1 Conventional Software Management

2.1.1 Introduction

The best thing about software is its flexibility: It can be programmed to do almost anything. The worst thing about software is also its flexibility: The "almost anything" characteristic has made it difficult to plan, monitor, and control software development. This unpredictability is the basis of what has been referred to for the past 30 years as the "software crisis".

In the mid-1990s, at least three important analyses of the state of the software engineering industry were performed. The results were presented in *Patterns of Software Systems Failure and Success* (Jones, 1996), in "Chaos" (Standish group, 1995), and in *Report of the Defence Science Board Task Force on Acquiring Defence Software Commercially* (Defence Science Board, 1994). All three analyses reached the same general conclusion: The success rate for software projects is very low. Although the analyses had some differing perspectives, their primary messages were complimentary and consistent. They can be summarised as follows:

1. Software development is still highly unpredictable. Only about 10% of software projects are delivered successfully within initial budget and schedule estimates.

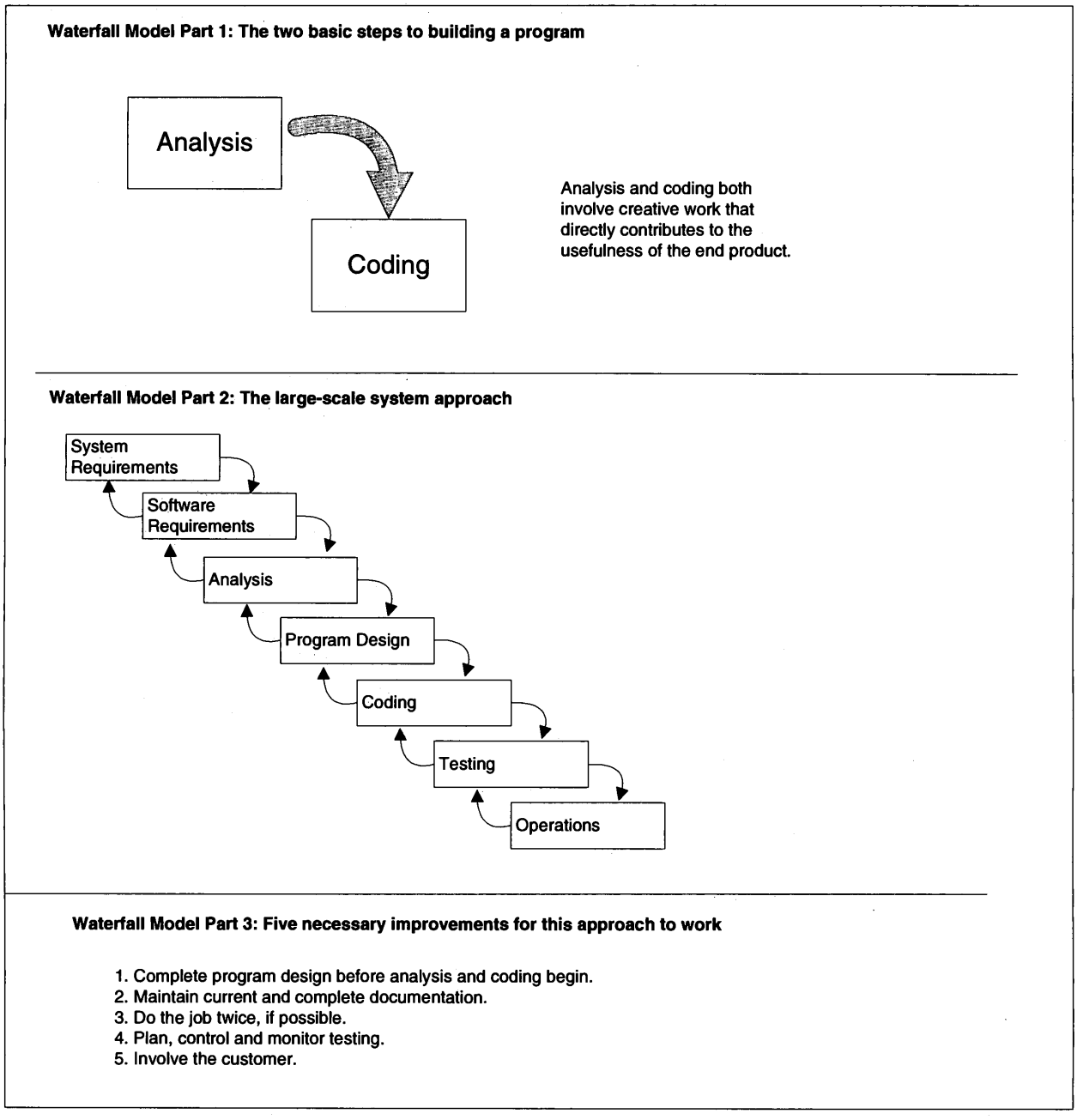

Figure 2.1 : The Waterfall model (Royce, 1998)

Boehm (1987) provides ten industrial software metrics, which is a good, objective characterisation of the state of software development. There is very little evidence of significant changes in the past decade. Although many of the metrics are gross generalisations, they accurately describe some of the fundamental economic relationships that resulted from the conventional software process practices over the past 30 years (Royce, 1998).

The 10 metrics are:

1. Finding and fixing a software problem after delivery cost 100 times more than finding and fixing the problem in early design phases.

the process exponent being greater than 1.0. Contrary to most manufacturing processes, the more software you build, the more expensive it is per unit item.

2.3 Improving Software Economics

Improvements in the economics of software development have been not only difficult to achieve but also difficult to measure and substantiate.

The key to substantial improvement is a balanced attack across several interrelated dimensions(Royce, 1998):

1. Reducing the size or complexity of what needs to be developed.
2. Improving the development process.
3. Using more-skilled personnel and better teams.
4. Using better environments (tools to automate the process).
5. Trading off or backing off on quality thresholds.

2.4 The Old Way and the New

2.4.1 The principles of conventional software engineering

Davis (1994) chooses the following principles as a benchmark, which is a good summary of the conventional wisdom within the software industry:

1. Make quality number 1.

Defining quality commensurate with the project at hand is important, but is not easily done at the outset of a project. Consequently, a modern process framework strives to understand the trade-offs among features, quality, cost and schedule as early in the life cycle as possible. Until this understanding is achieved, it is not possible to specify or manage the achievement of quality.

2. High-quality software is possible.

This principle is mostly redundant with the others.

3. Give products to customers early.

This is a key tenet of a modern process framework, and there must be several mechanisms to involve the customer throughout the life cycle.

12. Good management is more important than good technology.

This principle is of such importance, because without good management throughout the life cycle, no software system will succeed.

13. People are the key to success.

This principle is too low on the list. People are the "glue" to the process.

14. Follow with care.

This is sage advice, especially in a rapidly growing industry in which technology fads are difficult to distinguish from technology improvements. Trading off features, costs, and schedules does not always favour the most technologies.

15. Take responsibility.

This is a great corollary to 14. It takes more than good methods, tools and components to succeed. It also takes good people, good management ad learning culture that is focused on forward progress even when confronted with numerous and inevitable setbacks.

16. Understand the customer's priorities.

Understanding the customer's priorities is important, but only on balance with other stakeholder priorities.

17. The more they see, the more they need.

This principle is true, but it suggests that you would never want to show a user anything. It should read, "The more users see, the better they understand". The ramification of this principle on a modern process is that the software project manager needs to have objective data with which the inevitable change requests and maintain a balance of affordability, features, and risk.

18. Plan to throw one away.

You should not plan to throw one away. Rather, you should plan to evolve a product from an immature prototype to a mature baseline.

19. Design for change.

Simple statement that has proven to be exceedingly complex to realise. But attempting to predict the sorts of changes that are likely to occur is a useful exercise in risk management and a recurring theme of successful software projects.

20. Design without documentation is not design.

Anchored in the document-driven approach in the past. Should rather say "Software artefacts should be mostly self-documenting".

30. Expect excellence.

This principle applies to all disciplines, not just software management.

2.4.2 Principles of modern software management

Royce (1998) mentions top 10 principles:

1. Base the process on an architecture-first approach.

This requires that a demonstrable balance be achieved among the driving requirements, the architecturally significant design decisions, and the life-cycle plans before the resources are committed for full-scale development.

2. Establish an iterative life-cycle process that confronts risk early.

With today's sophisticated software systems, it is not possible to define the entire problem, design the entire solution, build the software, and then test the end product in sequence. Instead, an iterative process that refines the problem understanding, an effective solution, and an effective plan over several iterations encourages a balanced treatment of all stakeholder objectives. Major risks must be addressed early to increase predictability and avoid expensive downstream scrap and rework.

3. Transition design methods to emphasis component-based development.

Moving from a line-of-code mentality to a component-based mentality is necessary to reduce the amount of human-generated source code and custom development. A component is a cohesive set of pre-existing lines of code, either in source or executable format, with a defined interface and behaviour.

4. Establish a change management environment.

The dynamics of iterative development, including concurrent workflows by different teams working on shared artefacts, necessitates objectively controlled baselines.

5. Enhance change freedom through tools that support.

Round-trip engineering is the environment support necessary to automate and synchronise engineering information in different formats (such as requirement specifications, design models, source code, executable code, test case). Without substantial automation of this bookkeeping, change management, documentations, and testing, it is difficult to reduce iteration cycles to manageable time frames in which change is encouraged rather than avoided. Change freedom is a necessity in an iterative process, and establishing an integrated environment is crucial.

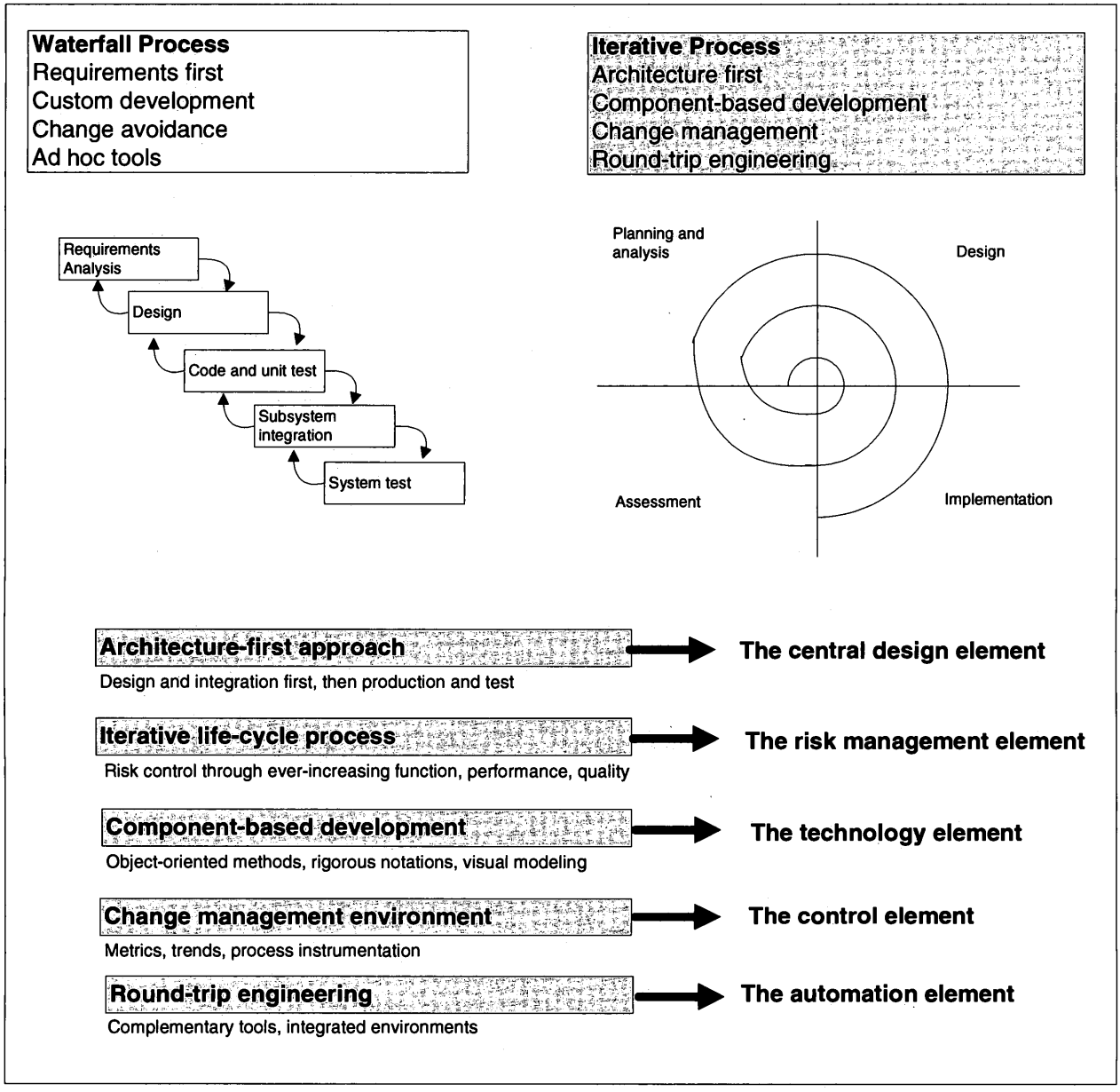

Figure 2.2 : The top five principles of a modern process (Royce, 1998)

2.5 Conclusion

Conventional software management practices are mostly sound in theory, but practice is still tied to archaic technology and techniques. Conventional software economics provides a benchmark of performance for conventional software management principles.

Economic results of conventional software projects reflect an industry dominated by custom development, ad hoc processes and diseconomies of scale. Today's cost models are based primarily on empirical project databases with very few modern iterative development success stories. Good software cost estimates is difficult to attain. Decision makers must deal with highly imprecise estimates.

3 SOFTWARE MANAGEMENT PROCESS FRAMEWORK

3.1 Life-Cycle Phases

3.1.1 Introduction

The most discriminating characteristic of a successful software development process is the well-defined separation between "research and development" activities and "production" activities. When software projects do not succeed, the primary reason is usually a failure to crisply define and execute these two stages, with proper balance and appropriate emphasis. This is true for conventional as well as iterative processes. Most unsuccessful projects exhibit on of the following characteristic (Royce, 1998):

1. **An overemphasis on research and development.** Too many analyses or paper studies are performed, or the construction of engineering baselines is procrastinated. This emphasis is typical of and promoted in the conventional software process.
2. **An overemphasis on production.** Rush-to-judgment designs, premature work by overeager coders, and continuous hacking are typical.

Successful modern projects – and even successful projects developed under the conventional process – tend to have very **well-defined project milestone** when there is a noticeable transition from a research attitude to a production attitude. Earlier phases focus on achieving functionality. Later phases revolve around achieving a product that can be shipped to a customer, with explicit attention to robustness, performance, fit, and finish. This life-cycle balance, which is somewhat subtle and still too intangibles, is one if the underpinnings of successful software project management.

A modern software development process must be defined to support the following (Royce, 1998):

1. Evolution of the plans, requirements, and architecture, together with well-defined synchronisation points.
2. Risk management and objective measures of progress and quality.
3. Evolution of system capabilities through demonstrations of increasing functionality.

Engineering stage		Production stage	
Inception phase	**Elaboration phase**	**Construction phase**	**Transition phase**
Primary objectives			
Establish project's software scope and boundary conditions	Baseline the architecture as rapidly as practical	Minimising development coast by optimising resources and avoiding unnecessary scrap and rework	Achieving user self-supportability
Discriminating the critical use cases	Baseline the vision	Achieving adequate quality as rapidly as practical	Achieving stakeholder concurrence that deployment baselines are complete and consistent with the evaluation criteria of the vision
Demonstrating at least one candidate architecture	Baseline a high-fidelity plan for the construction phase	Achieving useful versions	Achieving final product baselines as rapidly and cost-effectively as practical
Estimate cost and schedule	Demonstrating that the baseline architecture will support the vision at a reasonable cost in a reasonable time		
Estimate potential risks			
Essential activities			
Formulating the scope	Elaborating the vision	Resource management, control and process optimisation	Synchronisation and integration of concurrent construction increments into consistent deployment baselines
Synthesizing the architecture	Elaborating the process and infrastructure	Complete component development and testing against evaluation criteria	Deployment-specific engineering
Planning and preparing a business case	Elaborating the architecture and selecting components	Assessment of product releases against acceptance criteria of the vision	Assessment of deployment baselines

Table 3.1 : Details of the Engineering and Production Stages (Royce, 1998)

3.2 Artefacts of the Process

The artefacts of the process are organised into five sets: management, requirements, design, implementation, and deployment. The management artefacts capture the information necessary to synchronise stakeholder expectations. The requirements, design, implementation, and deployment artefacts are captured in rigorous notations that support automated analysis and browsing (Royce, 1998). Figure 3.1 below provides the details of each set and Figure 3.2 indicates the life cycle focus.

3.3 Model-Based Software Architectures

3.3.1 Introduction

Software architecture is the central design problem of a complex software system, in the same way that architecture is the central design problem of a complex skyscraper. However, software architecture has several additional dimensions of complexity. In contrast to the architecture of a large building, the critical performance attributes and features of a complex software system cannot be described through stale laws of physics (Pahl, 2004).

3.3.2 Architecture: A management perspective

According to Pahl (2004) there are three different aspects of architecture, from a management perspective, and they are:

1. An ***architecture*** (the intangible design concept) is the design of a software system, as opposed to the design of a component. This includes all engineering necessary to specify a complete bill of materials. Significant make/buy decisions are resolved, and all custom components are elaborated so that individual component costs and construction/assembly costs can be determined with confidence.
2. An ***architecture baseline*** (the tangible artefacts) is a slice of information across the engineering artefact sets sufficient to satisfy all stakeholders that the vision (function and quality) can be achieved within the parameters of the business case (cost, profit, time, technology, and people).
3. An ***architecture description*** (a human-readable representation of an architecture, which is one of the components of an architecture baseline) is an organised subset of information extracted from the design set model(s). It includes the additional ad hoc notation (text and graphics) necessary to clarify the information in the models. The architecture description communicates how the intangible concept is realised in the tangible artefacts.

The importance of software architecture and its close linkage with modern software development processes can be summarised as follows (Pahl, 2004):

1. Achieving stable software architecture represents a significant project milestone at which the critical make/buy decisions should have been resolved. This life-cycle event represents a transition from

3.4.1 Software process workflows

The term *workflow* is used to mean a thread of cohesive and mostly sequential activities. Workflows are mapped to product artefacts and to project teams. There are seven top-level workflows (Robertson, 2004):

1. **Management**

Controlling the process and ensuring win conditions for all stakeholders.

2. **Environment**

Automating the process and evolving the maintenance environment.

3. **Requirements**

Analysing the problem space and evolving the requirements artefacts.

4. **Design**

Modelling the solution and evolving the architecture and design artefacts.

5. **Implementation**

Programming the components and evolving the implementation and deployment artefacts.

6. **Assessment**

Assessing the trends in process and product quality.

7. **Deployment**

Transitioning the end products to the user.

3.5 Checkpoints of the process

Three sequences of project checkpoints are used to synchronise stakeholder expectations throughout the life cycle: major milestones, minor milestones and status assessments. The most important major milestone is usually the event that transitions the project from the elaborations phase into the construction phase. The format and content of minor milestones are highly dependent on the project and the organisational culture. Periodic status assessments are crucial for focusing continuous attention on the evolving health of the project and the dynamic priorities (Robertson, 2004).

4 SOFTWARE MANAGEMENT DISCIPLINES

4.1 Introduction

Projects can under plan and they can over plan. Once again, balance is paramount in the level of planning detail and the buy-in among stakeholders. The work breakdown structures are the "architecture" of the project plan. It must encapsulate change and evolve with the appropriate level of detail throughout the life cycle. Cost and schedule budgets should be estimated using macro-analysis techniques (top-down project level) and microanalysis techniques (bottom-up task level) to achieve predicable results (Robertson, 2004).

4.1.1 Work breakdown structures (WBS)

A good WBS and its synchronisation with the process framework are critical factors in software project success (Royce, 1998).

4.1.1.1 Conventional WBS Issues

Conventional WBS frequently suffers from three fundamental flaws:

- They are prematurely structured around the product design.
- They are prematurely decomposed, planned, and budgeted in either too much or too little detail.
- They are project-specific, and cross-project comparisons are usually difficult or impossible.

Example of WBS:

Management

System requirements and design

Subsystem 1

Component 11

Requirements

Design

Code

Test

Documentation

Component 1N

Requirements

AD Transition phase management

ADA

ADB

B Environment

BA Inception

BAA

BAB

BB Elaboration

BBA

BBB

BC Construction

BCA

BCB

BD Transition

BDA

BDB

C Requirements

.....

D Design

...

E Implementation

F Assessment

G Deployment

4.2 Project Organisations and Responsiblities

Organisational structures form the architecture of the teams. Organisations engaged in a software line of business; need to support projects with the infrastructure necessary to use a common process. Project organisations need to allocate artefacts and responsibilities clearly across project teams to ensure a balance of global (architecture) and local (component) concerns. The organisation must evolve with the WBS and the life-cycle concerns (Robertson, 2004).

4.5 Tailoring the Process

The process framework must be configured to the specific characteristics of the project. The scale of the project – in particular, team size – drives the process configuration more than any other factor. Other key factors include stakeholder relationships, process flexibility, process maturity, architectural risk, and domain experience. While specific process implementations will vary, the spirit underlying the process is the same (Robertson, 2004).

4.6 Conclusion

Organising the WBS around the process framework rather than the product framework, with a supporting organisational structure and clear responsibilities allocated across project teams, are crucial to the success of managing software projects. Furthermore process automation, change management, process control and metrics will ensure that quality and agility of software products.

1. Initial interview – a management member of each company was interviewed to obtain general information on the company, to personally make contact with the manager as the primary contact in coordinating the research, and to provide the company with information regarding the objectives and phases of the research. The purpose of the interview was to confirm that the research is relevant to the company, to make the company aware of the requirements of participating in the survey, and to ensure that they are committed and willing to cooperate and provide their necessary contribution. Refer to Appendix B for the aspects discussed in the initial interview.
2. Questionnaires – After analysing and synthesising the data in the literature review, the various success factors for software development applications were identified and questionnaires were developed to reflect these success factors:
    - A **Software development Delivery Survey** questionnaire was designed and distributed to certain software development team members, determined at the discretion of the manager/contact person. The questions were grouped into two logical areas: software development ratings and client relationship ratings. These two areas were then further broken down into more logical groupings which were derived from the success criteria for software development applications specified in the literature. Refer to Appendix C for the questionnaire.
    - A **Client Relationship Survey** questionnaire was designed and distributed to the clients/users of the software development applications provided by the software development company, determined at the discretion of the manager/contact person. The questions were grouped into three logical areas: the software application ratings, the support ratings and the software application provider ratings. These three areas were then broken down into more logical groupings which were derived from the success criteria for software development applications specified in the literature. Refer to Appendix D for the questionnaire sent to the clients/users.
3. Correspondence via e-mail – The management member of the company was the main contact person and all correspondence on the progress of the study, the different phases and distribution and collection of the surveys was done electronically via e-mail.

The questions posed to the software development team members and the clients/users was based on a 6 point numerical rating scale where 1 represents VERY POOR and 6 represents EXTREMELY GOOD.

The client and development survey questionnaires were constructed in such a manner that they encapsulated all the factors and aspects that the literature review considers as the critical success factors

6 DATA ANALYSIS AND INTERPRETATION

This chapter will focus on:

- Data preparation - which includes editing, coding, and data entry. These activities ensure the accuracy of the data and their conversions from raw form to reduced and classified forms that are more appropriate for analysis.
- Preparing a descriptive statistical summary to understand the data collected in the surveys.

To facilitate the answering of the research questions in Chapter one and testing the various hypothesis stated, general assumptions and limitations of the survey will be discussed and thereafter this chapter will be structured into 3 sections that will focus on exploring, displaying and examining the data collected from the surveys to address each hypothesis. Subsequently the open-ended questions posed to the clients and development are collated and synthesised to derive the factors that the clients and development team think is important for the success of software development applications. These factors can then be compared to the factors that the literature prescribes as important. Ultimately all the information are analysed and interpreted to reach the research objectives stated in Chapter one.

6.1 General Assumptions and Limitations

The companies that participated in the research study have a staff count of less than 20 with annual revenue between 1million and 10 million and software development experience of between 5 and 15 years, which indicates that they can be classified as small sized software development organisations. The type of software development varies from financial asset administration, accounting, manufacturing, and cosmetics distribution applications. **Therefore, any conclusions made are limited to small software development companies.**

Due to the size of the software development companies, the number of development team member respondents and client respondents are also relative small, with an average of 2 development members and 2 client respondents in each company, accumulating to a total of 9 development and 8 client respondents. Although this is a small sample, due to the relative same number of development and client respondents the comparisons between the development and client perspectives should not be distorted in any way.

Acronym	Client Relationship Survey Results	Weight
TrbAr	a. Turnaround time to implement changes	50%
InfOth	b. Influence of changes on the rest of the application (does change cause problems in other areas of the application)	50%
SSERV	**Support service**	**70%**
AdSup	a. Adequate support provided with regards to any queries – quick response and resolution	50%
AdComm	b. Adequate communication channels – ease of access to support team	50%
SAP	**Section C: Software Application Provider**	**25%**
PRWK	**Professional Work**	**20%**
PrMan	a. Professional manner in which the Provider liaise with you – friendly, helpful, respectful, integrity, responsiveness, leadership, commitment	15%
AtIW	b. Quality of work delivered by the Provider	40%
AdComm	c. Adequate communication provided by the Provider	30%
AdContin	d. Adequate continuity provided by the Provider	15%
QPER	**Qualified Personnel**	**20%**
AdBusKn	a. Adequate business knowledge by the Provider	30%
AdTechSk	b. Adequate technical skills by the Provider	30%
MapBusReq	c. Ability to understand and map business requirements	40%
VALP	**Value Provided**	**60%**
Vmoney	a. Value for money for the application delivered	35%
EasUndDoc	b. Easy to understand and useful documentation	15%
TimelyDel	c. Timely delivery by the Provider	20%
AdPM	d. Adequate project management provided by the Provider – progress reports, meetings, project plan and tracking	30%

Table 6.1 : List of the client questionnaire logical groupings and their acronyms

3. The questions, hierarchical structure, and their weighting percentages were captured in a spreadsheet.
4. Each survey questionnaire response was captured in the spreadsheet.
5. Each question has a value out of 6 (highest rating value in the scale), which was translated into a percentage, example a value of 4, translates in 66.67% (4/6 * 100).
6. For each company an average of the total number of respondents was determined for each question, for example where there are 2 respondents, and one rated a question as 5 and the other as 4, the average for the question for that particular company is equal to 75%, calculated as follows: [(5/6 *100) + (4/6 * 100)]/2.
7. For each company their score for each question was calculated as the average percentage multiplied by the weighting percentage, example Average = 75%, question aspect is User involvement, which has a weighting of 20%, therefore the company's score equals 15% calculated as follows: (20% * 75%).
8. Then the lowest level aspects in the hierarchy were aggregated by applying the weighting percentage of the particular level, for example all aspects under Business Fit were aggregated and weighted by the weighting percentage of 40%, results in a total of 30%.
9. Step 8 is repeated until the Overall total percentage is calculated for each company, example Company 2 has an overall score of 71.88%
10. To calculate a general total for all the companies, the average of the total number of respondents for all companies, for each question was calculated, for example there are a total of 8 respondents and 4 rated a question as 3 and 4 rated it as 5, the average will be [4 * (3/6 *100)] + [4 * (5/6 * 100) / 8], which results in an average of 66.67%. The Average was then multiplied by the weighting of the question, and then aggregated at each level, resulting into the general total score of 70.13%.
11. Each aspect general total score was also translated to a 100% ratio value, for example , the Business fit aspect total score is 30.83%, weighting of 40%, which translates to a survey result value of 77.08% (30.83/40.00 * 100).

The survey result value will assist to identify any aspects that indicate problem areas for the clients, and could be areas that software development organisations can improve on. Although 3/6, i.e. 50% is the cut-off, for success or failure, it is the difference between 3 (50%) and 4(66.67) that determines the range where an aspect requires investigation, therefore any aspect that scores less than 4/6, i.e. 66.67% is regarded as an area that needs to be investigated or improved on.

Acronym	Client Relationship Survey Results	Survey Result	Weight	TOTAL SCORE	TOTAL AVG	Company 1						Company 2					
						Score	Response 1		Response 2		Avg	Score	Response 1		Response 2		Avg
	Client Overall score			**70.13%**		**62.10%**						**71.88%**					
SA	**Section A: Software Application/Solution**	**72.77%**	**50%**	**36.39%**		**33.83%**						**36.79%**					
BF	**Business fit**	**77.08%**	**40%**	**30.83%**		**28.67%**						**30.00%**					
UsIn	a. User involvement in determining requirements	**79.17%**	20%	15.83%	79.17%	16.67%	5	83.33%	5	83.33%	83.33%	15.00%	4	66.67%	5	83.33%	75.00%
BuRe	b. Fulfills the business requirements	**79.17%**	30%	23.75%	79.17%	20.00%	4	66.67%	4	66.67%	66.67%	22.50%	4	66.67%	5	83.33%	75.00%
EaIn	c. Easy integration with other processes and/or applications	**70.83%**	5%	3.54%	70.83%	3.33%	4	66.67%	4	66.67%	66.67%	3.75%	4	66.67%	5	83.33%	75.00%
Mexp	d. Meets your expectations	**70.83%**	25%	17.71%	70.83%	16.67%	4	66.67%	4	66.67%	66.67%	18.75%	4	66.67%	5	83.33%	75.00%
Aval	e. Adds value to the business	**81.25%**	20%	16.25%	81.25%	15.00%	5	83.33%	4	66.67%	75.00%	15.00%	4	66.67%	5	83.33%	75.00%
DOC	**Documentation**	**69.79%**	**10%**	**6.98%**	0.00%	**6.92%**		0.00%		0.00%	0.00%	**6.17%**		0.00%		0.00%	0.00%
ASDo	a. Adequate supporting documentation/on-line help	**72.92%**	40%	29.17%	72.92%	26.67%	5	83.33%	**3**	50.00%	66.67%	26.67%	**3**	50.00%	5	83.33%	66.67%
Uunb	b. Understandable and un-ambiguous	**68.75%**	30%	20.63%	68.75%	22.50%	5	83.33%	4	66.67%	75.00%	17.50%	4	66.67%	**3**	50.00%	58.33%
QFCo	c. Quality of format and content	**66.67%**	30%	20.00%	66.67%	20.00%	4	66.67%	4	66.67%	66.67%	17.50%	4	66.67%	**3**	50.00%	58.33%
TRN	**Training**	**67.08%**	**10%**	**6.71%**	0.00%	**7.08%**		0.00%		0.00%	0.00%	**5.75%**		0.00%		0.00%	0.00%
AdTr	a. Adequate training provided	**77.08%**	40%	30.83%	77.08%	33.33%	5	83.33%	5	83.33%	83.33%	30.00%	5	83.33%	4	66.67%	75.00%
AdTM	b. Adequate training manuals provided	**60.42%**	30%	18.13%	60.42%	20.00%	**3**	50.00%	5	83.33%	66.67%	12.50%	**1**	16.67%	4	66.67%	41.67%
QFAFCTm	c. Quality of format and content of training manuals	**60.42%**	30%	18.13%	60.42%	17.50%	**3**	50.00%	4	66.67%	58.33%	15.00%	**2**	33.33%	4	66.67%	50.00%
APUS	**Application Usage**	**70.63%**	**40%**	**28.25%**	0.00%	**25.00%**		0.00%		0.00%	0.00%	**31.67%**		0.00%		0.00%	0.00%
UsFr	a. User friendliness of application (ease of use)	**72.92%**	20%	14.58%	72.92%	15.00%	5	83.33%	4	66.67%	75.00%	15.00%	4	66.67%	5	83.33%	75.00%
ResTi	b. Response times (quick response between screens/processes)	**77.08%**	10%	7.71%	77.08%	6.67%	4	66.67%	4	66.67%	66.67%	8.33%	5	83.33%	5	83.33%	83.33%
EfEc	c. Effectiveness and efficiency (good screen design with minimum cumbersome	**68.75%**	20%	13.75%	68.75%	11.67%	4	66.67%	**3**	50.00%	58.33%	15.00%	4	66.67%	5	83.33%	75.00%

Acronym	Client Relationship Survey Results	Survey Result	Weight	TOTAL SCORE	TOTAL AVG	Company 1						Company 2					
						Score	Response 1		Response 2		Avg	Score	Response 1		Response 2		Avg
AdComm	c. Adequate communication provided by the Provider	**62.50%**	30%	18.75%	62.50%	15.00%	3	50.00%	3	50.00%	50.00%	17.50%	2	33.33%	5	83.33%	58.33%
AdContin	d. Adequate continuity provided by the Provider	**64.58%**	15%	9.69%	64.58%	8.75%	4	66.67%	3	50.00%	58.33%	8.75%	2	33.33%	5	83.33%	58.33%
QPER	**Qualified Personnel**	**73.54%**	**20%**	**14.71%**	0.00%	**12.17%**		0.00%		0.00%	0.00%	**16.67%**		0.00%		0.00%	0.00%
AdBusKn	a. Adequate business knowledge by the Provider	**70.83%**	30%	21.25%	70.83%	17.50%	2	33.33%	5	83.33%	58.33%	22.50%	4	66.67%	5	83.33%	75.00%
AdTechSk	b. Adequate technical skills by the Provider	**77.08%**	30%	23.13%	77.08%	20.00%	3	50.00%	5	83.33%	66.67%	27.50%	6	100.00%	5	83.33%	91.67%
MapBusReq	c. Ability to understand and map business requirements	**72.92%**	40%	29.17%	72.92%	23.33%	3	50.00%	4	66.67%	58.33%	33.33%	5	83.33%	5	83.33%	83.33%
VALP	**Value Provided**	**63.44%**	**60%**	**38.06%**	0.00%	**32.75%**		0.00%		0.00%	0.00%	**36.00%**		0.00%		0.00%	0.00%
Vmoney	a. Value for money for the application delivered	**70.83%**	35%	24.79%	70.83%	23.33%	4	66.67%	4	66.67%	66.67%	26.25%	4	66.67%	5	83.33%	75.00%
EasUndDoc	b. Easy to understand and useful documentation	**56.25%**	15%	8.44%	56.25%	8.75%	4	66.67%	3	50.00%	58.33%	6.25%	1	16.67%	4	66.67%	41.67%
TimelyDel	c. Timely delivery by the Provider	**66.67%**	20%	13.33%	66.67%	10.00%	4	66.67%	2	33.33%	50.00%	15.00%	4	66.67%	5	83.33%	75.00%
AdPM	d. Adequate project management provided by the Provider – progress reports, meetings, project plan and tracking	**56.25%**	30%	16.88%	56.25%	12.50%	2	33.33%	3	50.00%	41.67%	12.50%	1	16.67%	4	66.67%	41.67%

Table 6.2 : Company 1 and 2 client responses mapped to the hierarchical components and the weighting calculations

Acronym	Client Relationship Survey Results	Survey Result	Weight	TOTAL SCORE	TOTAL AVG	Company 3								Company 4			
						Score	Response 1		Response 2		Response 3		Avg	Score	Response 1		Avg
Const	d. Consistency (Same look and feel in all screens and processes)	**79.17%**	20%	15.83%	79.17%	15.56%	6	100.00%	5	83.33%	**3**	50.00%	77.78%	13.33%	4	66.67%	66.67%
ErrFr	e. Error free (no run-time errors)	**62.50%**	20%	12.50%	62.50%	12.22%	5	83.33%	4	66.67%	**2**	33.33%	61.11%	13.33%	4	66.67%	66.67%
BuFr	f. Bug free (no inconsistency with business rules)	**62.50%**	10%	6.25%	62.50%	6.67%	5	83.33%	4	66.67%	**3**	50.00%	66.67%	6.67%	4	66.67%	66.67%
					0.00%			0.00%		0.00%		0.00%	0.00%			0.00%	0.00%
SUP	**Section B: Support**	**68.44%**	**25%**	**17.11%**	0.00%	**19.24%**		0.00%		0.00%		0.00%	0.00%	**14.17%**		0.00%	0.00%
CHN	**Changes**	**60.42%**	**30%**	**18.13%**	0.00%	**22.50%**		0.00%		0.00%		0.00%	0.00%	**10.00%**		0.00%	0.00%
TrbAr	a. Turnaround time to implement changes	**56.25%**	50%	28.13%	56.25%	36.11%	6	100.00%	5	83.33%	**2**	33.33%	72.22%	16.67%	**2**	33.33%	33.33%
InfOth	b. Influence of changes on the rest of the application (does change cause problems in other areas of the application)	**64.58%**	50%	32.29%	64.58%	38.89%	6	100.00%	4	66.67%	4	66.67%	77.78%	16.67%	**2**	33.33%	33.33%
SSERV	**Support service**	**71.88%**	**70%**	**50.31%**	0.00%	**54.44%**		0.00%		0.00%		0.00%	0.00%	**46.67%**		0.00%	0.00%
AdSup	a. Adequate support provided with regards to any queries – quick response and resolution	**66.67%**	50%	33.33%	66.67%	38.89%	6	100.00%	5	83.33%	**3**	50.00%	77.78%	33.33%	4	66.67%	66.67%
AdComm	b. Adequate communication channels – ease of access to support team	**77.08%**	50%	38.54%	77.08%	38.89%	6	100.00%	5	83.33%	3	50.00%	77.78%	33.33%	4	66.67%	66.67%
					0.00%			0.00%		0.00%		0.00%	0.00%			0.00%	0.00%
SAP	**Section C: Software Application Provider**	**66.54%**	**25%**	**16.64%**	0.00%	**19.19%**		0.00%		0.00%		0.00%	0.00%	**13.79%**		0.00%	0.00%
PRWK	**Professional Work**	**68.85%**	**20%**	**13.77%**	0.00%	**15.06%**		0.00%		0.00%		0.00%	0.00%	**13.33%**		0.00%	0.00%
PrMan	a. Professional manner in which the Provider liaise with you – friendly, helpful, respectful, integrity, responsiveness, leadership, commitment	**75.00%**	15%	11.25%	75.00%	11.67%	6	100.00%	5	83.33%	3	50.00%	77.78%	10.00%	4	66.67%	66.67%
AtlW	b. Quality of work delivered by the Provider	**72.92%**	40%	29.17%	72.92%	31.11%	6	100.00%	5	83.33%	3	50.00%	77.78%	26.67%	4	66.67%	66.67%
AdComm	c. Adequate communication provided by the Provider	**62.50%**	30%	18.75%	62.50%	21.67%	6	100.00%	5	83.33%	**2**	33.33%	72.22%	20.00%	4	66.67%	66.67%
AdContin	d. Adequate continuity provided by the Provider	**64.58%**	15%	9.69%	64.58%	10.83%	6	100.00%	5	83.33%	**2**	33.33%	72.22%	10.00%	4	66.67%	66.67%
QPER	**Qualified Personnel**	**73.54%**	**20%**	**14.71%**	0.00%	**15.56%**		0.00%		0.00%		0.00%	0.00%	**13.33%**		0.00%	0.00%

6.2.2 Summary

The general total score for all the companies came to 70.13%, which indicates that the overall view from the clients' perspective is that the software development application is successful, because the score is much greater than 50%. Comparing the individual companies scores, 2 companies are above the average (71.88% and 76.26%, indicated in green) and 2 are below (62.10% and 64.29%, indicated in orange), but those below the average are still higher than 50%, which shows that the clients rate the software applications developed as successful. The 2 companies whose scores are below the average could do things to improve their delivery due to certain aspects scores being marginal, i.e. have a rating between 3 and 4 (50 and 66.67%). Furthermore, the aspects that were rated 3 or less are highlighted in red, to indicate aspects that the clients did not find favourable, and which provide opportunities for the software development company to improve on. The impact of these aspects on the overall client survey can be seen in the survey result column, and they are also highlighted in red. The lowest scoring aspects are enumerated below (ordered from lowest to highest):

- Support – Changes – a. Turnaround time to implement changes – 56.25%
- Software Application Provider – Value Provided – b. Easy to understand and useful documentation – 56.25%
- Software Application Provider – Value Provided – d. Adequate project management provided by the Provider – progress reports, meetings, project plan and tracking – 56.25%
- Software Application – Training – b. Adequate training manuals provided – 60.42%
- Software Application – Training – c. Quality of format and content of training manuals – 60.42%
- Application Usage – e. Error free (no run-time errors) – 62.50%
- Application Usage – f. Bug free (no inconsistency with business rules) – 62.50%
- Software Application Provider – Professional Work – c. Adequate communication provided by the Provider – 62.50%
- Support – Changes – b. Influence of changes on the rest of the application (does change cause problems in other areas of the application) – 64.58%
- Software Application Provider – Professional Work – d. Adequate continuity provided by the Provider – 64.58%

hierarchical structure and the percentage weightings allocations for each logical grouping. Table 6.5 to 6.7 shows the development survey results mapped to the hierarchical structure and the weighting calculations for the different companies.

Acronym	Software Development	Weight
SDEV	**Section A: Software Development**	**65%**
APPL	**Applicability**	**15%**
AdBusKn	a. Adequate business knowledge	30%
UsIn	b. User involvement in determining their requirements	40%
AdTechSk	c. Adequate technical skills to translate business knowledge to a useful application	30%
SUST	**Sustainability**	**10%**
StrProc	a. Structured process followed – requirements definition, analysis, design, test, implementation	25%
UsTools	b. Usage of tools to assist with process - any case tools, data-modeling tools, re-engineering tools, etc.	20%
DocProv	c. Documentation provided – User Requirements, Specifications, Technical Specifications, training manuals, On-line help, etc.	35%
UsMeth	d. Usage of a specific methodology	20%
USAB	**Usability**	**15%**
StdAppl	a. Development and design standards applied – same look and feel across screens and certain coding standards in place	25%
AdTestBug	b. Adequate testing done for any defects/bugs (unit testing)	25%
AdTestBu	c. Adequate testing done for the fit to business requirements/functionality (integration/functionality testing)	25%
TestUser	d. Testing done by the users – user acceptance testing	25%
FUNC	**Functionality**	**15%**
Trace	a. Trace ability from requirements to design to development to testing	30%
AdComm	b. Adequate communication and liaising with users	40%
AbComp	c. Ability to make suitable compromises on requirements and development alternatives	30%
ADAB	**Adaptability**	**5%**
AdConChn	a. Adequate control of changes – Change management (change control forms, sign-offs and additional costs, etc)	40%
AdConfMng	b. Adequate Configuration management	30%
AdVersCont	c. Adequate Version control management	30%
QUAL	**Quality**	**10%**
DevDesStd	a. Existence of design and development standards	25%
ErrStd	b. Error handling standards applied	25%
Signoffs	c. Sign-offs obtained at different stages in the project	25%
CodeRev	d. Code reviews performed	25%
TIME	**Timely**	**10%**
OnTime	a. Timely delivery of application to the user	40%
AdPM	b. Adequate project management – progress reports, meetings, project plan and tracking	60%
PROF	**Professionalism**	**5%**
AdQualif	a. Adequate qualifications to perform your role	30%
AdCert	b. Adequate certifications of processes followed (e.g. ISO 9000, ISO 9001)	15%
AdContin	c. Adequate continuity by team members through all stages of the project	25%
AdSupComMng	d. Adequate support and commitment of management	30%
SUPSER	**Support Service**	**15%**
AvHlpDsk	a. Availability of a help-desk	20%
SupProv	b. Support is provided	20%
DefLogTrk	c. Defect logging and tracking system used	20%
SLA's	d. Service level agreements in place with the users/clients	20%
SufFeedbck	e. Sufficient feedback to users/clients provided – on any aspects such as problems, bugs or enhancements submitted	20%
CLRE	**Section B: Client Relationship**	**35%**

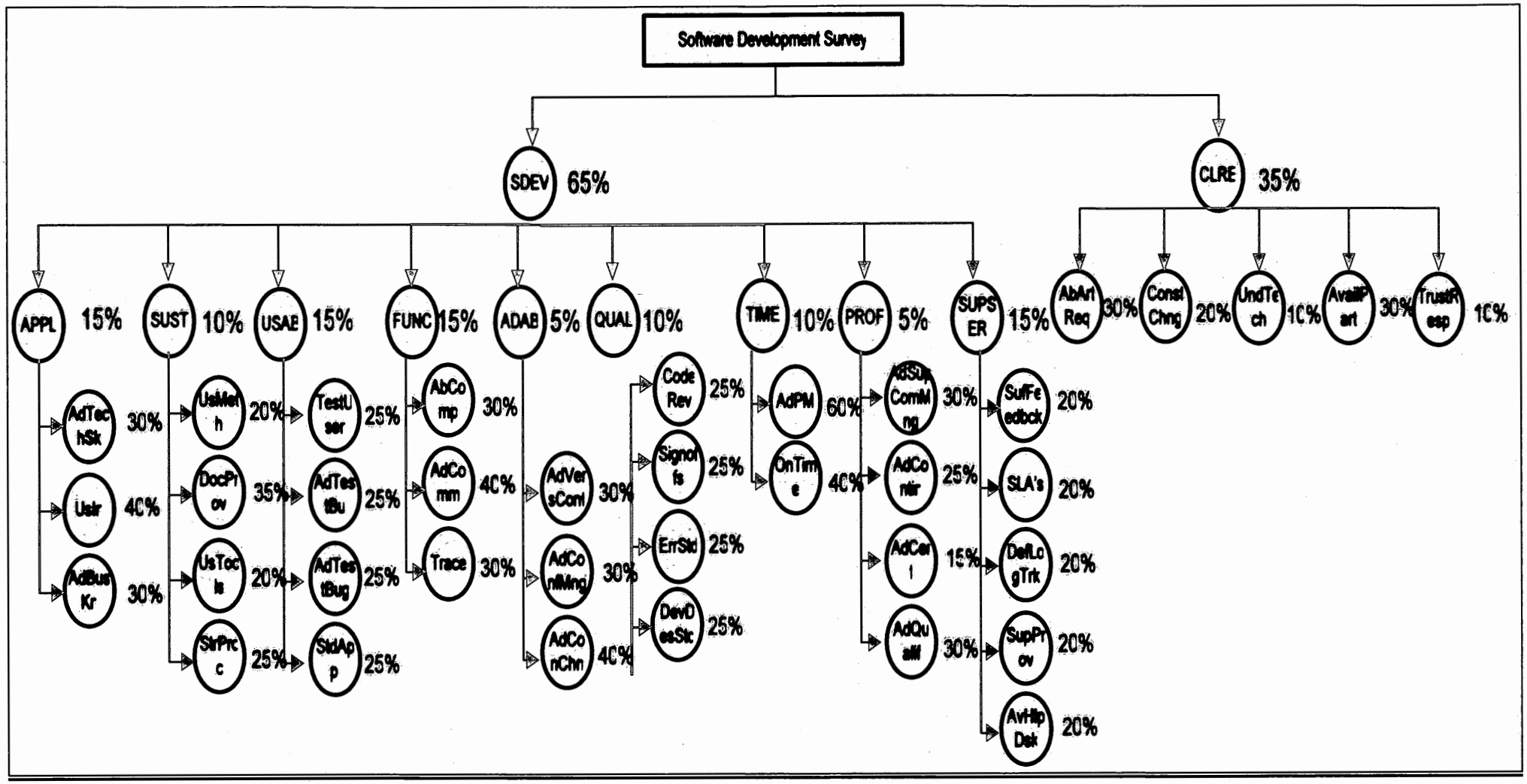

Figure 6.2 : Development Survey Hierarchical Structure and Percentage Weightings Allocation

Acronym	Software Development	100% Ratio	Weight	TOTAL SCORE	TOTAL AVG	Company 1				Company 2					
						Score	Response 1		Avg	Score	Response 1		Response 2		Avg
AdComm	b. Adequate communication and liaising with users	**72.22%**	40%	28.89%	72.22%	33.33%	5	83.33%	83.33%	40.00%	6	100.00%	6	100.00%	100.00%
AbComp	c. Ability to make suitable compromises on requirements and development alternatives.	**79.63%**	30%	23.89%	79.63%	25.00%	5	83.33%	83.33%	30.00%	6	100.00%	6	100.00%	100.00%
ADAB	**Adaptability**	**74.44%**	**5%**	**3.72%**	**0.00%**	**4.67%**		**0.00%**	**0.00%**	**3.33%**		**0.00%**		**0.00%**	**0.00%**
AdConChn	a. Adequate control of changes – Change management (change control forms, sign-offs and additional costs, etc)	**72.22%**	40%	28.89%	72.22%	33.33%	5	83.33%	83.33%	26.67%	4	66.67%	4	66.67%	66.67%
AdConfMng	b. Adequate Configuration management	**72.22%**	30%	21.67%	72.22%	30.00%	6	100.00%	100.00%	20.00%	4	66.67%	4	66.67%	66.67%
AdVersCont	c. Adequate Version control management	**79.63%**	30%	23.89%	79.63%	30.00%	6	100.00%	100.00%	20.00%	4	66.67%	4	66.67%	66.67%
QUAL	**Quality**	**64.81%**	**10%**	**6.48%**	**0.00%**	**5.00%**		**0.00%**	**0.00%**	**7.92%**		**0.00%**		**0.00%**	**0.00%**
DevDesStd	a. Existence of design and development standards	**68.52%**	25%	17.13%	68.52%	8.33%	2	33.33%	33.33%	20.83%	5	83.33%	5	83.33%	83.33%
ErrStd	b. Error handling standards applied	**81.48%**	25%	20.37%	81.48%	25.00%	6	100.00%	100.00%	20.83%	5	83.33%	5	83.33%	83.33%
Signoffs	c. Sign-offs obtained at different stages in the project	**66.67%**	25%	16.67%	66.67%	4.17%	1	16.67%	16.67%	25.00%	6	100.00%	6	100.00%	100.00%
CodeRev	d. Code reviews performed	**42.59%**	25%	10.65%	42.59%	12.50%	3	50.00%	50.00%	12.50%	3	50.00%	3	50.00%	50.00%
TIME	**Timely**	**70.37%**	**10%**	**7.04%**	**0.00%**	**5.00%**		**0.00%**	**0.00%**	**8.33%**		**0.00%**		**0.00%**	**0.00%**
OnTime	a. Timely delivery of application to the user	**81.48%**	40%	32.59%	81.48%	20.00%	3	50.00%	50.00%	33.33%	5	83.33%	5	83.33%	83.33%
AdPM	b. Adequate project management – progress reports, meetings, project plan and tracking	**62.96%**	60%	37.78%	62.96%	30.00%	3	50.00%	50.00%	50.00%	5	83.33%	5	83.33%	83.33%
PROF	**Professionalism**	**74.44%**	**5%**	**3.72%**	**0.00%**	**3.96%**		**0.00%**	**0.00%**	**3.79%**		**0.00%**		**0.00%**	**0.00%**
AdQualif	a. Adequate qualifications to perform your role	**81.48%**	30%	24.44%	81.48%	30.00%	6	100.00%	100.00%	20.00%	4	66.67%	4	66.67%	66.67%
AdCert	b. Adequate certifications of processes followed (e.g. ISO 9000, ISO 9001)	**50.00%**	15%	7.50%	50.00%	7.50%	3	50.00%	50.00%	10.00%	4	66.67%	4	66.67%	66.67%
AdContin	c. Adequate continuity by team members through all stages of the project	**72.22%**	25%	18.06%	72.22%	16.67%	4	66.67%	66.67%	20.83%	5	83.33%	5	83.33%	83.33%

Acronym	Software Development	100% Ratio	Weight	TOTAL SCORE	TOTAL AVG	Company 1				Company 2				
						Score	Response 1		Avg	Score	Response 1		Response 2	Avg
CLRE	Section B: Client Relationship	60.56%	35%	21.19%	0.00%	19.83%		0.00%	0.00%	29.75%		0.00%	0.00%	0.00%
	CRM				0.00%			0.00%	0.00%			0.00%	0.00%	0.00%
AbArtReq	a. Client's ability to articulate requirements	57.41%	30%	17.22%	57.41%	15.00%	3	50.00%	50.00%	25.00%	5	83.33%	5 83.33%	83.33%
ConstChng	b. Client's willingness to constrain changes	64.81%	20%	12.96%	64.81%	16.67%	5	83.33%	83.33%	20.00%	6	100.00%	6 100.00%	100.00%
UndTech	c. Client's understanding of technologies used	51.85%	10%	5.19%	51.85%	3.33%	2	33.33%	33.33%	6.67%	4	66.67%	4 66.67%	66.67%
AvailPart	d. Client's availability and participation in all the stages of the projects	61.11%	30%	18.33%	61.11%	15.00%	3	50.00%	50.00%	25.00%	5	83.33%	5 83.33%	83.33%
TrustResp	e. Client's trust and respect towards your abilities	68.52%	10%	6.85%	68.52%	6.67%	4	66.67%	66.67%	8.33%	5	83.33%	5 83.33%	83.33%

Table 6.5 : Company 1 and 2 development responses mapped to the hierarchical components and the weighting calculations

Acronym	Software Development	100% Ratio	Weight	TOTAL SCORE	TOTAL AVG	Company 3					
						Score	Response 1		Response 2		Avg
Trace	a. Trace ability from requirements to design to development to testing	**72.22%**	30%	21.67%	72.22%	15.00%	3	50.00%	3	50.00%	50.00%
AdComm	b. Adequate communication and liaising with users	**72.22%**	40%	28.89%	72.22%	33.33%	5	83.33%	5	83.33%	83.33%
AbComp	c. Ability to make suitable compromises on requirements and development alternatives.	**79.63%**	30%	23.89%	79.63%	30.00%	6	100.00%	6	100.00%	100.00%
ADAB	**Adaptability**	**74.44%**	**5%**	**3.72%**	**0.00%**	**3.83%**		**0.00%**		**0.00%**	**0.00%**
AdConChn	a. Adequate control of changes – Change management (change control forms, sign-offs and additional costs, etc)	**72.22%**	40%	28.89%	72.22%	26.67%	5	83.33%	3	50.00%	66.67%
AdConfMng	b. Adequate Configuration management	**72.22%**	30%	21.67%	72.22%	25.00%	5	83.33%	5	83.33%	83.33%
AdVersCont	c. Adequate Version control management	**79.63%**	30%	23.89%	79.63%	25.00%	5	83.33%	5	83.33%	83.33%
QUAL	**Quality**	**64.81%**	**10%**	**6.48%**	**0.00%**	**4.79%**		**0.00%**		**0.00%**	**0.00%**
DevDesStd	a. Existence of design and development standards	**68.52%**	25%	17.13%	68.52%	12.50%	3	50.00%	3	50.00%	50.00%
ErrStd	b. Error handling standards applied	**81.48%**	25%	20.37%	81.48%	14.58%	3	50.00%	4	66.67%	58.33%
Signoffs	c. Sign-offs obtained at different stages in the project	**66.67%**	25%	16.67%	66.67%	10.42%	3	50.00%	2	33.33%	41.67%
CodeRev	d. Code reviews performed	**42.59%**	25%	10.65%	42.59%	10.42%	3	50.00%	2	33.33%	41.67%
TIME	**Timely**	**70.37%**	**10%**	**7.04%**	**0.00%**	**5.33%**		**0.00%**		**0.00%**	**0.00%**
OnTime	a. Timely delivery of application to the user	**81.48%**	40%	32.59%	81.48%	33.33%	5	83.33%	5	83.33%	83.33%
AdPM	b. Adequate project management – progress reports, meetings, project plan and tracking	**62.96%**	60%	37.78%	62.96%	20.00%	2	33.33%	2	33.33%	33.33%
PROF	**Professionalism**	**74.44%**	**5%**	**3.72%**	**0.00%**	**3.92%**		**0.00%**		**0.00%**	**0.00%**
AdQualif	a. Adequate qualifications to perform your role	**81.48%**	30%	24.44%	81.48%	25.00%	6	100.00%	4	66.67%	83.33%
AdCert	b. Adequate certifications of processes followed (e.g. ISO 9000, ISO 9001)	**50.00%**	15%	7.50%	50.00%	2.50%	1	16.67%	1	16.67%	16.67%
AdContin	c. Adequate continuity by team members through all stages of the project	**72.22%**	25%	18.06%	72.22%	20.83%	6	100.00%	4	66.67%	83.33%

Acronym	Software Development	100% Ratio	Weight	TOTAL SCORE	TOTAL AVG	Company 4									
						Score	Response 1		Response 2		Response 3		Response 4		Avg
	Development Overall Score			**68.64%**		**64.08%**									
SDEV	**Section A: Software Development**		**65%**	**47.44%**		**44.69%**									
APPL	**Applicability**	**79.81%**	**15%**	**11.97%**		**10.94%**									
AdBusKn	a. Adequate business knowledge	**77.78%**	30%	23.33%	77.78%	20.00%	4	66.67%	**3**	50.00%	4	66.67%	5	83.33%	66.67%
UsIn	b. User involvement in determining their requirements	**75.93%**	40%	30.37%	75.93%	26.67%	**3**	50.00%	5	83.33%	5	83.33%	**3**	50.00%	66.67%
AdTechSk	c. Adequate technical skills to translate business knowledge to a useful application	**87.04%**	30%	26.11%	87.04%	26.25%	5	83.33%	6	100.00%	5	83.33%	5	83.33%	87.50%
SUST	**Sustainability**	**62.87%**	**10%**	**6.29%**	**0.00%**	**6.52%**		**0.00%**		**0.00%**		**0.00%**		**0.00%**	**0.00%**
StrProc	a. Structured process followed – requirements definition, analysis, design, test, implementation	**77.78%**	25%	19.44%	77.78%	18.75%	4	66.67%	5	83.33%	6	100.00%	**3**	50.00%	75.00%
UsTools	b. Usage of tools to assist with process - any case tools, data-modeling tools, re-engineering tools, etc.	**35.19%**	20%	7.04%	35.19%	8.33%	**1**	16.67%	4	66.67%	**3**	50.00%	**2**	33.33%	41.67%
DocProv	c. Documentation provided – User Requirements, Specifications, Technical Specifications, training manuals, On-line help, etc.	**64.81%**	35%	22.69%	64.81%	24.79%	4	66.67%	**3**	50.00%	5	83.33%	5	83.33%	70.83%
UsMeth	d. Usage of a specific methodology	**68.52%**	20%	13.70%	68.52%	13.33%	**3**	50.00%	4	66.67%	5	83.33%	4	66.67%	66.67%
USAB	**Usability**	**77.31%**	**15%**	**11.60%**	**0.00%**	**10.78%**		**0.00%**		**0.00%**		**0.00%**		**0.00%**	**0.00%**
StdAppl	a. Development and design standards applied – same look and feel across screens and certain coding standards in place	**85.19%**	25%	21.30%	85.19%	20.83%	5	83.33%	4	66.67%	6	100.00%	5	83.33%	83.33%
AdTestBug	b. Adequate testing done for any defects/bugs (unit testing)	**77.78%**	25%	19.44%	77.78%	17.71%	**3**	50.00%	4	66.67%	5	83.33%	5	83.33%	70.83%
AdTestBu	c. Adequate testing done for the fit to business requirements/functionality (integration/functionality testing)	**81.48%**	25%	20.37%	81.48%	16.67%	**3**	50.00%	4	66.67%	4	66.67%	5	83.33%	66.67%
TestUser	d. Testing done by the users – user acceptance testing	**64.81%**	25%	16.20%	64.81%	16.67%	**3**	50.00%	5	83.33%	4	66.67%	4	66.67%	66.67%
FUNC	**Functionality**	**74.44%**	**15%**	**11.17%**	**0.00%**	**8.63%**		**0.00%**		**0.00%**		**0.00%**		**0.00%**	**0.00%**

Acronym	Software Development	100% Ratio	Weight	TOTAL SCORE	TOTAL AVG	Company 4									
						Score	Response 1		Response 2		Response 3		Response 4		Avg
AdSupComMng	d. Adequate support and commitment of management	81.48%	30%	24.44%	81.48%	21.25%	4	66.67%	4	66.67%	5	83.33%	4	66.67%	70.83%
SUPSER	Support Service	73.33%	15%	11.00%	0.00%	10.00%		0.00%		0.00%		0.00%		0.00%	0.00%
AvHlpDsk	a. Availability of a help-desk	64.81%	20%	12.96%	64.81%	14.17%	4	66.67%	4	66.67%	6	100.00%	3	50.00%	70.83%
SupProv	b. Support is provided	90.74%	20%	18.15%	90.74%	16.67%	4	66.67%	5	83.33%	6	100.00%	5	83.33%	83.33%
DefLogTrk	c. Defect logging and tracking system used	61.11%	20%	12.22%	61.11%	10.00%	1	16.67%	4	66.67%	4	66.67%	3	50.00%	50.00%
SLA's	d. Service level agreements in place with the users/clients	77.78%	20%	15.56%	77.78%	15.00%	3	50.00%	5	83.33%	5	83.33%	5	83.33%	75.00%
SufFeedbck	e. Sufficient feedback to users/clients provided – on any aspects such as problems, bugs or enhancements submitted	72.22%	20%	14.44%	72.22%	10.83%	2	33.33%	4	66.67%	5	83.33%	2	33.33%	54.17%
					0.00%			0.00%		0.00%		0.00%		0.00%	0.00%
CLRE	Section B: Client Relationship	60.56%	35%	21.19%	0.00%	19.40%		0.00%		0.00%		0.00%		0.00%	0.00%
	CRM				0.00%			0.00%		0.00%		0.00%		0.00%	0.00%
AbArtReq	a. Client's ability to articulate requirements	57.41%	30%	17.22%	57.41%	17.50%	2	33.33%	3	50.00%	4	66.67%	5	83.33%	58.33%
ConstChng	b. Client's willingness to constrain changes	64.81%	20%	12.96%	64.81%	10.00%	2	33.33%	2	33.33%	3	50.00%	5	83.33%	50.00%
UndTech	c. Client's understanding of technologies used	51.85%	10%	5.19%	51.85%	5.00%	3	50.00%	2	33.33%	4	66.67%	3	50.00%	50.00%
AvailPart	d. Client's availability and participation in all the stages of the projects	61.11%	30%	18.33%	61.11%	17.50%	3	50.00%	4	66.67%	5	83.33%	2	33.33%	58.33%
TrustResp	e. Client's trust and respect towards your abilities	68.52%	10%	6.85%	68.52%	5.42%	2	33.33%	2	33.33%	4	66.67%	5	83.33%	54.17%

Table 6.7 : Company 4 development responses mapped to the hierarchical components and the weighting calculations

- Quality – c. Sign-offs obtained at different stages in the project – 66.67%

In summary, we can conclude that the development team think that software applications delivery are successful which confirms the hypothesis which states that developers think there is a low degree of software application failure, therefore the hypothesis is accepted and seen as true.

What we should be mindful of is:

- That these respondents are development team members of small software development organisations and their view may not reflect the views of developers of larger software development organisations.
- Only 9 development members, spread over 4 software development companies, responded, which is a limited sample size.
- One software development company's score was much higher than the rest, which influenced the general average and caused the other companies to fall below the average.

What is interesting to note is that the developers do not rate the client's ability to translate their business requirements and understanding of software development technologies highly, which is an area of concern because it could have a cascading effect on the quality and type of software applications developed, and therefore the software development companies should make it their responsibility of educating and informally training their clients on these aspects.

6.4 Hypothesis Three - Clients and Development Teams Perspectives Differ

Hypothesis 3 (HYP3):

"Development teams and clients perspective differs when it comes to defining success of software applications."

6.4.1 Analysis approach

The client and development survey questionnaires were constructed for their particular audience, taking into consideration their specific knowledge and 'language, which ultimately leads to different perspectives. To

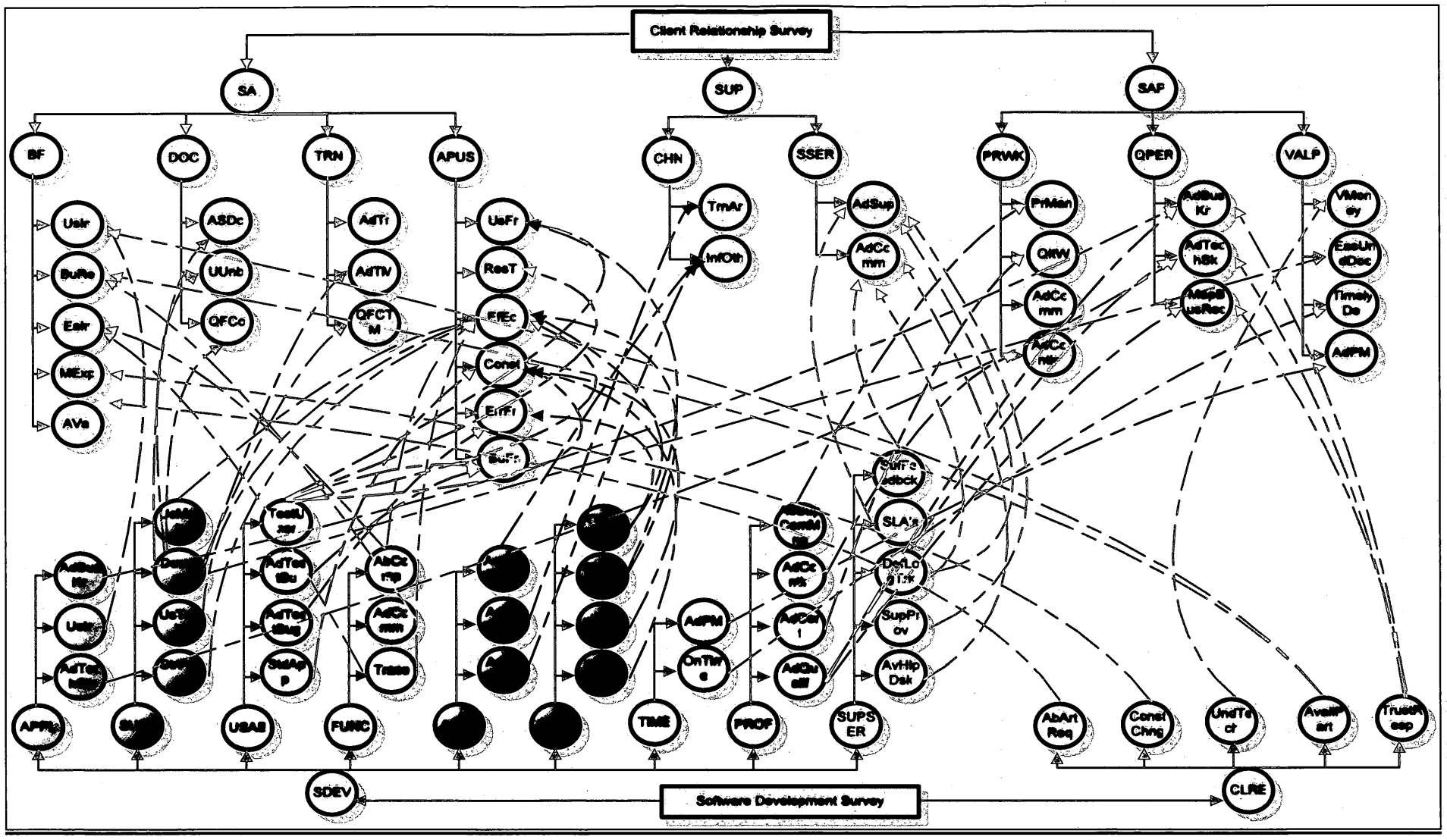

Figure 6.3 : Client and Development Mapping Diagram

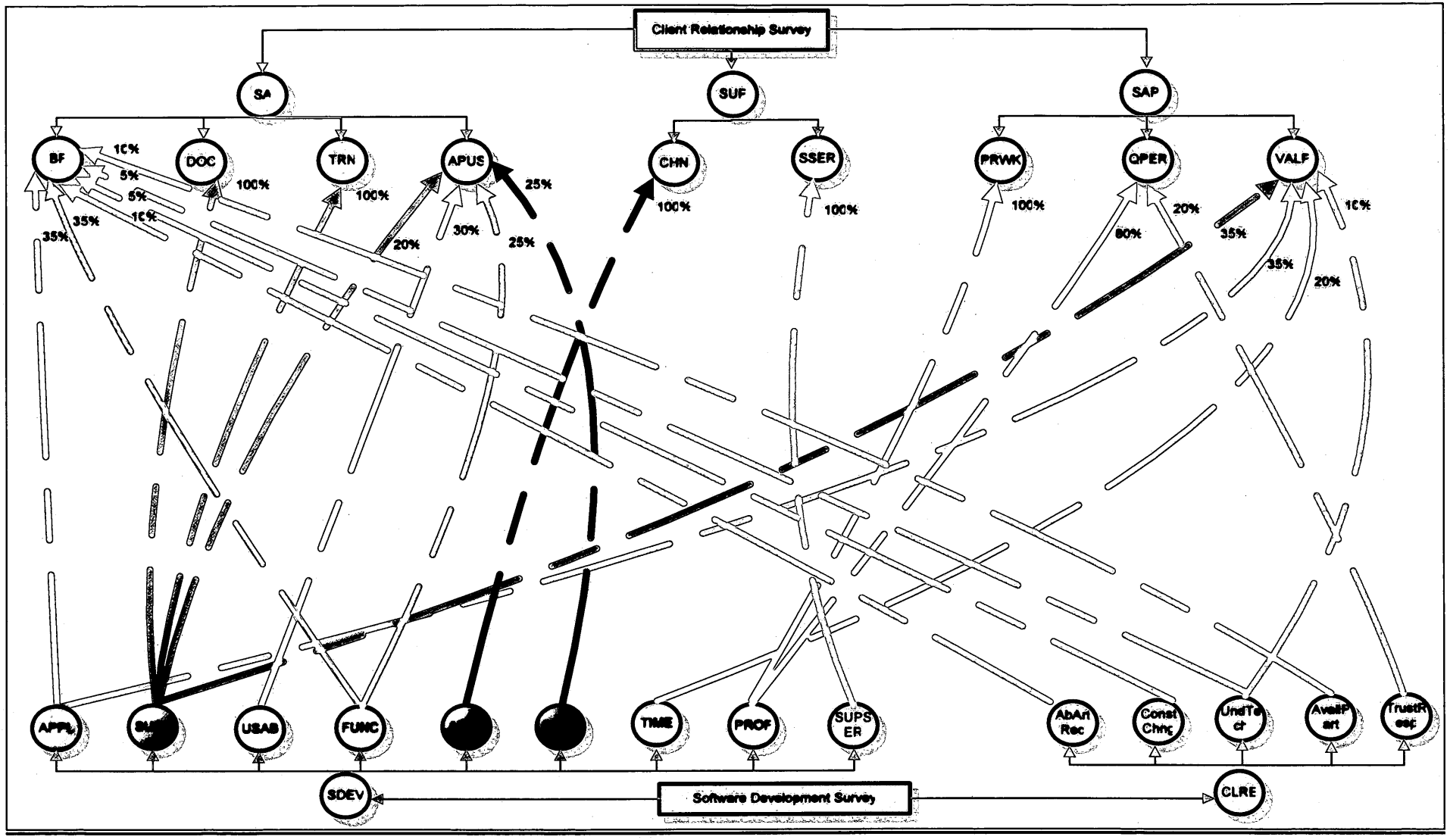

Figure 6.4 : Simplified mapping with new weighting allocations

Development components						Applicability			Sustainability			Usability		
						Weight	Survey Result of Dev	Calc Value	Weight	Survey Result of Dev	Calc Value	Weight	Survey Result of Dev	Calc Value
Acronym	Client Relationship Survey components	Client weight	Survey Result of Client	Dev Value	Audit Check									
	Client Overall score		70.13%	70.14%										
SA	Section A: Software Application/Solution	50%	72.77%	69.48%										
BF	Business fit	40%	77.08%	71.68%	100.00%	35.00%	79.81%	27.94%						
DOC	Documentation	10%	69.79%	62.87%	100.00%				100.00%	62.87%	62.87%			
TRN	Training	10%	67.08%	62.87%	100.00%				100.00%	62.87%	62.87%			
APUS	Application Usage	40%	70.63%	70.58%	100.00%				20.00%	62.87%	12.57%	30.00%	77.31%	23.19%
SUP	Section B: Support	25%	68.44%	73.67%										
CHN	Changes	30%	60.42%	74.44%	100.00%									
SSERV	Support service	70%	71.88%	73.33%	100.00%									
SAP	Section C: Software Application Provider	25%	66.54%	67.95%										
PRWK	Professional Work	20%	68.85%	74.44%	100.00%									
QPER	Qualified Personnel	20%	73.54%	73.26%	100.00%									
VALP	Value Provided	60%	63.44%	64.01%	100.00%	35.00%	79.81%	27.94%	35.00%	62.87%	22.00%			

Table 6.8 : Client and development components mapping calculations

Development components						Timely			Professionalism			Support Service		
						Weight	Survey Result of Dev	Calc Value	Weight	Survey Result of Dev	Calc Value	Weight	Survey Result of Dev	Calc Value
Acronym	Client Relationship Survey components	Client weight	Survey Result of Client	Dev Value	Audit Check									
	Client Overall score		70.13%	70.14%										
SA	Section A: Software Application/Solution	50%	72.77%	69.48%										
BF	Business fit	40%	77.08%	71.68%	100.00%									
DOC	Documentation	10%	69.79%	62.87%	100.00%									
TRN	Training	10%	67.08%	62.87%	100.00%									
APUS	Application Usage	40%	70.63%	70.58%	100.00%									
SUP	Section B: Support	25%	68.44%	73.67%										
CHN	Changes	30%	60.42%	74.44%	100.00%									
SSERV	Support service	70%	71.88%	73.33%	100.00%							100.00%	73.33%	73.33%
SAP	Section C: Software Application Provider	25%	66.54%	67.95%										
PRWK	Professional Work	20%	68.85%	74.44%	100.00%				100.00%	74.44%	74.44%			
QPER	Qualified Personnel	20%	73.54%	73.26%	100.00%				80.00%	74.44%	59.56%			
VALP	Value Provided	60%	63.44%	64.01%	100.00%	20.00%	70.37%	14.07%						

Table 6.10 : Client and development components mapping calculations(cont.)

Development components						AvailPart			TrustResp		
						Weight	Survey Result of Dev	Calc Value	Weight	Survey Result of Dev	Calc Value
Acronym	Client Relationship Survey components	Client weight	Survey Result of Client	Dev Value	Audit Check						
	Client Overall score		70.13%	70.14%							
SA	Section A: Software Application/Solution	50%	72.77%	69.48%							
BF	Business fit	40%	77.08%	71.68%	100.00%	10.00%	61.11%	6.11%			
DOC	Documentation	10%	69.79%	62.87%	100.00%						
TRN	Training	10%	67.08%	62.87%	100.00%						
APUS	Application Usage	40%	70.63%	70.58%	100.00%						
SUP	Section B: Support	25%	68.44%	73.67%							
CHN	Changes	30%	60.42%	74.44%	100.00%						
SSERV	Support service	70%	71.88%	73.33%	100.00%						
SAP	Section C: Software Application Provider	25%	66.54%	67.95%							
PRWK	Professional Work	20%	68.85%	74.44%	100.00%						
QPER	Qualified Personnel	20%	73.54%	73.26%	100.00%				20.00%	68.52%	13.70%
VALP	Value Provided	60%	63.44%	64.01%	100.00%						

Table 6.12 : Client and development components mapping calculations(cont.)

possible. Furthermore it does not require such strict configuration and version control and project management methods, because of the size and number of the projects.

- The management of smaller software development organisations also have a more hands-on approach because invariably they have to be involved in all stages of the software development life cycle as there is less specialisation and a smaller number of people are involved. This is also true for the rest of the software development team, because each team member is involved in all the phases, from requirement analysis, to design, development, testing and implementation, where in larger software development companies there tends to be more specialised roles and responsibilities of a team member to the different phases, i.e. an analyst only does analysis, a programmer only does coding, etc. This could lead to communication gaps in the transition process between phases and discontinuity in translating the requirements into a software application, which then results in a mismatch in the business fit.

Although the client views the business fit as more satisfactory than the development team, the values are quite similar, which indicates that their views on business fit do not differ much.

2. Documentation

The clients are slightly more satisfied with documentation than the development team, although the values are similar, which indicates that their views do not really differ much. The development value is less than 66.67% which does indicate an area which could be improved. This also corresponds with the areas in section 6.2.2 and 6.3.2 which were highlighted as areas of improvement.

3. Training

Again the clients are slightly more satisfied with training than the development team, but the values are very similar, which also indicates that their views do not differ that much. However both values are close to the marginal value of 66,67%, which corresponds with the section 6.2.2 summary of training being an area in the client survey analysis which could be improved on.

4. Application Usage

Here the values are virtually equal, which indicates that their views do not differ. Both values are quite high, which from the development team's perspective makes sense, because this is the core requirement of a software development organisation to produce useable applications.

5. Changes

The development team's view is higher than the client's view, which indicates a certain level of disagreement. The client's rating of 60.42% also corresponds with the summary in section 6.2.2 that this aspect is not

applications. What should be mentioned again is that this could possibly only apply to small software development organisations, where there is a much more client-orientated focus.

Client Relationship Survey Open-ended questions	Company 1	
	Response 1	**Response 2**
	Also, communication around progress is a small irritation. Programmers are expensive, and normally the job needs to be done in order to meet certain deadlines. Lack of feedback around progress leads to overruns in project costs, as well as the 'unknowing' of where the programmer is 'at'. In summary then: 1. The work must first 'spec'd' properly, understanding exactly what the client wants 2. the costs must be estimated and a deadline set for the project 3. progress reports, costs to date and demos (if possible) must be given to the client 4. questions must be asked by the developer to ensure they are still on track in terms of understanding the client, not just assuming that by doing something in a particular way is acceptable (eg – format of reports, not cross checking totals to see if they balance etc) 5. A final signoff should be done by way of a proper demo	
Software Application Provider on the whole Would you recommend the Software Application Provider to other businesses?	Yes	Yes
If No, why not?		
What does the Software Application Provider do well?	I find particularly that the manager himself has a good commitment toward solving our challenges – I enjoy his enthusiasm as well as respect his expertise. Company 1 is very proficient, and remains a great support to our business. All staff is friendly, courteous and has a great manner of working with people. They also have a depth to their business – IT, vendor, programming – they have a good understanding of process manufacturing which is most helpful. All round they are a very good outfit and I would recommend them.	Offer cost effective solutions to Plant Maintenance problems.

Table 6.14 : Open-ended responses of clients of Company 1

Client Relationship Survey Open-ended questions	Company 3		
	Response 1	**Response 2**	**Response 3**
Designation	Financial Director	Analyst IT	Regional Accountant
Your involvement	User	Project Manager	User
Section D: General			
Software Application/Solution on the whole Overall, would you rate the software application/solution as a success?	Yes	Yes	Yes
If No, what would you say were the main reasons for the failure?			
What do you consider the most important aspect of a software application/solution?		It gives the end user lots of flexibility and also less time spend.	User friendly and it must meet all the business needs that it is designed to cater for.
Software Application Provider on the whole Would you recommend the Software Application Provider to other businesses?	Yes	Yes	Yes
If No, why not?			
What does the Software Application Provider do well?		Communication between the provider and us is very satisfactory. The packages that we use from them are very user friendly	Created a product that integrates with our industry specific software for accounting applications.

Table 6.16 : Open-ended responses of clients of Company 3

- Project management aspects such as: planning, cost monitoring and estimations with specific deliverables e.g.: demos, user sign-offs at the end of each phase.
- Provide value to the business by providing flexibility and time savings to the business.
- Delivery of bug-free applications.
- User-friendly applications.
- Adequate support service.
- Changes to be implemented efficiently and effectively.
- Continuous improvements through enhancement and additional functionalities at acceptable costs.

The clients' view of what activities the software application provider does well can be summarised as follows:

- Long-lasting relationship between the clients and the software application providers.
- Management commitment.
- Professionalism of staff – friendly, courteous.
- Level of business knowledge.
- In depth Information Technology (IT) knowledge.
- Cost effective applications developed.
- Quality applications developed that easily integrates with other applications.

There seems to be a discrepancy between the aspects that the clients view as important for successful software development and the aspects their software application providers does well. This could potentially be due to 'softer people' issues where the clients value the relationship they have built with the software application providers and therefore did not want to influence this relationship negatively.

Furthermore, it also indicates that the software development organisations need to improve and concentrate on those factors important to their clients.

It is very clear that the factors the clients view as important for the success of software development applications matches those defined by the literature, and therefore we can surmise that the survey analysis done is a true reflection of the clients satisfaction levels belief in the success/failure of software development applications.

Software Development Survey Open-ended questions	Company 1	Company 2	
	Response 1	Response 1	Response 2
What do you consider the most important aspect of software development?	Planning and designing the software i.e. using Patterns to solve problems and using UML to assist in this. Testing the software.	Accuracy, without that the software would never work and the client would never be happy with your service	Meeting the clients exact and specific requirements coupled with a sound well tested and stable environment. You can meet the clients needs to the last bit but if the programs fall over they still can't use them
	Testing the software.		
What does the Software Development team do well?	Writing code.	Develop software	Analysis of requirements documents and coding accordingly

Table 6.18 : Open-ended responses of development team of Company 1 and 2

Software Development Survey Open-ended questions	Company 4			
	Response 1	**Response 2**	**Response 3**	**Response 4**
Designation	Programmer	Developer	Developer	
Industry application is developed for	Unit Trust	FINANCIAL – FUND ADIMINISTRATION	Financial / Asset Management	Financial
Your involvement	Developer	Developer	Developer	Software programmer, analyst.
Section C: General				
Software Development Approach				
Overall, would you rate the software development as a success?	Yes	Yes	Yes	Yes
If No, what would you say were the main reasons for the failure?				
What do you consider the most important aspect of software development?	Communication	Knowledge and of the participants involved in their different roles as well as in depth knowledge in the product and product domain as such. Lack of training on the actual project is most probably the downfall of most software products. A developer working on a financial package will be able to write the code without much knowledge of the product involved, but this lack of training on the product and subject which he is busy coding, definitely leaves scope for errors and logs being returned time and time again. Lack of communication and a lack of mutual respect between developers and analysts is also a great concern in this industry and the failure of many projects. Software development is supposed to be a team effort.	Quality Control & timely delivery	Quality

6.5.4 Development conclusion

In summary the factors that development team view as important aspects of software development applications are the following:

- Adequate business knowledge to translate the requirements to proper specifications and developing an application that fulfils the client's business requirements.
- Adequate analysis skills to determine the client's business requirements.
- Communication.
- Project management aspects such as: Planning.
- Development of bug and error-free applications
- Development of user-friendly applications
- Adequate testing of the applications (unit, functionality and integration testing).
- Development of simple and easy to use applications.
- Adequate logical problem-solving skills to code the applications.
- Proper training of developers on the business knowledge of the client's industry to ensure accurate mapping of requirements into applications.
- Respect between development colleagues.
- Timely delivery of the applications to the clients.
- Quality control.

The development teams' view of what activities they do well can be summarised as follows:

- Design and develop
- Analysis of business requirements.
- Quick delivery of base functionality to the clients.
- Version control.

The factors that the development view as important aspects in successful software development applications are very similar to the factors listed by the clients in section 6.5.2 (with some minor differences) and also matches the factors defined in the literature, and therefore we can conclude that the survey analysis done is a true reflection of the development teams' belief in its ability to deliver successful applications.

7 CONCLUSION AND RECOMMENDATIONS

In the previous chapter the data collected for this research study was explained and analysed according to a defined approach to facilitate the interpretation of the data. This chapter will focus on synthesising the previous chapter's findings to provide a succinct conclusion. This chapter will consist of the following sections:

- Client rating conclusions and recommendations.
- Development rating conclusions and recommendations.
- Comparing the clients and development teams' views in a graphical representation and recommendations.
- Mapping the different success factors of the literature, clients and development teams. The following mappings will be done:
  - The success factors as defined by the literature mapped to those defined by the clients in this survey
  - The success factors as defined by the literature mapped to those defined by the development teams in this survey.
  - The success factors defined by the clients mapped to those defined by the development teams.

7.1 Client Rating Summary

To calculate the overall score for each company, as well as the total score for the client perspective, the hierarchical weighting structure was used and each client's factor weighting was rolled up to a 100% contribution. The results of the total client perspective and each company's overall score broken into the different client factors are graphically represented in figure 7.1 below.

Furthermore the different boundary lines provide insight into the status of each company. The blue boundary line represents the total client score (70.13%), the yellow line represents the success/failure boundary of 50%, where anything above 50% indicates a success rate. The green line represents the marginal boundary of 66.67%, which implies that anything between 50% and 66.67% is only relative successful and the software development companies in this range should analyse the details contributing to their client's satisfaction levels.

Client Relationship Survey Results	Total	Company 1	Company 2	Company 3	Company 4
Business fit	**14.55%**	**13.53%**	**13.53%**	15.13%	**14.53%**
Documentation	**3.64%**	**3.38%**	**3.38%**	3.78%	**3.63%**
Training	**3.64%**	**3.38%**	**3.38%**	3.78%	**3.63%**
Application Usage	**14.55%**	**13.53%**	**13.53%**	15.13%	**14.53%**
Changes	**5.13%**	**4.19%**	5.56%	5.77%	**4.25%**
Support service	**11.98%**	**9.77%**	12.98%	13.47%	**9.92%**
Professional Work	**3.33%**	**2.86%**	**3.31%**	3.84%	**2.76%**
Qualified Personnel	**3.33%**	**2.86%**	**3.31%**	3.84%	**2.76%**
Value Provided	**9.98%**	**8.59%**	**9.93%**	11.52%	**8.28%**

Table 7.1 : Factors of each company that are below the total client score

In summary, the majority of the companies scores are not much lower than the total score, and the only aspects that have a significant gaps that could indicate areas where the software development companies needs to investigate and improve upon are:

- Business fit
- Application usage
- Changes
- Value provided

Only one company (Company 2) is above the total development score of 68.64%, all the companies are above 50%, the rest of the companies (Company 1, 3 and Company 4) are in the marginal range, which indicates that they should look at improving certain development aspects.

Furthermore each development factor for each company is recorded and this can be used as a yardstick to assist the software companies in prioritising and focusing on the areas that need attention. Table 7.2 indicates for each company those factors that have a lower score than the total development average. All values below the total score are in red.

Software Development	Total	Company 1	Company 2	Company 3	Company 4
Applicability	7.12%	6.71%	8.73%	6.53%	6.70%
Sustainability	4.74%	4.47%	5.82%	4.35%	4.47%
Usability	7.12%	6.71%	8.73%	6.53%	6.70%
Functionality	7.12%	6.71%	8.73%	6.53%	6.70%
Adaptability	2.37%	2.24%	2.91%	2.18%	2.23%
Quality	4.74%	4.47%	5.82%	4.35%	4.47%
Timely	4.74%	4.47%	5.82%	4.35%	4.47%
Professionalism	2.37%	2.24%	2.91%	2.18%	2.23%
Support Service	7.12%	6.71%	8.73%	6.53%	6.70%
Articulate Req	6.36%	5.95%	8.93%	5.08%	5.82%
Constrain Chnges	4.24%	3.97%	5.95%	3.38%	3.88%
Understand tech	2.12%	1.98%	2.98%	1.69%	1.94%
Participate in stages	6.36%	5.95%	8.93%	5.08%	5.82%
Trust & Respect	2.12%	1.98%	2.98%	1.69%	1.94%

Table 7.2 : Factors of each company that are below the total development score

In summary, looking at the factors that have a relative large gap between their values and the total average, there are a few that can be identified as areas on which the software development companies should focus and improve on (also listed in order of priority):

- Applicability
- Usability
- Functionality
- Support Service

There are also certain factors that the software development companies report as lacking on the clients side, such as:

- Articulation of their requirements
- Ability to constrain changes
- Their understanding of the technologies used
- Their trust and respect towards the software development staff

In summary the client and development view do not differ on software development success factor rates, because the overall client score is 70.13% and the overall development score is 70.14%, with only minor differences in the different contributing factors. The biggest gap exist within the 'Changes' factor, with the client score being 60.42% and the development score of 74.44%, which indicate that the development team have a more positive view than the clients have for this aspect. Change management has always been a difficult aspect and these values results highlight this problem-area.

The gap did not influence a major gap in the overall scores because the different weightings of the various aspects are applied.

not refer to directly, but it could also be that the literature does not include the support service as part of the software development cycle. Figure 7.5 below shows the mapping between the success factors defined by the literature and the success factors the development team mentioned in the survey.

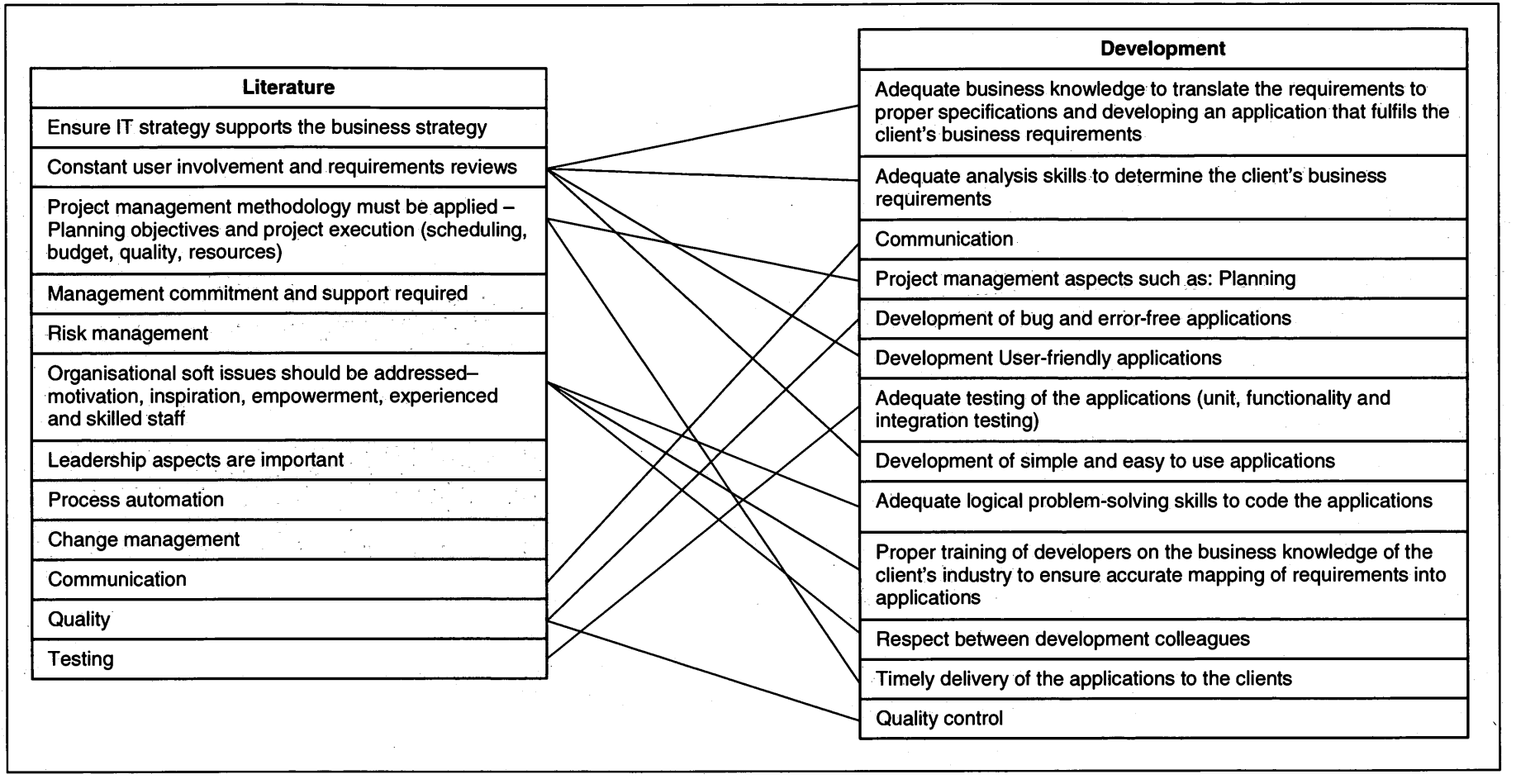

Figure 7.5 : Development view mapped to literature

The factors that do not map are highlighted in blue and orange. The unmapped factors imply that there are factors that the clients view is important which development teams do not view as important and visa versa or the sample size has caused these factors to be omitted. On the development side it seems that certain organisational soft issues should be addressed and that the highlighted client factors Also require attention from the software development companies and improvement.

7.5 Conclusion and Recommendations

In summary the clients and development view on what is important for the success of software development applications corresponds with what the literature defines.

The survey results indicate that the clients and development teams in South Africa rate software development applications as successful. The results therefore rejects hypothesis one stating *"Clients think there is a high degree of software application failure"* and accept hypothesis two stating *"Developers think there is a low degree of software application failure".*

The clients and development perspective also matches and correlate, which means that they do not differ and therefore hypothesis three is rejected, which states that *"Development teams and clients perspectives differs when it comes to defining success of software applications".*

The sample size was focused on small software development companies, which means that this result might not necessarily be applicable to larger software companies and it would be interesting to do the same research on larger software companies.

The Clients survey analysis do emphasise the following areas that software development companies need to investigate and improve upon:

- Business fit
- Application usage
- Changes
- Value provided

8 REFERENCES

Author Unknown. 2003. 'NZIM: When IT Projects Fail Blame Management.' New Zealand Management: 18-19.

Boehm, B. 1987. 'Industrial Software metrics top 10 list', *IEEE Software*, Sept87 4(5):84-85.

Boyd, A. 2001. 'The five maxims of project satisfaction', *Aslib Proceedings,* 53(10): 423-430.

Boyd, A. and Boyd, J. 1998. *The five pillars of satisfaction.* Unpublished lecture and consultancy notes, 1998-2001.

Davis, A.M. 1994. 'Fifteen principles of Software Engineering', *IEEE Software,* Nov94 11(6): 94-97.

Doherty, N.F., King,M. & Al-Mushayt,O. 2003. 'The impact of inadequacies in the treatment of organizational issues on information systems and development projects', *Information & Management,* 41(1): 49-55.

Fichter, D. 2003. 'Why Web projects fail', *Online,* 27(4): 43-45.

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

Grossman, I. 2003. 'Why so many IT projects fail...and how to find success', *Financial Executive,* 19(3): 28-29.

Joel, H. 2004. *Software project management: a real-world guide to success.* 1st Ed. Boston: Pearson Addison Wesley.

Jones, C. 1996. *Patterns of Software Systems Failure and Success.* 2nd Ed. New York: International Thomson Publishing.

APPENDIX A: PARTICIPATION REQUEST E-MAIL AND LETTER

E-Mail:

Dear Sir/Madam,

Please find attached a letter to explain all details of the Research paper.

I would really appreciate it if you would take the time to read the letter and hope that you will be willing to participate.

Hope to hear from you soon.

Regards

JEANINE VAN DER GUGTEN

CONTACT NUMBERS : +27 (0)82 950 1388 (Cell)

+27 (0)12 653 7885 (Home)

E-MAIL ADDRESS : JeanineVDG@ananzi.co.za

2. Completion of a questionnaire

Confidentiality of all information of the individual companies and their clients are guaranteed, and only the statistical results of the data analysis will be presented to the participating companies.

The paper and its results will be shared with the Software Development Company, which should provide the following:

- Identifying their success/failure rate compared to the current literature information.
- Providing an in-house perception on projects,
- Providing insight on their client's perceptions.
- Benchmarking their position to other small software development companies.
- Identifying improvement opportunities to serve their clients.

Much appreciated and thanks you for your time.

Regards,

Jeanine van der Gugten

JeanineVDG@ananzi.co.za

082-950-1388 (cell number)

APPENDIX C: SOFTWARE DEVELOPMENT DELIVERY SURVEY QUESTIONNAIRE

Note:
Place an X in the appropriate field below, where:
1 = Very poor
6 = Extremely good

Section A: Software Development

Topic	Description	1	2	3	4	5	6
1 Applicability	This topic includes the following:						
	a. Adequate business knowledge						
	b. User involvement in determining their requirements						
	c. Adequate technical skills to translate business knowledge to a useful application						
2 Sustainability	This topic includes the following:						
	a. Structured process followed – requirements definition, analysis, design, test, implementation						
	b. Usage of tools to assist with process - any case tools, data-modeling tools, re-engineering tools, etc.						
	c. Documentation provided – User Requirements, Specifications, Technical Specifications, training manuals, On-line help, etc.						
	d. Usage of a specific methodology						
3 Usability	This topic includes the following:						
	a. Development and design standards applied – same look and feel across screens and certain coding standards in place						
	b. Adequate testing done for any defects/bugs (unit testing)						
	c. Adequate testing done for the fit to business requirements/functionality (integration/functionality testing)						
	d. Testing done by the users – user acceptance testing						
4 Functionality	This topic includes the following:						
	a. Trace ability from requirements to design to development to testing						
	b. Adequate communication and liaising with users						
	c. Ability to make suitable compromises on requirements and development alternatives.						
5 Adaptability	This topic includes the following:						
	a. Adequate control of changes – Change management (change control forms, sign-offs and additional costs, etc)						
	b. Adequate Configuration management						
	c. Adequate Version control management						

Note:
Place an X in the appropriate field below, where:
1 = Very poor
6 = Extremely good

Section B: Client Relationship

Topic	Description	1	2	3	4	5	6
1 CRM	This topic includes the following:						
	a. Client's ability to articulate requirements						
	b. Client's willingness to constrain changes						
	c. Client's understanding of technologies used						
	d. Client's availability and participation in all the stages of the projects						
	e. Client's trust and respect towards your abilities						

APPENDIX D: CLIENT RELATIONSHIP SURVEY QUESTIONNAIRE

Note:
Place an X in the appropriate score field below, where:
1 = Very poor
6 = Extremely good

Section A: Software Application/Solution

Topic	Description	1	2	3	4	5	6
1 Business fit	This topic includes the following:						
	a. User involvement in determining requirements						
	a. Fulfills the business requirements						
	b. Easy integration with other processes and/or applications						
	c. Meets your expectations						
	d. Adds value to the business						
2 Documentation	This topic includes the following:						
	a. Adequate supporting documentation/on-line help						
	b. Understandable and un-ambiguous						
	c. Quality of format and content						
3 Training	This topic includes the following:						
	a. Adequate training provided						
	b. Adequate training manuals provided						
	c. Quality of format and content of training manuals						
4 Application Usage	This topic includes the following:						
	a. User friendliness of application (ease of use)						
	b. Response times (quick response between screens/processes)						
	c. Effectiveness and efficiency (good screen design with minimum cumbersome processes)						
	d. Consistency (Same look and feel in all screens and processes)						
	e. Error free (no run-time errors)						
	f. Bug free (no inconsistency with business rules)						

Note:
Place an X in the appropriate field below.

Section D: General

	Topic	Description	Yes	No
1	**Software Application/Solution on the whole**	Overall, would you rate the software application/solution as a success?		
2	**If No, what would you say were the main reasons for the failure?**			
3	**What do you consider the most important aspect of a software application/solution?**			