ISSN 1744-1986

Technical Report N° 2006/ 24

How Can Defect Analysis Help To Improve Risk Management Techniques In IT Projects?

Roger Swaby

30 September, 2006

Department of Computing
Faculty of Mathematics, Computing and Technology
The Open University

Walton Hall, Milton Keynes, MK7 6AA
United Kingdom

http://computing.open.ac.uk

How Can Defect Analysis Help To Improve Risk Management Techniques In IT Projects?

A dissertation submitted in partial fulfilment
of the requirements for the Open University's
Master of Science Degree
in Computing for Commerce and Industry

Roger Swaby
(W0281408)

20 February 2007

Word Count: *15,117*

Preface

I would like to thank all those who supported me throughout a difficult final year of the MSc, in particular, Peter Greatrex, my project supervisor, for his guidance and encouraging comments. He has been a great help to me.
I'd also like to thank my employers for allowing access to resources for this project, and all the project personnel who played a part in the research.
Finally, I'd like to thank my wife Nicole for her hard work and support during the last year, my son for understanding that some weekends daddy has to study, and my mother and brother for instilling in me, an eagerness to learn.

Table of CONTENTS

CD CONTENTS – SUPPORTING DOCUMENTS FOR DISSERTATION

TESTCASES.XLS
Details of test cases pertaining to the projects discussed in this dissertation as retrieved from a "Quality Centre" logging system. And details of their associated testing categories.

DEFECTS.XLS
Details of defects raised during system testing of the projects discussed in this dissertation as retrieved from a "Quality Centre" logging system. And details of their reclassification to the Hewlett Packard Defect Scheme.

CALCULATIONS.DOC
In-depth details of calculations used in the dissertation.

Table of TABLES

Table of Figures

Abstract

Historical data has shown us that a high proportion of IT projects do not meet the success criteria on which they are based. There are many reasons why a project may fail, some within the direct control of the project manager and others upon which he has less influence. One such reason for failure is ill preparation for circumstances that detrimentally affect the project. Risk management provides a powerful tool for improving project success rates, as good risk management techniques can counteract many of the root causes of project failure.

Defects provide a quantitative measure for known problems that occurred during a project. The analysis of these defects is often used to feedback into the development cycle in an attempt to improve project management and process techniques.

Much of the research surrounding defect analysis schemes is based on causal analysis, looking at how defect analysis can improve processes for future and current projects. It could be argued that risk management aims to reduce the impact of unforeseen circumstances by eliminating or decreasing the root-causes of these unforeseen circumstances. Not much research has been performed on the linking these two techniques; hence, this dissertation will investigate a possible link between these two.

1 Introduction

1.1 Problem overview

This dissertation aims to gain an insight as to how defect analysis can help to improve risk management techniques used in IT projects in general, and in particular those used by my department at work. This research question has originated from background research on "Why IT projects fail". This section summarises the findings on project failure.

1.1.1 What is the problem with IT projects

In 1995, U.S. companies alone, spent $59 billion on IT project overspends and another $81 billion on cancelled projects from a total IT project spend of $250 billion (Johnson, 1995). In 1998, industry data indicated that 26 percent of IT projects were outright failures and 46 percent came in over budget with regard to cost or schedule (Pressman, 2000). So why is it so difficult to deliver a successful IT project? Before we can attempt to answer this we must first understand what defines an IT project, what constitutes its failure or success and what are the possible causes of project failure?

1.1.2 An IT project – A definition

Goal Directed Project Management (GDPM), a model for successful project management describes a project as "*a unique task, designed to attain a specific result, is limited in time and requires a variety of resources*" (Anderson, Grude and Haug, 2004, p. 10). From this we can infer that a project is something not previously undertaken and hence cannot be performed by simply addressing routine tasks. It has an end-goal in that it is performed to attain some specific result in some limited time and in order to achieve this end-goal it requires a variety of resources, what GDPM refers to as a "*temporary organisation*" (ibid).

Turner (1999) goes further to say it may not be wise to attempt to give a precise definition to something that by nature is not precise. Instead, it is more appropriate to identify the features of a project that differ from the features of other projects and from routine tasks. Hence, we can look at a project as a mixture of the unique and the routine performed on the path to attaining our end-goal.

1.1.3 Success or failure – On whose say-so?

In order to illustrate the significance of the terms success and failure, we can consider a hypothetical example of a completed software project: -

> Senior management of service company ABC sponsor a project to be undertaken by their internal IT department to improve the information systems of their customer service centres. After discussions with both senior management and customer services, the IT department agree a budget and timescale with senior management and the requirements for delivery with customer services.
>
> At the time of delivery, IT deliver the proposed solution to customer services on time and to specification, but due to under estimating the size of the task, they had to cut some corners on their quality standards and had to add extra resources in order to complete the work on time.

So, is the outcome of this project a success or a failure? Fortune and Peters (2005 p. 26) provide us with a realistic answer to this question when they state "*Failure and success are subjective assessments that vary in time and with the standpoint of the person making the judgement*". Hence, customer services may have deemed the project a success, as their original requirements were delivered to them in accordance with their schedules. Senior management and IT may have judged this a failure due to the project exceeding the agreed budget, and additionally IT may not have been satisfied with the quality built into the delivery. In time senior management may begin to view the project as a success if the revenue returned by introducing the new system exceeds the expected return on investment by enough to make the initial overspend seem insignificant. So as Fortune and Peters assert, success is not only subjective, but also the same individual may judge the success of a project differently depending on other external factors.

Moreover, as a manager of IT projects, one cannot simply state that success is judged subjectively and hope for the best. Therefore, it is necessary to provide some criteria on which to judge the success of a project. Fortune and Peters (2005), Wateridge (1997) and Turner (1999) all describe frameworks to judge success in which a series of questions are posed about the project. These questions such as "Was the project completed within budget?" (Fortune and Peters 2005), "Did the product match the user requirements?" (Wateridge 1997), "Did the project meet quality standards?" (ibid.), "Did the product make a profit for its customer?" (Turner 1999) must be answered affirmatively for the project to be deemed a success.

1.1.4 The possible causes of project failure

Given the expenditure on project failures and cancellations, it would be feasible to say that a large proportion of IT projects do not match these criteria for success. Following is a discussion of some of the reasons why failure might occur.

Unclear objectives/requirements

Andersen et al (2004) talk about the human tendency to rush into solving problems without putting sufficient effort into first defining the problem to be solved. They discuss how the failure of many IT projects can be traced back to this tendency. We find the same message in Open University literature – "*If the requirements are not clearly and completely set out, any project or design based on them cannot succeed*" (M865 Unit 1, p.24).

This assertion is exhibited in Addison and Vallabh's empirical study of risk mitigation methods used by experienced project managers, where the top two risk factors identified by the project managers taking part in their survey were

"Unclear or misunderstood scope/objectives" and "Misunderstanding of Requirements". (Addison T, Vallabh S (2002))

Poor management

The abilities to motivate and inspire others, and to plan, organise and control work methodically are seen as qualities of a good project manager (Pressman 2000, Andersen et al 2004). Both Pressman and Andersen et al go further to say that good practitioners often make poor managers as they lack these managerial qualities, but these are often the people who eventually find themselves in charge of projects after gaining promotion as a reward for their good work.

Poor estimation

Andersen et al (2004) describe poor estimation as a possible pitfall in the planning process of an IT project. They talk of how the lack of trust in proven empirical estimation methods based on historical data can be overwhelmed by the "*lack of realism*" of an over-optimistic project manager.
Brooks (1995 p. 14) in his seminal book, 'The mythical man-month' again backs up this assertion when he refutes the optimistic assumption that "*each task will only take as long as it ought to take*".

Individual motivations

Turner (1999 p. 52) talks of overt and covert objectives. Overt objectives are the stated aims of the project whereas covert objectives are the hidden agenda of the individuals involved in the project. He discusses how these two types of objectives may be in conflict with each other and therefore cloud the thinking or reduce the motivation of an individual. He illustrates this concept of covert objectives by giving examples of individual motivations that might clash with project objectives, for example: -

- Project managers wishing to enhance their careers
- Managers wishing to widen their sphere of influence
- People wanting to protect their jobs

Social pressures

Staw and Ross (1987) discuss how the idea of "staying the course" is associated with "strong leadership" within our culture. That is to say, that a strong leader is one who is seen to work through the hard times until the situation becomes successful. This implies that withdrawal from an unnecessarily difficult situation is a sign of weakness. They talk of how this attitude can sometimes push managers to stay with a course of action for too long rather than considering alternative action.

Unforeseen circumstances

Risk management is aimed at preventing project uncertainty or reducing the impact of such uncertainty on a project. Pressman (2000) discusses reactive versus proactive risk strategies and discusses how project personnel can prepare themselves for uncertainty by a process of proactive risk identification, assessment and subsequent management. He warns of the dangers of leaving possible problem areas to fate and the detrimental impact on project success should such a problem actually transpire.

Kiel et al (1998) follow this line of reasoning and talk of the need to establish the importance of each uncertainty in order to focus effort on the most critical.

The importance of good risk management can be illustrated by examining its relationship with the other pitfalls highlighted here. Addison and Vallabh (2002) and Keil et al (1998) both discuss the use of risk management techniques to reduce the impact or likelihood of such pitfalls as unclear objectives/requirements, poor estimation and a lack of effective management.

1.1.5 Summary

Historical data has shown us that a high proportion of IT projects do not meet the success criteria on which they are based. The actual degree of success of a project is subjective and may vary with time and changes in the environment within which they are based; hence it is necessary to create a framework on which success is based in order to consistently judge success. There are many reasons why a project may fail, some within the direct control of the project manager and others upon which he has less influence. One such reason for failure is ill preparation for circumstances that detrimentally affect the project. There is a strong relationship between this type of project failure and the others mentioned, as good risk management can counteract the causes of these other types of project failure. This makes risk management a powerful tool for improving project success rates, and it is for this reason the primary research will be based on this type of project management function, i.e. risk management.

1.2 Aim of the research project

The aim of this research project is to explore a method for improving the success rates of IT projects. More specifically I will investigate how defect analysis can be used to improve risk management techniques, using the IT projects of the UK Internet Banking department of a well-known global bank as the raw data to this research. If viable evidence can be gleaned through this research then the research may be adapted to investigate the existence of this relationship on a wider scale.

Projects of this type have been selected because: -

- Much research has been performed on associating defects analysis with process improvements and project processes. On the contrary, very little work has been published on associating defect analysis with risk analysis.
- Some of the aspects of the projects undertaken by this department apply to many other departments in the company, e.g. types of software tools used, project management methodologies. Any conclusions pertaining to these aspects may be of use to similar departments.
- I am a project leader within the department, so my research efforts to improve success rates will be of benefit to me both academically as well as my day-to-day work.

With regard to improving project success rate, much research has been performed on defect analysis techniques (Basili and Rombach (1987), Li Plet al (2004), Chillarege R et al (1992)). Also good work has been carried out on identifying risks and their associated mitigation techniques (Keil et al (1998), Addison T, Vallabh S (2002)).

The aim of this dissertation is to continue these lines of work focusing on the abilities of defect analysis to provide insight into the shortcomings of the risk management techniques used by project managers. Hence, I propose to ask and answer the question: -

How can defect analysis help to improve risk management techniques in IT projects?

1.3 Contribution to knowledge

Process improvement strategies such as the Institute of Electrical and Electronics Engineers' (IEEE) Software Quality Assurance (SQA) plan (Pressman 2000 pp. 216 -217), the Capability Maturity Model Integration (CMMI) model (SEI 2006) and the Software Process Improvement and Capability Determination (SPICE) initiative (M880 SQM pp.29 -33) have led to an improved awareness of project and process management within the industry. Such schemes give a company the ability to predict project success and to learn from mistakes via continuous improvement. These schemes, although successful require a large commitment to implement involving senior manager sponsorship and organisational change.

The purpose of this research is not to suggest a less complicated process improvement model, but instead take one aspect of the software engineering process in an attempt to derive a method to enhance this aspect and as a by-product improve the success rate of Internet banking projects undertaken by my department. In the case of this research project, I have chosen risk mitigation as the software engineering aspect on which I would like to concentrate.

Advocates of risk management believe that improved risk identification and mitigation techniques can lead to increased project success rates (Andersen et al

(2004), Addision and Vallabh (2002), Pressman (2000)). I hope to show that defect analysis not only provides a concrete illustration of problems with the products of an IT project but can also provide feedback into improving the risk mitigation techniques used by my department. My intention is to contribute theoretically to the existing body of knowledge with regard to how risk mitigation can be improved, but more specifically contribute to the improvement of risk mitigation techniques used by my department and myself.

The nature of the projects that I will study will limit any findings to the scope of my department, the source of the projects. This is a necessary limitation, as I will need to study project information and interview staff to a very detailed level. Access to this level of information for projects outside of my department is not possible. In addition, I will need to set the number of projects to be studied to a nominal size to ensure that the work can be completed within the timescales of the M801 research project and dissertation.

In saying this, I do realise that the types of problems faced by my department may be similar to those faced by others external to the department. Certainly, within the company, many of the tools used by other departments and the development environments within which they work may be similar, hence I would hope that my findings might be of some use to these parties and possibly act as a starting point for their own improvement strategies.

1.4 Summary

Project failure is commonplace in the IT industry, the reasons for which can be inside or outside of the project manager's control. Risk management provides techniques that can improve the chances of project success by eliminating or reducing some of the uncertainty of a project.

Many schemes such as CMMI (SEI 2006) and SPICE (M880 SQM pp.29 -33) use defect analysis as a method for improving IT processes. This paper attempts to use defect analysis to understand the relationship between defects raised and risk management techniques used in order to provide feedback into improving risk management techniques.

2 Review of current knowledge

2.1 Introduction

Much research has been undertaken on how organisations and their workforces can improve the rate of success for IT projects. This review of current knowledge will attempt to provide the reader with a concise review of some of the strategies currently being used.

The discussion will commence with a look at project management methodologies. These methodologies aim to provide methods and tools to guide projects towards meeting their objectives. Next, the discussion will move to risk analysis/management, in which the current thinking on risk identification, assessment and management will be discussed. Finally, discussion will move onto defect analysis, investigating the process of defect identification and its uses. It is on these two final topics that the primary research will be based.

2.2 Project management

The Open University literature (M865 Unit 7, p.63) defines project management as "*the planning, monitoring and control of all aspects of a project and the motivation of all those involved in it to achieve the project objectives on time and to the specified cost, quality and performance*". This achievement of objectives needs to be systematically carried out in order to produce successful results; project management provides the foundation for this work (Anderson, 2004).

2.2.1 The factors that can affect project success

Pressman (2000) believes that projects cannot be successful unless project management focuses on what he refers to as the four P's: -

- People: The actual personnel who hold some stake in the project
- Product: The actual product to be delivered

- Process: The framework on which a plan for the delivery of the product will be based
- Project: The actual realisation of the process

He talks of the relationships between them and how over-emphasizing or compromising on any of them will have an impact on the others. This is illustrated in the Four P's interrelations diagram below.

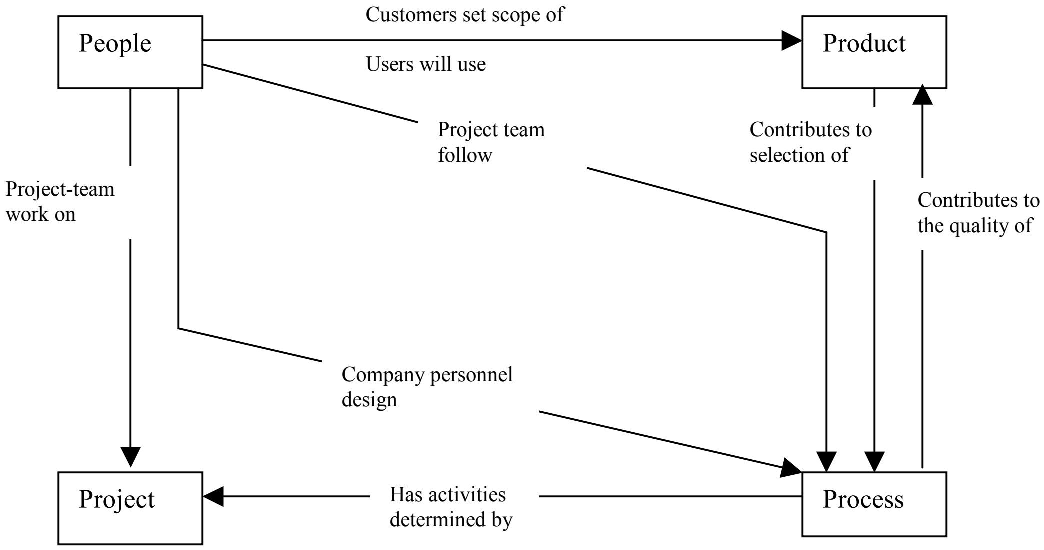

Figure 1 The Four P's interrelations (adapted from M880 Project Management, p. 9)

GPDM (Goal Directed Project Management) widens this approach to project management, describing IT projects as being People, System and Organisation (PSO) projects. PSO projects recommend the setting of goals for the development of the affected personnel (P – People), and the necessary organisational changes (O – Organisation) as well as the more common goals set for the technical requirements of system development (S – System) (Anderson, Grude and Haug, 2004). The authors talk of the habitual IT danger of overstressing the 'S' goals, leading to unstructured changes in work-processes and procedures 'O', and a negative attitude or understanding by the people concerned 'P' (ibid.).

Turner (1999) describes the five system objectives that need to be managed if project managers are to be successful in their endeavours, namely, scope of work, the temporary organisation created to perform the project work, the quality to be built into the project and the time and cost of completing the project. He makes a clear distinction between the importance of the scope and organisation with the remaining three objectives, which he describes as "*important constraints*". This is supported by the assertion that scope defines the work to be done, whilst the organisation provides effort for this work; these are the essential tasks without which there is no project. The importance of the three other objectives will differ from project to project depending on the needs of the base organisation (ibid).

2.2.2 Project Management Summary

The overriding message sent out by these project management models is that project success is not solely about delivering a solution that works. Other factors must be considered to ensure not only the creation of a quality system, produced on time to cost, but also to ensure the smooth integration of the resulting product into an organisation.

2.3 Risk analysis and management

I have already mentioned that project management allows us to realise our project objectives. In order to commit to attaining these objectives, the project manager should also recognize that there will be uncertainties that may prevent these goals from being achieved.

Risk analysis is the process of identifying uncertainty and assessing the probability of these uncertainties occurring and the consequences to the project should they occur. Risk management details the methods used to counteract or prepare for these uncertainties.

2.3.1 Risk identification

Pressman (2000) recommends the creation of a *risk item checklist*. This checklist can be used to identify predictable risks, he gives a list of generic risk attributes: -

- Product Size – Risks associated with the size of the project, such as getting and retaining the required workforce
- Business Impact – Risks associated with the constraints imposed by management or the market place, such as the need to comply with certain regulations.
- Customer characteristics – Risks associated with how often and at what level project staff communicate with the customer.
- Process Definition – Risks associated with the capability of the software engineering approach and how strictly it is adhered to.
- Development Environment – Risks associated with the availability of environments and the quality of the tools to be used
- Technology to be built – Risks associated with the complexity of the system to be built and the newness of the system technology
- Staff experience and size – Risks associated with the overall experience of staff (Technically and project-wise)

Turner (1999) also advocates the use of checklists as well as proposing other methods that may be used to aid risk identification. He advises decomposing plans to find bottlenecks. He recommends studying the external interfaces of these critical tasks to discover any potential risks. Turner also talks about assumption analysis and the drivers behind certain decisions. Assumption analysis considers events that we expect to occur but may not, as well as the negative events that we do not wish to occur. Decisions drivers are examined to determine if decisions are made for the wrong reason. For example, tools may have been chosen for budgetary rather than technical reasons leading to possible technical problems later in the project.

2.3.2 Risk assessment

Risk assessment provides a means to judge the relative importance of each risk identified. Pressman (2000) calls this ranking measure risk-exposure and regards it as being the product of the probability of occurrence of a problem and its impact on the project. Turner (1999 pp. 238 - 240) brings the public perception of risk into play as well, and argues that although this perception may sometimes be irrational, it is a vital ingredient in assessing risks if projects are to get the backing of the wider audience. However, I would argue that a tactic such as this could cloud the issue somewhat. Giving credence to the opinion of those who do not have access to the complete set of facts can falsely inflate the impact of one risk, and detract from risks that deserve far greater attention.

Kiel et al (1998) report on the results of a Delphi study, where experienced project managers identified and ranked the most important risks. From these results, they produced a risk categorisation framework, identifying attributes of risk and ranking them by importance and the level of control for this risk by the project manager. Their findings showed that failure to gain commitment and involvement by both senior management and customers alike proved to be the most worrying risk, as it was perceived that whilst failure to involve these parties would be of great detriment to the project, project managers have a low level of control over this commitment. Their studies also revealed that risks pertaining to the execution of the project, such as staffing/skills issues or process methodology were of least concern. Such risks were perceived to be of moderate impact to the project, but were very much under the control of the project manager.

Addison and Vallabh (2002) back up this assertion. Their study is based on a survey of experienced project managers. As with Kiel et al (1998), their findings show the most important risks pertain to customer and senior management involvement, such as a lack of senior management input or misunderstanding customer requirements. These risks also are described as occurring "*most of the time*", although it was noted that the importance of these risks decreased with the level of experience of the project manager (ibid.). The findings also show that the least worrisome risks also have the lowest occurrence rates. These risks are those that pertain to the execution of the project, such as inadequate knowledge/skills, gold plating and a lack of effective project management methodology. Although the results can be catastrophic if these risks are left unchecked, as Kiel et al (1998) stated, project managers viewed these project execution risks to be within their control.

2.3.3 Risk management

Risk management is the process of defining and taking counter measures to decrease or nullify the impact of uncertainty on a project. Andersen et al (2004) recommend the creation of an uncertainty matrix marrying the risk probability and risk impact with the action to be taken should the risk occur, including who should be responsible for this action. Pressman (2000) talks of RMMM (risk mitigation, monitoring and management). He describes the three issues necessary to deal with uncertainty in projects: -

- Risk avoidance – Avoidance or reduction of the risk; Pressman regards the former as the best practice when adopting a proactive approach to risk management.
- Risk monitoring – Tracking the changing probability and impact of the risk as the project progresses.
- Risk management and contingency planning – Course of action should the mitigation efforts fail and the risk becomes a reality.

M880 (Unit PM, p.42) provides more insight into how risks may be mitigated. The literature suggests four possible strategies, namely: -

- Risk avoidance – Changing the planned work so that the project no longer carries the risk, such as changing from a new programming language to an old language in which staff are more knowledgeable.
- Risk retention – Accept the risk due to the low probability and impact, or because nothing can be done about it.

- Risk reduction – Using controls to lower the impact and the likelihood of a risk.
- Risk transfer – Passing the risk on to someone else. E.g. using a penalty clause in a contract.

Addison and Vallabh's (2002) survey detailed which controls experienced project managers used to mitigate important risks. Their findings provide the user with a list of commonly encountered risks and the controls used to combat them. They conclude that the relationship between these risks and controls is a many-to-many relationship. Hence, one control could be used to mitigate multiple risks and each of these risks could be mitigated by the use of more than one control.

This idea of a many-to-many relationship between controls and risks could cause difficulties when performing retrospective analysis on the effectiveness of control strategies used. The influence over a particular risk would be shared by many controls, which in turn would each have influence over other risks. The resulting picture of which control mitigated which risk, and did so effectively, would therefore be unclear.

2.3.4 Risk Summary

Risk identification, analysis and management are activities that allow a project manager to understand and manage the impacts of uncertainty on a project. Tools such as checklists as well as processes such as plan-decomposition, assumption-analysis and the examination of decision-drivers aid the identification of project risks. Once identified, differing theories exist on how the impact of a risk should be measured, but the same underlying theory is common to all these theories in that the measurement formulated should take into account the likelihood of the risk and its impact on the project should it occur. Risk management provides the theories to define and take counter measures to decrease or nullify the impact of a risk on a project. Generally these measures take the form of avoiding the risk altogether, retaining the risk due to negligible impact or likelihood or both, reducing the impact or likelihood of a risk, or transferring the risk to another party.

2.4 Defect analysis

The Oxford English dictionary defines the term defect as a "*shortcoming*" or a "*failing*" (Oxford 2002). In IT project terms, defects can be viewed as problems with the intermediary or final products of a project. The purpose of defect analysis in IT is to identify these shortcomings, understand their root causes and finally use this information in some way that is useful to the organisation.

Throughout this topic, I will use the following definitions extracted from the IEEE Guide to Classification for Software Anomalies (IEEE 1996): -

- **Category**: An attribute of a defect to which a group of classifications belongs. A specifically defined division in a system of classification.
- **Classification**: A choice within a category to describe a defect and the environment where it was encountered.

2.4.1 Defect Identification and Classification Schemes

The idea behind a defect classification scheme is to provide categories within which defects can be grouped. This grouped information can then provide the basis for additional analysis activities such as estimating resources, process improvements and measuring product progress.
Basili and Rombach (1987) discuss the selection of an appropriate classification scheme for a given task and name three criteria on which this decision should be based: -

1. Is it possible to decide the category for each defect?
2. Can the information for this decision be collected easily?
3. For each category, are there methods and tools to detect, prevent, isolate and correct the defects of that category?

Kelly and Shepard (2001) mention two common uses of defect classification schemes: -

- Simple classifications - used to assign priorities to fixing defects by identifying the severity of defects. Such a classification could be Minor, Major, Severe and Critical.
- More in-depth classification schemas - used to identify key information from defects for use in the assessment or improvement of development processes.

I would also add product progress to this list of uses. Schemes such as IBM's Orthogonal Defect Classification (ODC) scheme (IBM 2002) can also be used to illustrate the current state of a project. The ODC scheme is discussed in further detail below along with some of the more widely used schemes.

IBM Orthogonal Defect Classification (ODC)

Chillarege et al (1992) describe ODC, as bridging the gap between statistical defect models – used to provide defect metrics for a project (e.g. detection rate, failure rate) and causal analysis – qualitative analysis of the individual defects of a project.

The thinking behind ODC is to classify defects into categories that collectively point to the part of the development process that needs attention. However, as Chillarege et al (ibid) point out, this classification is a human process and so is subject to human error. ODC attempts to rectify this problem by classifying defects using a small distinct set of pre-defined defect types to represent the type of problem, such as documentation, interfacing or variable assignment. It also defines a set of pre-defined triggers that identify the condition that allowed the defect to surface, such as moving programs to a different hardware configuration or database recovery (ibid). The aim here is to remove subjectivity from the defect classifier and developer. The relatively small number of defect types and triggers means that the classifier and developer can choose more accurately (Kelly and Shepard, 2001).

Chillarege et al (1992) discuss the change of the defect type distribution over each phase of the project and the expected peaks and troughs of the defect type classifications during the process. This implies that this defect type distribution can provide an indication of where the development is logically, known as process inference. Should the actual development phase be different to this logical development phase, an alarm should be raised. For example, a project development up to a certain point has led to various defects being captured. The distribution of the defect type classifications of these defects would suggest that the product is currently in the integration-testing phase, although the actual phase of this project in the "real- world" is system testing. This would imply that the project has prematurely moved into system testing, meaning an inadequate integration test has been performed.

The defect trigger distribution provides an insight into the verification process by indicating which verification processes are capturing which defects. Chillarege et al (1992) suggest that a huge discrepancy between the defect trigger distribution in system test and that in the field would identify potential holes in the system test environment, and I would add to this, possible problems with the system-knowledge or testing-skills of the system testers and the areas focused on during testing.

More recently, IBM has updated ODC to include additional types and triggers. The current standing shows defects categorised by triggers and impact, namely the categories available on finding the defect, and by categories such as the type, age and source, those available on fixing a defect (IBM 2002).

Hewlett Packard Defect Scheme

The goal of the Hewlett Packard Defect Scheme is to "*report, analyse and focus efforts to eliminate defects and their root causes*" (Grady 1992.). The scheme provides three categories on which defects are logged: -

- Origin – The first activity where the defect could have been prevented. E.g. **Design, Code**
- Type – The area within a particular origin, that is responsible for the defect. E.g. **Error Checking** *occurring within the Code origin*
- Mode - The reason why the defect occurred. E.g. **Missing** *Error Checking in Code*

These three defect categories are not independent; instead, the selection of a classification in the Origin category influences the selection of classifications in a Type category. This interdependency between these categories is illustrated in the diagram below.

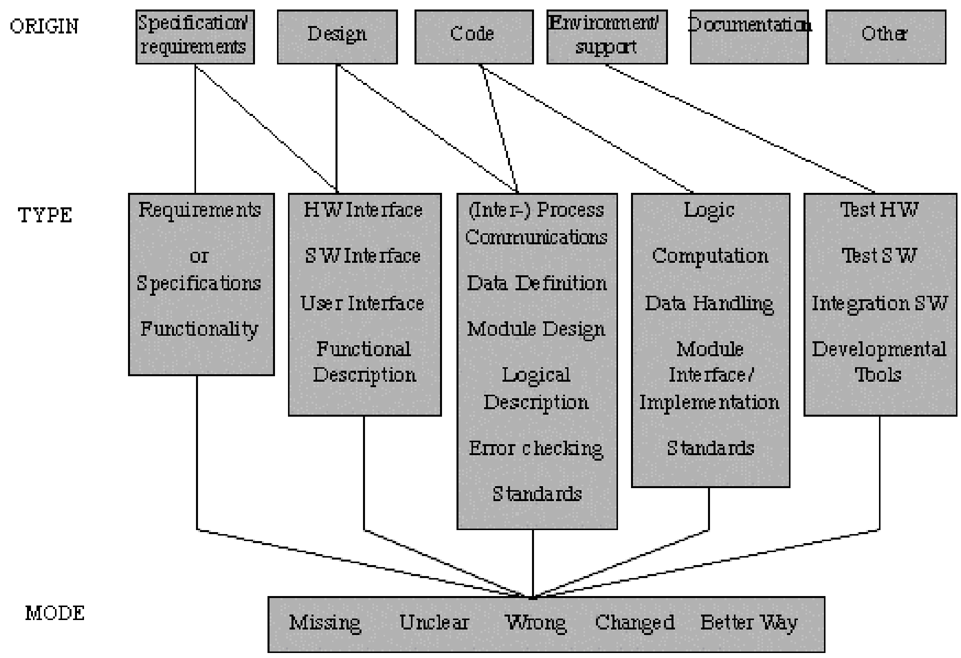

Figure 2 Hewlett-Packard Defect Categorization Scheme (Grady 1992))

Based on this categorisation, analysis can be performed to discover the major kinds of defects being captured, and hence provide insight into the process improvements required to prevent these defects occurring in future developments.

More recently, a software testing extension was introduced to the HP model (Huber 1999). This extension links the HP type category to a test level, showing what kinds of testing might be appropriate to discover defects of this type. The table below illustrates this relationship.

Type Classification	**Test Level**
Hardware/software Interface	Sub-system integration test
User interface Requirement specification Functionality Functional description Error checking	System test
Process/Inter-process communication	High-level component integration test
Module Design	High-level component test
Error checking Data definition Logic description Logic Computation Data Handling	Unit test
Module/Interface implementation	Unit integration and low level component integration test

Table 1 HP Model Testing Extension Table

IEEE Standard Classification for Software Anomalies

The IEEE classification scheme works on the principle that more detailed defect information can provide greater insight into the project and process activities. It requires the user of this scheme to provide greater and more detailed levels of information pertaining to each defect in order to point to problem areas in the process (IEEE 1993). Analysis of this data provides insight into common errors made, the areas where defects were introduced, and the most effective methods for discovering defects.

The scheme consists of four steps. The first step, Recognition, occurs when the defect is found. During the second step, Investigation, the defect is investigated to find the actual cause of the defect. In the third step, Action, a plan of action is established to resolve the defect. The final step, Disposition, is performed when the defect has been resolved and long-term corrective actions have been completed (Freimut 2000).

IEEE defines defect categories for each of these steps plus additional categories for assessing defect impact. For organisations wishing to implement schemes based on this standard, IEEE defines each of these categories as optional or mandatory, allowing the implementers to start with a core scheme of mandatory categories and add to it those optional categories that fit their purpose. All of these categories are described below: -

Recognition Categories

- The project activity that was taking place when the defect was found. E.g. Analysis
- The project phase that was taking place when the defect was found. E.g. Prototype Design
- The suspected cause and whether the error is repeatable
- The symptom that caused the anomaly to be identified. E.g. system crash
- The resulting status of the product. E.g. Unusable, Workaround available

Investigation Categories

- The actual cause and source of the defect describing where the product has performed inadequately; e.g. Test Plan in the testing system documentation
- The type of the defect pinpointing the actual problem. E.g. Forgotten steps in program logic

Action Categories

- The resolution detailing how the analyst decided to deal with the defect. E.g. Immediate software fix, Deferred to later release
- The corrective action details actions to prevent the occurrence of similar defects in the future. E.g. Revising Department procedures

Impact Categories

- The severity and the priority of the defect.
- The value of the fix to the customer

- Project related impact, such as additional cost, effect on schedule and risk to project success.

Disposition Categories

- Classification of the defect status upon closure of a defect; e.g. Closed with resolution implemented, Incorrectly diagnosed as a problem, Deferred to a later release

This large selection of categories permits the IEEE scheme to provide insight into project management activities, such as evaluating the impact of defects on the project plan. It also provides measures for product assessment, such as evaluating the Impact-Severity and the Impact-Customer Value categories to provide an insight as to whether a product is ready for release (IEEE 1996).

2.4.2 Other schemes

Other work has been performed on defect analysis in order to modify existing schemes or create new schemes tailored to specific needs.

Kelly and Shepard (2001) create a modified version of ODC in their study of the use of defect classifications in the comparison of software inspection techniques. Their findings show that their defect classification scheme can be used as a metric for such inspections, although they state that further work needs to be carried out on the re-usability of their scheme.

Hvannberg and Law (2003) define a classification scheme that specifically extracts information from usability problems - Classification of Usability Problems (CUP) scheme. The authors used the kappa coefficient to verify the agreement of data classified by two different analysts using this scheme. The results show that CUP needs further refinement to conclude that it is a repeatable scheme, although it has the potential for being a useful tool in the field of usability.

Basili and Rombach (1987) provide several classification schemes for collating information on: -

- Errors – Defects in the human thought process when trying to understand information or solve problems, i.e. a misuse of the system
- Faults – The actual manifestation of these errors within the software
- Failures – Departures of the software system from the software requirements

Leszak et al (2000) also define their own classification scheme in their case study in root cause defect analysis, where they attempt to use this scheme to provide countermeasures to prevent defects occurring in the development process. One novelty of this scheme is the creation of a four-dimensional root cause category. Each defect raised must provide a classification for each of the root cause categories, allowing a greater richness to logging defect root causes. The four root-cause categories are as follows: -

- Phase - The phase or document where the defect was injected
- Human - The cause of the defect in relation to skills of the project staff. E.g. lack of knowledge, individual mistake
- Project – Regards the way the project is managed. E.g. Time pressure
- Review – Pertaining to the reviewing procedures for project deliverables

Leszak et al (2000) also provides a fifth root cause category “Other” to handle root causes not covered by those above.

2.4.3 Fit for purpose

As mentioned, some defect analysis is simple in nature, such as assigning bug-fix priorities (Kelly and Shepard, 2001). Moreover, much defect analysis is taken on as a method of statistical quality assurance and other more extensive tasks.

The actual classification scheme used to record these defects is dependent on the actual use to which the defect data will be put. Both Huber (1999) and

Freimut (2001) describe ODC as a scheme whose primary purpose is to provide feedback on the progress of the current project, as illustrated in Chillarege et al (1992). The Hewlett-Packard scheme and the IEEE Standard Classification for Software Anomalies both aim to improve the development process by reducing the number of defects injected over time (Freimut, 2001). The IEEE scheme differs slightly in that it is aimed at audiences who wish to implement their own scheme compliant to a proven standard. In addition, the Hewlett-Packard scheme is typically used as a retrospective scheme, after development and testing is complete (Huber 1999) whereas both ODC and the IEEE scheme can be used throughout the development cycle.

The Hewlett-Packard scheme has an advantage when compared with ODC and the IEEE scheme, in that it is a very small scheme and hence easier to implement (Huber 1999). On the contrary, such schemes as ODC and IEEE provide a wealth of data allowing avenues other than process improvement to be investigated e.g. Product assessment or review effectiveness. The price for this richness is that these schemes require more effort to implement due to the complexities of the categories and classifications.

Both the IEEE scheme and the Hewlett Packard scheme could be regarded as incomplete. IEEE is a standard and it is expected that the scheme be augmented with classifications used by the implementing organisation (IEEE 1996, p.4). The Hewlett Packard scheme provides an “Other” classification for the Origin category. However, if we follow Chillarege et al’s (1992 p. 946) discussion of “Sufficient Conditions”, which asserts that the classifications of a category should span all possible scenarios, it could be argued that defining this ‘Other’ classification is tantamount to taking the easy way out. However, Freimut’s (2001 p.17) concept of “Complete Attributes Values” in his paper on developing and using defect classification schemes does allow for this, although only as a temporary measure on a scheme which is not yet complete.

The scheme described by Leszak et al (2000) might also face problems by providing an “Other” root-cause category. Chillarege et al (1992 p. 945) state that a good measuring system should be orthogonal and therefore the “*choices* *should be uniquely identified and easily classified*". The addition of this "Other" category could be seen to overlap the other root-cause categories and hence complicate this selection process.

2.4.4 Analysis

A reduction in the number of defects can be brought about by performing statistical analysis on the data collected using a classification scheme, and using the resulting data to feed back into the development process. This concept allows an organisation to discover and remove vital defect root causes, a process known as defect prevention (Pressman, 2002 p.207). Defect prevention is a key process area of the Capability Maturity Model Integration (CMMI) level 5 – Optimising, the highest level of a widely-used approach employed to guide process improvement across a project, a division, or organization (SEI 2002); hence, it is employed by all high-maturity organisations.

Examination of the frequency distribution and Chi-squared tests of uniformity can help to indicate if the classifications within a category are equally likely to occur at a given level of significance (IEEE 1996). Correlation techniques such as the Chi-squared test of independence can then be used to identify causation between categories, where a positive test of dependence would imply a significant relationship between at least one of the classifications in each category (ibid). Once a dependency has been recognised additional statistical tests can be carried out to decipher any predictive relationships between the two categories, allowing the user to discover any trends that require deeper investigation.

Freimut (2001) also recommends visual aids such as Pareto charts to examine classification frequencies. The advantage here being that they are easily and quickly understood. Chillarege et al (1992) recommends comparing the defect type distribution with pre-determined expected frequencies to understand the current state of a development. Freimut (2001) also discusses data-mining techniques such as the "Attribute Focusing Method", which compares the actual frequency of classifications or classification pairs with expected frequencies and pinpoints those that deviate the most.

2.4.5 Defect Analysis Summary

Defect analysis provides a means to identify shortcomings in a project, understand their root causes and finally use this information in some way that is useful to the organisation. Several schemes exist to capture these defects and the choice of scheme depends on what goals you wish to achieve, whether the analysis will be retrospective or performed during the development, and the amount of effort an organisation wishes to expend on initial set-up, logging and data-analysis.

2.5 Summary

Project management provides tools and methods to work towards project objectives in a structured and organised way. However, project management alone does not provide a means to prepare for and handle unexpected circumstances that may arise along the way. This is the job of risk identification, analysis and management.

Defects provide a quantitative measure for known problems that occurred during a project. The analysis of these defects is often used to feedback into the development cycle in an attempt to improve project management and process techniques.

Much of the research surrounding defect analysis schemes is based on causal analysis, looking at how defect analysis can improve processes for future and current projects. It could be argued that risk management aims to reduce the impact of unforeseen circumstances by eliminating or decreasing the root-causes of these unforeseen circumstances. Not much research has been performed on the linking these two techniques; hence, the primary research of this dissertation will attempt to find a possible link between these two, by investigating if defect analysis can also be used to improve risk management techniques.

3 Primary Research - Method

3.1 The purpose of the research

The purpose of this research is to test statistically whether or not the strengths of the risk management techniques used in IT projects impact on the resulting defects found. In particular, it looks at the root causes of these defects, and tests for correlation between these root causes and the strengths of the corresponding risk management techniques that could have successfully mitigated against them.

3.2 The approach to the research

The research question lends itself to the study of two variables and the existence of a correlation between them: -

- The strength of the risk management technique used.
- The resulting failure rate of test cases aimed at producing defects whose root cause could be mitigated by this technique.

Hence, the null hypothesis of the research is: -

There is no relationship between the strength of the risk management technique and the subsequent failure rate of tests designed to produce defects whose root cause could be mitigated by this technique.

In order to compare these two variables, research was undertaken to perform a retrospective study of projects undertaken by an Internet Banking development area of a global bank. The steps taken were as follows: -

- Selection of the projects to provide the raw data
- Selection of the project risks to be analysed
- Use of appropriate methods for collecting and measuring the strength of the risk management techniques used in these projects
- Use of appropriate methods for collecting and measuring test case failure rate

- Statistically proving/disproving the correlation of these two variables

3.3 Project Selection

3.3.1 Selection Criteria for projects

Kitchenham et al (1995 p.942) state that when attempting to corroborate a causal relationship we need to control the attribute that we believe causes this relationship. This control variable, also known as a predictor variable, is easier to control in laboratory experiments, but it must be said that in an experiment such as this, it would be unethical to control the predictor variable (strength of risk management technique) in order to intentionally bring an increase in the response variable (the number of defects captured). Instead, projects have been selected with differing strengths of risk management techniques in order to see how the defect numbers differ.

In their paper "Preliminary Guidelines for Empirical Research in Software Engineering' Kitchenham et al (2002 p. 723) state "*researchers need to identify particular factors that might affect the generality and the utility of the conclusions*". Considering this, efforts were made to minimise all other project characteristics that might have some influence on the response variable, by selecting projects where these other characteristics remain near to constant.

The projects studied were selected from the projects delivered by the UK arm of a global bank's Internet Banking development centre. Below is a summary of characteristics of these projects that were considered a possible influence on the response variable, and the project selection criteria used to minimise their impact.

- Methodologies used – Select projects that use the same development methodologies.
- Experience of development team – Select projects where the members of the project team are from the same department.
- Outsourcing of work – Select projects where the extent to which outsourcing of work to the bank's coding centres in India is the same. In

the majority of the bank's Internet Banking projects, the guideline is that at least 25% of the development work be carried out in India.

- Experience of testing team – Select projects where the testing is performed by the same teams and testing takes place at the same stage of the project.

3.3.2 Selecting the projects

Release 9 of the bank's Internet Banking software had major UK development involvement. All of these projects used the bank's proprietary project management and development methodologies ruling out any effect resulting from different methodologies being used. All UK and India development involvement was handled solely by two IT departments, one in each country; hence, the skill level of the personnel on each project remained relatively constant. The following chart depicts the global development involvement for each project in the release, with the selected projects shown in red.

PROJECT	TYPE	% UK Involvement	% India Involvement	% Other involvement
Single sign-on	System	100		
Customer Alerting	Application	75	25	
Balance and Transaction enhancements	Application	75	25	
User Interface Management System updates	System	75	25	
Private Bank	Application	85	15	
New Bank	Application	75	25	
Foreign Exchange	Application	70	30	
Supply chain solutions enhancements	Application	10		90
Image Retrieval phase 3	Application	5		95

Table 2 Global development involvement for release 9 of the bank's Internet Banking software

The Single sign-on and User Interface Management System projects were disregarded as possible projects for this research as they both pertain to changes to the core functionality used by application programs. Core functionality changes were considered very different in nature from application programs, as the injection of a defect in a core program would ripple through to application programs and result in a disproportional number of logs being raised for what is effectively one defect.

Supply chain solutions and Image Retrieval phase 3 projects were both developed mainly outside of the UK and India, with a small proportion of work carried out in the UK. This resulted in a very different workforce for these two projects and hence it would be harder to ensure that the development team experience was not a contributing factor to the number of defects raised for a particular project.

The Customer Alerting, Balance and Transaction enhancements, Private Bank integration, New Bank and Foreign Exchange projects were all application type projects, sharing roughly the same split of development effort between UK and India, and hence formed the basis for my primary research.

3.4 Strength of risk mitigation technique

The company's project management methodology suggests that risks of the following types are identified and managed; those associated with size, structure, distribution, impact, technical complexity, conformity, skills, volatility, testing, and business functions. The table below shows the list of risks identified for the release 9 projects and in which of the selected projects they were identified and managed.

TYPE OF RISK	Risk	Customer Alerting	Balance & Transactions enhancements	Private Bank	New Bank	Foreign Exchange
Risk associated with size	Insufficient time to deliver full functionality	X				
Risk associated with structure	Minimal overlap of working hours between UK and India	X			X	X
Risk associated with distribution						
Risk associated with impact	All Internet banking tools will be impacted by core services					
Risk associated with technical complexity	Insufficient knowledge of the product's core services	X	X	X	X	X
	Complex data retrieval from more than one back-office system					
Risk associated with conformity	Lack of knowledge of new User Interface standards	X	X	X		X
Risk associated with skills	Lack of deployment knowledge	X	X			
	Lack of experience in India due high staff turnover	X	X			
Risk associated with volatility	Lack of frozen requirements	X	X	X		X
Risk associated with testing	Core changes may effect Internet banking tools containing obscure constructs					
	Lack of environment knowledge	X	X	X	X	X
Risk associated with business functions						
Other risks						

Table 3 Risks identified for release 9 projects

The risks shown in red were chosen as the input into this research. Each of these risks were identified and managed in all or most of the selected projects. This is an important factor, allowing risk management information pertaining to the same risk to be obtained from different projects, all of which may use very different mitigating techniques. Hence, one would be more likely to find the existence of varying strengths of mitigation techniques used in the selected projects.

To conclude, a short synopsis of the selected risks is shown below: -

- Lack of knowledge of new User Interface standards - Prior to this release of the Internet banking system, new standards to the user interface "look and feel" were set out. Release 9 would be the first release to accommodate these new standards. As the changes were made at such a late stage and were very unfamiliar to the developers, the majority of project leaders recognised that implementing these new standards represented a risk to the project.

- Lack of frozen requirements - The IT department is funded by a very busy core-banking department. The result is that often user involvement is hard to come by, hence requirements are not always stabilised early in the project. Coupled with this, the Internet banking system is a global product and hence is subject to the opinions of a global management team vying for their own personal agenda. This has led to late requirement changes in the past; hence, this was identified as a possible risk in the majority of projects.

- Insufficient knowledge of the product's core services - The product provides a range of tools used for implementing standard functions (e.g. entitling a user to a new tool, logging user activity, automated email/SMS). Being such a large product developed by many IT staff worldwide, this information is often unfamiliar to application developers accustomed to working within one area of the product. For this reason it was considered a risk that development teams might use existing services incorrectly.

- Lack of environment knowledge - The interdependency between systems and the data required to ensure the integrity and correct processing of the system is very complex. An example is the user-id, which needs to be synchronised in multiple areas - front end sign-on, tool entitlements and the various back-ends which hold data for that user. This must be set-up correctly for each testing environment if tests are to be run correctly. As the architecture of the environment is complex, the setting up of this test data can be considerably complicated; hence, this was considered a high risk.

3.4.1 Quantifying strength of risk management technique

Strength of a technique is a very subjective measurement. This subjectivity does not lend itself to a process being repeatable, as different people will have different insights and opinions. Hallowell DL (n.d.) also suggests that a measurement scale based on subjectivity is "*just ordinal at best*" and hence no interpretation can be made from their values. That is to say, if the strength of a risk management technique was given a value of 10, it does not imply that this technique was rated twice as highly as one given a rating of 5. In order to successfully use strength of risk mitigation in any correlation technique, this subjective measure must be converted into a continuous measure. The Analytical Hierarchy Process (AHP) suggests an approach to the quantification of subjective judgements pertaining to the relative importance of various features of the items being measured. If we were to take the quality of a car as an example, an actual ranking of quality might be based on subjective judgements of the importance of various quality features such as comfort of seats, spaciousness and engine noise.

In order to quantify the strength of risk management techniques used on these projects, selected members of the project teams were invited to: -

- Decide upon the features that would influence the strength of risk management techniques
- Perform subjective judgements in order to derive weightings for the importance of each of these features
- Perform subjective judgements to rank specific features for specific risks on each project. This would give a measure of strength for a particular

feature with regard to a specific risk on a specific project, in relation to the same feature for the same risk on all other selected projects.

A brainstorming session with selected members of the project team provided the following list of features that would influence the strength of a risk management technique: -

- Perceived importance of the project by senior management
  The selected members all agreed that the perceived importance of the project to senior management would mean that the project progress and results would come under more scrutiny than most. The impact of this scrutiny could manifest itself as a more rigorous risk management plan in order to safeguard against a very public failure.

- Dependencies with other projects
  The selected members all agreed that the greater the level of dependencies with other projects then the likelihood of mistakes being kept internal would decrease. As a result, the actual risk management techniques would be strengthened to avoid publicised failings.

- Experience of risk project leader
  As Addison and Vallabh (2002) discuss, many risks are considered less problematic as the experience of the project manager grows, suggesting that the methods for managing these risks become second nature with experience. The selected members agreed that the experience of the project leader would therefore go some way to enhancing the method used to mitigate each risk.

- Risk Exposure of the risk
  The actual measure of risk exposure for each risk as recorded in the risk logs would also provide a measure of the strength of the risk management technique to be used. A logical argument would be that the bigger the exposure the stronger the management technique would need to be.

- Effectiveness of the risk management technique used
  The selected members felt that analysis of the risk logs would provide evidence of the risk management used and as such would provide an insight into the suitability of the technique. However, such a measurement was considered too hard to judge comparatively and hence was disregarded as a possible feature of risk strength.

The Release 9 project co-ordinator and the senior solutions architect for the release were then asked to complete: -

- The risk feature questionnaire to provide comparative judgements of the importance of each feature with regard to the strength of a risk management technique
- The risk comparison questionnaire to provide comparative judgements of the strength of these features in the application of a risk management technique for a given project / risk

These two personnel were chosen because of the extent of their involvement in the day-to-day running of all of the UK run release 9 projects. Not having any allegiance to a specific project, they were able to supply a more holistic view of the release and in this way, any specific project bias was minimised.

The results of these questionnaires would then feed into AHP calculations. In order to improve the level of agreement between these two assessors, both were invited to discuss these features and the strengths of the risk management techniques used, allowing discussion of any difference of opinion prior to completing the questionnaires.

3.5 Measuring test case failure rate

In order to calculate the test case failure rate, a relationship was explored between test cases and the types of defects that they could uncover. In summary: -

- Raw data was collated; test cases and defect logs from the defect logging system used
- The test cases were categorised by type of test performed
- The defects were categorised according to a classification schema
- The types of defects which would be uncovered by test cases of a particular type were specified
- The test case failure rate was calculated as: -

$$\frac{\text{number of defects that would be uncovered by test cases of type X}}{\text{number of test cases of type X}} * 100$$

This process is now discussed in more detail.

3.5.1 The raw data

The raw data for this primary research was collected during the system-acceptance testing phase of the selected projects. All test cases and their associated defects for the selected projects were originally captured in a "Quality Centre" logging system. The test case data consisted of details of 362 tests to be run including any preconditions that first needed to be met. The defect data consisted of descriptions of 97 defects found during these tests and a root cause decided at the point of defect creation. This root cause was sometimes updated by the coder on further analysis of the defect. Other categories were stored for each defect such as priority and status, but such categories were used to improve the ability of the development area to fix defects quickly, as opposed to providing any long-term process improvement information. The split of test cases and defects between the five selected projects are illustrated in the table below: -

Projects	Number of test cases	Number of defects found
Private Bank	68	28
Foreign Exchange	41	14
Customer Alerting	136	27
New Bank	77	11
Balance and Transactions Enhancements	40	17
TOTAL	362	97

Table 4 Split of test cases and defects between the selected projects

3.5.2 Categorisation of the test cases

During discussion with members of the testing team, the test cases were categorised as to understand the relationship between the test cases written and the categories of errors that the test case author was attempting to catch. The discussion was geared towards creating categories relating to the risks selected in the previous section, allowing easier correlation between strength of risk management techniques used and defect failure. These relationships are illustrated below

Risk	Testing Category	Description of tests
Lack of knowledge of new User Interface standards	Non-conformance of user interface	Those tests which specifically pertain to the conformity of applications to User Interface standards
Lack of frozen requirements	Functional testing	Those tests which pertain to checking the business rules of the system
Insufficient knowledge of the product's core services	Subsystem Interface testing	Those tests involving communication between more than one subsystem
Lack of environment knowledge	Data set-up/Environment testing	Those tests involving the set-up or synchronisation of data

Table 5 Relationship between risks and category of test

It should be noted that each test could belong to more than one category. That is to say, a test case might detail a functional requirement but also detail that the results adhere to the user interface standards. The results of this categorisation process gave rise to the following distribution of testing categories.

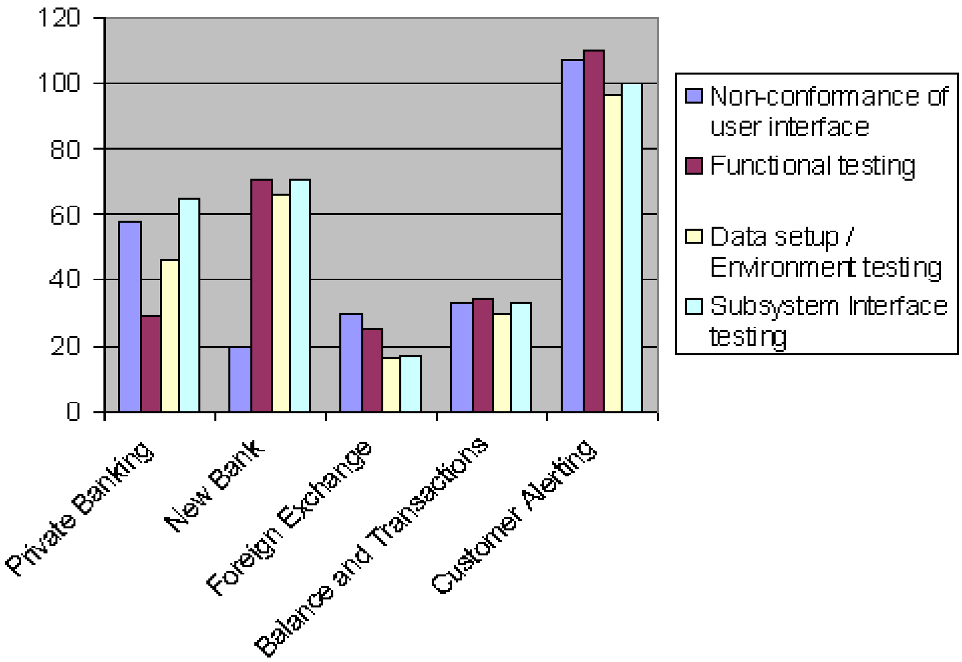

Figure 3 Distribution of test cases amongst selected projects

3.5.3 Re-classification of the defect root causes

Analysis of the defects logged for the selected projects showed the following root cause entries; Bad Test Data / Set Up, Code Error, Design Error, Enhancement Request, Existing Feature, Expected Results Incorrect, External Error, Network/Infrastructure Error, Requirements Error and Tester Error.

Further analysis showed these classifications to be inadequate for the following reasons: -

- As there was no predetermined classification schema, different testers had assigned defects with classifications of their own making; hence, defects of a similar nature were incorrectly assigned to different root causes.
- Some of the classifications for root cause are vague and therefore make the selection of the correct classification ambiguous

To combat this, the defects were re-classified using a more robust, industrial classification schema. IBM's ODC scheme was considered, but felt to be more suited to analysis of project progress as opposed to the retrospective analysis to be performed here. Both the IEEE Standard Classification for Software

Anomalies and the HP Defect scheme provide good process improvement classifications; the Hewlett-Packard schema was eventually chosen for its ease of use. This re-classification was performed in conjunction with selected testing and development personnel, and where necessary analysis of requirements, specification and design documents was undertaken.

One new classification was added to the Type category for the Code origin. This classification NLS Data set-up is specific to the setting up of static data for National Language Support (NLS) translations. This static data set-up function was vital if a project was to conform to the new UI standards put in place before release 9.

The classification was deemed necessary to differentiate between NLS static data set-up issue and other data set-up issues. The new NLS Data set-up classification would therefore change the scope of the existing Hewlett Packard Data Handling Problems classification, as NLS data problems could no longer be attributed to it.

The tables below summarise the results of the re-classification of defects to the Hewlett Packard scheme.

Project	Origin	Type	Total
Private Bank	Code	Logic	13
	Code	Standards	7
	Code	NLS Data set-up	3
	Environmental Support	Integration Software	2
	Specification Requirements	Specification	3
	SUB-TOTAL		**28**
Foreign Exchange	Code	Computation	1
	Code	Logic	3
	Code	Standards	3
	Code	NLS Data set-up	1
	Design	Module Design	2
	Environmental Support	Integration Software	3
	Specification Requirements	Requirements	1
	SUB-TOTAL		**14**
Customer Alerting	Code	Logic	10
	Code	Standards	4
	Design	Module Design	2
	Design	S/W Interface	1
	Environmental Support	Integration Software	3
	Environmental Support	Test Software	2
	Specification Requirements	Requirements	5
	SUB-TOTAL		**27**
New Bank	Code	Logic	4
	Code	Standards	5
	Environmental Support	Integration Software	1
	Specification Requirements	Requirements	1
	SUB-TOTAL		**11**
Balance & Transactions enhancements	Code	Logic	4
	Code	Standards	7
	Code	NLS Data set-up	1
	Design	Module Design	1
	Environmental Support	Integration Software	2
	Specification Requirements	Requirements	2
	SUB-TOTAL		**17**
TOTAL			**97**

Table 6 Re-classification of defects

3.5.4 Relating test cases to defects

With an aim to mapping each testing category to specific origin/type combinations of defects likely to be discovered by these types of test, testing personnel were engaged in further discussion. This discussion gave rise to the following relationships between test cases and defects.

<table>
<tr><th rowspan="2">Testing Category</th><th colspan="2">Hewlett Packard Classification</th><th rowspan="2">Comment on selected origin/type combination</th></tr>
<tr><th>Origin</th><th>Type</th></tr>
<tr><td rowspan="6">Data set-up / Environment testing</td><td>Environmental Support</td><td>Integration Software</td><td rowspan="6">Defects of these origin/type combination deal with the set-up of the testing environment and associated data</td></tr>
<tr><td>Environmental Support</td><td>Test Software</td></tr>
<tr><td>Environmental Support</td><td>Test Hardware</td></tr>
<tr><td>Environmental Support</td><td>Development Tools</td></tr>
<tr><td>Environmental Support</td><td>Integration software</td></tr>
<tr><td>Design</td><td>H/W Interface</td></tr>
<tr><td rowspan="3">Subsystem Interface testing</td><td>Design</td><td>Software Interface</td><td rowspan="3">Defects of these origin/type combination deal with the way related sub-systems interact</td></tr>
<tr><td>Design</td><td>Inter-process comms.</td></tr>
<tr><td>Code</td><td>Module/Interface Implementation</td></tr>
<tr><td rowspan="3">Non-conformance of user interface</td><td>Design</td><td>Standards</td><td rowspan="3">Defects of these origin/type combination deal with UI standards and UI specific data set-up</td></tr>
<tr><td>Code</td><td>Standards</td></tr>
<tr><td>Code</td><td>NLS Data set-up</td></tr>
<tr><td rowspan="3">Functional testing</td><td>Specification Requirements</td><td>Requirements</td><td rowspan="3">Defects of these origin/type combination pertain to products that are agreed with the customer throughout the project life</td></tr>
<tr><td>Specification Requirements</td><td>Specification</td></tr>
<tr><td>Specification Requirements</td><td>Functionality</td></tr>
</table>

Table 7 Relationship between testing category and Hewlett Packard classification (Origin and Type)

Using these relationships reclassified defects could then be matched with the corresponding testing categories, which would allow defects of those types to be discovered. It should be noted that some defects belonged to classifications that did not relate to the selected testing categories. These defects were therefore excluded from further calculations, as without a related testing category it would be impossible to relate these defects back to specific risks. The tables below illustrate this mapping for each of the selected projects.

Private Bank				
Testing Category	**Origin**	**Type**	**Total**	**Total per Category**
Other	Code	Logic	13	-
Non-conformance of user interface	Code	Standards	7	10
	Code	NLS Data set-up	3	
Data set-up / Environment testing	Environmental Support	Integration Software	2	2
Functional testing	Specification Requirements	Specification	3	3

Table 8 Mapping of defects for the Private Bank project

Foreign Exchange				
Testing Category	**Origin**	**Type**	**Total**	**Total per Category**
Other	Code	Computation	1	-
	Code	Logic	3	
Non-	Code	Standards	3	4
	Code	NLS Data set-up	1	
Subsystem Interface testing	Design	Module Design	2	2
Data set-up / Environment testing	Environmental Support	Integration Software	3	3
Functional testing	Specification Requirements	Requirements	1	1

Table 9 Mapping of defects for the Foreign Exchange project

Customer Alerting				
Testing Category	**Origin**	**Type**	**Total**	**Total per Category**
Other	Code	Logic	10	-
Non-conformance of user interface	Code	Standards	4	4
Subsystem Interface testing	Design	Module Design	2	3
	Design	S/W Interface	1	
Data set-up / Environment testing	Environmental Support	Integration Software	3	5
	Environmental Support	Test Software	2	
Functional testing	Specification Requirements	Requirements	5	5

Table 10 Mapping of defects for the Customer Alerting project

New Bank				
Testing Category	**Origin**	**Type**	**Number**	**Total per Category**
Other	Code	Logic	4	-
Non-conformance of user interface	Code	Standards	5	5
Data set-up / Environment testing	Environmental Support	Integration Software	1	1
Functional testing	Specification Requirements	Requirements	1	1

Table 11 Mapping of defects for the New Bank project

Balance & Transactions Enhancements				
Testing Category	**Origin**	**Type**	**Number**	**Total per Category**
Other	Code	Logic	4	-
Non-conformance of user interface	Code	Standards	7	8
	Code	NLS Data set-up	1	
Subsystem Interface testing	Design	Module Design	1	1
Data set-up / Environment testing	Environmental Support	Integration Software	2	2
Functional testing	Specification Requirements	Requirements	2	2

Table 12 Mapping of defects for the Balance & Transactions Enhancements project

3.6 Correlation

Correlation of the resulting variables was then undertaken to measure the existence of a relationship between the strength of the management technique and the subsequent failure rate of the category of tests that could be mitigated by these techniques.

3.7 Summary

This research has been based on projects of the ninth release of an Internet banking system developed mostly in the UK with some work being outsourced to India. Care was taken with the selection of these projects to ensure no external factors, such as level of outsourcing would affect the outcome of the experiment.

The project risk logs were then analysed to identify the risks that were common to all or most of these projects. Those that were identified by most of the projects were used as the basis of this research.

In order to provide an objective measure of strength of each risk management technique, an Analytical Hierarchy Process (AHP) calculation was undertaken. This involved the selection of features that would affect the strength of a risk management technique, and a series of comparative judgements to rank the importance of each feature with regard to technique strength, and rankings for these features for each project/risk. As the risk logs provided numerical rankings for risk exposure, no comparative ranking judgements were produced for this feature. Instead, these numerical values were normalised and fed into the AHP calculations. The outcome was a measure of the strength for each technique used to mitigate each risk on each project.

Raw data for the projects was then gathered from a Quality Centre logging system to provide details of test cases used for the system-acceptance testing phase of this release and the resulting defects raised. The test cases were categorised, with each test case being associated with one or more testing categories. These test categories were linked back to the originally selected risks. For example, the testing category "Non-conformance of user interface" was linked back to the risk "Lack of knowledge of new User Interface standards".

Due to the unsuitability of the defect classification used by the release, the defects were re-classified to a more robust scheme, the Hewlett Packard scheme. A mapping was decided to map the origin and type categories of each defect to a testing category, i.e. the testing category likely to discover defects of this origin/type combination. This mapping enabled the calculation of a test case failure rate.

$$\frac{\text{number of defects that would be uncovered by test cases of type X}}{\text{number of test cases of type X}} * 100$$

Failure rates were then calculated for each testing category. These failure rates along with the measures of technique strength were then input to correlation calculations for each risk.

4 Primary Research - Results

4.1 Analytical Hierarchy Process calculation

The actual AHP calculation consisted of four steps: -

1. Calculation of a weighting factor for each feature affecting the strength of a risk management technique
2. Calculation of rankings for the *perceived importance, level of dependencies* and *project leader experience* features with regard to each project
3. For each risk identified, calculation of rankings for the *risk exposure* feature with regard to each project
4. For each risk, addition of the weighted rankings giving a figure for the strength of each risk management technique used

Note: Steps 1 and 2 were completed for both assessors and then averaged to give due consideration to the measurements from both assessors.

The diagram below illustrates the entities to be used for the comparative judgements of each step.

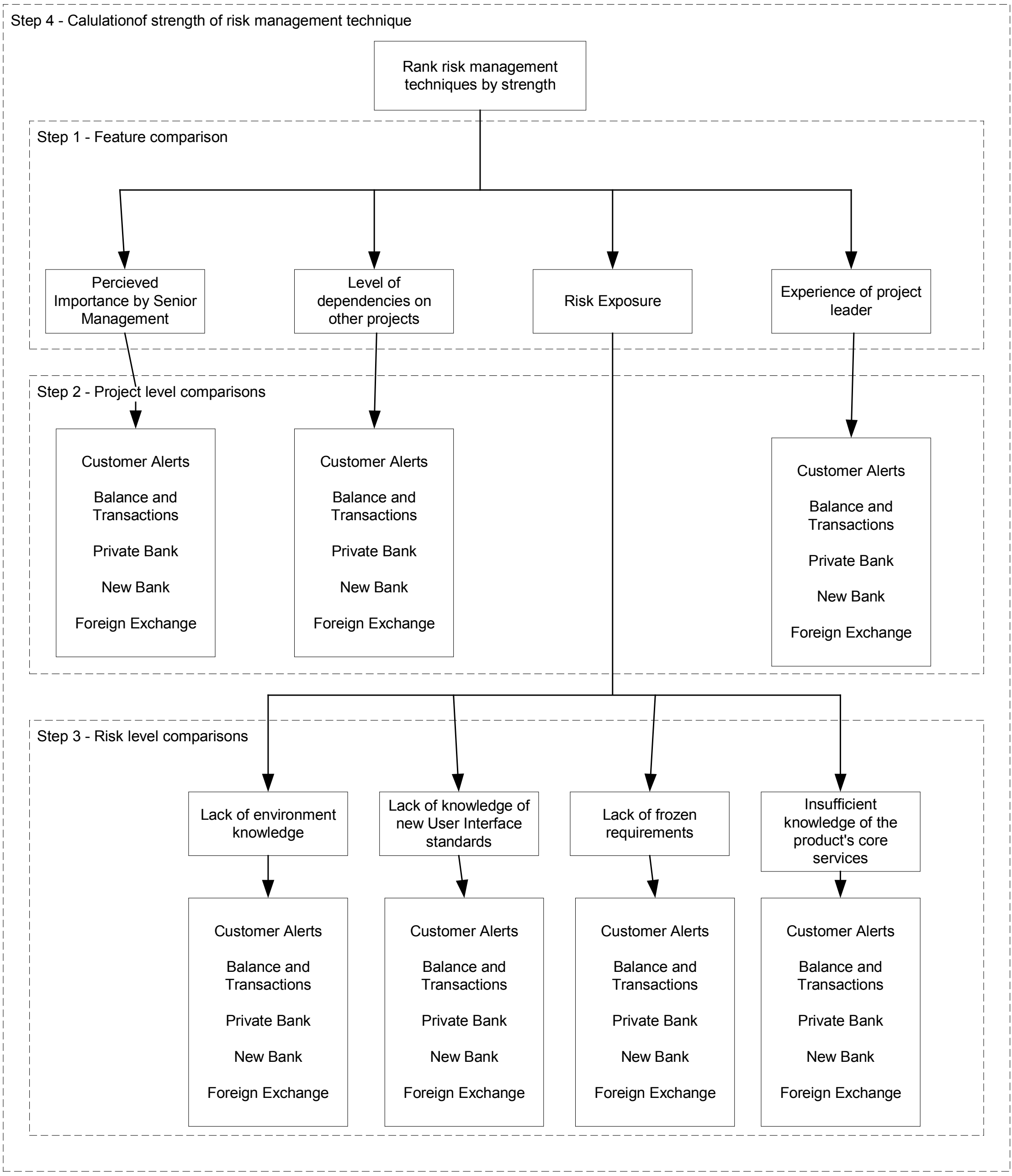

Figure 4 Steps of the AHP calculation

Calculation of the weightings for each feature provided a factor by which individual feature rankings for a project or risk can be multiplied. Hence, it follows that the more important a feature is perceived the higher its weightings will be. Weightings were calculated using the geometric means method (see Appendix B for details). The table below shows the results of this calculation.

	Project coordinator	Snr. Solutions Architect	Average
Perceived importance	0.079	0.056	0.068
Dependencies	0.079	0.112	0.096
Project Leader Experience	0.407	0.332	0.369
Risk Exposure	0.435	0.5	0.467

Table 13 Weighted Priorities for feature comparison

Geometric means were also used to determine the normalised ranking for the *perceived importance, level of dependencies* and *project leader experience* features. The tables below summarise these calculations

	Project coordinator	Snr. Solutions Architect	Average
Customer Alerting	0.088	0.128	0.108
Balance & Transactions Enhancements	0.522	0.512	0.517
Private Bank	0.244	0.25	0.247
New Bank	0.105	0.068	0.086
Foreign Exchange	0.041	0.0415	0.041

Table 14 Normalised ranking for the perceived importance feature

	Project coordinator	Snr. Solutions Architect	Average
Customer Alerting	0.489	0.4582	0.474
Balance & Transactions Enhancements	0.105	0.166	0.136
Private Bank	0.271	0.166	0.218
New Bank	0.095	0.166	0.13
Foreign Exchange	0.039	0.045	0.042

Table 15 Normalised ranking for the level of dependencies feature

	Project coordinator	Snr. Solutions Architect	Average
Customer Alerting	0.531	0.485	0.508
Balance & Transactions Enhancements	0.221	0.205	0.213
Private Bank	0.042	0.074	0.058
New Bank	0.105	0.08	0.093
Foreign Exchange	0.101	0.156	0.1283

Table 16 Normalised ranking for the project leader experience feature

Risk values were recorded directly from the project logs and then normalised (see Appendix B for details). By doing so, the need to provide comparative judgements for risk exposure is negated. The following table details the results of the ranking process.

	Customer Alerting	Balances and Transactions enhancements	Private Bank	New Bank	Foreign Exchange
Lack of knowledge of new User Interface standards	0.357	0.286	0.159	0	0.198
Lack of frozen requirements	0.195	0.395	0.351	0	0.059
Insufficient knowledge of the product's core services	0.393	0.238	0.306	0.044	0.019
Lack of environment knowledge	0.273	0.445	0.109	0.136	0.036

Table 17 Normalised ranking for the risk exposure feature

Calculation of the techniques strength was calculated as the summation of the feature ranking for a risk multiplied by the weighting for that particular feature. Hence, for a risk R1 on a project P1, the strength would be calculated as: -

P1R1 Risk exposure ranking * risk exposure weighting

+

P1 Perceived Importance ranking * Perceived Importance weighting

+

P1 Dependencies ranking * Dependencies weighting

+

P1 Project Leader Experience ranking * Project Leader Experience weighting

The following table and bar chart show the risk strengths obtained from these calculations (see Appendix B for details)

	Customer Alerting	Balances and Transactions enhancements	Private Bank	New Bank	Foreign Exchange
Lack of knowledge of new User Interface standards	0.407	0.26	0.133	0.053	0.147
Lack of frozen requirements	0.332	0.311	0.223	0.053	0.082
Insufficient knowledge of the product's core services	0.424	0.238	0.202	0.073	0.063
Lack of environment knowledge	0.368	0.335	0.11	0.116	0.071

Table 18 Strengths of risk management techniques (calculated using AHP)

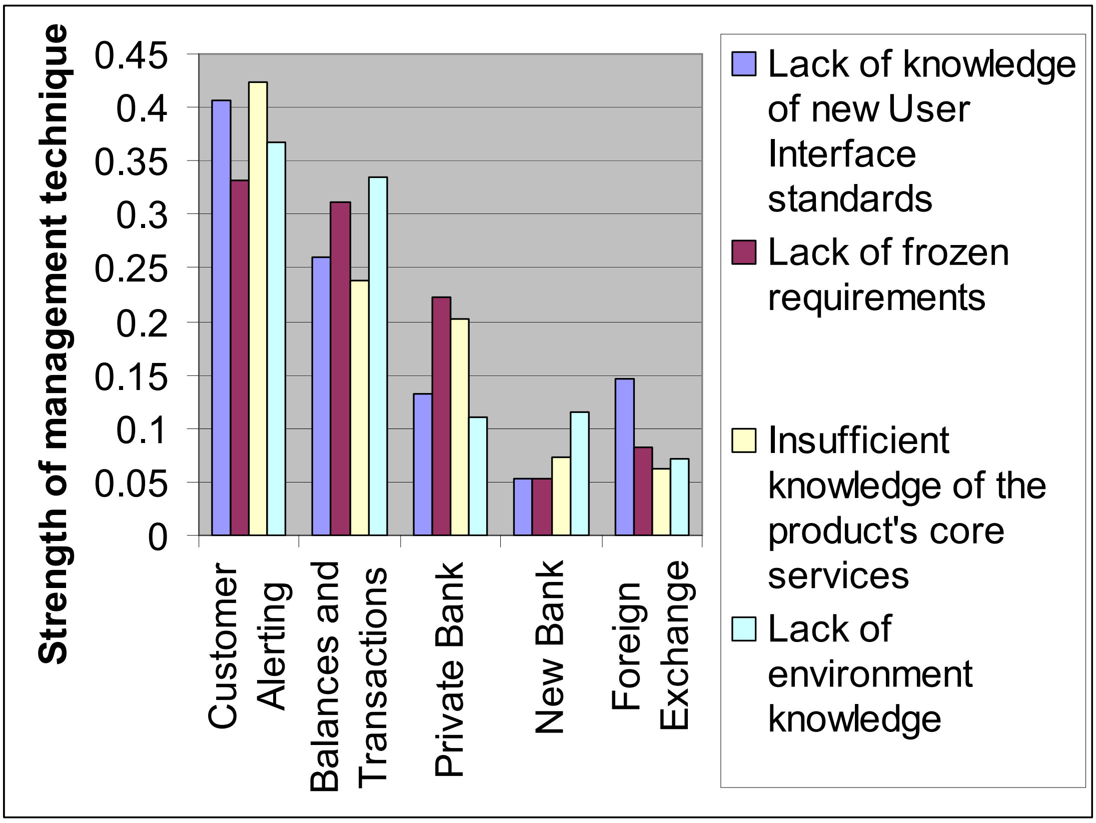

Figure 5 Strengths of risk management techniques

4.2 Calculating the test case failure rate

In order to provide a useful measurement for the number of defects, the test case failure rate was chosen as the most accurate measure with regard to judging successful prevention of a particular type of defect. Solely using the number of defects of that type found would mean that the findings would differ depending on the size of the project. Take as an example the following two projects: -

- Project A where 20 defects of type X were found from 40 test cases aimed at producing type X defects
- Project B, where 20 defects of type X were found from 200 test cases aimed at producing type X defects

Using the number of defects as a measure, the results would be recorded as the same i.e. 20. In reality project A had a 50% failure rate whilst project B had only a 10% failure rate. Thus, failure rate gives a much more insightful measure of defects across different projects.

Also simply taking a percentage of the number of defects of a specific type from all tests run does not take into account the bias of the testing. An example would be if test team were particularly good at creating tests for adherence to user interface standards, but not so proficient at defining tests that tested the interfaces between sub-systems

The following table shows the failure rate for each testing category using the formula

$$\frac{\text{number of defects that would be uncovered by test cases of type X}}{\text{number of test cases of type X}} * 100$$

Project		Type of test			
		Non-conformance of user interface	**Functional testing**	**Data set-up / Environment testing**	**Subsystem Interface testing**
Private Bank	Number of tests	58	29	46	65
	Defects pertaining to test	10	3	2	0
	Failure rate	**17.24%**	**10.34%**	**4.35%**	**0.00%**
New Bank	Number of tests	20	71	66	71
	Defects pertaining to test	5	5	5	3
	Failure rate	**25.00%**	**7.04%**	**7.58%**	**4.23%**
Foreign Exchange	Number of tests	30	25	16	17
	Defects pertaining to test	4	1	3	2
	Failure rate	**13.33%**	**4.00%**	**18.75%**	**11.76%**
Balance and Transactions Enhancements	Number of tests	33	34	30	33
	Defects pertaining to test	8	2	2	1
	Failure rate	**24.24%**	**5.88%**	**6.67%**	**3.03%**
Customer Alerting	Number of tests	107	110	96	100
	Defects pertaining to test	4	5	5	3
	Failure rate	**3.74%**	**4.55%**	**5.21%**	**3.00%**

Table 19 Failure rate of the testing categories

The results show a good range of failure rates amongst the categories, ranging from 0% failure rate to a 1 in 4 (25%) failure rate.

4.3 Statistical correlation

Correlation was undertaken to judge the existence of a relationship between the strength of the management technique and the subsequent failure rate of the category of tests that could be mitigated by these techniques. These correlations are illustrated in the following scatter graphs: -

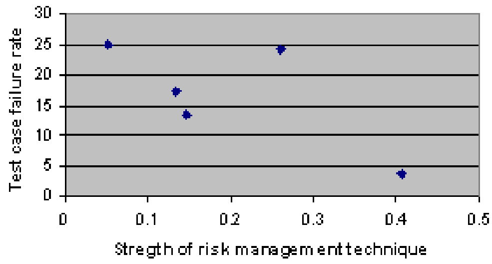

Figure 6 Scatter graph for risk "Lack of knowledge of new user interface standards"

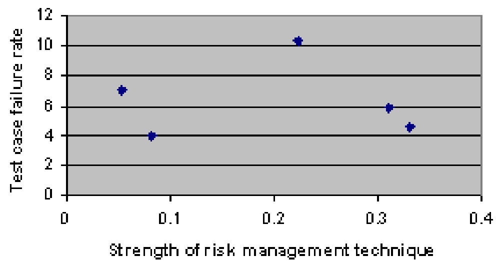

Figure 7 Scatter graph for risk "Lack of frozen requirements"

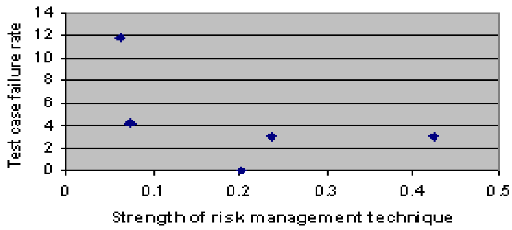

Figure 8 Scatter graph for risk "Insufficient knowledge of the product's core services"

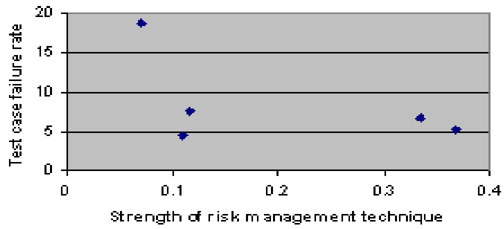

Figure 9 Scatter graph for risk "Lack of environment knowledge"

Visually there appears to be some small measure of correlation for three of the four risks, the exception being the "Lack of frozen requirements" risk. The correlation also seems to be negative, as one would hope, showing a decrease in test case failure rate with increases in the strength of the risk management technique. McNaughton (1997) points out numerical techniques are more likely to detect a relationship than graphical techniques if the relationship is weak or subtle, as well as also providing an "*objective measure of believability*". Hence, the correlation coefficient as described by Bee and Bee (1990) was used to test for the existence of a relationship. The coefficient (r) is calculated as: -

$$r = \frac{\sum xy - (\sum x \sum y)/n}{\sqrt{(\sum x^2 - ((\sum x)^2/n) * (\sum y^2 - ((\sum y)^2/n))}}$$

Using this calculation the correlation between the strengths of the risk management techniques (x value) and the resulting related test category failure rate (y value) was calculated. In addition, in order to determine the probability that these correlations occurred by chance, a two tailed level of significance test was performed as detailed by the British Medical Journal (BMJ.com, 2006).

Risk	Correlation coefficient	Percentage of failure rates that can be explained by the strength of the associated risk management technique	T test value	Significance level	Statistically significant (5%)
Lack of knowledge of new User Interface standards	-0.69	47.55%	22.6253	0.1%	Y
Lack of frozen requirements	-0.016	0.03%	0.14407	>50%	N
Insufficient knowledge of the product's core services	-0.525	27.56%	9.00482	1%	Y
Lack of environment knowledge	-0.51	25.98%	8.3843	1%	Y

Table 20 Correlation of technique strength and test category failure rate

The American College of Physicians (ACP 2006) discuss the power of an outlier, a single extreme data point, on the correlation coefficient when the size of the sample is small. In order to understand the effect of outliners on this research, the Spearman rank correlation was calculated. This technique suggests that values are ranked prior to the calculation of the coefficient as described in Appendix B. This coefficient is calculated as: -

$$r = 1 - \frac{6\sum d^2}{n(n^2 - 1)}$$

The results of these correlations and the resulting significance tests are illustrated below: -

Risk	Spearman Correlation coefficient	Percentage of failure rates that can be explained by the strength of the associated risk management technique	T test value	Significance level	Passed 5% significance level
Lack of knowledge of new User Interface standards	0.7	49.00%	24.2215	0.1%	Y
Lack of frozen requirements	0.2	4.00%	1.95313	50%	N
Insufficient knowledge of the product's core services	0.7	49.00%	24.2215	1%	Y
Lack of environment knowledge	0.5	25.00%	8	1%	Y

Table 21 Spearman rank correlation of technique strength and test category failure rate

4.4 Summary

The results of these correlation calculations showed virtually no correlation for "Lack of frozen requirements" and quite poor correlation for the remaining three risks. The correlation coefficients were then recalculated using the Spearman rank coefficient to counteract the influence of outliners. This calculation provided slight improvements to three of the coefficients and a worsening to the "Lack of environment knowledge" coefficient.

Weak negative correlation is shown by both sets of calculations. The direction of these correlations is encouraging as it hints towards improved risk management techniques giving lower test case failure rates. However, neither the original correlation calculations nor the Spearman rank correlations could be described as giving strong enough results to show an association between these two variables, hence, the null hypothesis must be accepted. Therefore, this experiment cannot prove a relationship between the strength of the risk management technique and the subsequent failure rate of tests designed to produce defects whose root cause could be mitigated by this technique.

5 Conclusions

5.1 Introduction

The motivation behind this research project was to explore a method of improving IT project success rates. Risk management was considered a major factor in the success or failure of an IT project due to its ability to mitigate many of the pitfalls of project success (Addison and Vallabh (2002), Kiel et al (1998)). Defect analysis provides an effective way of identifying shortcomings of an IT project, and as such, provides a quantitative measure for known project problems.

This project aimed to examine the relationship between these two areas, risk management and defect analysis, by examining the correlation between the strength of a risk management technique and the test-case failure rate (i.e. percentage of test cases which discovered a defect).

5.2 Project Review

The preceding chapters show that this research project has gone some way to understanding how these two variables can be derived, and hence, allow simple correlation calculations to be made upon them. The techniques used are based on industrial-strength concepts such as the Analytical Hierarchy Process and the Hewlett Packard classification scheme. However the research may lack some depth with regards to the size of the data and the number of personnel involved in some of the decision making processes, such as the creation of a feature list with which to rank risk mitigation techniques. In addition, there was a distinct lack of secondary data available that related risk management techniques with defect analysis. Below is a synopsis of some of the possible factors that may have affected the accuracy of this research.

5.2.1 Relevance of defect data

The raw defect data for this research was collected during the system testing cycle for a major release. However, the Hewlett Packard software testing extension (Huber 1999) would suggest that by collecting data from a single testing phase we are more likely to find defects of certain types than of others. This is also asserted by Chillarege et al (1992) who discuss the change of the defect type distribution over each phase of a project.

The effect of this defect distribution on this research may mean fewer defects being found for test cases relating to the "Non-conformance of user interface" and "Subsystem interface" testing categories. This would be due to likelihood of finding defects of these types in earlier testing phases (i.e. the unit testing and integration testing phases).

5.2.2 Sample size

Three issues arose when selecting the data: -

- The number of projects whose data and personnel were available
- The time required for analysing each project
- The time that could be afforded by project personnel to aid the research

McNaughton (1997 p. 24) states that "*generally, the more entities in the sample, the more powerful the statistical tests*". With this in mind, the ideal consideration would be to minimise the requirement for staff involvement whilst still providing sufficient data to feed into correlation calculations. Initial thoughts were to examine one risk over many projects. This would provide one large set of data to feed into correlation calculations, however in order to keep control over external factors (Kitchenam et al 2002 p. 723), the choice of projects was reduced. This reduction of the number of projects available meant a reduction in the size of the data to feed into a single correlation calculation and hence, a reduction in the significance of the tests.

One possible means of avoiding this small sample size would be to combine the samples for all four risks (5 data points each) into one sample (20 data points).

However, in the light of section 5.2.1-Relevance of defect data, combining these samples would not give a true indication of the correlation. This is because different defects, which map to different testing categories and therefore to different risks, are likely to be detected in different testing phases. Hence, the likelihood of finding defects relating to the "Non-conformance of user interface" and "Subsystem interface" testing categories might lessen, as many of these defects might be found during earlier testing phases. Therefore, by including all risks in one sample, the correlation might be skewed due to the lack of defects found for these testing categories during the system-testing phase.

As a compromise, several risks were examined over a limited number of projects in order to provide a more varied insight into this correlation.

5.2.3 Measurement Accuracy

Incorrect or incomplete feature list

The selection of the feature list upon which the strength of a risk management technique is based, was a critical part of this project. If important features were missed, the calculation of the strength variable could be misleading and therefore invalidate the final correlation calculations. The selection of these features was achieved through discussion with the various project team members, but there was no means to validate that all features had been identified. To add to this, one feature "the effectiveness of the technique used" was disregarded because collectively it was deemed too difficult to judge comparatively. This decision may have skewed the results and in doing so, may have affected the final correlation calculation.

Reliability of Risk Exposure values

The risk exposure values were retrieved form the individual project logs. For each project, the actual exposure values would have been determined by a different project leader, leading to inconsistencies in this ranking between projects. This influence over the risk exposure value would also give disproportionate weighting to the experience of the project leader, another identified feature, and could again skew the results of the AHP calculations.

Subjectivity in mapping tests to test categories

One of the biggest challenges of the research project was the mapping of tests to test categories. The original tests were not documented in such a way as to define the types of defects against which they were testing, so an arduous, retrospective process was carried out with the testing team to understand the nature of each test. This process introduced a certain amount of subjectivity into the research.

A consequence of this subjectivity is the possibility of inadequate test cases being recognised for a certain test category. An example of this is the New Bank project, where a relatively small number of UI non-conformance tests, in comparison with other types of tests, could imply inadequate UI test cases being identified. Defects of this type can sometimes be discovered during ad hoc testing or in tests of other categories. With such a low number of tests, only a few UI defects would need to be found in order to give a disproportionately high test-case failure rate.

5.3 Future Research

Any future research should increase the size of the sample in order to provide greater statistical significance to the results. By increasing the number of projects to be studied and decreasing the number of risks, larger samples can be created for correlation calculations without an inordinate increase in effort.

Improvements should be made to the selection of features upon which the strength of a risk management technique is based. A Delphi study could provide a more in-depth version of this list by engaging a larger number of experts to participate. As this is such a crucial part of the research, I would suggest this type of study be included in a future design. By doing so, a more complete list of features affecting the strength of a risk management technique could be discovered, resulting in greater accuracy in the calculation of the strength variables.

At the time of designing tests cases, additional thought should be given to how the details of these test cases can feed into this type of research. Test cases should be defined with a clear vision of which type of defects they are expected to find and what category of test they are attempting. In this way the mapping processes of the research method could be simplified and reduce the scope for error.

Any research should also aim to capture defects over the whole project lifecycle in order to eliminate any bias regarding the types of defects likely to be caught during any single project phase.

6 References

1. ACP (2006), Effective Clinical Practice - Primer on Correlation Coefficients [online], http://www.acponline.org/journals/ecp/mayjun01/primer_correlcoeff.htm [Accessed 21 October 2006]

2. Addison T, Vallabh S (2002), ‘Controlling software project risks: an empirical study of methods used by experienced project managers’, Proceedings of South African institute of computer scientists and information technologists on enablement through technology, Vol. 30, pp. 128 – 140

3. Anderson ES, Grude V, Haug T (2004), Goal Directed Project Management, London, Kogan Page

4. Basili VR, Rombach HD (1987), 'Tailoring the software process to project goals and environments', Proceedings of the 9th international conference on Software Engineering, pp. 345 – 357

5. Bee R, Bee F (1990), Management Information Systems and Statistics, London, IPD

6. BMJ.com (2006) Statistics at Square One [online], http://bmj.bmjjournals.com/collections/statsbk/11.shtml [Accessed 10 October 2006]

7. Brooks F (1995), The Mythical Man-Month, USA, Addison Wesley Longman Inc

8. Chillarege R, Bhandari I S, Chaar J K, Halliday M J, Moebus D S, Ray B K, Man-Yuen Wong (1992), ‘Orthogonal Defect Classification - A Concept for In-Process Measurements’, IEEE Transactions on Software Engineering, Vol. 18, No. 11 (November), pp. 943 – 956

9. Fortune J, Peters J (2005), Information Systems - Achieving Success by Avoiding Failures, West Sussex, John Wiley and Sons Ltd

10. Freimut B (2001), 'Developing and Using Defect Classification Schemes', Fraunhofer IESE, Report. No. 072.01/E Version 1.0

11. Grady RB(1992), Practical Software Metrics for Project Management and Process Improvement, Prentice-Hall Inc

12. Hallowell DL (n.d.) Analytical Hierarchy Process (AHP) - Getting Oriented [online], http://software.isixsigma.com/library/content/c050105a.asp [Accessed 30 May 2006]

13. Huber JT (1999), 'A comparison of IBM's Orthogonal Defect Classification to Hewlett Packard's Defect Origins, Types and Modes', Hewlett Packard Company

14. Hvannberg ET, Law LC (2003), 'Classification of Usability Problems (CUP) Scheme', 'Human-Computer Interactions -- INTERACT'03', IOS Press, 655 – 662

15. IBM (2002) Details of ODC v 5.11 [online], http://www.research.ibm.com/softeng/ODC/DETODC.HTM [Accessed 29 April 2006]

16. IEEE (1996), 'IEEE Guide to Classification for Software Anomalies', New York, 'Institute of Electrical and Electronic Engineers'

17. IEEE (1993), 'IEEE Standard Classification for Software Anomalies', New York, 'Institute of Electrical and Electronic Engineers'

18. Johnson J, (1995), quoted in Keil M, Cule E, Lyytinen K, Scmidt R (1998), 'A framework for identifying software project risks', Communications of the ACM, Vol. 41, No. 11 (November), p. 76

19. Keil M, Cule E, Lyytinen K, Scmidt R (1998), 'A framework for identifying software project risks', Communications of the ACM, Vol. 41, No. 11 (November)

20. Kelly D, Shepard T (2001), 'A case study in the use of defect classification in inspections', 'Proceedings of the 2001 conference of the Centre for Advanced Studies on Collaborative research'

21. Kitchenham B, Pfleeger SL, Fenton N (1995), 'Towards a framework for software measurement validation', IEEE Transactions on Software Engineering, Vol. 21, No. 12 (December) pp. 929 – 944

22. Kitchenham BA, Pfleeger SL, Pickard LM, Jones PW, Hoaglin DC, El Emam K, Rosenberg J (2002), Preliminary Guidelines for Empirical Research in Software Engineering ', IEEE Transactions on Software Engineering, Vol. 28, No. 8 (August) pp. 721 – 734

23. Leszak M, Perry DE, Stoll D (2000), 'A Case Study in Root Cause Defect Analysis', Software Engineering - Proceedings of the 2000 International Conference, pp. 428 – 437

24. Li PL, Shaw M, Herbsleb J, Ray B, Santhanam P (2004), 'Empirical evaluation of defect projection models for widely-deployed production software systems', Proceedings of the 12th ACM SIGSOFT twelfth international symposium on Foundations of software engineering, pp.263-272

25. McNaughton DB, (1997) The Entity-Property-Relationship Approach to Statistics: An Introduction for Students [online] http://www.matstat.com/teach/eprstdnt.pdf [Accessed 29 May 2006]

26. Oxford University Press (2006) Oxford English Dictionary [online] http://dictionary.oed.com [Accessed 29 April 2006]

27. Pressman R. (2000), Software Engineering – A Practitioner's Approach, Glasgow, McGraw-Hill

28. SEI (2002) Capability Maturity Model® Integration Version 1.1 [online], http://www.sei.cmu.edu/cmmi/models/ss-cont-v1.1.doc#_Toc1970149 [Accessed 29 April 2006]

29. SEI (2006) CMMI [online], http://www.sei.cmu.edu/cmmi/general/general.html [Accessed 5 March 2006]

30. Staw B.M., Ross J. (1987), 'Knowing when to pull the plug', 'Havard Business Review', Issue March 1987, pp. 68-74

31. The Open University (2002), M865 Project Management: Block1 - Project Initiation, Milton Keynes, Open University

32. The Open University (2002), M865 Project Management: Block7 – Standard Methods, Milton Keynes, Open University

33. The Open University (2002), M880 Software Engineering: Software Quality Management, Milton Keynes, Open University

34. Turner JR, (1999), The handbook of project-based management, Glasgow, McGraw-Hill

35. Wateridge (1997), quoted in Turner JR, (1999), The handbook of project-based management, Glasgow, McGraw-Hill

A.1 Appendix A - Questionnaires

A.1.1 Risk Feature Questionnaire

Introduction

Following is a form requesting information regarding your personal feelings on the various features that make a risk management technique either strong or weak. The answers you give will be fed into an Analytical Hierarchy Process (AHP) calculation. The results of this calculation will be used to determine a ranking for each feature with regard to their importance when evaluating the strength of a risk management technique.

Each question will ask you to compare two features of a risk management technique with regard to their importance when evaluating the strength of a risk management technique. The questions can be regarded as requiring two thought processes: -

1. Which of the features is more important when evaluating the strength of a risk management technique?
2. How much more important is that feature in comparison with the other?

Possible Inconsistencies

You should be aware of the possibility of inconsistency in your answers when comparing features in this way. As an example, take three features A, B and C to be compared against each other. If you rank feature A as more important than feature B, and feature C as more important than feature A, then it follows that feature C will also be more important than feature B. Therefore relative importance of the three features is as follows: -

C > A > B

If you were to again rank feature A as more important than feature B, and feature C as more important than feature A, but then rank feature B more important than feature C, you will have inconsistent ranking. That is to say, feature C cannot be less important than feature B and more important than feature A, if feature A is more important than feature B.

The Scale

You will be required to use the following comparative scale when comparing features in this questionnaire: -

feature A	x								y		z							*feature B*
	9		**7**		**5**		**3**		**1**		**3**		**5**		**7**		**9**	

Relative Importance	Definition
1	The two features (A and B) are of equal importance
3	One feature is slightly more important than the other
5	One feature is moderately more important than the other
7	One feature is strongly more important than the other
9	One feature is absolutely more important than the other
2,4,6,8	Intermediate values

Hence, the three points (x, y and z) on the scale would represent the following comparative judgements: -

- x : Feature A is **absolutely more important** than feature B
- y : Features A and B are of **equal importance**
- z : Feature B is **slightly more important** than feature A

The Questionnaire

Now, please provide your comparative judgements below: -

Please compare the following features with regard to their importance when evaluating the strength of a risk management technique: -

1. Perceived importance of the project by senior management vs. Dependencies with other projects

Perceived importance	9	7	5	3	1	3	5	7	9	*Dependencies*

2. Perceived importance of the project by senior management vs. Experience of project leader

Perceived importance	9	7	5	3	1	3	5	7	9	*Project Leader Experience*

3. Perceived importance of the project by senior management vs. Risk Exposure of the risk

Perceived importance	9	7	5	3	1	3	5	7	9	*Risk Exposure*

4. Dependencies with other projects vs. Experience of project leader

Dependencies	9	7	5	3	1	3	5	7	9	*Project Leader Experience*

5. Dependencies with other projects vs. Risk Exposure of the risk

Dependencies	9	7	5	3	1	3	5	7	9	*Risk Exposure*

6. Experience of project leader vs. Risk Exposure of the risk

Project Leader Experience																		*Risk Exposure*
	9		**7**		**5**		**3**		**1**		**3**		**5**		**7**		**9**	

And finally …

Thank you for your time in completing this questionnaire. I am very grateful for your efforts. As mentioned your answers will be used in Analytical Hierarchy Process calculations designed to give quantitative measurements of strength for a risk management technique. In turn, the results from these AHP calculations will provide some data for my research into the relationship between strengths of risk management techniques and the resulting number and types of defects raised.

Thank you again for your effort.

A.1.2 Risk Comparison Questionnaire

Introduction

Following is a form requesting information regarding your personal feelings on the strengths of various risk management techniques with regard to selected risk management features. These risk management techniques have all been used to mitigate risks identified in projects selected from the release 9 of the XXXX Internet banking system. The answers you give will be fed into an Analytical Hierarchy Process (AHP) calculation, which in turn will be used to determine a ranking for the strength of each of these techniques.

Each question will ask you to compare the strength of a feature with regard to a particular project. An example would be to compare the experience of senior management's perceived importance of project A and project B.
The questions can be regarded as requiring two thought processes: -

3. In which project is the feature stronger?
4. How much stronger is that feature in one project in comparison with the other project?

Possible Inconsistencies

You should be aware of the possibility of inconsistency in your answers when comparing features in this way. As an example, take three projects A, B and C. The strength of feature X is to be compared in each of the projects. If you rank feature X as being stronger in project A than in project B, and stronger in project C than in project A, then it follows that feature X will also be stronger in project C than in project B. Therefore relative importance of the three features is as follows: -
CX > AX > BX
If you were to again rank feature X as being stronger in project A than in project B, and stronger in project C than in project A, but then rank it stronger in project B than in project C, you will have inconsistent ranking.

Equally, if you were to rank feature X as being equally strong in project A and project B, and ranked feature X being slightly stronger in project A than in project C, then it would be inconsistent to rank the feature being absolutely stronger in project B than in project C. This is because feature X is equally as strong in projects A and B; hence the difference in strength of this feature between projects A and C should be the same as the difference in strength of this feature between projects B and C.

The Scale

You will be required to use the following comparative scale when comparing features in this questionnaire: -

Compare the strength of feature F1 in projects A and B

project A	x								y		z							*project B*
	9		**7**		**5**		**3**		**1**		**3**		**5**		**7**		**9**	

Relative Importance	Definition
1	Feature F1 is equally as strong in projects A and B
3	Feature F1 is slightly stronger in one of the projects
5	Feature F1 is moderately stronger in one of the projects
7	Feature F1 is much stronger in one of the projects
9	Feature F1 is absolutely stronger in one of the projects
2,4,6,8	Intermediate values

Hence, the three points (x, y and z) on the scale would represent the following comparative judgements: -

- x : Feature F1 is **absolutely stronger** in project A than in project B
- y : Feature F1 are of **equally strong** in both projects
- z : Feature F1 is **slightly stronger** in project B than in project A

The Questionnaire

Now, please provide your comparative judgements in the following sections: -

Section 1 - Perceived importance of the project by senior management

Please compare the "*perceived importance of the project by senior management*" between the following pairs of projects: -

	9	7	5	3	1	3	5	7	9	
Customer Alerting										*Balance and Transaction Enhancements*
Customer Alerting										*Private Bank*
Customer Alerting										*New Bank*
Customer Alerting										*Foreign Exchange*
Balance and Transaction Enhancements										*Private Bank*
Balance and Transaction Enhancements										*New Bank*
Balance and Transaction Enhancements										*Foreign Exchange*
Private Bank										*New Bank*
Private Bank										*Foreign Exchange*

New Bank	**9**	**7**	**5**	**3**	**1**	**3**	**5**	**7**	**9**	*Foreign Exchange*

Section 2 - Level of dependencies with other projects

Please compare "the level of dependencies with other projects" between the following pairs of projects: -

Customer Alerting	9	7	5	3	1	3	5	7	9	*Balance and Transaction Enhancements*
Customer Alerting	9	7	5	3	1	3	5	7	9	*Private Bank*
Customer Alerting	9	7	5	3	1	3	5	7	9	*New Bank*
Customer Alerting	9	7	5	3	1	3	5	7	9	*Foreign Exchange*
Balance and Transaction Enhancements	9	7	5	3	1	3	5	7	9	*Private Bank*
Balance and Transaction Enhancements	9	7	5	3	1	3	5	7	9	*New Bank*
Balance and Transaction Enhancements	9	7	5	3	1	3	5	7	9	*Foreign Exchange*
Private Bank	9	7	5	3	1	3	5	7	9	*New Bank*
Private Bank	9	7	5	3	1	3	5	7	9	*Foreign Exchange*

New Bank																		*Foreign Exchange*
	9		**7**		**5**		**3**		**1**		**3**		**5**		**7**		**9**	

Section 3 - Experience of project leader

Please compare "the level of dependencies with other projects" between the following pairs of projects: -

Customer Alerting	9	7	5	3	1	3	5	7	9	*Balance and Transaction Enhancements*
Customer Alerting	9	7	5	3	1	3	5	7	9	*Private Bank*
Customer Alerting	9	7	5	3	1	3	5	7	9	*New Bank*
Customer Alerting	9	7	5	3	1	3	5	7	9	*Foreign Exchange*
Balance and Transaction Enhancements	9	7	5	3	1	3	5	7	9	*Private Bank*
Balance and Transaction Enhancements	9	7	5	3	1	3	5	7	9	*New Bank*
Balance and Transaction Enhancements	9	7	5	3	1	3	5	7	9	*Foreign Exchange*
Private Bank	9	7	5	3	1	3	5	7	9	*New Bank*
Private Bank	9	7	5	3	1	3	5	7	9	*Foreign Exchange*

New Bank	**9**	**7**	**5**	**3**	**1**	**3**	**5**	**7**	**9**	*Foreign Exchange*

And finally …

Thank you for your time in completing this questionnaire. I am very grateful for your efforts. As mentioned your answers will be used in Analytical Hierarchy Process calculations designed to give quantitative measurements of strength for a risk management technique. In turn, the results from these AHP calculations will provide some data for research into the relationship between strengths of risk management techniques and the resulting number and types of defects raised.
Thank you again for your effort.

Index