WeDRisk: An Approach to Managing Web and Distributed Software Development Risks

Ayad Ali Keshlaf

School of Computing Science

Newcastle University

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

February 2013

Declaration

I certify that no part of the material offered has been previously submitted by me for a degree or other qualification in this or any other University.

Published Work

Part of the work presented in this thesis has or will have appeared as follows:

- Parts of Chapters 1, 2 and 3 appeared in papers 1 and 2.
- Parts of Chapter 4 and 6 appeared in papers 2 and 3.
- Parts of Chapters 4 and 8 appeared in paper 4.

1. Keshlaf, A. and Riddle, S. Risk Management for Web and Distributed Software Development Projects, The Fifth International Conference on Internet Monitoring and Protection (ICIMP10), IEEE Computer Society, May 9 - 15, Spain, pp. 22-28, 2010. (*Won a **Best Paper** award at the conference*).

2. Keshlaf, A. and Riddle, S. Web and Distributed Software Development Risks Management: WeDRisk Approach, International Journal on Advances in Software, vol. 3 no. 3 - 4, pp 447-460, 2010. (*Won a **Best Paper** award in the Dependability Group in the School of Computing Science, Newcastle University*).

3. Keshlaf, A. and Riddle, S. An Empirical Study of Web and Distributed Software Risks from Three Perspectives: Project, Process and Product. The Sixth International Conference on Internet Monitoring and Protection (ICIMP11), March 20-25, St. Maarten, 2011.

4. Keshlaf, A. and Riddle, S. The WeDRisk Approach for Management of Web and Distributed Risks: An Evaluation Experiment, Poster Session at The Newcastle Connection, August 2012.
http://www.cs.ncl.ac.uk/publications/trs/papers/1348.pdf

The research was carried out by Keshlaf with guidance and under supervision of the co-author.

Dedication

I would like to dedicate my thesis to my beloved mother, for continuous support and encouragement and waiting. To my father soul for being my first teacher, who worked hardly and suffered a lot to see me completing my study.

I also dedicate this thesis to my wife, who did more than help around the house as I sat at the computer. Without her support, love and patience it will be difficult for me to continue.

To my children (Saleh, Rafik, Ashwak, Abdalbari and Ahmed) for their patience during my study.

To my brothers for their endless support and encouragement.

Acknowledgements

First of all, I would like to thank my thesis advisor Dr. Steve Riddle. For his continuous support, advises, comments, encouragement and suggestions throughout the work of this thesis.

As well as I thank my thesis committee Dr. Lindsay Marshall Dr. Neil Speirs for their comments during the thesis committee meetings.

I also would like to thank my classmates at School of Computing Science, particularly Nur Haryani Zakaria, Su-Yang Yu, Ilya Lopatkin and Maryam Mehrnezhad for the support and comments.

I would like to thank experts, developers, researchers and students whom participated in the evaluation cycles of my approach. In particular, I thank Dr. Jacek Cala for running the evaluation case study and his adoption of my approach and building a prototype based on my approach.

Finally, my thank goes also to staff of School of Computing Science and Centre of Software Reliability and dependability group members at Newcastle University for their support during my research. I also thank my workplace Industrial Research Centre and Libyan government-Higher Education for the scholarship.

Abstract

Web and distributed software developments are risky and face specific challenges like time zone and cultural differences. These challenges have resulted in new risks and risk management needs. In this thesis, a systematic review of existing software risk management approaches was conducted to investigate their ability to satisfy the risk management needs of web and distributed developments. The review identifies a number of weaknesses in existing approaches. Examples are the lack of consideration for web and distributed factors and lack of preparation for atypical risks. A new approach called WeDRisk is introduced to manage the risks from project, process and product perspectives. The WeDRisk approach addresses the weaknesses of existing approaches to risk management, which are less able to deal with the specific challenges of web and distributed development. A key part of the approach is flexibility to deal with the rapid evolution which is typical of such developments. This flexibility is achieved by customizing the risk management and providing a method for coping with atypical risks. WeDRisk also provides an improved risk estimation equation to consider web and distributed factors. The novel aspects of the WeDRisk approach were subjected to a series of evaluation cycles, including peer review, two controlled experiments, expert evaluation and a case study. In addition to a number of improvement suggestions, the evaluation results illustrate how WeDRisk is useful, understandable, flexible, easy to use, and able to satisfy many web and distributed development risk management needs.

Contents

List of Tables

List of Figures

Glossary

3P	Three Perspectives (Project,process and Product)
Atypical Risk	Unknown,unpredictable and completely new risk
Deep RM	Implementation of maximum number of risk management steps
Evolution Approval Board	A board consist of a group of experts, managers and developers whom aware/involved in the implementation WeDRisk approach and responsible/able to make evolving decisions to modify or improve the approach or the risk management processes
GSD	Global Software Development
I/O	Input and Output (for the modules)
Module	Refers to a part of RM phase in the WeDRisk approach. It consists of a number of steps, procedures or components to perform a specific risk management task(s)
Plain RM	Implementation of minimum steps of risk management
Project Card	A record consists of all important risk management data for a specific project
RE	Risk Exposure
Risk Card	A record consists of all important data that are related and describe a specific risk
Risk Factor	A factor that has an effect on risk impact
RiskMag.	Risk Magnitude
RiskProb.	Risk Probability

RM	Risk Management
RM Customization Matrix	A matrix helps to specify what risk management type is suitable
RM Phase	This refers to WeDRisk approach risk management stage which consists of a number of modules
SRM	Software Risk Mangement
Stakeholder Card	A record consists of all important data (bibliographic and privileges) that are related to risk management involved stakeholder
T	Task
TREV	Total Risk Estimation Value
WD development	Web and Distributed development: This refers to distributed types of software development including developing over the web
WDF	WD risk factor
WDF Estimation Matrix	A matrix to estimate web and distributed risk factors
WD-R	Web and Distributed risk management Requirmrnt
WeDRisk	WeDRisk approach
WeD-RM	WeD-RM prototype

Chapter 1

Introduction

1.1 Introduction

Levels of risk and complexity in the software industry have been increasing continuously with its growth. This includes Web and Distributed (WD) development. WD development has sharply accelerated over the last few years, presenting specific challenges and risk areas to the software industry which need to be considered and managed [16]. This growth in WD development has occurred due to the incremental demand on software applications in all of today's activities and technologies, and the ubiquity of the Internet, which has increased its deployment and development. The new challenges and risks include: inadequate informal communications; lack of trust; cultural differences (e.g. languages, corporate cultures); time-zone difference, ineffective communication synchronicity; development process differences; knowledge management challenges; and security issues, which are related to electronic transmissions confidentiality and privacy [16; 17; 18; 19]. Nevertheless, software Risk Management (RM) practice in general and in WD development in particular is still imperfect, since there are still many development projects facing budget and schedule overruns [20; 21]. This research aims to study how to manage the risk of WD development effectively. This chapter presents the research motivation, aims and objectives, methodology, contributions, research questions and thesis structure.

Definitions

Software development is a complicated development type, since it depends on evolving technologies and development methodologies and tools which could involve high risks. Software development involves three main perspectives (project, process and product, hereafter called "3P"). In all these perspectives, the people, including managers, developers, stakeholders, customers and users, are the key actors [22; 23; 24; 25]. The following definitions are used for the purpose of this study:

Definition 1: *Project perspective concerns aspects such as, budgets, plans, goals, responsibilities and schedules*

Definition 2: *Process perspective concerns the methods, tasks and activities involved in producing the software etc.*

Definition 3: *Product perspective concerns the final product functionality, maintenance, market competence, security, customers and users.*

Although in the collocated software development, there might be some overlap between these 3P perspectives, this overlap could be slightly higher in WD development. In fact, different challenges and risks affect the 3P perspectives and thus there is a need to manage the risks from all of these perspectives.

Risky situation criticality and available RM resources could have a significant impact on RM implementation. Ordinarily, the developers/managers perform specific RM steps, which could be considered as "Plain" or "Deep" RM types [26]. The "Plain" or "Deep" RM types are introduced and used in this thesis and are defined as follows:

Definition 4: *Plain RM is a simple type of RM where only the minimum and essential steps of RM (e.g. identification and simple estimation of risks) are performed.*

Definition 5: *Deep RM is the ordinary type of RM where the maximum number of steps of RM (e.g. additional estimation options, RM performance evaluation, documentation) are performed.*

Another term which is introduced and used in this thesis is "Atypical Risk" which is different from ordinary risk. Atypical risk is defined as follows:

Definition 6: *Atypical Risk: Unknown, unpredictable and completely new risk that occurs for the first time and cannot be predicted. It might stop the development progress totally or partially.*

1.2 Research Motivation

WD development grows fast due to the advantages of development cost reduction, time to market and ICT infrastructure improvements, including communications and Internet availability [7; 27; 28]. Extra new challenges and risks are faced by WD development compared with collocated software development, which existing software risk management approaches may not be able to address.

The limitations of the existing RM approaches in managing WD development risks are due to lack of flexibility to accommodate the evolving nature of WD development. Therefore, there is a need to simplify and increase the practice of RM in WD development by providing a suitable methodology and approach for this purpose.

WD development faces challenges and risks (e.g. differing time zones, inadequate informal communications and cultural differences), which have lower impact on col-located software development and need to be addressed effectively to manage WD development risks. The hypothesis of this research is that WD development has specific challenges and risks which require a new approach to manage them. This hypothesis has been broken down into research questions and introduced in a section below.

1.3 Research Aim and Objectives

The aim of this research is to propose an approach to manage the risks of WD development effectively from the 3P perspectives. The objectives of this research to achieve its aim are to:

- Identify challenges, potential risks and their importance to WD development from the 3P perspectives to explore the RM needs for WD development.
- Review the existing RM approaches to see how they are able to accommodate the challenges and satisfy the RM needs of WD development and to identify the weaknesses of the existing approaches in this regard.
- Propose an approach to tackle the weaknesses in the existing approaches and to consider the WD development factors; and
- Evaluate the approach by conducting some empirical studies and applying methods such as peer reviews, experiments, a case study and expert evaluation.

1.4 Research Questions

Three research questions have been specified to reflect the research hypothesis, aim and objectives and were answered during the research stages. The three questions of this research are as follows:

R-Q1: *What are the challenges of WD development for risk management?*

R-Q2: *Can existing software risk management address the identified challenges?*

R-Q3: *How can risk management approaches be adapted to tackle the weaknesses of the existing approaches?*

1.5 Research Methodology

In order to achieve the research aim, a research methodology was prepared which covers all research aspects, from collecting the preliminary research data to the evaluation of the work. The research methodology is summarized as follows:

I) Literature review : This stage of the study aims to identify the challenges and risks of WD development. In addition, it aims to identify the RM challenges for WD development.
II) Exploring the State of the Art: This is to review the existing RM approaches (method, models, techniques, frameworks and tools) in order to investigate their ability to accommodate the RM challenges for WD development, which come from the first research stage. This includes identifying the strengths and weaknesses of the existing approaches in managing the WD development risks. As a result of this stage, the requirements of risk management for the WD development are specified.
III) Proposing a new Approach: As a result of this research, an approach will be proposed, which is expected to address the weaknesses of the existing approaches in managing WD development and which includes some aspects that have not yet been covered.
IV) Evaluation: In this stage, the proposed approach will be evaluated in different evaluation cycles. A number of methods will be used to evaluate the approach based on the availability of the resources and suitability of the methods.

1.6 Research Contributions

The major contribution of this research is the WeDRisk approach, which is intended to manage WD development risks. This approach aims to address the weaknesses of the existing RM approaches in managing WD development risks and cover new RM aspects. It consists of establishment, implementation and evaluation and evolution phases. It is also supported with a communication channel. Each phase in the WeDRisk contains one or more modules. This modular structure of WeDRisk is to ensure maximum flexibility and the ability of the approach to evolve. The novel and contribution aspects of the WeDRisk approach are summarized below.

1.6.1 *The WeDRisk Approach*

Although the WeDRisk approach which is introduced in this thesis is intended to manage WD development risks in particular, it can also be used to manage software risks in general. The approach implies a number of novel aspects and contributions to software risk management. These contributions are described in detail, discussed and evaluated in the following chapters and are summarized thus:

- Introducing a new RM style which depends on customization of the RM by offering two RM options: Plain and Deep. This is in order to encourage the developers/managers to practise RM even with limited resources or time.
- Ability to address atypical risk types by means of a module to deal with atypical risks.
- Consideration of the risks from 3P perspectives; for this purpose, the WeDRisk approach is provided with 3P perspectives clustering criteria.
- Consideration of WD factors, since the WeDRisk is supported by an upgradable matrix to estimate the WD factors.
- Introducing an improved risk estimation equation called Total Risk Estimation Value (TREV), which incorporates the WD factors.
- WeDRisk initiation with a list of potential WD risks to help developers and managers to establish and start the RM easily.
- Evolvability and flexibility: WeDRisk has the ability to evolve and accommodate changes and new needs. This is ensured via the modular structure of the WeDRisk, minimum dependencies between the modules and its Evolution Module.
- Risk evaluation module in the WeDRisk which deals with two risk estimation equations (RE and TREV).

As an example of WeDRisk applicability, one of the developers involved in the WeDRisk evaluation case study (see Chapter 9), has adopted the WeDRisk by utilizing a prototype tool that has been built based on the WeDRisk, and embedded this into his distributed project management system with some collaboration with the author in testing, providing related material and initializing the risks.

1.6.2 *Produced Papers*

Four contributions including three papers (i.e. two conferences papers and one journal paper) and one poster have been produced and published from this research. The first conference paper won a Best Paper award at the The Fifth International Conference on Internet Monitoring and Protection (ICIMP10) conference and the journal paper won a Best Paper award in the Dependability Group in the School of Computing Science, Newcastle University. Moreover, the final draft of the second journal paper is ready. These papers are listed below [16; 23; 26]:

1. Keshlaf, A. and Riddle, S. Risk Management for Web and Distributed Software Development Projects, The Fifth International Conference on Internet Monitoring and Protection (ICIMP10), IEEE Computer Society, May 9 - 15, Spain, pp. 22-28, 2010.

2. Keshlaf, A. and Riddle, S. Web and Distributed Software Development Risks Management: WeDRisk Approach, International Journal on Advances in Soft- ware, vol. 3 no. 3 - 4, pp 447-460, 2010.

3. Keshlaf, A. and Riddle, S. An Empirical Study of Web and Distributed Software Risks from Three Perspectives: Project, Process and Product. The Sixth International Conference on Internet Monitoring and Protection (ICIMP11), March 20-25, St. Maarten, 2011.

4. Keshlaf, A. and Riddle, S. The WeDRisk Approach for Management of Web and Distributed Risks: An Evaluation Experiment, Poster Session at The Newcastle Connection, August 2012.

1.7 Structure of Thesis

The thesis consists of 10 chapters and each chapter addresses some objectives of the research. The following is an outline of the chapter contents.

Chapter 2 (Background): This chapter reviews the WD development and identifies its challenges and potential risks, giving an overview of the software risk management concept and its related definitions. It specifies the challenges and needs of risk management in WD development. The chapter addresses the first research question R-Q1.

Chapter 3 (Related Work): This chapter reviews the existing state of the art software risk management approaches and identifies their weaknesses and strengths in managing WD development risks. The criteria for the review come from Chapter

2 based on the specified challenges and needs. This chapter addresses the second research question R-Q2. A case study is presented and used as an example in this chapter to illustrate the limitations of the existing approaches.

Chapter 4 (WeDRisk Approach): This chapter introduces the WeDRisk approach, which is designed to manage WD development risks and tackle the weaknesses of the existing approaches in this regard. The chapter describes the WeDRisk approach structure and modules, how it works, and presents the main contributions and novel aspects of the approach. The case study which is presented in Chapter 3 is used in this chapter to demonstrate how the WeDRisk approach deals with WD development risk situations. This chapter addresses part of the third research question R-Q3.

Chapter 5 (Evaluation): This chapter is an introduction to the methods, which are used to evaluate the WeDRisk approach. In addition to evaluation questions, the chapter provides a brief description of each method. The methods are reported and discussed in detail in the following chapters (i.e. Experiment One, Expert Evaluation, Experiment Two and a Case Study). This chapter also addresses part of the third research question R-Q3.

Chapter 6 (Experiment One): This chapter is an extension to the evaluation chapter and it reports the hypothesis design, results, analysis and findings of the first experiment, which is used to evaluate some novel aspects of the WeDRisk approach. This chapter addresses part of the third research question R-Q3.

Chapter 7 (Expert Evaluation): This chapter is an extension of the evaluation chapter and reports the expert evaluation of the WeDRisk approach. This evaluation was performed by a number of researchers and scientists in software development from around the world from both academia and industry, who participated in the ICIMP 2011 conference. This chapter addresses part of the third research question R-Q3.

Chapter 8 (Experiment Two): This chapter is an extension of the evaluation chapter and reports in detail the second evaluation experiment for the WeDRisk approach modules (i.e. estimation, customization and atypical modules). This chapter address part of the third research question R-Q3.

Chapter 9 (Case Study): This chapter reports a case study that was used to evaluate WeDRisk modules and the approach overall. Three developers in three different distributed software projects participated in this case study. The chapter describes a prototype that was built by one of the developers based on the WeDRisk approach. This chapter also reports part of WeDRisk evaluation work, and addresses part of the third research question R-Q3.

Chapter 10 (Discussion and Conclusion): This chapter discusses the previous chapters and how the research questions are answered. It also concludes the thesis and presents the contributions, limitations and implications for future work.

Finally, this work contains four main appendices which are:

Appendix A : Contains experiment one materials.

Appendix B : Contains experts evaluation materials.

Appendix C : Contains experiment two materials.

Appendix D : Contains WeD-RM-Prototype Screenshots.

Chapter 2

Background

2.1 Introduction

Before proceeding to the next stages of this research, it is important to give a background on software risk management and understand the nature of the web and distributed software developments and challenges associated. This chapter gives a background on software risk management and its related definitions, importance and principles and describes the web and distributed software developments and their challenges before combining them under one umbrella and sheds light on their potential risks. At the end of the chapter, needs and challenges of managing web and distributed development risks are highlighted.

2.2 Software Risks

The Software Engineering Institute (SEI) defines risk as the possibility of suffering loss [29] and it defines loss in a development project as the impact on the project, which could be in the form of diminished quality of the end product, increased costs, delayed completion, loss of market share, or failure [29]. Sherer [30] pointed out that the risk in software development projects is probable loss that could have an effect when software is developed, used or maintained. For the purpose of this work, the definition of software risk is that introduced by SEI:

Definition 7: *Risk. The possibility of suffering loss* [29]

For each risk, there are two aspects: risk probability and risk magnitude. These aspects are used to estimate the impact or Risk Exposure (RE) of the risk [4], as follows:

$$RE = RiskProb. * RiskMag \tag{2.1}$$

Where,
RE is the Risk Exposure
RiskProb. is the probability of risk occurring
RiskMag. is the magnitude of the losses if the risk occurs

Risk probability estimation is not a straightforward task and cannot be 100% accurate. Some probability estimation techniques use qualitative data and then convert them into equivalent quantitative data using some equations, risk probability tables, checklists or relative scales [4; 11].

The top ten software risk items (listed below), which are introduced by Boehm, are examples of risks for software development projects [4].

- Personnel shortfall
- Unrealistic schedules and budget
- Developing wrong software functions
- Developing wrong user interface
- Gold Plating
- Continuing stream of requirements change
- Shortfalls in externally furnished components
- Shortfalls in externally performed tasks
- Real-time performance shortfalls
- Straining computer-science capabilities

A further identified list of software risks includes the following items [31]:

- Bad traceability
- Insufficient verification and validation
- System complexity
- Customer dissatisfaction at project delivery
- Risk reducing techniques producing new risk
- Catastrophe/Disaster

Any list of software risk items needs to be updated from time to time, when there are new changes or challenges in software development technology and environment (e.g. aspects related to social and cultural issues, geographic dispersal or changes to new technologies). The significance and type of risks and their sources will also inevitably evolve over time. As an example, a review in [32] found that different authors have identified or proposed different software risks, which means that the number and types of software risks are not fixed. Therefore, new or improved methodologies, techniques and tools to identify, measure and control them are needed.

2.3 Software Risk Management

Software development projects are, by their nature, risky, complicated and multi-dimensional endeavour [16; 17; 18; 33]. Many software development projects miss their goals of delivering acceptable software products within agreed constraints of time, budget and quality, due to a combination of the risks themselves, and absent or poor Software Risk Management (SRM) [11; 34]. SRM is still evolving, and many software managers have only a limited understanding of its concepts [18]. Industrial risk management practice tends to lag behind recommended risk management best practice, although there are exceptions [18; 35; 36]. This lag is clearer with WD software development, where the level of SRM practice is still low. In this research, the definition of SRM given by Boehm in [4] is used.

Definition 8: *"Software Risk Management [is] a discipline whose objectives are to identify, address, and eliminate risk items before they become either threats to successful software operation or major sources of software rework"*[4].

There are many methodologies and techniques to manage software development risks. Figure 2.1 shows the basic steps of software risk management as introduced by Boehm in [4], which are still the foundation of many software risk management approaches.

As the figure shows software risk management consists of two primary steps (Risk Assessment and Risk Control). Risk Assessment consists of three sub steps: Risk Identification, Risk Analysis, and Risk Prioritization. In these sub steps, the risks are identified and their severities are analyzed based on probability and magnitude of each risk before they are prioritized based on their severities. The Risk Control step consists of three steps: Risk-Management Planning, Risk Resolution, and Risk Monitoring. In these sub steps, plans are prepared to address the identified risks, risk items are eliminated or resolved and project's progress and risks are monitored [4].

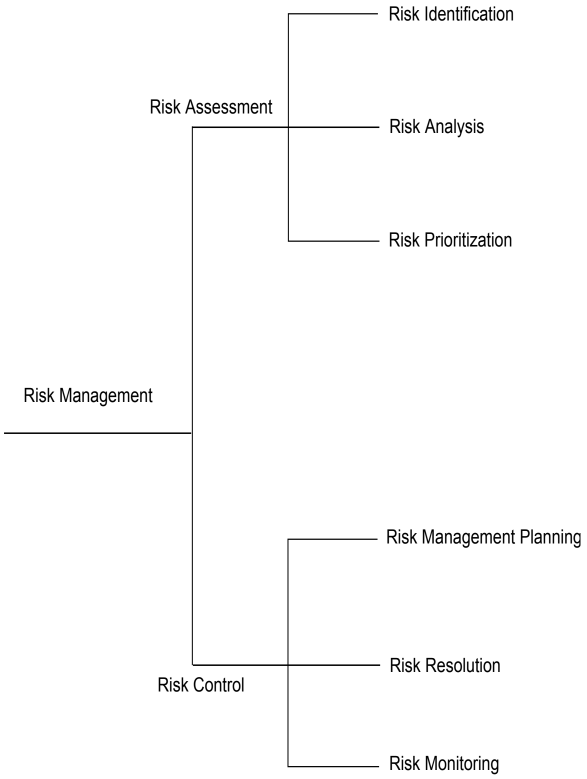


Figure 2.1: SRM basic steps [4]

2.4 Web and Distributed Software Development

2.4.1 *Web Development*

Web applications surround us from everywhere, i.e. in houses, at work and even on our mobiles. In the early 1990s, scientists started using the World Wide Web as a working tool at the European Centre for Nuclear Research "Centre Europeen de Recherche Nuclaire" (CERN) [37] before it became a huge repository of information for the people; later on in 1999, a new concept called the "web application" was introduced in Java [38; 39]. In 2003, people started using the World Wide Web (WWW) for business and services as it became a platform for developing and delivering web applications (known as "Web 2.0") [40]. Beal [41] differentiates the WWW (or simply web) from the Internet by considering the WWW as a way of sharing and accessing information over the medium of the Internet. The rapid advancement of web technologies and increased demand for web applications led to a trend for web developments which are no longer limited to Java.

A web application has become a common type of modern software application. It is an essential part of daily business (e.g. marketing, reservations, research, and planning and media platforms). Baresi and Morasca [42] describe web applications as complex distributed systems which depend on the Internet for communication and interaction. In contrast to desktop software applications, web applications may be instantly deployed worldwide without any installation or a need for upgrading manuals [43].

Kappel [44] defines a web application as “a software system based on technologies and standards of the World Wide Web Consortium (W3C) that provides Web specific resources such as contents and services through a user interface, the Web browser”.

This research focuses on “Web Development” and for this research purpose it is defined as:

Definition 9: *Web Development is development and deployment software applications on the web, which range from simple to complex applications.*

Web development growth has been very fast compared with traditional software developments and has become a significant part of today’s software industry and business [45; 46]. Web development is not free from risk and many web development projects fail due to the involved risks. In fact, web developments are complicated, distributed and need suitable implementation models and techniques [42] and have unique characteristics (e.g. high usability, rapid technology changes, short life cycle and continuous maintenance) [47; 48]. In the current situation, the lack of suitable web development models, process and methodologies lead many web developers to adapting the available conventional software approaches (models, processes and methodologies) and use them for web development [28].

2.4.2 *Web Development Challenges*

Even though there are similarities in many characteristics between web development and distributed software development, some authors claim that the speed of evolution and deployment, as well as the frequency of new releases is higher in web application development than other software development, whereas the maintenance cycle is shorter, the availability is usually 24/7, and the web developments face many challenges [49; 50], all of which can be summarized as follows [43; 44; 45; 48; 51; 52; 53; 54; 55; 56; 57; 58; 59]:

- The evolutionary nature and rapid changes of the web development and its associated technologies and environment.
- Demand for availability as the users/developers can work at any time.
- Security and reliability, which are vital issues and therefore need to be tested well.
- High frequency of releases, which implies rapid and continuous requirement changes.

- Involvement of developers and stakeholders from different backgrounds.
- Legal, social, ethical and localization issues, which are always important.
- Anonymity of users (numbers, backgrounds, when, where).
- Parallelism or subgroups development (due to short development cycles). This needs good communications between the groups to ensure development consistency and avoid duplications.

The above challenges can be considered as sources of risk. Therefore, the deployment environment and the significance of associated challenges and risks of web development need to be considered carefully. Features such as diversity and rapid changes present new challenges for the developer and manager, as well as to traditional project management approaches [46; 52; 60; 61]. More effective development and management methodologies, models and tools should be introduced to deal with these challenges [36; 62; 63], the importance of which is different from others in a number of ways:

- Their impact and significance are different. For example, exposure to security threats is higher in web development [64; 65; 66; 67].
- As web applications may be deployed instantly worldwide [43], their risks can affect a wider range of components and applications simultaneously in a very short period of time.
- Additional risk sources related to the distributed environment include communication, culture, diversity and geographical location [14; 44; 48; 54].

Ideally, assessment and management of web development risks should be performed during the whole life cycle of the projects [68], but unfortunately, many web developers use a reactive risk strategy and do not act until something goes wrong. Although this strategy could be cost-effective when the risk is minimal, this cannot be ensured all the time and makes software projects vulnerable to any type of risk at any time without effective assessment and control [69].

Mikkonen and Taivalsaari [70] argued that, although the web is rapidly becoming a platform for the software industry, there is a growing gap between web development and software engineering, which needs to be filled with a new set of software architectures, methodologies and systematic approaches covering security, modularity and legal aspects.

Jeary *et al.* [71] urged that the adoption and adaptation of conventional software methodologies to the web development is a severe problem. Thus, further research is required to improve the capabilities of the methodologies in order to deal with exceptional web development needs [56; 71].

2.4.3 *Distributed Software Development*

The trend of Distributed Software Development, sometimes called Global Software Development (GSD) was started in the late 1980's and early 1990's. Factors such as cost reduction and seeking for skilled people were behind this trend [72; 73]. Distributed Software Development can be defined as described by Jimenez and others [72; 74] as follows:

Definition 10: *Distributed Software Development is a type of development that allows team members to be located in various remote sites during the software lifecycle, thus making up a network of distant sub-teams.*

Distributed software projects are usually developed by teams working collaboratively via communication channels (e.g. networks, Internet, emails) across many locations, usually in different countries [5; 6]. Software developers have adopted distributed software development as a strategy to reduce costs and increase their projects' productivity [75]. Generally, the rapid growth of distributed software development is gained from technological progress, low development cost, time differences, quick formation of virtual corporations and teams of skilled people, flexibility, and breaking into markets across the developing world, especially in countries like India, Russia and Brazil [1; 72; 76; 77].

2.4.4 *Distributed Software Development Challenges*

Despite the benefits of adopting distributed software development, there are different challenges that could become a source of problems and risks to this type of development [78]. As shown in Table 2.1 Khoshroo and Rashidi in [1] summarized and categorized a number of challenges and concluded that there is a significant need for training, new management and compatible tools to deal with the distributed development challenges.

Damian and Moitra in [79] argued that distributed software development is a fast growing phenomenon, but it faces unique technical, organizational and cultural issues, complexities and challenges. Time zones, languages, and geographical location differences are potential sources of these challenges. In fact, there is still a necessity to develop methods, techniques and practices to accommodate the challenges and evolvement issues of distributed development [79]. The adoption of distributed development has an effect on different levels and issues such as [72]:

Table 2.1: Distributed Software Development Management, Cooperation and Technical Challenges [1]

<table>
<tr><th colspan="3">Management Challenges</th></tr>
<tr><td><b><i>Initiation and Planning:</i></b>
Strategic issues; distributing the work ; Process definition; Cost overheads related to travel, Communications; Risk management</td><td><b><i>Execution and Control:</i></b>
Controlling remote activities ; Risk management</td><td><b><i>Leadership:</i></b>
Leadership issues</td></tr>
<tr><th colspan="3">Cooperation Challenges</th></tr>
<tr><td colspan="2"><b><i>Communication and Coordination:</i></b>
Trust issues; language differences; informal communication; delays; isolation</td><td><b><i>Awareness:</i></b>
lack of shared context ; different organizational culture; expertise identification and selection</td></tr>
<tr><th colspan="3">Technical Challenges</th></tr>
<tr><td colspan="2"><b><i>Communication and Coordination:</i></b>
Requirement engineering; changing architecture; integration issues; knowledge management; change and configuration; management; incompatible processes</td><td><b><i>Tools and Environment:</i></b>
Standards; infrastructure</td></tr>
</table>

Strategic issues: e.g. an organization's resistance due to misalignment in the management between different management levels.

Cultural issues: e.g. different cultural backgrounds have a significant effect on distributed developments due to issues like sense of time and communication styles.

Inadequate communication: e.g. weak formal and informal communications due to lack of face-to-face meetings, fear of losing intellectual property rights and restrictions on information.

Knowledge management: e.g. ineffective collaboration due to poor information sharing or poor documentation.

Project and process management issues: e.g. lack of synchronization.

Technical issues: e.g. slow networks.

Smite and Borzovs [27] identified the main distinguishing factors of distributed software development including multisourcing, temporal diversity, socio-cultural diversity, linguistic diversity and political and legislative diversity. Generally, developing software across distributed sites presents many challenges which can be summarized as follows [80; 81]:

- Inadequate informal communications
- Lack of trust
- Culture differences (e.g. different language, different corporate culture and dif-

ferent developers' backgrounds)

- Time-zone difference (leading to ineffective synchronous communication)
- Development process differences
- Knowledge management challenges (most of the existing management approaches are designed for collocated teams)
- Technical issues: Incompatible data formats and exchanges
- Security issues (ensuring electronic transmissions are confidential and private).

Development environment, characteristics and challenges of distribute software development could be sources of many risks, which need to be managed effectively to avoid an unacceptable budget or delay in the schedule [21].

Finally, it can be seen that there are many common challenges for both web and distributed developments (e.g. cultural differences, developers' background differences, importance of communication and security issues). These challenges could be sources of many risks. Thus, these need to be considered in any proposed risk management approach that intends to manage web and distributed development risks. The next sections discuss potential web and distributed development risks and their management challenges.

2.5 Potential Risks for Web and Distributed Development

Al-Rousan[59] highlighted that the most important reasons for web projects delays are the risks involved, including the poor communications within the development team members or with the customers. Al-Rousan listed 21 risks as the most important risks to web development. Some of these risks are: Difficult to define content and functional requirements; Threats from competitors; Time and location from where the applications are accessed are unpredictable; Little development consideration on safety, security and reliability; New technologies not well developed yet and lack of matrurity; Web developers have variety of background, experience and age; Lack of define user categories, continually change project/scope/ objectives; and Many external suppliers involved in the development project and legacy systems are poor documented.

Reed and Knight conducted a comparison study [82] which aimed to understand the significance of the impact of a number of risks on IT projects that use virtual (distributed) teams. In this study, seven risks, are filtered from 55 risks as they are more important to virtual teams than to collocated teams. The risks are: Insufficient knowledge transfer; Lack of project team cohesion; Cultural or language differences; Inadequate technical resources; Inexperience with company and its processes; Loss of key resource(s); and Hidden agendas. The study illustrated a significant difference in risk importance between the distributed and the collocated software developments. In particular, the study showed that the seven evaluated risks have a lower impact (i.e. no impact or simply did not occur) on collocated development teams compared with the distributed teams [82].

This thesis deals with the risks of both web development and distributed development as one chunk, and refers to them as Web and Distributed (WD) development risks. Reasons to deal with the web and distributed development risks as one chunk are:

- The identified web and distributed development challenges, illustrate that there are many shared challenges (e.g. communication failures, time to market, developers' background differences and others) between the web and distributed developments and they thus share many risks.
- Many web developers follow the parallelism or subgroups development strategy because of web development short cycles. This is an adoption of distributed development strategy and indirectly inherits its challenges and risks [59].
- Many distributed development projects are web development projects and share the same risks.
- Providing one approach that is able to deal with risks of both web and distributed developments reduces training time and makes the risk management more easy for the developers since there is no need to switch between different risk management approaches.

In this section, the types of potential risks that WD developments face are identified. In fact, as mentioned above, all the identified challenges could pose risks to WD development. Aspects related to diversity, evolution, technology, availability and anonymity of users' types and numbers could be sources of many risks.

From a review of the available WD development literature [14; 43; 45; 48; 52; 58; 59; 64; 68; 69; 74; 76; 77; 80; 81; 83], a list of potential risks to WD development is recognized and presented in Table 2.2. The ticks in the table illustrate the majority (about 70%) of the risks are shared for web and distributed development. In addition to this list, almost all of the traditional and collocated software risks are also risks to WD development (although their impact and significance might be different). Another type of risk that could also affect WD developments is the atypical risk type [26] (see Atypical Risk definition in Section 1.1). Any completely new risk (unknown risk), when it happens for the first time, is considered an atypical risk. This type of risk could stop the progress of the development partly or totally, as the developers do not know what to do.

Table 2.2: WD Potential Risks

Risk No.	Risk Name	Web	Dis.
1	Unfamiliarity with international and foreign contract law	✔	✔
2	Inadequate customer requirement (see and change strategy)	✔	
3	Poor documentation	✔	✔
4	Low visibility of project process	✔	✔
5	Inadequate process development	✔	✔
6	Insufficient measurement and estimations	✔	✔
7	Lack of security precautions	✔	✔
8	Weaknesses in protection procedures for Intellectual Property rights	✔	
9	Vendor feasibility		✔
10	Insufficient competence	✔	✔
11	Communication failures	✔	✔
12	Poor sites management control		✔
13	Failure to manage user expectations	✔	
14	Insufficient project stakeholder involvement	✔	✔
15	Process instability	✔	✔
16	Poor performance	✔	✔
17	Poor UI	✔	
18	Insecure communication channels	✔	✔
19	Lack of requirement specification	✔	✔
20	Inadequate user involvement	✔	✔
21	Difficulties in ongoing support and maintenance	✔	✔
22	Unrealistic estimation of the number of users	✔	
23	Differences in the development methodologies and processes	✔	✔
24	Weak or inadequate contracts	✔	✔
25	Complicated development dependencies between project sites		✔
26	Cross-cultural differences and influence	✔	✔
27	Poor product functionality	✔	✔
28	Market fluctuations	✔	✔
29	Scalability limitations	✔	
30	Poor availability	✔	✔
31	Lack of top management commitment	✔	✔
32	Instability in other project sites		✔
33	Lack of face-to-face meetings	✔	✔
34	Lack of Management availability and efficiency	✔	✔
35	Unfamiliarity with customer type	✔	
36	Constraints due to time zone differences	✔	✔

2.6 Needs and Challenges of Managing WD Development Risks

Risk management in software development is still evolving, which is clear from the continuous stream of new software risk management methodologies and the high failure rate in software development projects [20; 84]. Usually, the reasons behind introducing new methodologies are weaknesses or inappropriateness of existing ones, or a need to cover new conditions. All these reasons seem to be valid for WD development risk management. WD development is a rapidly evolving development area and most of its developers depend on collocated software risk management methodologies or modified approaches to manage its risks. WD development involves some unique characteristics, challenges and risks which impose the following challenges and needs to risk management:

- Ability to evolve to accommodate to the continuous and rapid evolution of WD development.
- Consideration of WD development environment, challenges, characteristics, risks and factors.
- Simplicity and flexibility to cope with the different backgrounds of WD stakeholders and developers.
- Consideration of risks from 3P perspectives.

2.7 Summary

In order to understand the challenges and specific needs of WD development, the literature on WD development was reviewed. This review showed that WD development faces some unique challenges and risks that are not faced by collocated development. Based on this, WD development needs, and challenges to risk management, are specified. In the next chapter, existing risk management strategies are reviewed to explore their ability to deal with these challenges and satisfy their needs.

Chapter 3

Related Work

3.1 Introduction

Before proceeding to improve any of the existing risk management approaches or developing a new approach, the existing risk management approaches need to be reviewed and investigated, so that their strengths and weaknesses can be detected and then the real requirements for managing WD development can be specified. This chapter reviews a number of existing software risk management approaches to explore their ability to accommodate risk management needs for WD development. These approaches were either designed specifically to manage web risks or distributed risks or have abilities in this regard. The chapter presents the evaluation criteria, which were prepared for this purpose, describes the selected approaches, and presents the evaluation results in terms of weaknesses and strengths of the evaluated approaches. A case study called "Security System Case Study" is presented and used in this chapter as an example to illustrate the limitations of the existing approaches in dealing with WD development risk situations. Based on the strengths and weaknesses, the gap in the management of WD development risks is identified and the requirements of risk management in WD development are specified. An early version of this chapter has been published in [16].

3.2 Evaluation Criteria

The survey of WD development in the previous chapter showed the unique challenges and needs of risk management in WD development. In order to fully investigate the abilities of the existing software risk management approaches, evaluation criteria were first established to explore the abilities of existing approaches to satisfy the requirements of risk management for WD development. The criteria have been derived from the needs and risk management challenges of WD development, the risk management literature review, and the research aim [14; 24; 25; 43; 44; 45; 48; 54; 58; 60; 61; 62; 63; 64; 65; 66; 67; 68; 75; 80; 81; 83; 85]. Although the criteria focus on WD development, they also include some aspects that should be included to improve any software risk management in general. In order to get a consistent list of criteria factors initially, a list of all criteria factors was created and then filtered according to those most related to WD software development. In addition, other factors are specified in order to cover aspects which were previously not touched upon. The factors cover important risk management aspects (e.g. Perspectives, Communications, Evolving Environment and Risk Management Evolution). The evaluation criteria are outlined in Table 3.1, which is used to review the abilities of existing risk management approaches to manage WD development risks in the forthcoming sections.

Table 3.1: Approaches Evaluation Criteria

Criteria	**Rationale**
Managing risks from the 3P perspectives	In WD development, there is a significant involvement of the three perspectives.
Preparedness for Atypical Risks	It is expected that WD development is vulnerable to atypical risks, which are usually unpredictable risks. Risk management approaches should have some abilities and techniques to deal with such types of risk.
Ability to support risk management communications	Communication is an essential factor for any successful risk management in WD development.
RM performance evaluation	Risk management could cost too much or become ineffective; therefore, its performance should be evaluated and corrective actions need to be taken when there is a need.
Consideration of the unique characteristics and challenges of WD development	WD development involves some unique characteristics, challenges and risks which need to be considered (e.g. time zone differences).
Initiative with Potential WD Risks	Providing an initial list of potential WD risks helps the devlopers/managers to identify the risk earlier and faster and thereby increase their awareness regarding the risks.
Ability to evolve	The rapid nature of WD development makes this industry evolve very fast over time due to the technologies used, market influences and development environment. This high rate of evolving needs to be accommodated.
RM customization	Providing more than one option to manage the risks could encourage more developers to practise risk management especially when time is critical. This can be done by offering plain and deep RM types.
Learning from Mistakes	Learning from mistakes helps to improve the risk management approaches.
Consideration of WD risk factors	In WD development, there are some factors (e.g. level of dependencies), which could have an effect on risk evaluation, but these might not be considered during risks estimation as they are not part of risks estimation equations.

3.3 Existing SRM Approaches

There are many different Software Risk Management (SRM) approaches. Some of these approaches are named "models" and others are named "methodology", "frameworks" or "methods", but they have the same target, which is managing software risks. The following definitions show the distinct meanings of these terms [86]:

Definition 11: *Model: "Something such as an object, plan, or set of rules that is used to show what something else is like or how it works."*

Definition 12: *Methodology: "A system of ways of doing, teaching, or studying something."*

Definition 13: *Framework: "A system of rules, ideas, or beliefs that is used to plan or decide something."*

Definition 14: *Method: "A particular way of doing something. "*

Existing SRM approaches are reviewed in this chapter. Each of the approaches uses some steps, components or techniques, which may be different or have some similarities with other approaches. Twelve of the existing approaches have been selected for detailed evaluation in this study. The selected approaches are those that are expected to satisfy the needs of risk management for WD development. These approaches were selected because they are either dedicated to management of web risks or distributed risks, or have some abilities in this regard. The selected approaches are described below.

3.3.1 *A Risk-driven Model for Work Allocation*

Lamersdorf *et al.* [5] analyzed the work allocation in Global Software Development (GSD) projects and identified a number of problems that are related to task assignments across the sites (e.g. a complicated task could be assigned to a site with insufficient experience or allocating of works depending on manager experience only), which result in low productivity and negative impact on the project goals. Lamersdorf *et al.* [5] assert the importance of managing risks associated with work allocation in GSD and presented a model called the Risk-driven model for systematic work allocation in GSD. The Risk-driven model (See Figure 3.1) consists of two main steps [5]: 1) assigning tasks based on project and site characteristics, resources, and their impact on project goals; and 2) analyzing the assignment decision with respect to the possible risks. As seen in Figure 3.2 the model contains three main sub-models, i.e. the strategic assignment model; common causal model, which is able to store organization-specific experiences; and risk identification model, to predict project risks.

This model is a step on the way to considering distributed development risks, but it concentrates only on the work allocation aspect and mainly on project perspectives. The Risk-driven Model does not deal with atypical risks and does not evaluate the performance of the RM process. It provides one type of RM (i.e. Deep type - see Definitions 4 and 5 in Chapter 1) and is not ready for any evolution. Finally, the model is not initiated by any potential WD risks, but considers WD risk factors partially.

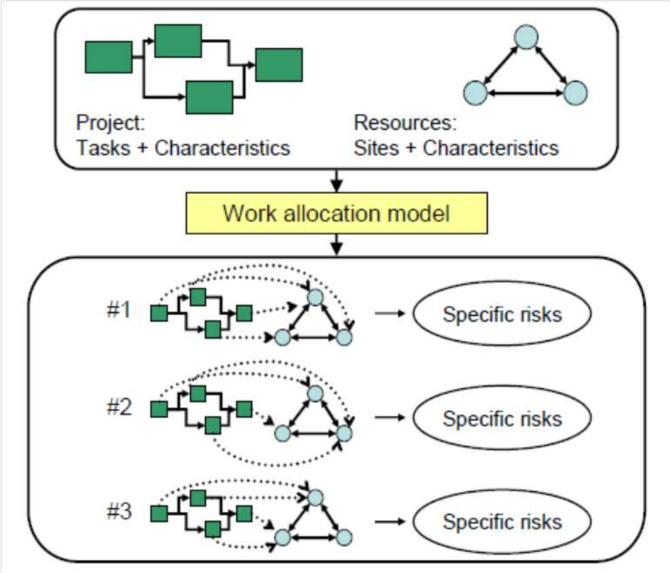


Figure 3.1: A Risk-driven Model for Work Allocation Input-Output [5]

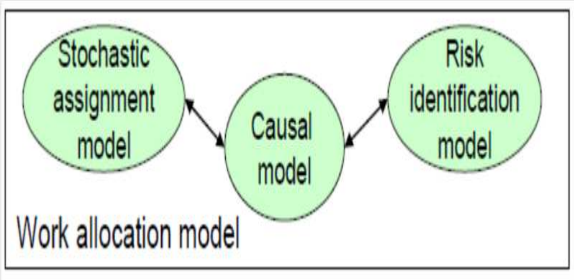
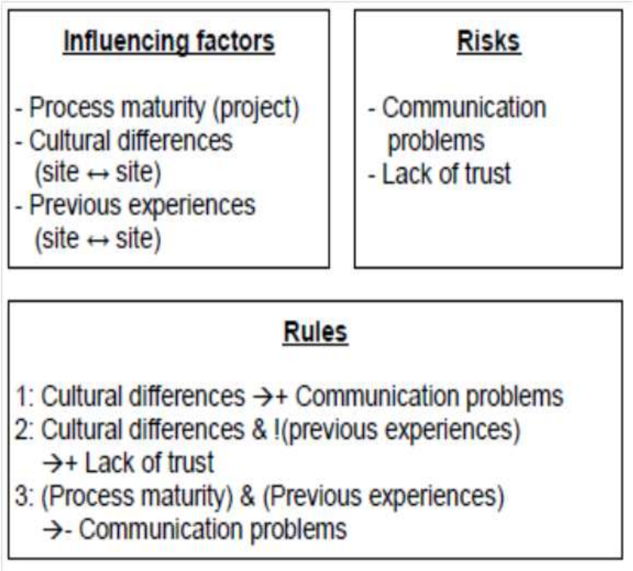


Figure 3.2: A Risk-driven Model for Work Allocation-Basic Structure [5]

3.3.2 *Rule-based Model*

The Rule-based Model, is introduced by Lamersdorf [6]. This model is designed based on previous experiences and aims to identify and assess specific risks in distributed software projects at an early stage. The model rules are derived from a number of interviews with distributed development experts. In the model, the project characteristics and lessons learned are used as an input for risk prediction purposes. The

Rule-based model considers the relationship between the risk and influence characteristic factors (characteristics of remote sites, sites relationships, tasks, and overall project characteristics) in the GSD, based on systematic design rules. In the model, the "Risk" is the possible problem; the "Influencing factors" are the characteristics of the project environment that have an impact on the problem, and the "Rules" are to formalize the impact of the influencing factors on the risks. Figure 3.3 illustrates an example of the model and how it structures the relationship between the influencing factors, risks and rules. As shown in the figure, the cultural differences between sites increase communication problems and the process maturity and previous experiences decrease communication problems [6].



Influencing factors

- Process maturity (project)
- Cultural differences (site ↔ site)
- Previous experiences (site ↔ site)

Risks

- Communication problems
- Lack of trust

Rules

1: Cultural differences →+ Communication problems
2: Cultural differences & !(previous experiences) →+ Lack of trust
3: (Process maturity) & (Previous experiences) →- Communication problems

Figure 3.3: A Rule-based Model [6]

In conjunction with the introduction of this model, Lamersdorf [6] has introduced a table of a number of identified rules which consist of logical rules and their textual description. The advantage of this model is its use of previous experiences, which are an important element for predicting and manages new risks. However, the model focuses only on software development within one organization and it does not have a systematic process or steps to estimate and evaluate the risks. In particular, it partially considers the process and product perspectives and does not deal with atypical risks. It also does not support RM communications or the evaluation of RM performance. The model provides a deep type of RM only and is not initiated for potential WD risks. Finally, it does not consider the WD risk factors.

3.3.3 *Process-Planning Methodology*

Betz *et al.* [7] presented a methodology to manage risks in GSD process planning. The methodology aims to improve planning by identifying its associated risks. As can be seen in Figure 3.4, the methodology consists of four steps, which are:

1. Process Modelling to identify risks
2. Risk Assessment to assess risks
3. Simulation and Evaluation of Process Improvements; followed by
4. Selection and Transformation

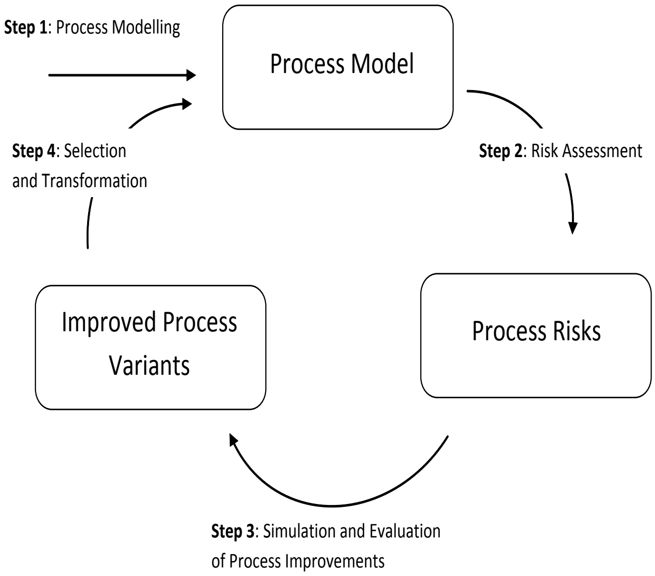

Step 1: Process Modelling
Process Model
Step 2: Risk Assessment
Process Risks
Step 3: Simulation and Evaluation of Process Improvements
Improved Process Variants
Step 4: Selection and Transformation

Figure 3.4: Process-Planning Methodology [7]

The Process-Planning methodology is supported with a tool, which is developed based on the methodology steps. The methodology depends on experience to estimate the identified risks, which makes the estimated values subjective to the user experience. Process-Planning Methodology supports partial evaluation of the RM performance, but does not consider the product perspective. The methodology does not deal with atypical risks, does not consider characteristics, challenges and risk factors of WD developments, but is initiated for some risks. It is not ready for any evolution and provides only a deep type of RM.

3.3.4 *RIAP Model*

Al-Rousan *et al.* [2] have introduced a model called the Risk Identification Architecture Pattern (RIAP), and a tool called the Web Project Risk Management Assessment

(WPRiMA) [8] which is built based on the RIAP model. Both the model and the tool aim to manage risks in web based applications. The RIAP model has been built in order to consider stakeholders', developers' and web projects' characteristics. As Figure 3.5 illustrates, the development process of the RIAP model consists of two main sections (Theoretical model and Operational model).

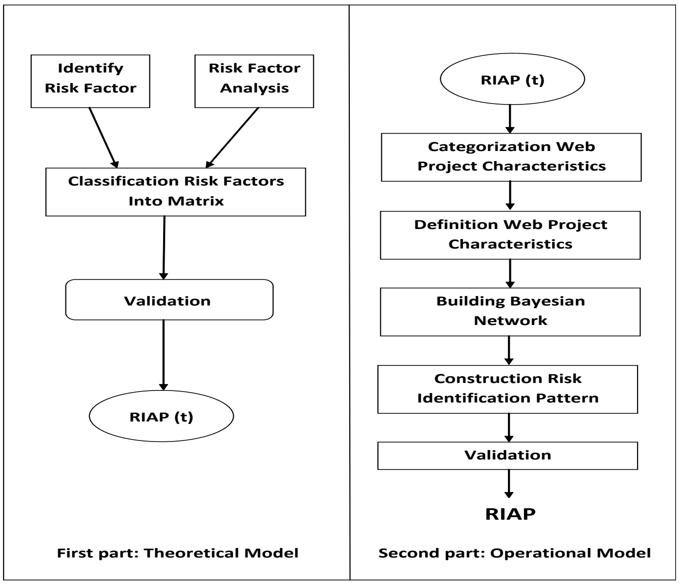

Identify Risk Factor
Risk Factor Analysis
Classification Risk Factors Into Matrix
Validation
RIAP (t)
First part: Theoretical Model
RIAP (t)
Categorization Web Project Characteristics
Definition Web Project Characteristics
Building Bayesian Network
Construction Risk Identification Pattern
Validation
RIAP
Second part: Operational Model

Figure 3.5: The Development Process of RIAP [8]

The theoretical model section aims to identify and assess the relevance of the risk factors of web development projects. This section involves four phases (survey and explore risk factors, assess relevance of risk factors to web project characteristics (see Table 3.2), classify the risk factors and validate performance) [2]. The operational model section uses the output of the first model as input and involves two phases (constructing a Bayesian Network and constructing a RIAP model).

RIAP has been supported with a prototype tool called WPRiMA, which supports its practising dimensions. The RIAP model and the WPRiMA tool try to link a list of risks with the characteristics of the web projects. However, RIAP focuses only on the project perspective; it is not able to deal with atypical risks; it partially supports RM communications and does not support RM performance evaluation. It does not deal with distributed perspectives of web development and is not prepared for RM evolvements. RIAP offers only one type of risk management (deep type). Although it is supported with a list of potential risks, it does not consider the WD factors (e.g. Sites dependencies).

Table 3.2: Risk Factors in Web Projects [2]

Characteristics	**Risk Factors**
Content	1. The Standard of project quality is not clearly defined 2. The structure of contents lacks understanding
Navigation	3. Hard to navigate and detect problems 4. Large amount of information
Presentation	5. Hard to operate the web application, lacks simplicity 6. Hard to specify the aesthetics
Social Context	7. Difficult to define contents and functional requirements 8. Hard to express possible threats from competitors
Technical Context	9. Complexity of design models increases with the use of mobile devices 10. Lack of understanding of delivery medium concepts
Natural Context	11. Time and location from where the applications are accessed are unpredictable and at random 12. Meets user's expections to have accessibility at any time
Development Team	13.Web developers have varied backgrounds, experience and age 14. Built on emerging technology and methodology 15. Little development concern of safety, security, reliability
Technical Infrastructure	16. New techniques not well developed yet and lack maturity 17. Hard to predict operational environment
Process	18. Constantly evolving project/scope/objectives 19. Lack of defined user categories
Integration	20. Legacy systems are poorly documented 21. Many external suppliers involved in the development project

3.3.5 *DS-RM-Concept*

Janusz Gorski and Jakub Miler [9] introduced a concept of risk management in distributed software development projects. The concept is built based on the idea that communication is an essential issue in managing risks. Therefore, this approach is supported with a communication channel (see Figure 3.6). The communication channel is open for all stakeholders at any time. The channel is intended to collect data from different sources, including checklists, questionnaires, brainstorming sessions and individual observations. During the analysis, it is also used to exchange views for risk evaluation. All collected data are stored in an information repository called the Black Box for further analysis and assessment [9].

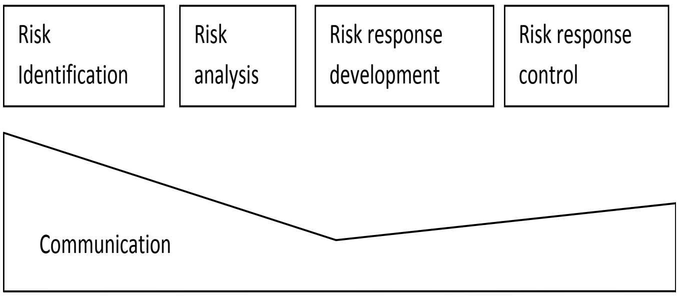


Figure 3.6: DS-SM-Concept [9]

As can be seen in Figure 3.7, this approach has three layers for processing, analyzing and assessing the risk data: Review to identify the risks, Snapshots to pass the identified data for analysis, and Reports to share the assessment results.

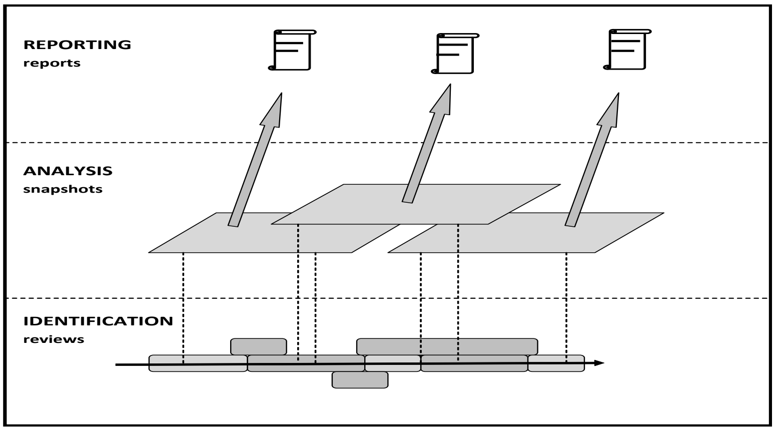


Figure 3.7: RiskAssessmentLayers of DS-SM-Concept [9]

This approach is supported with a tool called "Risk Guide". The approach has the advantage of having a risk communication channel, which continuously collects data for risk analysis and assessment. It is claimed that this channel is supported with unlimited memory [9]. However, gathering all this data is not an easy task and puts more pressure on managers and developers as it needs more resources and extra effort and time for sorting, filtering and analysing the collected data. This needs a lot of time, which is a critical issue with WD software projects. Moreover, the quality and credibility of the collected data, and their level importance, might not be taken into account. In this approach, the assessment and evaluation operations are subjective, as they depend on a rating process by stakeholders. Like most of the previous approaches, this approach focuses only on the project perspective and is not ready to deal with atypical risks. It supports RM communications and is initiated for a list of potential distributed software risks. It partially considers the characteristics and challenges of WD development and RM performance evaluation. It offers only a deep type of RM and is not ready for RM process evolution. The DS-RM-Concept has the advantage of learning from mistakes and partially considers some WD risk factors.

3.3.6 *ProRisk Framework*

Roy [10] has proposed a software risk management framework called ProRisk. As illustrated in Figure 3.8, the ProRisk framework focuses on business and operational management dimensions. The business dimension focuses on identifying the economic environment, and the operational dimension focuses on the main risk management steps (e.g. identifying risks, describing action plans, implementing the plans and re-assessment).

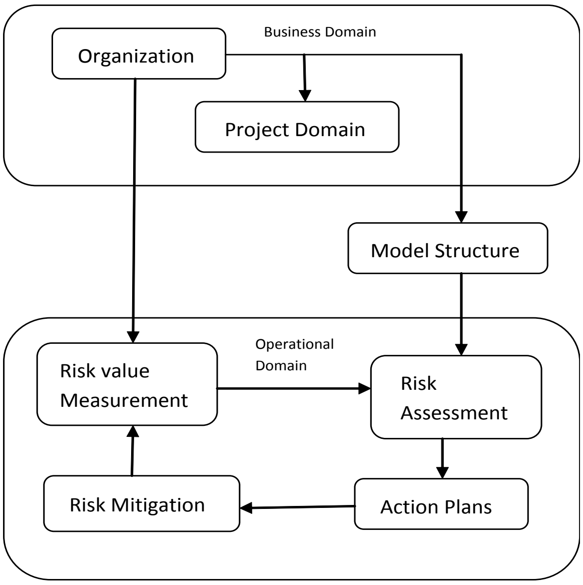


Figure 3.8: ProRisk Framework [10]

The ProRisk Framework is built with consideration of the hierarchical structure of the Software Engineering Institute's (SEI) risk taxonomy [87]. It involves a number of activities, including Stakeholder identification, Risk factor identification, Risk tree model construction, Calibration of the model, Estimation of the risk event probabilities, Computation of combined risk values, Development of action plans, Monitoring of the progress, and Framework operation. Roy [10] claimed that this framework is open for users to calibrate and use alongside other models. However, ProRisk involves deep analysis and a large number of activities that need to be implemented in order to manage the risks. The used models are integrated under a number of constraints (e.g. risk factors are largely independent and use appropriate methods for computing risk values [10]), but this could raise a question about the level of credibility and reliability of these models. The ProRisk framework does not consider the characteristics and risk factors of WD development and concentrates on the project perspective only. It involves a large number of activities which could increase complexity and consume more time. The framework is not ready to deal with atypical types of risk, but partially supports RM communications and RM performance evaluation. The ProRisk framework is not provided with a list of potential WD risk; it has some abilities for evolution. Finally, consideration of WD risk factors and learning from mistakes are not supported by this framework.

3.3.7 *Riskit Method*

Riskit is a method to manage software risks introduced by Kontio [3; 88]. The Riskit method links between project risks and stakeholders with a focus on identifying the impact of risk scenarios on project goals [3; 88]. Figure 3.9 illustrates the Riskit process diagram, and Table 3.3 describes the steps of the Riskit method [3; 88].

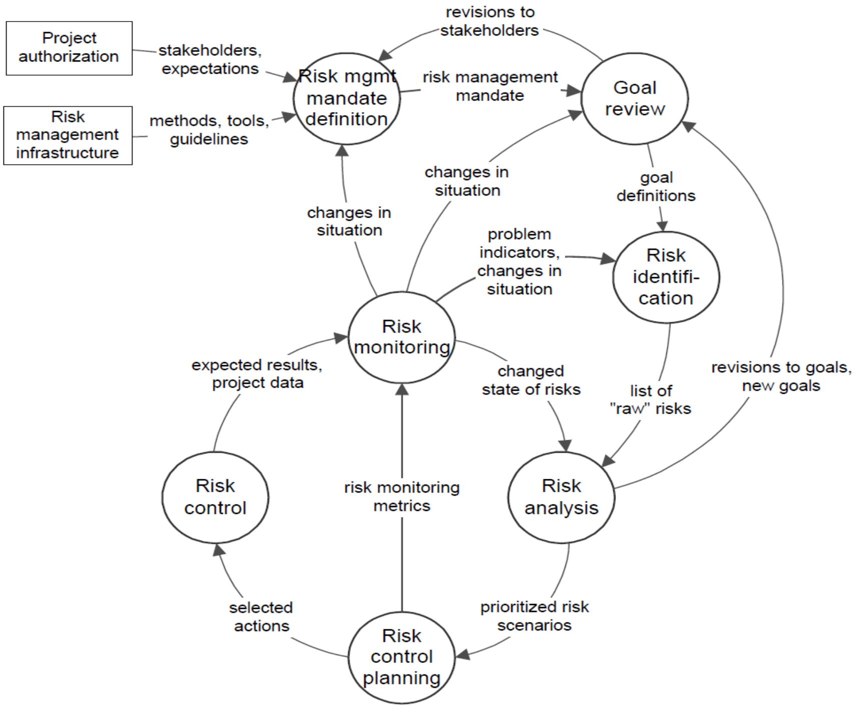


Figure 3.9: The Process Diagram of Riskit Method [3]

The Riskit method is also provided with a Riskit Analysis Graph to represent risk scenarios. This method has some advantages, such as visualising risk scenarios, simplifying estimating techniques and making available training material. However, it does not consider the potential risks of WD development or its associated characteristics and challenges. In fact, it is a very detailed method which makes it difficult for use in rapidly evolving WD development. Like other existing SRM approaches, it does not have a mechanism to deal with atypical risks and supports only the deep RM type. The method does not consider the product perspective or RM communications, but supports RM performance evaluation. The Riskit method is not ready for RM evolution or learning form mistakes, nor does it consider the WD risk factors.

3.3.8 *SoftRisk*

SoftRisk [11; 89] is a model developed to manage software risks. It is intended to manage risks of all project sizes and is supported with a prototype tool which is

Table 3.3: Riskit Steps Overview [3]

Riskit step	**Description**	**Output**
Risk management mandate definition	Define the scope and frequency of risk management. Recognize all relevant stakeholders	Risk management mandate: why, what, when, who, how, and for whom?
Goal review	Review the stated goals for the project, refine them and define implicit goals and constraints explicitly. Analyze stakeholders' associations with the goals	Explicit goal definitions
Risk identification	Identify potential threats to the project using multiple approaches.	A list of "raw" risks.
Risk analysis	Classify and consolidate risks. Complete risk scenarios for main risk events. Estimate risk effects for all risk scenarios Estimate probabilities and utility losses of risk scenarios.	Completed Riskit analysis graphs for all analyzed risks. Ranked risk scenarios.
Risk control planning	Select the most important risks for risk control planning. Propose risk control actions for most important risks. Select the risk control actions to be implemented.	Selected risk controlling actions
Risk control	Implement the risk controlling actions.	Reduced risks.
Risk monitoring	Monitor the risk situation.	Risk status information.

built based on model steps. As shown in Figure 3.10, the SoftRisk model consists of 7 iterative steps (Identify, Estimate, Document, Assess, Prioritize, Monitor and Control).

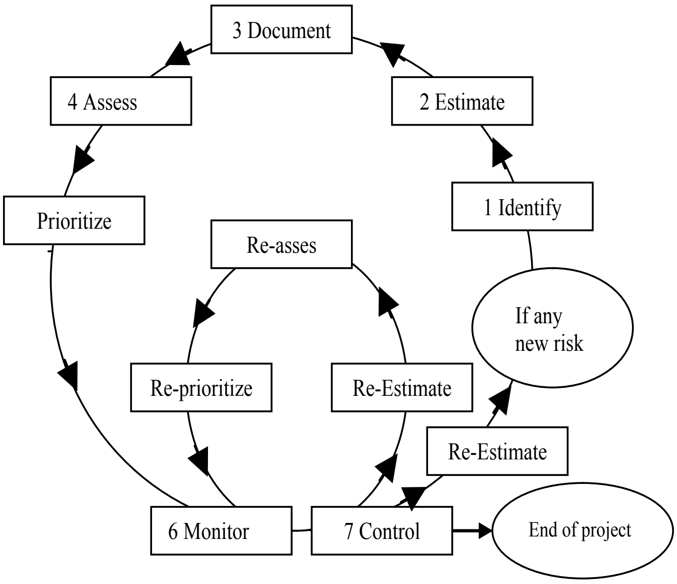


Figure 3.10: The Main Steps of SoftRisk Model [11]

SoftRisk focuses on top risks in each management cycle, concentrating on risk documentation and using historical risk data as a strategy to avoid risks and produce statistics for future use. However, the SoftRisk model does not consider any aspects of WD development, including its risk factors, and is not provided with an initial list of the WD potential risks. It focuses only on project perspective. This model is not ready for atypical risks and does not provide any RM communications. The SoftRisk model supports deep RM only, but is not ready for any RM evolution. Generally, the model supports the learning from mistakes concept.

3.3.9 *CMMI-RSKM*

Capability Maturity Model Integration (CMMI) [90] is a process improvement approach, introduced by SEI [90]. CMMI covers 22 process areas, including (CMMI-RSKM) risk management. As Figure 3.11 illustrates, the risk management process is supported with a risk repository. CMMI-RSKM covers three main goals (G1-G3) [12] which are:
G1: Prepare for Risk Management: Determine risk source and categories, Define risk parameters, Establish a risk management strategy.
G2: Identify and Analyze Risks: Identify risks, Evaluate and prioritize risks.
G3: Mitigate Risks: Develop mitigation plans, Implement mitigation plans.

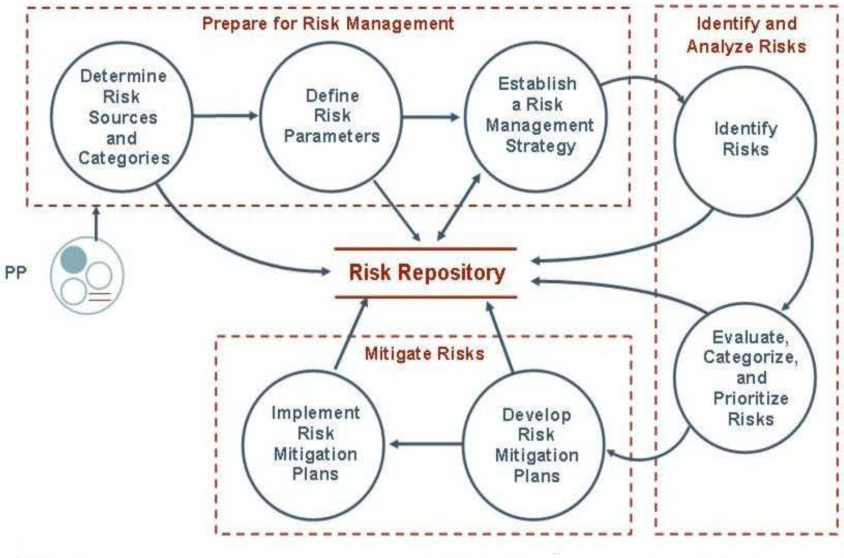

Prepare for Risk Management
Identify and Analyze Risks
Determine Risk Sources and Categories
Define Risk Parameters
Establish a Risk Management Strategy
Identify Risks
PP
Risk Repository
Evaluate, Categorize, and Prioritize Risks
Mitigate Risks
Implement Risk Mitigation Plans
Develop Risk Mitigation Plans

Figure 3.11: CMMI-RSKM Structure [12]

Generally, the CMMI-RSKM focuses on training, which is essential to practitioners and instructors. However, the CMMI-RSKM offers only the deep type of risk management and, like some other approaches, does not consider the unique characteristics and challenges of WD development; furthermore, it is not flexible enough for evolving [12; 90; 91; 92]. CMMI-RSKM considers the three perspectives (project, process and product) but is not ready to deal with atypical risk types. It supports

RM communication and RM performance evaluation, but it is not provided with WD potential risks and does not consider the WD risk factors.

3.3.10 *PMBOK*

PMBOK is a project management methodology introduced by the Project Management Institute [13]. PMBOK has 44 project management processes and 9 knowledge areas for project management (Integration, Scope, Time, Cost, Quality, Human Resources, Communications Risk and Procurement). PMBOK consists of four process phases (Initiating, Planning, Executing, and Closing). This methodology can be considered as a standard for project management [91; 93; 94]. The PMBOK methodology supports the distribution of information between project stakeholders. As is illustrated in Figure 3.12, the methodology supports the ordinary project risk management steps (i.e. Plan risk management, Identify risks, Perform qualitative analysis, Perform quantitative risk analysis, Plan risk management response and Monitor and control risks).

However, PMBOK is designed for general project management purposes and is not ready for use in WD development as it does not consider any of the WD development specific challenges and characteristics and is not initiated for any of the WD development potential risks. Moreover, PMBOK is not flexible enough to accommodate the rapid changes of WD development.

Aspects like preparedness for atypical risks and consideration of WD risks factors are not supported by the PMBOK methodology and it supports the deep RM type only.

3.3.11 *GDSP Integrated Framework*

The Geographically Distributed Software Projects (GDSP) Integrated Framework [14] is a framework designed to manage distributed software project risks. A systematic survey of available literature to refine and specify risk areas and risk factors of distributed software projects was conducted before building the framework. The risk areas were linked to resolution techniques in order to specify which technique is suitable for which risk area.

As can be seen in Figure 3.13, the framework consists of three main risk management components, namely: risk assessment, risk control and risk management planning focusing on distributed risks.

The GDSP Integrated Framework is supported with a web based tool called the Distributed Project Management System (DPMS). The risk management plans of this framework are supposed to be linked and integrated with other project plans. In fact, this framework is a step towards managing distributed risks; however, it depends

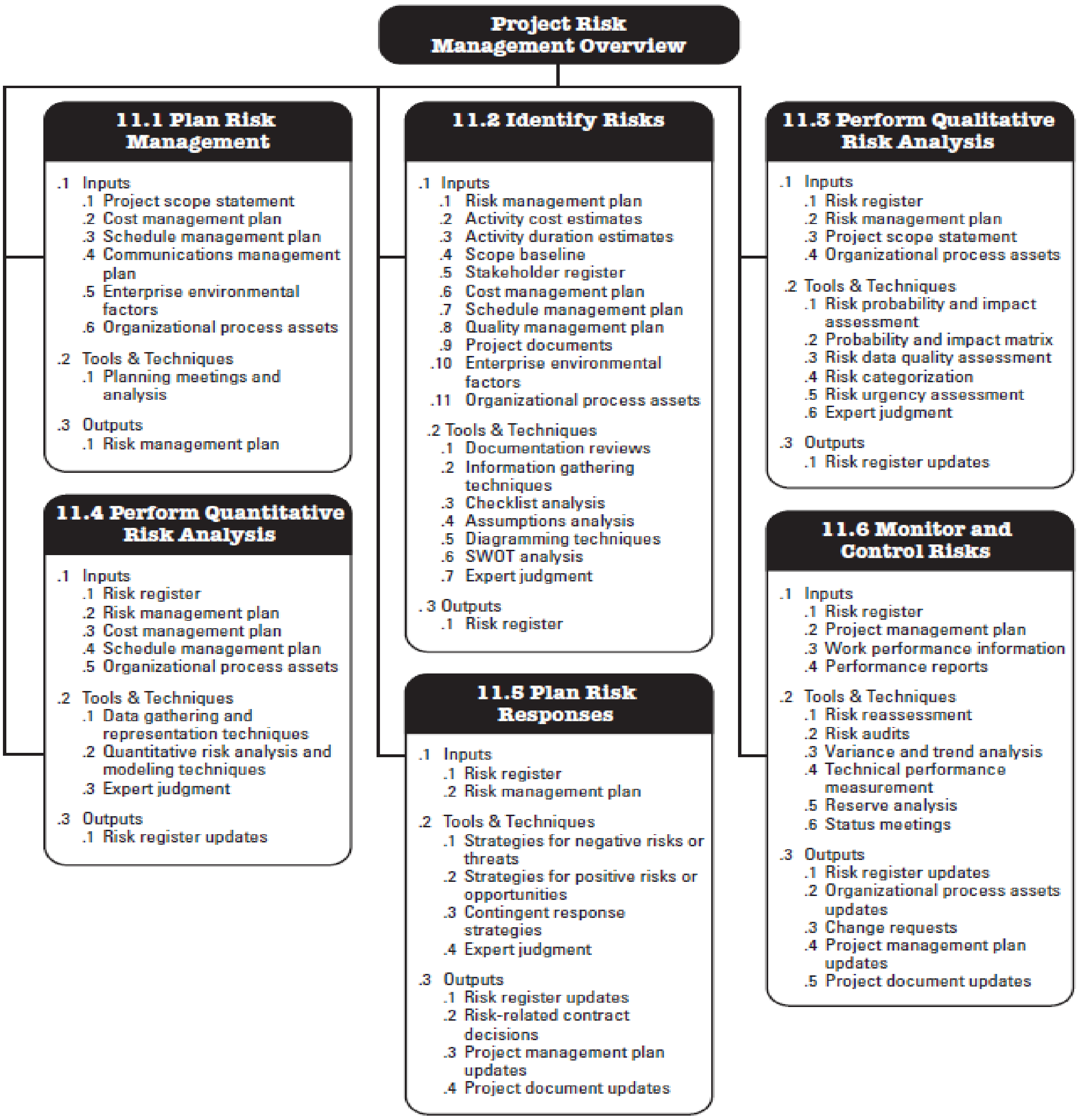

Project Risk Management Overview
11.1 Plan Risk Management
.1 Inputs
.1 Project scope statement
.2 Cost management plan
.3 Schedule management plan
.4 Communications management plan
.5 Enterprise environmental factors
.6 Organizational process assets
.2 Tools & Techniques
.1 Planning meetings and analysis
.3 Outputs
.1 Risk management plan
11.2 Identify Risks
.1 Inputs
.1 Risk management plan
.2 Activity cost estimates
.3 Activity duration estimates
.4 Scope baseline
.5 Stakeholder register
.6 Cost management plan
.7 Schedule management plan
.8 Quality management plan
.9 Project documents
.10 Enterprise environmental factors
.11 Organizational process assets
.2 Tools & Techniques
.1 Documentation reviews
.2 Information gathering techniques
.3 Checklist analysis
.4 Assumptions analysis
.5 Diagramming techniques
.6 SWOT analysis
.7 Expert judgment
.3 Outputs
.1 Risk register
11.3 Perform Qualitative Risk Analysis
.1 Inputs
.1 Risk register
.2 Risk management plan
.3 Project scope statement
.4 Organizational process assets
.2 Tools & Techniques
.1 Risk probability and impact assessment
.2 Probability and impact matrix
.3 Risk data quality assessment
.4 Risk categorization
.5 Risk urgency assessment
.6 Expert judgment
.3 Outputs
.1 Risk register updates
11.4 Perform Quantitative Risk Analysis
.1 Inputs
.1 Risk register
.2 Risk management plan
.3 Cost management plan
.4 Schedule management plan
.5 Organizational process assets
.2 Tools & Techniques
.1 Data gathering and representation techniques
.2 Quantitative risk analysis and modeling techniques
.3 Expert judgment
.3 Outputs
.1 Risk register updates
11.5 Plan Risk Responses
.1 Inputs
.1 Risk register
.2 Risk management plan
.2 Tools & Techniques
.1 Strategies for negative risks or threats
.2 Strategies for positive risks or opportunities
.3 Contingent response strategies
.4 Expert judgment
.3 Outputs
.1 Risk register updates
.2 Risk-related contract decisions
.3 Project management plan updates
.4 Project document updates
11.6 Monitor and Control Risks
.1 Inputs
.1 Risk register
.2 Project management plan
.3 Work performance information
.4 Performance reports
.2 Tools & Techniques
.1 Risk reassessment
.2 Risk audits
.3 Variance and trend analysis
.4 Technical performance measurement
.5 Reserve analysis
.6 Status meetings
.3 Outputs
.1 Risk register updates
.2 Organizational process assets updates
.3 Change requests
.4 Project management plan updates
.5 Project document updates

Figure 3.12: PMBOK Risk Management Diagram [13]

on open discussions to prioritize and control the risks, which depend on subjective judgement, and it is affected by the experience of managers. This could be also a time-consuming strategy. The GDSP Integrated Framework could be much better if it considered the dependencies of risks and if it could be customized by offering more options for risk management. The framework focuses on the project perspective of distributed development only; supports RM communications; considers the many characteristics and challenges of distributed developments; and is initiated with a list of distributed development potential risks. The GDSP Integrated Framework is not ready to deal with atypical risk types or make any evolvements in the RM process. It partially supports the learning from mistakes concept.

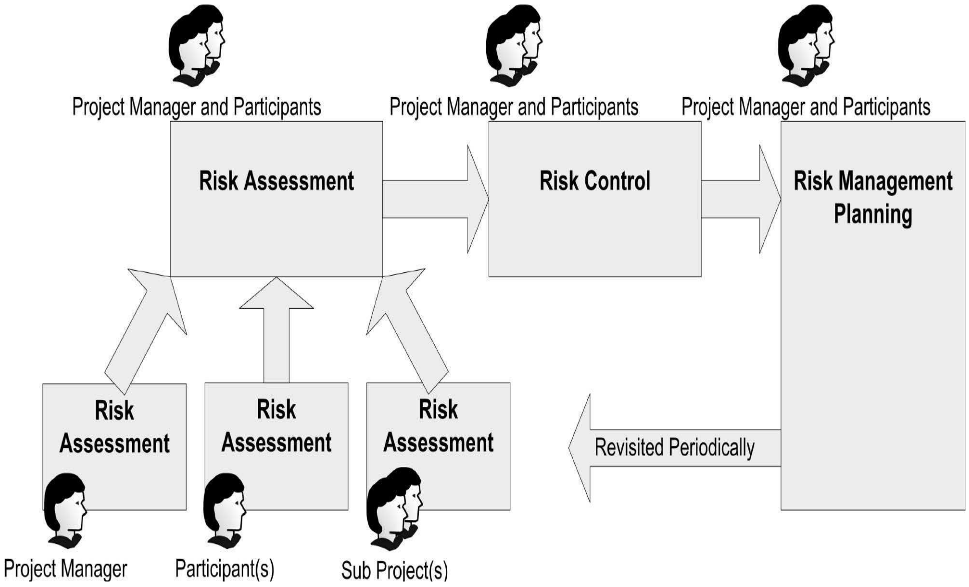


Figure 3.13: Elements of GDSPs Risk Management Framework [14]

3.3.12 *Risk and Performance Model*

As can be seen in Figure 3.14, the Risk-Performance-Model [15] uses six risk dimensions (organizational environment, user, requirements, project complexity, planning-control and team risks). The model considers the relationships between the risks, and project performance includes product and process performance. One of the model implications is that the six dimensions can be used as a technique for early identification and management of the risks [15].

The Risk-Performance-Model is built on a depth defined background, but is not easy to follow. The three perspectives are considered by the model, but with respect to social subsystem risk, and the model does not consider the other specific WD development challenges and risks. The preparedness for atypical risks and RM evolution are not supported by the model. Meanwhile, it is not provided with a RM communication channel and does not consider the WD risk factors. Similar to previous approaches, it has only one type of RM, which is the deep type, and it is not initiated by a WD development potential risk list.

3.4 Analysis

The approaches were reviewed for their ability to manage the risks of modern software development in the WD environment and how they can deal with the challenges. In order to see their weaknesses and strengths, the approaches have been evaluated based on the predefined criteria factors (see Table 3.1). This evaluation has been conducted based on available literature. Table 3.4 shows the results of the evaluation.

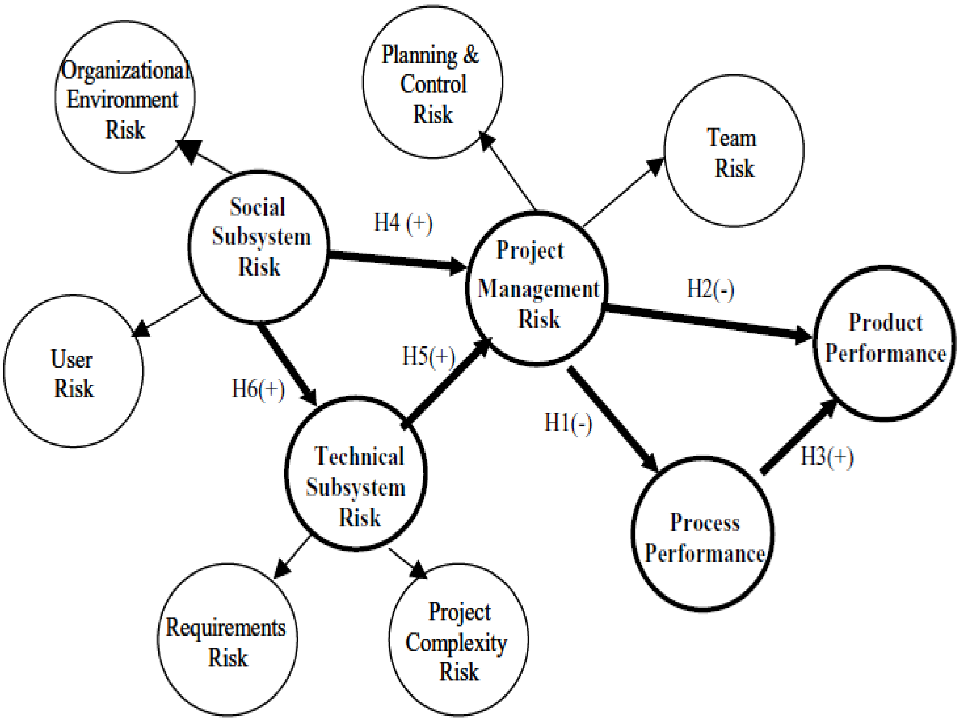


Figure 3.14: Risk-and-Performance-Model [15]

In Table 3.4, there are three options for each criteria factor:

✔ when the factor is supported by the approach.
✖ if the factor is not supported by the approach.
P if it is partially supported by the approach.

Table 3.4 can be read either horizontally or vertically. If it is read horizontally, then the numbers on the table represent the total points that each factor has from all of the approaches for each one of the above three options. If the table is read vertically then the numbers represent the total points each approach has for each one of the above three options.

From the numbers that appear in Table 3.4, it can be noticed that the total number of criteria factors that are supported or agreed by the approaches has 47 points from the total points, which is 156 (with percentage 30.1%). The ones that are partially supported or partially agree have 23 points (with a percentage of the total points of 14.7%), whereas the factors that have the lowest support by the existing approaches have the highest number of points, at 86 (with a percentage of 55.1%). The criteria factors that have the lowest support are:

- Covering of process and product perspectives

Table 3.4: Approach Evaluation

	Approaches						Sub Totals		
Evaluation Criteria	Risk-Driven	Rule-Based	Process-Planning	RIAP	DS-RM-Concept	ProRisk	✔	✖	**P**
Perspectives									
- Project	✔	✔	✔	✔	✔	✔	6	0	0
- Process	✖	**P**	✔	✖	✖	**P**	1	3	2
- Product	✖	**P**	✖	✖	✖	✖	0	5	1
Preparedness for Atypical risks	✖	✖	✖	✖	✖	✖	0	6	0
RM Communication	✔	✖	✔	**P**	✔	**P**	3	1	2
RM Performance Evaluation	✖	✖	**P**	✖	**P**	**P**	0	3	3
Consideration of WD Challenges and Characteristics	✔	✔	✖	**P**	**P**	✖	2	2	2
Initiative with Potential WD Risks	✖	✖	**P**	✔	✔	✖	2	3	1
Ability to Evolve	✖	**P**	✖	✖	✖	✔	1	4	1
Offered RM Types									
- Plain	✖	✖	✖	✖	✖	✖	0	6	0
- Deep	✔	✔	✔	✔	✔	✔	6	0	0
Learning from mistakes	✔	**P**	✖	**P**	✔	✖	2	2	2
Consideration of WD risks factors (e.g. Dependencies)	**P**	✖	✖	✖	**P**	✖	0	4	2
Columns Totals									
✔ Supported	5	3	4	3	5	3			
✖ Not Supported	7	6	7	7	5	7			
P Partially Supported	1	4	2	3	3	3			
Continued on Next Page..									

Continued from previous Page..									
	Approaches						**Sub Totals**		
Evaluation Criteria	Riskit	SoftRisk	CMMI-RSKM	PMBOK	GDSP-Integrated	Risk-Performance	✔	✖	**P**
Perspectives									
- Project	✔	✔	✔	✔	✔	✔	6	0	0
- Process	**P**	✖	✔	✔	✖	✔	3	2	1
- Product	✖	✖	**P**	✖	✖	**P**	0	4	2
Preparedness for Atypical risks	✖	✖	✖	✖	✖	✖	0	6	0
RM Communication	✖	✖	✔	✔	✔	✖	3	3	0
RM Performance Evaluation	✔	**P**	✔	✖	**P**	✔	3	1	2
Consideration of WD Challenges and Characteristics	✖	✖	✖	✖	✔	✖	1	5	0
Initiative with Potential WD Risks	✖	✖	✖	✖	✔	✖	1	5	0
Ability to Evolve	✖	✖	✖	✖	✖	✖	0	6	0
Offered RM Types									
- Plain	✖	✖	✖	✖	✖	✖	0	6	0
- Deep	✔	✔	✔	✔	✔	✔	6	0	0
Learning from mistakes	✖	✔	✖	**P**	**P**	✖	1	3	2
Consideration of WD risks factors (e.g. Dependencies)	✖	✖	✖	✖	✖	✖	0	6	0
Columns Totals									
✔ Supported	3	3	5	4	5	4			
✖ Not Supported	9	9	7	8	6	8			
P Partially Supported	1	1	1	1	2	1			
Summary of Table Totals									
Ticks	✔	✖	**P**	Over all points total					
Points	47	86	23	156					

- Preparedness for atypical risks
- Consideration of WD challenges
- Plain risk management type
- Ability to evolve
- Initiative with Potential WD Risks
- Consideration of WD risks factors (e.g. Dependencies)

As can be seen in Table 3.4, the points are different from one approach to another. This means that a weak aspect in one approach could be a strong aspect in another. This is clear from the total points at the end of each approach. On the other hand there are many similarities between many approaches in many aspects as they have the same selections for some criteria factors.

3.5 Approaches Weaknesses Summary

In general, the associated weaknesses of the existing approaches that have resulted from the evaluation can be summarized in the following points:

- The existing approaches concentrate on the project perspective of software development and do not pay enough attention to process and product perspectives.
- They are not ready to accommodate the continuous evolution of WD development.
- They do not consider the WD development risk factors (e.g. sites distribution and dependencies).
- They lack preparedness for atypical risk types.
- They are not flexible enough and they offer only the deep type of risk management. Plain risk management is not offered.
- They have no or weak evaluation of the RM performance.

3.6 Security System Case Study

This case study depends on a simplified WD risk situation (an example), which is designed in this research to demonstrate how the existing approaches act with WD development risk situations, and it will be used with the proposed approach for the same purpose in the following chapter.

The example involves atypical risk. This type of risk is not predictable, occurs suddenly and was previously unknown. In the example, we emulate an atypical risk situation in a WD development scenario. In brief, this risk situation emulates a web service software company, which has three distributed development sites (i.e. KL-Malaysia, New Delhi-India and Paris-France), and its headquarters are in London-UK. These distributions of the sites impose time zone, cultural and background differences and development dependencies. The development progress in all of the sites depends on the New Delhi site. This dependency is because there are key programmers working only at the New Delhi site that have special skills, abilities and experience to link system application modules with hardware components. This means any problems or delays at the New Delhi site will affect the rest of the development sites' progression.

The manager of this company has signed a very important contract to build a significant software component for a new security system for a famous airline company. According to the contract terms, the system should meet the required quality and reliability and should be ready for work within 63 working days. Any submission delay will cost extra (5%) per day as a fine.

The development progress of the system was going exactly as planned. Suddenly, an unexpected risk happened and stopped development progress at the New Delhi site. All development team members, including the programmers at the New Delhi site, had severe food poisoning. The programmers at the New Delhi site have unique skills in linking radio stations of the security system with software applications and cannot be replaced by others; the company has invested lots of money to train them.

This is an atypical risk as it involves the whole development team who suddenly become unable to perform development work. The manager was aware of the ordinary personnel shortfall risk of one, two or even part of the development team becoming unable to carry out the development, but not all of them and suddenly. The problem is that this risk is not limited to the desired site only. It affects all other sites as their progress depends on the New Delhi site. It also affects the company in general. The effect comes in the shape of schedule and cost overruns, loss of reputation, loss of market, and payment of fines and reparations.

The question is what to do. The project needs to be completed on time; the company was not ready for this type of risk and the work was almost suspended, with time not on the company's side. With the existing risk management approaches, there a little that could done, as discussed below:

- The existing RM approaches are designed to deal with ordinary risks when the probability and magnitude of the risk can be estimated. Thus, they are not ready to deal with the current atypical risk. Everything with this risk is different; it is unknown, happened suddenly, and no one expected it. Therefore, with the existing RM approaches, the managers could act unsystematically and might take the wrong action or decisions, which might cost more than the risk itself would cost.

- The existing RM approaches might able to manage the risks individually, but they cannot build a complete picture of site distribution and dependencies which affect the risks. Relationships can change the importance of any risk, regardless of its RE value or its importance for a single site. For example, in the above case study, any risk which affects the New Delhi site will have extra importance compared with other sites' risks as the other sites' progress depends on the New Delhi site. Existing RM approaches are not able to draw a real picture about WD risks in relation to importance. Moreover, more distribution and a higher number of sites could make the risks affect more sites, or lead to a combination of risks which become more complicated or even produce new risks

3.7 Requirements to Manage WD Development Risks

In order to enhance risk management in WD development, it is necessary to specify the requirements that need to be considered for improving or building any approach that is intended to manage WD development risks. The list of requirements in Table 3.5 were prepared after reviewing the existing approaches and specify their weaknesses and strengths, experience with software risk management and the specified challenges of WD development and its risk management needs.

3.8 Summary

In this chapter, existing software risk management is reviewed. Twelve approaches were selected for deep investigation to explore their abilities to manage WD development risks. The investigation was based on specific criteria that were prepared carefully for this purpose, which considered WD development risk management needs

Table 3.5: WD-RM Requirements

No	Requirements description	Rationale
WD-R1	Usability (easy to learn, simple).	It is expected that WD developers have different backgrounds. Therefore, different types of users with different types of background should be able to use the approach.
WD-R2	It should produce quick results.	WD development is rapidly evolving; therefore, the users are keen to see fast results from the used approaches
WD-R3	It should have the ability to evolve.	WD development is evolvable and involves rapid changes. Any risk management approach should be ready for any evolution if it is needed.
WD-R4	It should consider the characteristics of WD development.	This is to provide better awareness to potential risks and challenges
WD-R5	It should consider the WD factors that might have an affect on risks (e.g. dependencies).	This is to have an overall evaluation of the relationships that have an effect on risks
WD-R6	It should be a customisable approach.	To encourage developers/managers to practise RM, even when resources or time are limited.
WD-R7	Ability to use minimum information to manage the risks.	Sometimes there is a lack of information with WD developments.
WD-R8	Ability to deal with atypical risks.	Because there is always a chance of atypical risks
WD-R9	It should be initiated with a list of potential WD development risks.	To help manager/developers to start with them and to establish risk management awareness
WD-R10	Consideration of risks from 3P perspectives.	Because the risks can affect the 3P perspectives

(identified in Chapter 2). Weaknesses and strengths of the reviewed approaches were identified. Generally, many of the identified strengths are related to collocated software development and are spread among the approaches. As the review shows, there are some approaches that have some extra abilities than other approaches that can be engaged to manage WD development risks (e.g. DS-RM-Concept), these approaches still have many weaknesses in different aspects. From Table 3.4, the following points can be concluded:

- There is no single approach that is able to manage software risks in WD environments alone; unfortunately the strengths of the approaches are dispersed between them. In the current situation, developers must either use more than one approach or omit some aspects and support.
- Due to the weaknesses involved, any plan to improve any of the approaches means making the approach more complicated or too large.
- Tackling the weaknesses of the approaches and combining the strengths of them in a new approach is a step toward improving risk management in WD environment.
- Features such as the ability to evolve, flexibility and customization of RM are difficult to add to any of the approaches as this would entail a total restructure of the approach as a whole.

It can be concluded that the reviewed approaches have added significant value to the RM of traditional software development, but it is clear that WD developments are not yet well covered. Thus, there is still a gap that needs to be bridged in order to improve RM for WD development. In this chapter, the requirements of any approach that is intended to manage WD development risks are specified.

Chapter 4

WeDRisk Approach

4.1 Introduction

In the previous chapter, existing software risk management is reviewed. Furthermore, strengths and weaknesses of existing software risk management approaches in managing WD development risks are identified. Thus, the gap in the risk management of WD development and its requirements is specified and the need for a new approach is asserted.

In this chapter, an approach called WeDRisk (Web and Distributed Risk) is introduced. The WeDRisk approach is proposed in order to tackle some of the existing approaches' weaknesses with more emphasis on WD development. WeDRisk is intended to be flexible, evolvable, understandable and easy to use, and it considers the risks from the 3P perspectives. The same "Security System Case Study", which is introduced in the previous chapter is used again in this chapter as a running case study to illustrate how the WeDRisk approach works.

4.2 WeDRisk Structure

Simplicity, flexibility and the ability to evolve were considered during the design and building of the WeDRisk approach, which consists of three phases (RM Establishment phase, RM Implementation phase and RM Evaluation and Evolution phase), as well as a communication channel. The phases consist of modules, which contain components, steps, techniques and guidelines. Table 4.1 describes in summary the phases and their inputs and outputs.

Before describing in depth the WeDRisk structure, the three phases and the communication channel are briefly described as follows:

Table 4.1: WeDRisk Phases Summary

Phases	Description	Inputs	Outputs
Establishment	Concerns establishing risk management cycle. Involved Modules: Project Card, Stakeholder and RM Customization modules	Project Identification data, RM cycle data, Available risks data; Stakeholders' data; RM Avaliable recourses, Time Criticality and Experience with RM	Project Card; Stakeholders cards; Suggested RM type (Plain/Deep)
RM Implementation	To identify, estimate, evaluate, plan and control the risks and also to absorb atypical risks. Involved Modules: Risks Repository, Clustering, Estimation, Evaluation, Atypical Risks and Planning & Controlling Modules	Current risks data and Project card	Identified, estimated and prioritized risks, Risks cards, Control plans, Management and atypical risks absorbing actions and Updating data of risk and project cards
RM Evaluation & Evolution	To evaluate and audit RM process and to make improvements or modifications on the approach or risk management cycle. Involved Modules: Evaluation and Auditing and RM Evolution Regulator module	Progress data during RM cycle Evolution Box (requirements, suggestions, problems, performance reports..)	Performance report and suggestions for corrective actions, Steps, Techniques, corrective actions
Communication Channel ensures RM communications and data exchange during the RM cycle			

RM Establishment phase

The establishment phase is the foundation for any RM cycle in the WeDRisk approach. It is triggered before starting any of other phases (the first time only). The aim of this phase is to build a project card (see Table 4.3) and stakeholders' cards, and to customize the RM type (optional). The cards and phase modules are described in detail in the following sections. Building the project card is essential for all the RM cycle modules as all risk data updates are stored in the project card. Stakeholders' cards are used when there is any involvement of the stakeholders in the development or RM processes. Risk management customization specifies a suitable RM type (Plain or Deep) for the RM cycle in general or for specific risks. Plain or Deep scope is flexible and can be re-defined by the developer/manager.

RM Implementation phase

The RM implementation phase is the core of RM operations in the WeDRisk approach. It concerns the implementation of RM steps, such as identification, estimation, evaluation, planning and controlling, and documenting of potential risks in the risks repository. For example, the potential risks repository helps to identify the risks that could attack the desired development perspective and a risk card is issued for any new ordinary or atypical risk. All the risks in this repository are clustered from the 3P perspectives. Once the risk is identified, the next step is to specify the type of RM (Plain or Deep) which also can be triggered from this phase. Following this is the estimation of the risks based on the RM type needs. WeDRisk supports more than one type of risk estimation. All the estimated risks are subjected to evaluation, which prioritizes the risks based on their estimated impact values and produces a list of the most critical risks. Once the risks are evaluated, they need to be controlled using the controlling plans. Planning section could be triggered when the risk is identified to include the plan in the risk card or could be prepared just before the controlling process. Finally, the project card is updated and this phase's activities continue until the end of the RM cycle.

RM Evaluation and Evolution phase

This phase collects the data during the whole RM cycle and produces a performance evaluation report. This report is useful for monitoring risks and RM efficiency. The evaluation result is also used to specify and take corrective actions and make any needed evolutions or improvements to the WeDRisk approach components.

Communication Channel

This channel is designed to provide and ensure all internal and external RM communications and data exchanges during the RM cycles.

For simplicity and standardization purposes, special notations were designed and used to represent and structure the WeDRisk phases, modules and other components (the notations are shown in Figure 4.1). In order to ensure simplicity, flexibility and the ability to evolve of the WeDRisk approach, a modular design strategy was chosen to build the approach. This can be seen in Figure 4.2 which illustrates the main architectural design of the WeDRisk approach. This structure gives the modules the ability to:

- Work with minimum dependency.
- Make improvements and changes (i.e. adding, modifying or removing com- ponents) easily.
- Exchange data directly using the communication channel.

The following sections describe the WeDRisk phases and its modules in more detail.

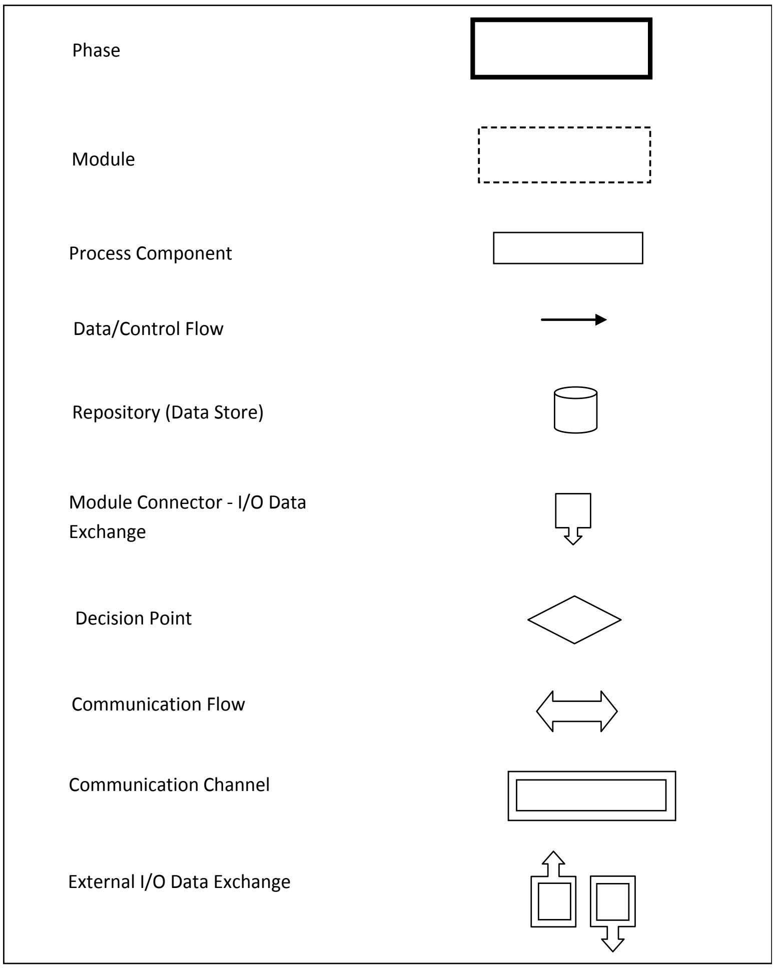


Figure 4.1: WeDRisk Notations

4.2.1 *RM Establishment Phase*

The establishment phase in the WeDRisk approach is responsible for establishing the risk management cycle. It handles all project data, stakeholder data and customizes the type of risk management (Plain or Deep). As illustrated in Table 4.2, this phase consists of three modules, namely: the project card module, stakeholder module and RM customization module. The three modules are described in detail below.

Project Card Module

The aim of this module is to produce and update risk management project cards from any development perspective. The project card is the foundation of any risk management cycle and is continuously updated over the time when there are any changes in the risk management data. The card can be used for a single or multisite WD development project type. In addition to project bibliographic data, the project card contains current and previous identified risk data (see Table 4.3). These data are used at different stages of the risk management cycle and also as historical data

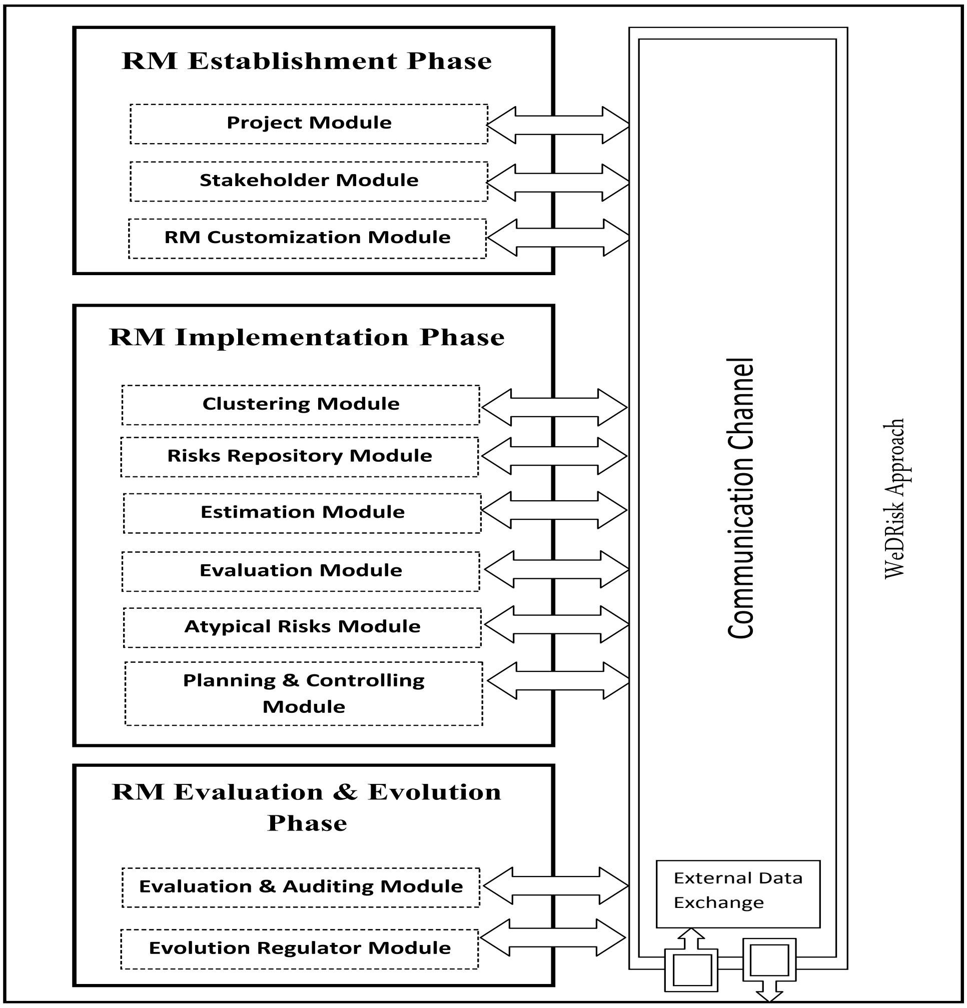


Figure 4.2: WeDRisk Approach Main Structure

Table 4.2: RM Establishment Phase

Module	Description	Inputs	Outputs
Project Card Module	A module to establish a risk management cycle project produces a card that contains all the data that are relevant to the risk management process	Bibliographic project data, risk data, management data and monitoring data	Project Card
Stakeholders' Module	A module to ensure and control the involvement of project stakeholders in the risk management process	Stakeholder bibliographic data and their tasks and privileges and permissions in the RM process	Stakeholders' Cards
RM Customization Module	A module to customize the RM (helps to decide what type of risk management is suitable (Plain or Deep) for the desired risk based on available information.	information about the risk management resources (Time, Staff, Budget and Experience)	Suggested decision (Plain or Deep)

for future and statistical purposes and learned lessons. Figure 4.3 is a diagram of the project card module.

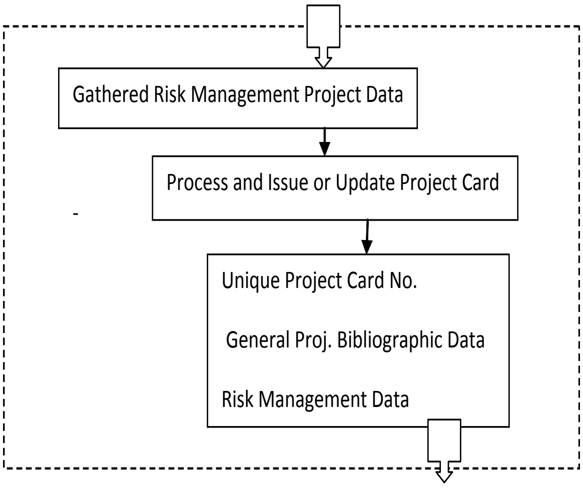
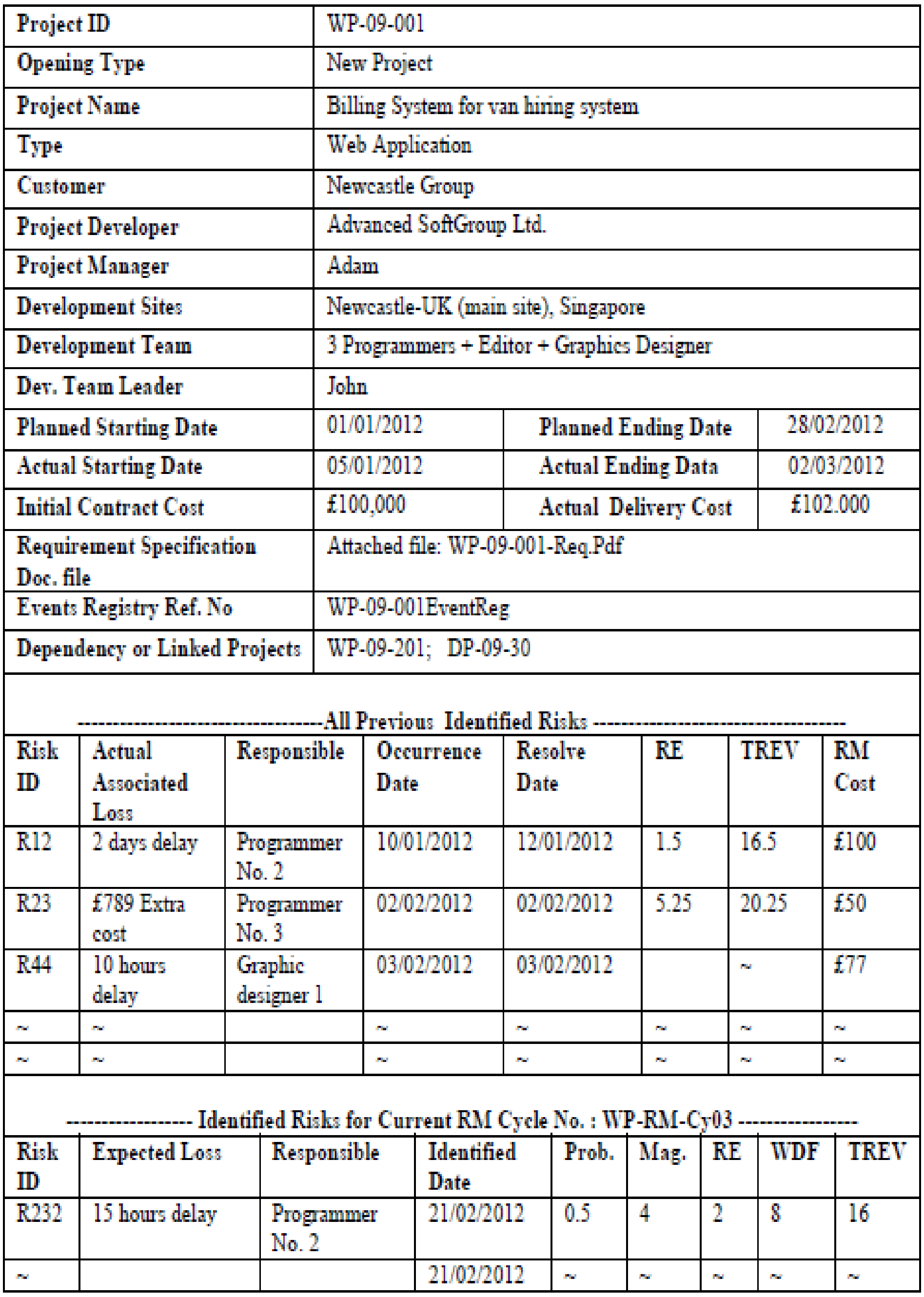

Gathered Risk Management Project Data
Process and Issue or Update Project Card
Unique Project Card No.
General Proj. Bibliographic Data
Risk Management Data

Figure 4.3: Project Card Module

Stakeholder Module

The involvement of project stakeholders in the risk management process is one of the success factors of WeDRisk. Therefore, one of its essential modules is the stakeholder module. This module aims to produce cards for all stakeholders involved in the risk management cycle as they might become involved in some RM aspects (e.g. perform tasks or be given relevant information). Stakeholder cards contain the main stake-

Table 4.3: Sample of Project Card



Project ID	WP-09-001		
Opening Type	New Project		
Project Name	Billing System for van hiring system		
Type	Web Application		
Customer	Newcastle Group		
Project Developer	Advanced SoftGroup Ltd.		
Project Manager	Adam		
Development Sites	Newcastle-UK (main site), Singapore		
Development Team	3 Programmers + Editor + Graphics Designer		
Dev. Team Leader	John		
Planned Starting Date	01/01/2012	**Planned Ending Date**	28/02/2012
Actual Starting Date	05/01/2012	**Actual Ending Data**	02/03/2012
Initial Contract Cost	£100,000	**Actual Delivery Cost**	£102.000
Requirement Specification Doc. file	Attached file: WP-09-001-Req.Pdf		
Events Registry Ref. No	WP-09-001EventReg		
Dependency or Linked Projects	WP-09-201; DP-09-30		

---------------------------------All Previous Identified Risks ---------------------------------

Risk ID	**Actual Associated Loss**	**Responsible**	**Occurrence Date**	**Resolve Date**	**RE**	**TREV**	**RM Cost**
R12	2 days delay	Programmer No. 2	10/01/2012	12/01/2012	1.5	16.5	£100
R23	£789 Extra cost	Programmer No. 3	02/02/2012	02/02/2012	5.25	20.25	£50
R44	10 hours delay	Graphic designer 1	03/02/2012	03/02/2012		~	£77
~	~		~	~	~	~	~
~	~		~	~	~	~	~

---------------- Identified Risks for Current RM Cycle No. : WP-RM-Cy03 ----------------

Risk ID	**Expected Loss**	**Responsible**	**Identified Date**	**Prob.**	**Mag.**	**RE**	**WDF**	**TREV**
R232	15 hours delay	Programmer No. 2	21/02/2012	0.5	4	2	8	16
~			21/02/2012	~	~	~	~	~

holder bibliographic data, privileges and permissions in the RM process, as well as any assigned tasks (role) related to the risk management cycle (Table 4.4 is an example of a stakeholder card).

As can be seen in Figure 4.4, the stakeholder module consists of four components. The first is an input component which gathers stakeholder related data and passes it to the second component which issues and updates the stakeholder cards. The third component produces and updates a list of involved stakeholders. All stakeholder

Table 4.4: An Example of Stakeholder Card

Stakeholder ID	StW101
Stakeholder Name	Rose
Stakeholder Type	Person
Stakeholder Projects	WP-09-001
Stakeholder Role	Financial
Stakeholder Privileges	Costs Related Risks

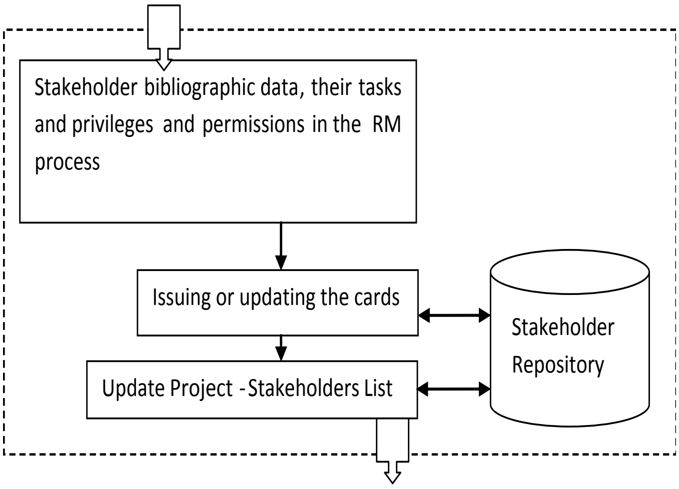

Stakeholder bibliographic data, their tasks and privileges and permissions in the RM process
Issuing or updating the cards
Update Project -Stakeholders List
Stakeholder Repository

Figure 4.4: Stakeholder Module

data are stored in a component called the stakeholders' repository which is the fourth component of this module.

RM Customization Module

As demonstrated in the previous chapter, one of the existing software risk management weaknesses is the offering of a single type of risk management, which is usually the deep risk management type. In WD development, many developers/managers ignore the practice of risk management if there is a resource shortage or criticality in time [95]. Hence, in order to encourage them to practise RM, even when there are shortages in time and resources, the WeDRisk approach offers Plain and Deep types of RM (defined in Chapter 1) to manage the risks involved. The Plain type is supposed to be suitable for RM when the resources (e.g. time, budget or staff) are limited or critical, whereas the Deep type is intended to be used when there are sufficient resources. However, it is not easy to decide when to use each type. Therefore, the WeDRisk approach is supported with a RM customization module, which aims to help the developers/managers to decide when to use Deep and when to use Plain RM types. The module is designed to be simple and gives quick results. It can be used to deal with the risks individually (one risk each time) or a risk situation (complete RM cycle). As can be seen in Figure 4.5, the core of the RM Customization Module is a matrix called the RM Customization Matrix. The matrix uses available information about risk management resources (time, staff, budget and experience) as input. For each resource, there are two options (Enough or Limited). In the matrix (see Table 4.5), the *Limited* option under the *Time, Staff* and *Budget* resources is coloured in red, and the *Limited* option under the *Experience* with the RM is coloured in blue. This colouring is used to simplify the utilization of the matrix and it is also linked with the matrix conditions to give fast results. Based on the applicability of the options and conditions, the matrix helps the users to make decisions regarding the suitable RM type. Three conditions work in conjunction with the applicable options to help the user to make the right decision, which are:

- If there is any tick under RED, go for the Plain RM type
- If all ticks are under BLACK, then go for Deep RM
- **Exception:** If there is a tick under the "Enough" RM Experience in Blue (i.e. three years' or more experience in managing related risks) then decisions can be made based on experience.

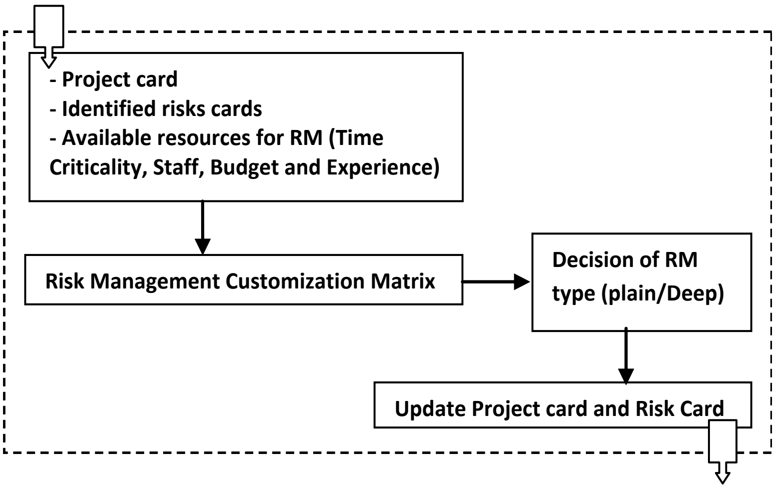

- Project card
- Identified risks cards
- Available resources for RM (Time Criticality, Staff, Budget and Experience)
Risk Management Customization Matrix
Decision of RM type (plain/Deep)
Update Project card and Risk Card

Figure 4.5: Customization Module

In the Plain RM type, only minimal and essential steps of RM (i.e. estimating RE, quick evaluation based on RE, and control of the risk) are to be implemented. In contrast, in the deep RM type, all RM steps including additional steps, such as estimating Total Risk Estimation Value (TREV) (see equation 4.1) values, prioritizing the risks based on TREV values, evaluating the RM performance, building/updating all cards and RM evolution are to be implemented. In fact, to ensure more flexibility, the steps for the Plain or Deep risk management types can be refined and they are subject to situation criticality and manager experience.

Table 4.5: Example of Customization Matrix

		RM Time		RM Staff Availability		RM Budget Availability		RM Experience		RM Type Decision	
RM Cycle / Situation No.	Risk No.	Enough	Limited	Enough	Limited	Enough	Limited	Enough	Limited	**Plain**	**Deep**
W1-C1	R33		✓	✓		✓			✓	✓	
W1-C1	R5	✓		✓		✓			✓		✓

If there is any tick under RED go for **Plain** RM type.
If all ticks are under BLACK then go for **Deep** RM.
Exception: If there is a tick under "Enough" RM Experience which is BLUE,(i.e. three years' or more experience in managing related risks) then decisions can be made based on experience.

Notes:
It is not necessary for all fields to be ticked to make the decision. This is based on the available information.
Enough: Means there is sufficient of the particular resource (time, budget or staff) for RM.
Limited: Means there is insufficient (Shortage/ Criticality) of the resource (time, budget or staffs) for the RM operation.

4.2.2 *RM Implementation Phase*

All risk management steps (e.g. identification, estimation, evaluation, planning and controlling) are included in the modules of this phase. However, the techniques used and the manner of conducting these steps could be different from other approaches, as they are designed to accommodate the WD development nature and needs. For instance, aspects such as sites dependencies, which are related to WD development, are considered. Atypical risks are also treated in this phase. The RM implementation phase ensures continuous risk management implementation for the 3P perspectives of the WD development. As Table 4.6 shows, the RM implementation phase consists of six modules (Clustering Module, Risks Repository Module, Estimation Module, Evaluation Module, Atypical Risks Module and Planning and controlling Module). These modules are described in detail in the following subsections.

RM Clustering Module

WeDRisk includes several concepts, which could help in tackling some of the identified weaknesses in the existing SRM approaches. One of these concepts is the consideration of the risks from the 3P perspectives. This concept depends on a clustering strategy which uses special criteria to deal with the risk from these three perspectives. The clustering strategy is intended to save time and effort. It locates fewer resources for each perspective, as the management of risks will focus on the relevant perspective risks each time. WeDRisk suggests some factors that could help to cluster the risks from the 3P perspectives (they are shown in Figure 4.6).

The suggested criteria group the risks based on some characteristics and the nature of the perspectives. The proposed criteria are identified based on experience, available literature and previous research results [23]. Table 4.7 shows a number of WD potential risks that are clustered based on the 3P clustering criteria. WeDRisk focuses on the 3P perspective risks equally and does not ignore any of them. The clustering criteria are used simply to help the developers/managers to categorize the risks from the 3P perspectives. The clustering from the 3P perspectives should be performed with any newly identified risk and before or during its inclusion into the risks repository.

Risks Repository Module

The Risks Repository Module has a vital role in the risk management process as it is the core of the risk identification process. It provides a preliminary list of WD development potential risks. The risks are clustered from the 3P perspectives and made available for use during any RM cycle. Any risk has a unique card, called a

Table 4.6: RM Implementation Phase

Module	Description	Inputs	Outputs
Clustering Module	Any potential risks to WD should be clustered from the 3P perspectives before being saved in risks repository	Risk data , clustering criteria	Clustered risks from 3P perspectives
Risks Repository Module	Cards are issued for all risks and saved in this repository. Each card has a unique number and contains all main data about the risk. To help developers/managers to identify the risks, the risks repository is initiated with cards for all known WD potential risks.	Current cycle identified risks and any potential risks	Risk cards clustered from 3P perspectives and made available for use during the risk management cycles
Estimation Module	This module estimates the risks with consideration to WD factors. It uses two estimation equations RE and TREV and WDF estimation matrix	Risks cards and related information that could be used to estimate risks probabilities , magnitudes WD factors	RE, WDF, TREV values
Evaluation Module	To evaluate the risks	Estimated identified risks (RE/TREV values) and atypical risks, project and risk card	Top risks (most critical) and prioritized risks based on RE/TREV values
Atypical Risks Module	To deal with and absorb new unpredictable risks (atypical risks)	Atypical risk	Absorbing actions, and risk card
Planning and Controlling Module	This module deals with the preparation of plans and precautions to deal with the risks	Ideas, experience, historical experiment, learned lessons, risks cards	Update risks card with plans (Who, What to , needs ,..)

“Risk Card” (see Table 4.8), which contains the main risk data (e.g. risk reference number, name, perspective, potential impact and suggested control plan). In addition to the initial list of WD potential risks, the risks repository module receives risks from current RM cycles, previous and similar development, available literature and other related sources. A risk card needs to be built for any new identified risk before adding it to the risks repository. All the risks in the repository are clustered from the 3P perspectives using the module clustering criteria. This repository is supposed to deal with ordinary risks and atypical risks. Ordinary risks which are known risks might be new to the desired RM cycle, whereas atypical risks are completely new and unpredictable risks. In the case of ordinary risks, the cards are created before dealing with risks, or they might already exist as part of the initial potential risks list; however, in the case of atypical risks, the priority is to deal with the risk and then build its card,

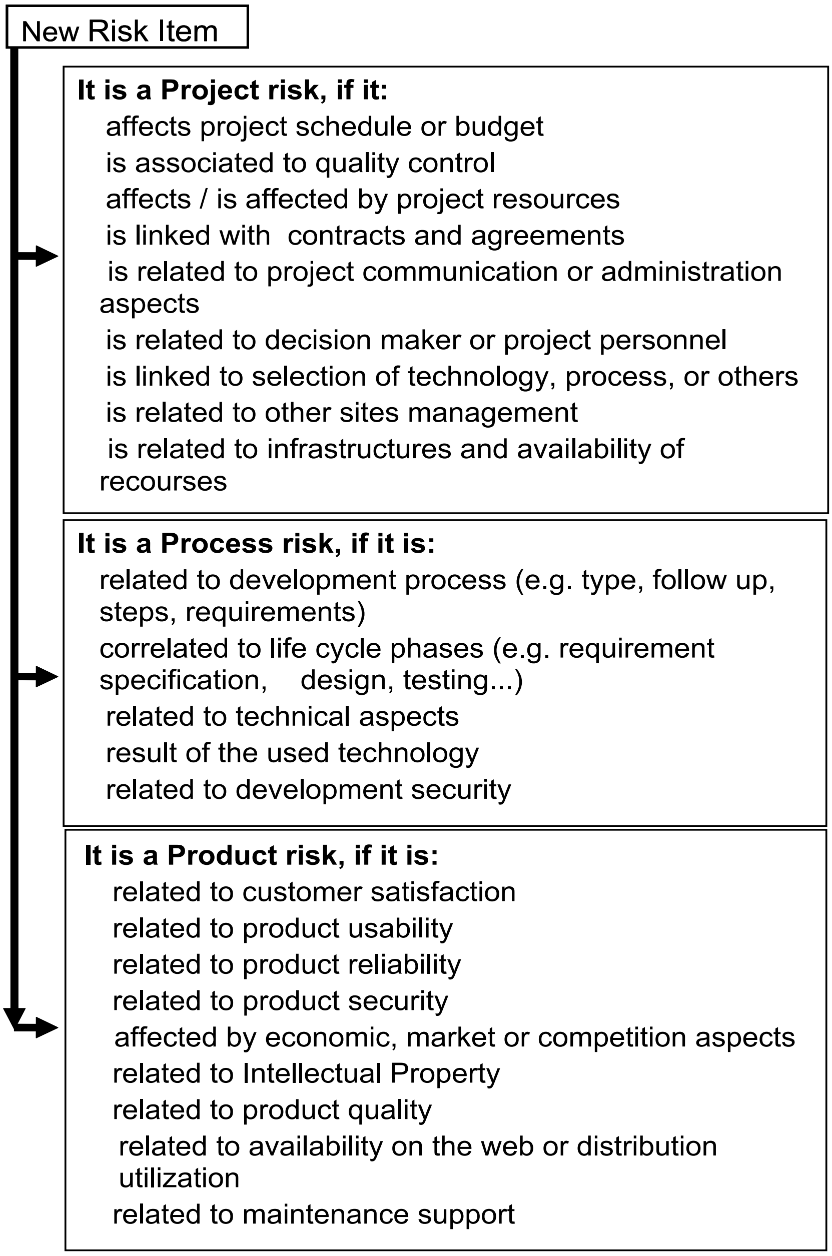

New Risk Item
It is a Project risk, if it:
affects project schedule or budget
is associated to quality control
affects / is affected by project resources
is linked with contracts and agreements
is related to project communication or administration aspects
is related to decision maker or project personnel
is linked to selection of technology, process, or others
is related to other sites management
is related to infrastructures and availability of recourses
It is a Process risk, if it is:
related to development process (e.g. type, follow up, steps, requirements)
correlated to life cycle phases (e.g. requirement specification, design, testing...)
related to technical aspects
result of the used technology
related to development security
It is a Product risk, if it is:
related to customer satisfaction
related to product usability
related to product reliability
related to product security
affected by economic, market or competition aspects
related to Intellectual Property
related to product quality
related to availability on the web or distribution utilization
related to maintenance support

Figure 4.6: Clustering Criteria

because time is critical. The risk repository can be used by all stakeholders, sites, developers and managers and could also be used for statistics and learned lessons. Generally, the data in the risk card are almost fixed data (descriptive data), but they might be updated if there are any changes related to the risk (e.g. controlling strategies). Table 4.8 is an example of a risk card. As can be seen in the table, the risk card contains all essential description data.

Table 4.7: WD Potential Risk - Clustered Based on 3P Clustering Criteria

Project Perspective Potential Risks
Communication Failures
Cross-cultural differences / influence
Lack of Face-To-Face meetings
Poor sites management control
Weak or inadequate contracts
Lack of top management commitment
Constraints due to time zone differences
Instability in other project sites
Unfamiliarity with international and foreign contract law
Failure to manage user expectations
Lack of Management availability and efficiency
Inadequate customer requirement (see and change strategy)
Insufficient project stakeholder involvement
Process Perspective Potential Risks
Poor documentation
Lack of requirement specification
Process instability
Low visibility of project process
Differences in development methodology / process
Complicated development dependencies between project sites
Inadequate process development
Insecure communication channel
Insufficient measurement and estimations
Product Perspective Potential Risks
Unrealistic estimation of the number of users
Poor performance
Lack of security precautions
Poor product functionality
Poor availability
Poor UI
Weaknesses in protection procedures for intellectual property rights
Inadequate user involvement
Vendor feasibility
Market fluctuations
Difficulties in ongoing support and maintenance
Insufficient competence
Unfamiliarity with customer type
Scalability limitations

Estimation Module

Before controlling the risk, it is essential to estimate its impact. There are some techniques that are used to estimate the risk impact (e.g. Risk Exposure). The estimation module in the WeDRisk approach aims to consider the WD factors during risk estimation and offer more than one estimation option to fit the RM type. Therefore, as shown in Figure 4.7, the WeDRisk estimation module offers two options for risk estimation. The first option uses the ordinary Risk Exposure (RE) equation (see equation 2.1) and is supposed to be used for the Plain RM type, whereas the second

Table 4.8: Risk Card Example

Risk ID	R-Cu-011
Risk Name	Not enough experience with web services
Risk Source	Programmer 3
Aspect	Technical Risks
Perspective	Process
Risk Description	The programmer should have enough experience with Java and web services, but he has only experience with Java applications.
Risk Factors	The time is too short to learn web services; Not enough time to hire programmers; Not enough experience.
Potential Impact	Extra Cost (e.g. it costs 300 per a day for any delays)
Potential Affected Areas	Web related aspects
Dependency	All linked sites could be affected
RM plan	Plan Ref. No.: P-Cu-011 Summarry: Fast training course, postponed web service part, changing the type of the application or hire programmer
Primary Precautions Plan	Provide necessary training early Hire extra programmers if the time is short, but if there is enough time and less dependency train the existing programmers.
Card Issue Date	18/11/2012
Risks combination consequence	There is no other risk which has a combination effect with this risk

option uses the Total Risk Estimation Value (TREV) equation (see equation 4.1), which is intended to include the WD Factors (WDF) and is supposed to be used for the Deep RM type. The module uses a special estimation matrix to estimate the total value of the WDF. The two equations and WDF estimation matrix are described below.

Risk Exposure (RE) Equation

RE is a famous equation (see equation 2.1) and has been used for many years to estimate software risks. It depends on the estimation of the probability and magnitude values of the risk [4; 96].

There are different ways (qualitative and quantitative methods) to estimate the probability and magnitude. Although quantitative estimation is much more precise than qualitative estimation, people usually prefer to use qualitative estimation, because they find it much easier. In the WeDRisk estimation module, an attempt was made to mix qualitative and quantitative estimations. For this purpose, a ranked line, as shown in Figure 4.8, is used by the WeDRisk estimation module to estimate the probabilities and magnitudes of the risks. In fact, this is still subjective, but it is

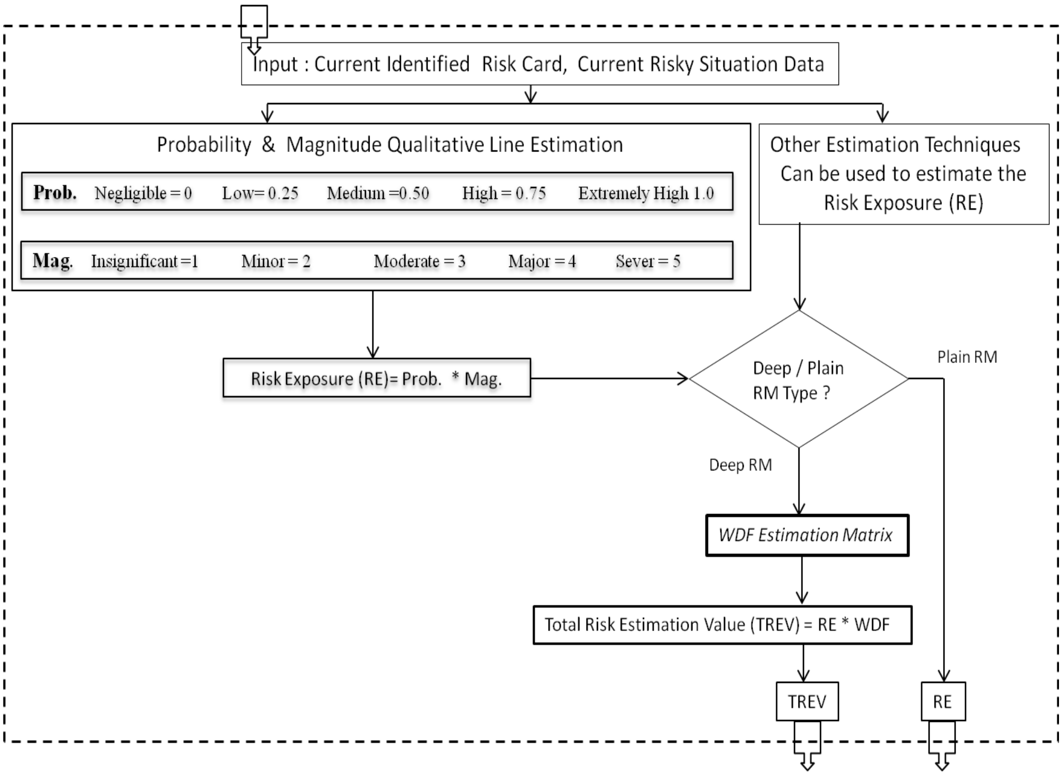

Input : Current Identified Risk Card, Current Risky Situation Data
Probability & Magnitude Qualitative Line Estimation
Prob. Negligible = 0 Low= 0.25 Medium =0.50 High = 0.75 Extremely High 1.0
Mag. Insignificant =1 Minor = 2 Moderate = 3 Major = 4 Sever = 5
Other Estimation Techniques Can be used to estimate the Risk Exposure (RE)
Risk Exposure (RE)= Prob. * Mag.
Deep / Plain RM Type ?
Plain RM
Deep RM
WDF Estimation Matrix
Total Risk Estimation Value (TREV) = RE * WDF
TREV
RE

Figure 4.7: Estimation Module

internally consistent for one manager.

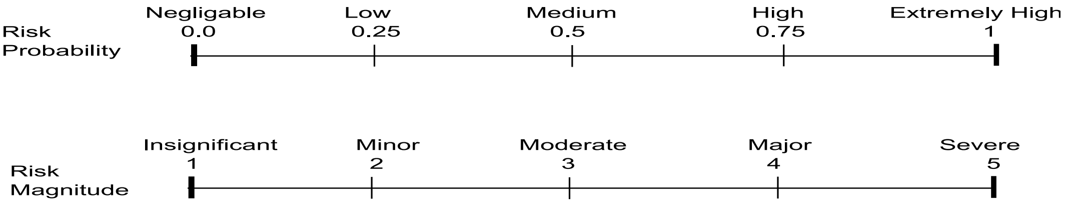


Figure 4.8: Probability and Magnitude Estimation Line

As can be seen in Figure 4.8, the probability ranges from Negligible to Extremely High (*qualitative data values*) with values from 0 to 1 (*equivalent quantitative values*). The values for risk magnitude ranges from 1, when the risk associated loss is “insignificant”, to 5 when it is “severe”. This technique was used for its simplicity and is in fact adapted from a simple technique used in [97].

The RE equation has been used for the assessment of collocated software development since the late 1980s [4; 96]. However, the software industry is an evolving and rapidly growing industry, especially with the new phenomenon of WD software development. Therefore, a new set of factors are involved which could have an effect on the risks and need to be considered in the estimation equations. For WD development risk estimation, the RE equation could be improved by including the WD factors. The TREV is an attempt to produce an improved equation for this purpose with consideration of the WD factors.

To make the probability and magnitude estimation easy and to avoid any subjective and confusing issues, Table 4.9 is designed to help the users to estimate the probabilities and magnitudes of risks and can be used alongside the estimation line. The table is adapted from the Qualitative Risk Analysis [98].

Table 4.9: Probability and Magnitude Estimation Guide

Risk Probability Estimation Guide	
Negligible	Seldom occurs
Low	Unlikely to occur
Medium	Could occur
High	Will probably occur
Extremely High	Will almost certainly
Risk Magnitude Estimation Guide	
Insignificant	Lowest impact on goals and functions
Minor	Would threaten an element of the function
Moderate	Necessitating significant adjustment to overall function
Major	Would threaten functional goals / objectives
Severe	Highest impact on goals and functions

Table 4.10 establishes an example of using the estimation line to estimate risk probability, risk magnitude and risk exposure.

Table 4.10: Prob. and Mag. Estimation Line and Risk Exposure Example

Risk ID	Probability of the risk	Prob. Value	Magnitude of the risk	Mag. Value	Risk Exposure
R32	There is a high chance of the risk occurring but not certain	High = 0.75	Not worth mentioning impact on any of the project aspects	Insignificant = 1	RE=Pro.*Mag RE= 0.75 * 1 = 0.75

Total Risk Estimation Value (TREV) Equation

In addition to the RE value, the TREV equation (equation 4.1) includes the WDF as a part of the question. A special matrix is designed to estimate the total WDF (see Table 4.11). The WDF estimation matrix is described below.

$$TREV = RE * WDF \tag{4.1}$$

Where,
RE is the Risk Exposure and
WDF is the WD Factor

WDF Estimation Matrix

The WDF Estimation matrix (Table 4.11 is an example of the matrix) estimates three

WD factors (Sites Dependency, Sites Distribution and Communication Availability), which could have considerable and changeable effects on risks exposure. For instance, the importance of a risk could be changed in consideration of the dependency level on other sites or risks, even if its RE value is low. Using the estimation matrix, the total WDF value can be estimated using equation (equation 4.2).

$$WDF = \sum_{n=1}^{3}(ColTicksNo * FactorLevel_n) \tag{4.2}$$

Where,
n is the number of WD factors
$ColTicksNo$ is the number of ticks in each column
$FcatorLevel$ is the factor rank during the estimation time

The WD factors and ranking system used in the matrix are described below:

I) *Sites Dependency Level*

In distributed development, the progress of one site could depend on that of another. This means any delay (due to a risk) in that site will affect the other dependent site. This will have the worst effect when there is a large number of sites which depend on each other or there is a cross-dependency between them. Usually, the dependency is not considered in the probability or the magnitude estimations of the risks as the developers do not see the big picture in terms of the relationships between the sites and just deal with the risks individually.

II) *Sites Distribution*

The number and distribution of the development sites has specific influence on the risks in terms of type, number and significance. Multisite projects which have sites in different countries are much more vulnerable to distribution risks (e.g. time zone and cultural differences) than those which are multisite but in one country. Meanwhile, the importance or the impact of a desired risk could differ if there is a high level of sites distribution in terms of number and distance.

III) *Communication Availability*

Communication plays vital role in WD development. Occasionally, availability of communication is unreliable and differs from one situation to another and from one time to another. Therefore, the effect of communication availability on risk importance needs to be considered as a part of risk estimation.

Table 4.11: Example of WDF Estimation Matrix

	Factors Levels				
WD Factors	***1**	***2**	***3**	***4**	***5**
Sites Dependency Level n=1	NO Dependency (D)	Low D. Affects One Node	✔ Medium D. Affects One Node + It is Cross D.	High D. Affects Multi Nodes	Very High D.Affects Multi Nodes + it is Cross D.
Sites Distribution n=2	1 site	✔ more than one 1 site but in the same city	Sites are in different cities but in the same country	Sites are in different countries but on the same continent	Sites are in different cities, countries and different continents
Communication Availability n=3	Excellent 24/7/12 available, excellent history and infrastructure	Good Good History and infrastructure. Very rare to face problems	✔Acceptable The history and infrastructure are fine but there is a very small chance of problems	Bad faces problems from time to time and either the history or infrastructure are bad	Totally Unavailable Currently not available and both the history and infrastructure are very bad
Sub-Totals = (No. of Ticks * Factor Level)		1*2=2	2*3=6		
WDF = $\sum Sub-Totals$	WDF= 2 +6 =8				

The above three WD factors have resulted from a review of the challenges and risks associated with WD development (see Tables 3.4 and 3.5). These factors are selected and considered due to the following reasons:

- The consideration of these factors could change the importance of risk priority.
- These factors are changeable from risk to risk, from situation to situation and from time to time.
- Developers/Managers should not just deal with the WD development risks individually; they should see the big picture for their relationships and dependencies.

Matrix Ranking Technique

As can be seen in the WDF estimation matrix (Table 4.11), based on the degree of effect, the factors' levels are ranked from 1 to 5. The value 1 means the factor has a negligible effect on the risk and 5 means it has the worst effect on the risk. In order to estimate the WDF, one ranking value (1 to 5) is assigned to each WD factor. If the factor is not applicable, a default value of 1 is assigned. In the end, the total of the assigned values are added to each other to obtain the total WDF for the desired risk matrix (see Table 4.11).

Table 4.12 is an example of estimating WD development risk. As can be seen in the table, the previous estimated value of RE in Table 4.10 and the WDF estimated value in Table 4.11 are used to estimate the TREV value.

Table 4.12: Example of WD Risk Estimation Example

Risk ID	Prob. Value	Mag. Value	Risk Exposure	WDF	TREV
R32	High= 0.75	Insignificant= 1	RE= 0.75	WDF= 2 +6 =8	TREV= RE * WDF 0.75*8=6

Evaluation Module

The evaluation module aims to evaluate the estimated risks in order to control them. The evaluation could be based on RE or TREV values. Since there are two types of RM (Plain and Deep), the estimated risks could be mixed (RE and TREV) in the same RM cycle. Therefore, the evaluation module evaluates the risks separately based on the type of RM and estimate equation used. The separation is due to the following considerations:

- The types of RM are different (e.g. if it is Plain type, it means the time could be critical).
- RE and TREV values cannot be mixed as the calculation methods are different; otherwise, the TREV values will be always higher.

The evaluation module as it is exhibited in Figure 4.9 receives the estimated RE or TREV values for the current RM cycle risks with their RM type as input. Atypical Risks (if any) are also input to this module. Then, it prioritizes the risks based on RE and TREV values separately. Finally, and before updating the project card, the module issues an updated statement with current most critical risks which include:

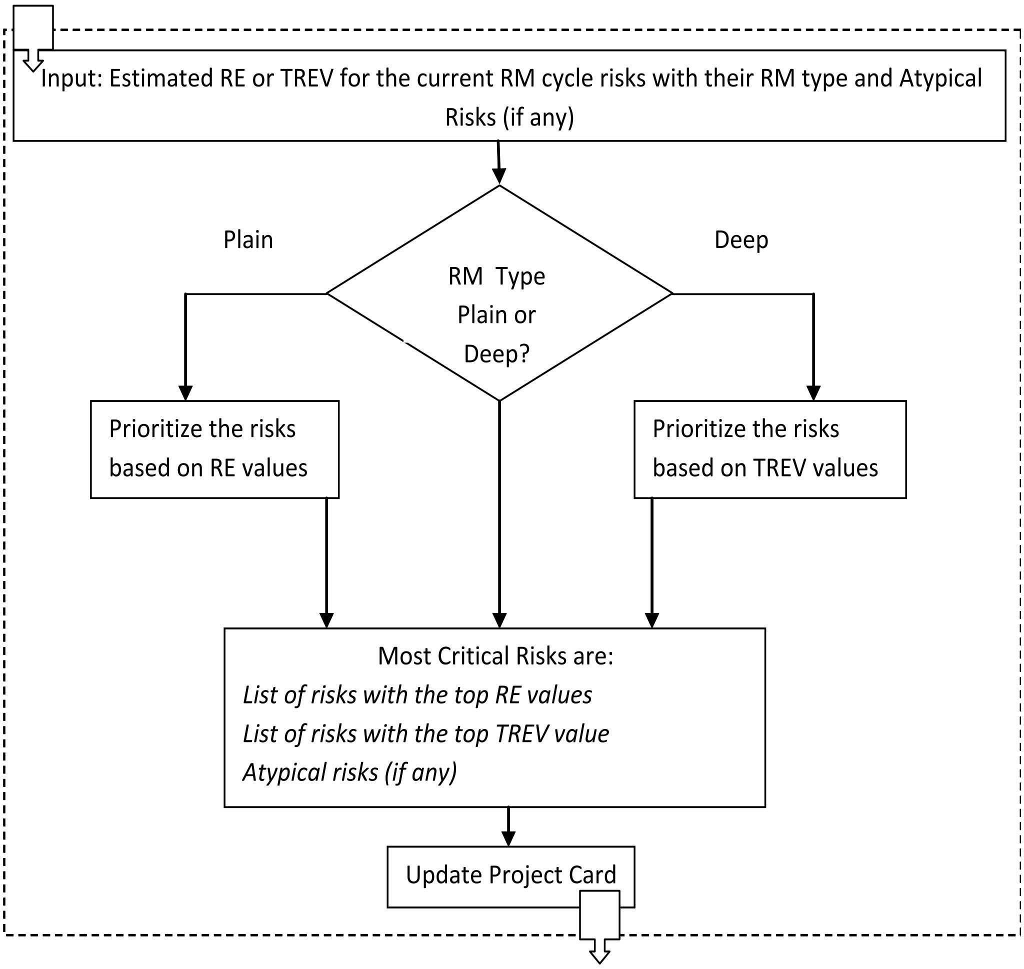

Input: Estimated RE or TREV for the current RM cycle risks with their RM type and Atypical Risks (if any)
Plain
Deep
RM Type Plain or Deep?
Prioritize the risks based on RE values
Prioritize the risks based on TREV values
Most Critical Risks are:
List of risks with the top RE values
List of risks with the top TREV value
Atypical risks (if any)
Update Project Card

Figure 4.9: Evaluation Module

- List of risks with the top RE values
- List of risks with the top TREV values
- Atypical risks (if any)

The atypical risks are always included with the most critical risks and they should be treated as top risks, because usually there is not enough time or information to estimate their RE or TREV and thus they might have higher priorities than others. The remaining identified risks can be accessed through the project card when it is needed, and updated with the evaluation module output. Focusing on the management of the top risks first is intended to save developers/managers time and effort, especially when the resources are limited.

Table 4.13: Example of All Estimated Risks

RM Cycle	Risk ID	Estimation Equation	Estimated Value
W1-C1	R7	RE	2.25
	R32	RE	0.5
	R18	TREV	6.25
	R5	RE	l
	R21	TREV	16.5
	R9	TREV	20.25
	R2	RE	1.5
There are no atypical risks			

Table 4.14: Example of Prioritized Estimated Risks

	Prioritized based on RE		Prioritized based on TREV	
RM Cycle	Risk ID	RE	Risk ID	TREV
W1-C1	R7	2.25	R9	20.25
	R2	1.5	R21	16.5
	R5	1	R18	6.25
	R32	0.5		
There are no atypical risks				

The examples in Tables 4.13 and 4.14 demonstrate how a number of identified risks are evaluated in this module. The first table (Table 4.13) shows all the estimated risks before the evaluation. In this table, all estimated risks are listed randomly without any sorting (mixed from RE and TREV). In the second table (Table 4.14), the risks are prioritized and grouped based on the evaluation equation (RE and TREV).

Atypical Risks Module

An atypical risk (see its definition in Section 1.1) is a risk that cannot be predicted, new, happens suddenly, and of which no-one has any previous experience. Existing risk management approaches are ready to deal with ordinary known risks (those for which estimates of probability and magnitude can be made), but they are unable to act against atypical risks. The nature of an atypical risk is different from an ordinary risk in terms of the fact that:

- it is totally new to the field and no one can expect it;
- it suddenly appears without any indication and could suspend work progress partially or totally.

Any risk, when it happens for the first time and has never happened or been identified by anyone before, is considered an atypical risk; however, thereafter it becomes known and an ordinary risk. Thus, the effect of an atypical risk could be harmful, either because of its high impact or, more commonly, because of the lack of experience of such types of risk. Because of the nature of WD development and its rapid development environment and technologies, it is vulnerable to this type of risk (atypical risk). In this research, existing software risk management approaches were reviewed and this led to the realization that there is a lack of ability to deal with atypical risks. Therefore, the developers/managers either do nothing or might act unsystematically when faced with this type of risk. In this research, the acts were formalized in order to design a module to absorb atypical risks. The introduced module is ready for activation when any atypical risk is faced. As Figure 4.10 illustrates, the atypical risk module has four stages to deal with atypical risks, which are:

- Activating Emergency Plan
- Assessment (quick estimation and evaluation)
- Taking actions based on situation criticality
- Back to normal RM cycle to issue a risk card and consider the atypical risk as an ordinary risk in future risk management cycles

Planning and Controlling Module

Any identified and evaluated risks need to be managed before they become a threat to the development progress (e.g. schedule overrun, low quality or extra cost). The WeDRisk approach provides a planning and control module to manage the identified risks. To maintain the flexibility of the WeDRisk modules, the planning or control sections can be activated individually based on need (see Figure 4.11). For instance, the planning section can be activated early simultaneously with the building of risk cards to include the plans as a part of the risk cards.

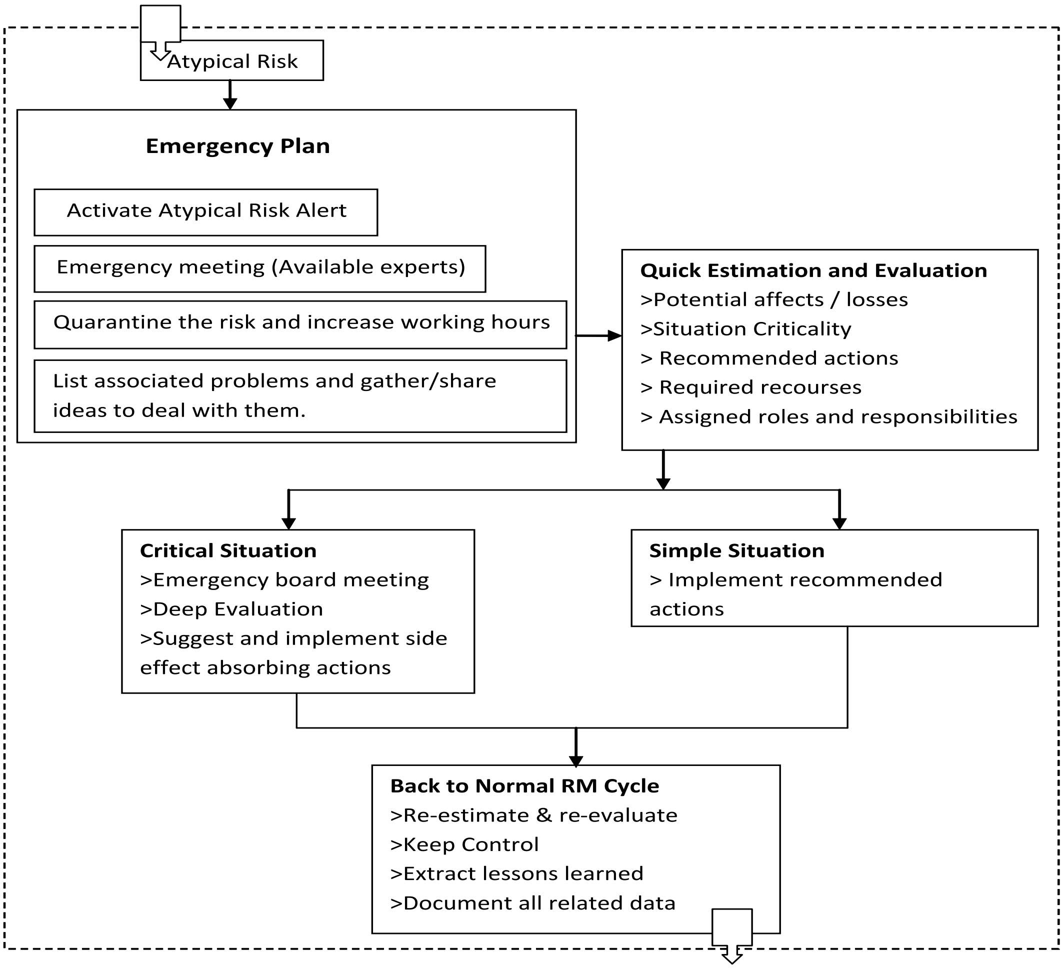

Atypical Risk
Emergency Plan
Activate Atypical Risk Alert
Emergency meeting (Available experts)
Quarantine the risk and increase working hours
List associated problems and gather/share ideas to deal with them.
Quick Estimation and Evaluation
>Potential affects / losses
>Situation Criticality
> Recommended actions
> Required recourses
> Assigned roles and responsibilities
Critical Situation
>Emergency board meeting
>Deep Evaluation
>Suggest and implement side effect absorbing actions
Simple Situation
> Implement recommended actions
Back to Normal RM Cycle
>Re-estimate & re-evaluate
>Keep Control
>Extract lessons learned
>Document all related data

Figure 4.10: Atypical Module

As can be seen in Figure 4.11, the planning section involves two types of plans: precautions and reduction. Precaution plans are simple and could be valid for more than one risk when there are similarities between them. They involve some precautionary measures that are usually taken before the risk has occurred. These precautions are intended to avoid the occurrence of risks before they attack the development perspectives. These precautions should be simple, not costly and be carried out at any time. It is advisable that they are designed early and become available for use quickly. The history of similar risks and development is helpful for the preparation of precautions.

The second type of risk management plan is the reduction plan, which is intended to be used when the risk has already occurred. The reduction plans are designed carefully to control the risks and reduce their impact. These reduction plans consist of a number of steps that are performed systematically when a risk has occurred and tells the user what to do, how to do it, and which resources are required. Experience, brain storming, historical data and learned lessons help to design these plans.

The control section (see Figure 4.11) in the planning and control module is respon-

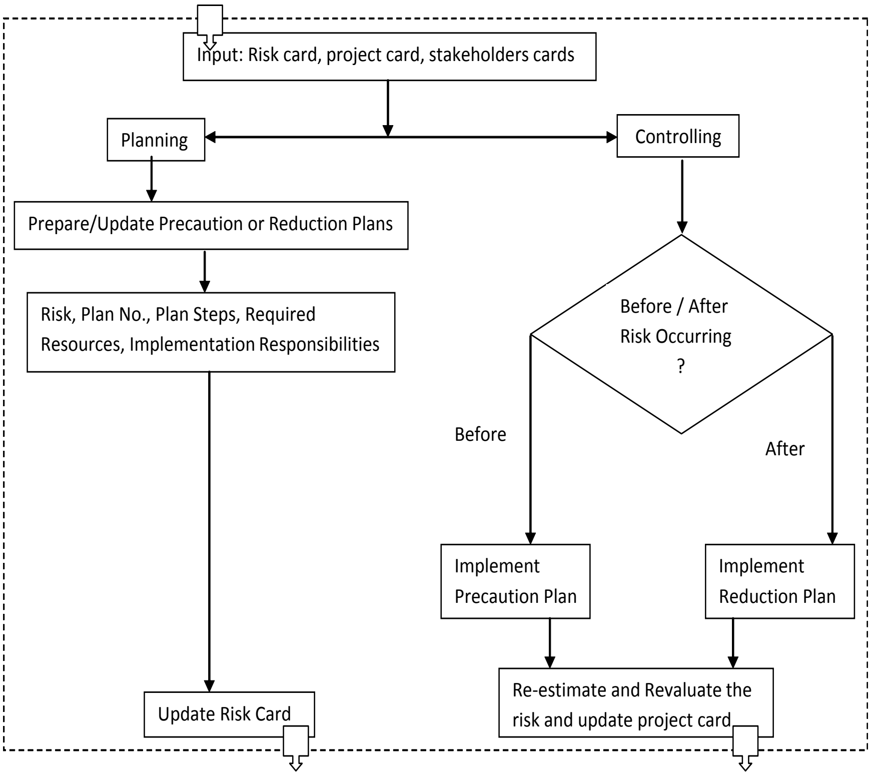

Input: Risk card, project card, stakeholders cards
Planning
Controlling
Prepare/Update Precaution or Reduction Plans
Risk, Plan No., Plan Steps, Required Resources, Implementation Responsibilities
Before / After Risk Occurring ?
Before
After
Implement Precaution Plan
Implement Reduction Plan
Update Risk Card
Re-estimate and Revaluate the risk and update project card

Figure 4.11: Planning and Controlling Module

sible for the implementation of the risk management plans. The precaution plans are implemented before the risks have occurred, but the reduction plans are performed when the risks have already occurred. The control strategy in the WeDRisk approach focuses on the most critical risks first (at the top of the RE, TREV and atypical risk list). The selection of the most critical first is because of their expected higher impact on the project compared with others. This does not mean ignoring the other risks. In fact, all the identified risks must be controlled, but because of the resource availability and limitations, the most critical risk should be controlled first. At the end of any controlling operation, the risks need to be re-assessed and then re-evaluated, and the project card should be updated with the newer results and learned lessons could be extracted as well. Controlling the risks does not mean the end of the risk management process. RM is a continuous operation and new risk management cycles will be conducted until the risk management project is closed.

4.2.3 *Evaluation and Evolution Phase*

As Table 4.15 demonstrates, the RM Evaluation Auditing phase consists of two modules: the RM Evaluation and Auditing Module and RM Evolving Regulator Module. The aim of RM Evaluation and Auditing Module is to monitor the performance of the RM process and to monitor the risks of any desired risk management cycle. Monitoring the risks gives information about the risk threat levels during the RM cycle. This ensures that all the risks are always monitored before and after controlling them. Monitoring of RM processes performance gives information about the efficiency of the RM process in general. The RM Evolving Regulator Module is responsible for making any required improvements or modifications to the WeDRisk approach.

Table 4.15: Evaluation and Evolution Phase

Module	Description	Inputs	Outputs
RM Evaluation and Auditing Module	A module to evaluate the progress of the risk management process which is used for monitoring purposes and taking necessary corrective actions if necessary	Project card, any gathered comments or suggestions	Performance report
RM Evolving Regulator Module	Evolving module is responsible for making any required improvements or modifications to the WeDRisk approach	Performance report	Evolution Plan

Evaluation and Auditing Module

The Evaluation and Auditing module gathers data during the RM cycle via the input component (see Figure 4.12). The input data include project card, auditing evaluation and developers'/managers' comments and suggestions.

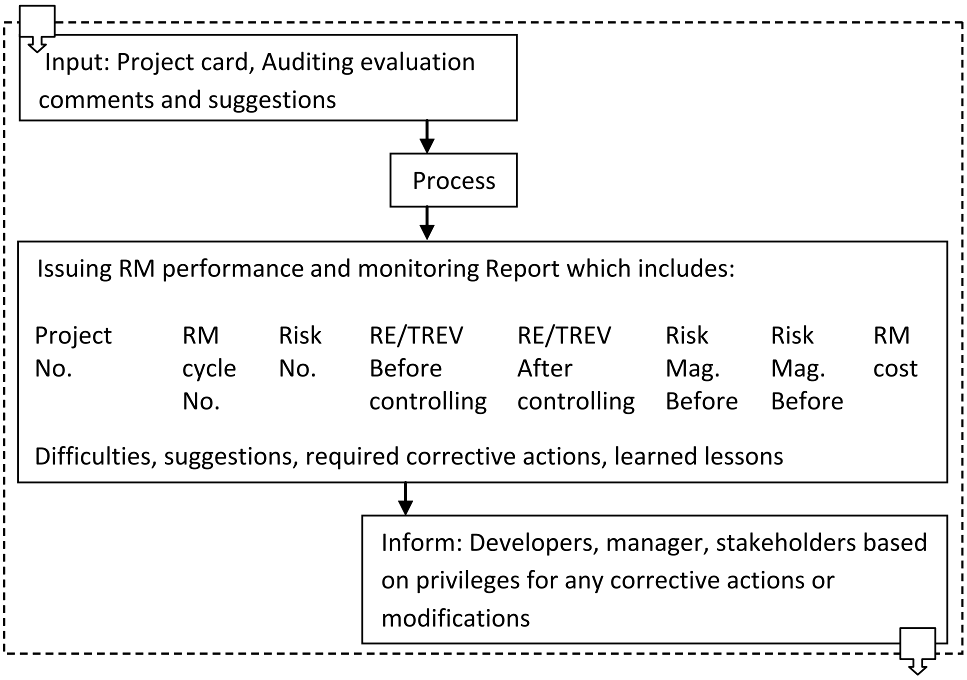

Input: Project card, Auditing evaluation comments and suggestions
Process
Issuing RM performance and monitoring Report which includes:
Project No.
RM cycle No.
Risk No.
RE/TREV Before controlling
RE/TREV After controlling
Risk Mag. Before
Risk Mag. Before
RM cost
Difficulties, suggestions, required corrective actions, learned lessons
Inform: Developers, manager, stakeholders based on privileges for any corrective actions or modifications

Figure 4.12: Evaluation and Auditing Module

The comments and suggestions reflect problems and difficulties that are faced during the RM process and any improvement ideas to enhance the approach or RM process. After inputting the related data, the next step is the processing of the collected data, which leads to producing a RM Performance and Monitoring Report. This report, as shown in Figure 4.12, contains important information about the RM performance, risk situations before and after being controlled, and any suggestions or comments. All of this information is linked with the project, RM cycle and risk numbers. In fact, the report is intended to monitor RM efficiency and also helps to effect any necessary change (corrective actions) to the WeDRisk approach. Finally, the developers, managers and stakeholders involved in the desired RM cycle receive a report based on their privileges, so that they can decide to take any corrective actions or even suggest some modifications and evolvement to improve the WeDRisk approach, to improve the RM process or tackle any weaknesses. Any suggestions in this regard are passed to the RM evolving phase.

RM Evolving Regulator Module

The WeDRisk approach is designed to be ready for any necessary future modification or improvement. It has a special module to handle such modifications, called the RM Evolving Regulator module. As Table 4.15 established, the evolving module is intended to receive improvement and modification needs and suggestions and make the decision to evolve the RM process and WeDRisk approach. The RM Evolving Regulator Module (see Figure 4.13) is responsible for regulating all evolving operations on the WeDRisk.

As input, the module collects all evolution needs and suggestions in a repository called the “Evolution Box”. All of the evolution box contents, including the performance report, are discussed by an evolution approval board. Periodically, this board has scheduled meetings to analyze the contents of the evolution box and decide what sort of evolutions need to be made to the RM process. The evolutions occur as new or modified steps, components and techniques. The board issues an evolution plan (see 26 in Scenario B in the following section) which indicates the implementation priorities, required cost, required resources, affected layer/components, necessary training, responsibilities and implementation schedule. The next step in this module is the implementation and evaluation of the evolution plan, which is carried out by the manager/developer who uses the WeDRisk approach.

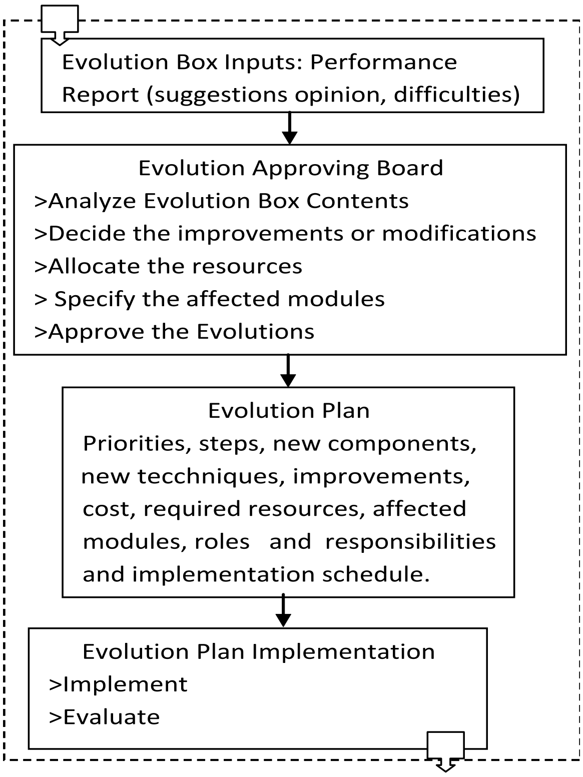

Evolution Box Inputs: Performance Report (suggestions opinion, difficulties)
Evolution Approving Board
>Analyze Evolution Box Contents
>Decide the improvements or modifications
>Allocate the resources
> Specify the affected modules
>Approve the Evolutions
Evolution Plan
Priorities, steps, new components, new teccniques, improvements, cost, required resources, affected modules, roles and responsibilities and implementation schedule.
Evolution Plan Implementation
>Implement
>Evaluate

Figure 4.13: Evolution Module

4.2.4 *Communication Channel*

RM in WD development depends more on communications than RM in collocated software development. This is due to the nature of WD development, where communication plays a vital role in all phases and perspectives of the development. Therefore, the WeDRisk approach supports RM communication via a special channel called the communication channel (see Figure 4.2). The purpose of the channel is to ensure internal and external RM communication and data exchanges during the RM cycle. The communication could be internal communication between the phases or modules, or it could be external communication with the other related approaches or sites. For this purpose, all electronic media can be used. Furthermore, all exchanged data must be documented and controlled based on privileges and permissions. The communication channel provides this support continuously during all RM stages with consideration to security restriction issues.

In the following section, a simple case study is presented to demonstrate how the WeDRisk approach works and deals with the WD risks.

4.3 WeDRisk Running Case study

The same "Security System Case Study" which was used in Section 3.6 is used again in this chapter as a running Case study to illustrate how the WeDRisk approach phases and modules work and integrate to deal with risks. In fact, the "Security System Case Study" is used, because a suitable set of data could not be found in the available literature or historical risk data to fit and run all of the WeDRisk approach modules and features (i.e. most of the available risk data are collocated development risk data). Generally, this case study depends on a simplified WD risk situation, which is used in this research to demonstrate how the WeDRisk approach modules run.

As is illustrated in the case study in Section 3.6, time is a critical issue; therefore, in order to maintain the required quality and development timetable, the software company decided to use the WeDRisk approach to manage any involved risks that might affect the project. The selection of the WeDRisk approach is due to a time criticality issue, the nature of the distributed development environment, the fact that the system is web based, the flexibility that WeDRisk offers, and its ability to consider WD factors.

Case name: *Security System Case Study*

Case actors: *Risk manager; WeDRisk RM Establishment Phase (its modules); WeDRisk RM Implementation Phase (its modules) and RM Evaluation and Evolution Phase (its modules)*

The risk manager has implemented the WeDRisk approach to manage the risks involved in the project. Hereafter is the implementation flow (case flow):

1- The manager decided to start the RM cycle (triggered the first WeDRisk phase) "***RM Establishment Phase***" and ordered the building of a project card.
2- The WeDRisk Project Module built the project card (see the first section in Table 4.16).
3- The manager ordered the Stakeholder Module to build stakeholders' cards.
4- The WeDRisk Stakeholder Module built the stakeholder card (see Table 4.17).
5- The manager decided to specify a suitable type of RM for the current RM cycle. Thus, he triggered the Customization Module for this purpose.
6- The WeDRisk Customization Module activated the RM customization Matrix.
7- The manager applied the matrix to the current situation and ticked the matrix options.

8- The matrix result suggested using the deep RM type (see Table 4.18).
9- The manager accepted the suggested deep RM type and decided to apply it to the current RM cycle.
10- The manager decided to continue with the second WeDRisk phase, the "***Implementation phase***" by triggering the Risks Repository Module.
11- The WeDRisk Risk Repository Module listed the WD potential risks for the current "project" perspective (see Table 4.19).

Here two scenarios are given as examples for the running of the WeDRisk approach which are:

Scenario A: The risk happened for the first time and was then considered an atypical risk. In the "Security System Case Study" it happened suddenly, was not expected and was totally new. The manager named this risk "suddenly losing all highly qualified development staff and then an essential development site becoming out of work". For managing this atypical risk, the manager triggered the Atypical Module and followed its instructions to deal with the risk:

- Emergency plan: (atypical risk alert, held a quick emergency meeting, identified associated problems and extracted some ideas to deal with the situation). As a result, all the sites' developers became aware of the situation and the problem size was identified.

- Quick estimation and evaluation: As a result, it was confirmed that the situation was very critical and affected all the development sites; the losses would be too high if there were a delay of more than one day.

- Critical Action: The company emergency board was called for a meeting and, as a result, it was decided to seek help from some other companies, contact the customers to ask for some extra time, and activate the insurance agreement immediately. Then, the manager resumed the ordinary implementation of the WeDRisk after this risk was resolved and its effects absorbed.

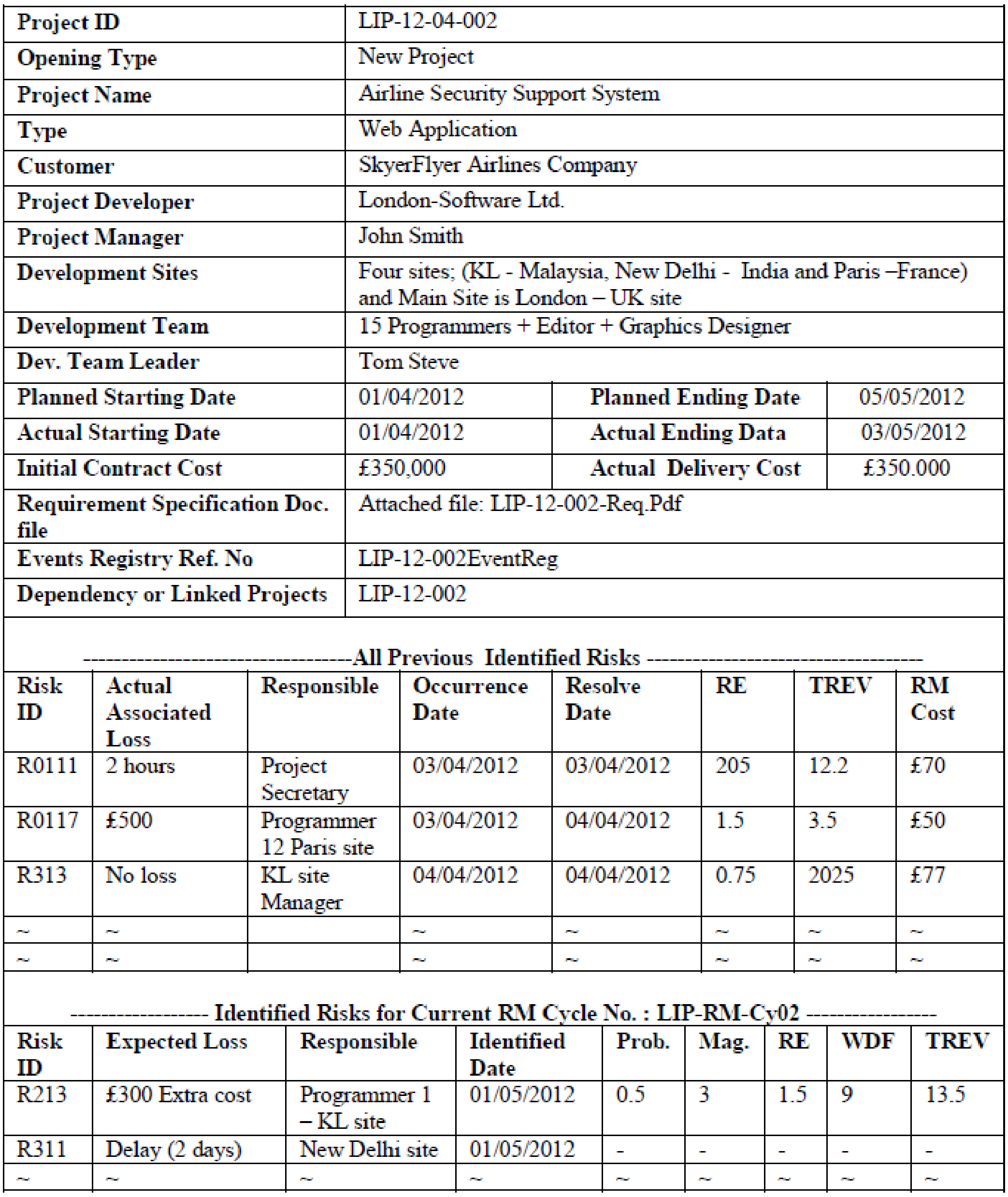
Scenario B: When the manager identified a new risk, the risk was new to the current project and it was an ordinary risk, as it had been identified by others before. In this case, the manger implemented the WeDRisk approach as follows:

12- The manager read the listed risks; however, a new risk was identified and was not listed among the potential risks; therefore, he ordered the Clustering Module to cluster the identified risk.
13- According to the clustering criteria the Clustering Module suggested to add it to the "project" perspective risks list.

14- The manager ordered the Risk Repository Module to build a card for the new identified risk and include it under "project" perspective risks.
15- The Risk Repository Module created a card for the identified risk (see Table 4.20) and then added it to the Risk Repository.
16- The manager decided to estimate the new identified risk, so he triggered the Estimation Module (input data: risk card, RM type).
17- The WeDRisk Estimation Module estimated: Risk Probability, Magnitude, Risk Exposure, WDF and TREV values for the desired risk. This was done by using the estimation line for risk probability and magnitude, RE equation for Risk Exposure, the WDF estimation matrix to estimate the total WDF value, and TREV equation to estimate the TREV value. (see Tables 4.21 and 4.22).
18- After estimation, the manager triggered the Evaluation Module to evaluate the estimated risks. Only two risks were estimated in the current RM cycle. (see Table 4.23).
19- The manager triggered the Planning and Control Module to prepare a management plan and to control the identified risks. Firstly, the management plan was produced (Note: the plan section could be also triggered in conjunction with the building of the risk card and could be updated by or used for other RM cycles).
20- The manager ordered the Controlling section to implement the risk control plan.
21- After each implemented step the project card was updated with the identified risk data. If the risk is still in the management process, it is considered as a current risk, but if it is already controlled it goes into the history list (previously identified risks list).
22- Steps 10 to 21 were continuously repeated till the end of the RM cycle (Note: based on RM customization decisions, the manager might switch between deep or plain risk management types).

23- Now, the manager decided to switch into the third WeDRisk phase, the "***Evaluation and Evolution Phase***" to get a report about the RM cycle performance and to take any corrective actions on the RM processes if necessary.
24- The Evaluation and Auditing Module used the collected data during the RM cycle as input and produced the RM performance evaluation report (see Table 4.24).
25- The manger read the RM performance report and found that the progress of the RM cycle was fine, though some of RM plans needed to be updated to accommodate changes in WD development.
26- The manager triggered the Evolution Module. The required changes were approved by the evolution board and will be ready for use in the next RM cycles.
27- Finally, the software project was submitted two days before the deadline and met the required quality.

Table 4.16: Project Card - Case Study Example



Project ID	LIP-12-04-002		
Opening Type	New Project		
Project Name	Airline Security Support System		
Type	Web Application		
Customer	SkyerFlyer Airlines Company		
Project Developer	London-Software Ltd.		
Project Manager	John Smith		
Development Sites	Four sites; (KL - Malaysia, New Delhi - India and Paris –France) and Main Site is London – UK site		
Development Team	15 Programmers + Editor + Graphics Designer		
Dev. Team Leader	Tom Steve		
Planned Starting Date	01/04/2012	**Planned Ending Date**	05/05/2012
Actual Starting Date	01/04/2012	**Actual Ending Data**	03/05/2012
Initial Contract Cost	£350,000	**Actual Delivery Cost**	£350.000
Requirement Specification Doc. file	Attached file: LIP-12-002-Req.Pdf		
Events Registry Ref. No	LIP-12-002EventReg		
Dependency or Linked Projects	LIP-12-002		

--------------------------------**All Previous Identified Risks** --------------------------------

Risk ID	**Actual Associated Loss**	**Responsible**	**Occurrence Date**	**Resolve Date**	**RE**	**TREV**	**RM Cost**
R0111	2 hours	Project Secretary	03/04/2012	03/04/2012	205	12.2	£70
R0117	£500	Programmer 12 Paris site	03/04/2012	04/04/2012	1.5	3.5	£50
R313	No loss	KL site Manager	04/04/2012	04/04/2012	0.75	2025	£77
~	~		~	~	~	~	~
~	~		~	~	~	~	~

----------------- **Identified Risks for Current RM Cycle No. : LIP-RM-Cy02** -----------------

Risk ID	**Expected Loss**	**Responsible**	**Identified Date**	**Prob.**	**Mag.**	**RE**	**WDF**	**TREV**
R213	£300 Extra cost	Programmer 1 – KL site	01/05/2012	0.5	3	1.5	9	13.5
R311	Delay (2 days)	New Delhi site	01/05/2012	-	-	-	-	-
~	~	~	~	~	~	~	~	~

Finally, Figure 4.14 illustrates the running scenario of the WeDRisk approach phases in general.

Table 4.17: Stakeholder Card - Case Study Example

Project ID	LlP-12-04-002
RM Cycle	LiP-RM-Cy01
Stakeholder ID	LlP-Stk-12-04-002
Stakeholder Name	SkyerFlyer Airlines company stakeholders
Stakeholder Type	Company Owners
Stakeholder Role	Providing necessary support for RM
Stakeholder Privileges	RM Performance Monitorig

Table 4.18: Customization Matrix - Case Study Example

RM Cycle / Situation No.		RM Time		RM Staff Availability		RM Budget Availability		RM Experience		RM Type Decision	
	Risk No.	Enough	Limited	Enough	Limited	Enough	Limited	Enough	Limited	**Plain**	**Deep**
LiP-RM-Cy01	-	✔		✔		✔			✔		✔

If there is any tick under RED go for **Plain** RM type.
If all ticks are under BLACK then go for **Deep** RM.
Exception: If there is a tick under "Enough" RM Experience which is BLUE,(i.e. three years' or more experience in managing related risks) then decisions can be made based on experience.

Notes:
It is not necessary for all fields to be ticked to make the decision. This is based on the available information.
Enough: Means there is sufficient of the particular resource (time, budget or staff) for RM.
Limited: Means there is insufficient (Shortage/ Criticality) of the resource (time, budget or staffs) for the RM operation.

Table 4.19: WD Potential Project Perspective Risks - Case Study Example

Project Perspective Potential Risks
Communication Failures
Cross-cultural differences / influence
Lack of face-to-face meetings
Poor sites management control
Weak or inadequate contracts
Lack of top management commitment
Constraints due to time zone differences
Instability in other project sites
Unfamiliarity with international and foreign contract law
Failure to manage user expectations
Lack of Management availability and efficiency
Inadequate customer requirement (see and change strategy)
Insufficient project stakeholder involvement

Table 4.20: Risk Card - Case Study Example

Risk ID	R311
Risk Name	Suddenly losing all highly qualified development staff and then an essential development site becoming out of work
Aspect	Management
Perspective(s)	Project and Process
Risk Description	Losing all key people (e.g. programmers) and there is a high dependency on the affected site
Risk Factors	The time is too short and critical There is a high dependency between the sites Not enough experience
Potential Impact	Some days delay and extra costs
Potential Affected Areas	The whole project
Dependency	All dependable sites will be affected
RM Plan	Plan Ref. No.: P311 Summary: Allocate some funds for hiring an expert , provide the data sheet
Primary Precautions	Try to build alternative skills early and prepare enough documentation
Controlling Steps	Implement the RM plan
Card Issue Date	01/05/2012
Risks Combination Consequence	There is no other risk that has a combined effect with this risk

Table 4.21: Prob. and Mag. Estimation Line and Risk Exposure - Case Study Example

Risk ID	Probability of the risk	Prob. Value	Magnitude of the risk	Mag. Value	Risk Exposure
R311	There is a good chance of the risk occurring but it is not certain	High = 0.75	Would threaten an element of the function	Minor = 2	RE=Pro.*Mag. RE= 0.75 * 2 = 1.5

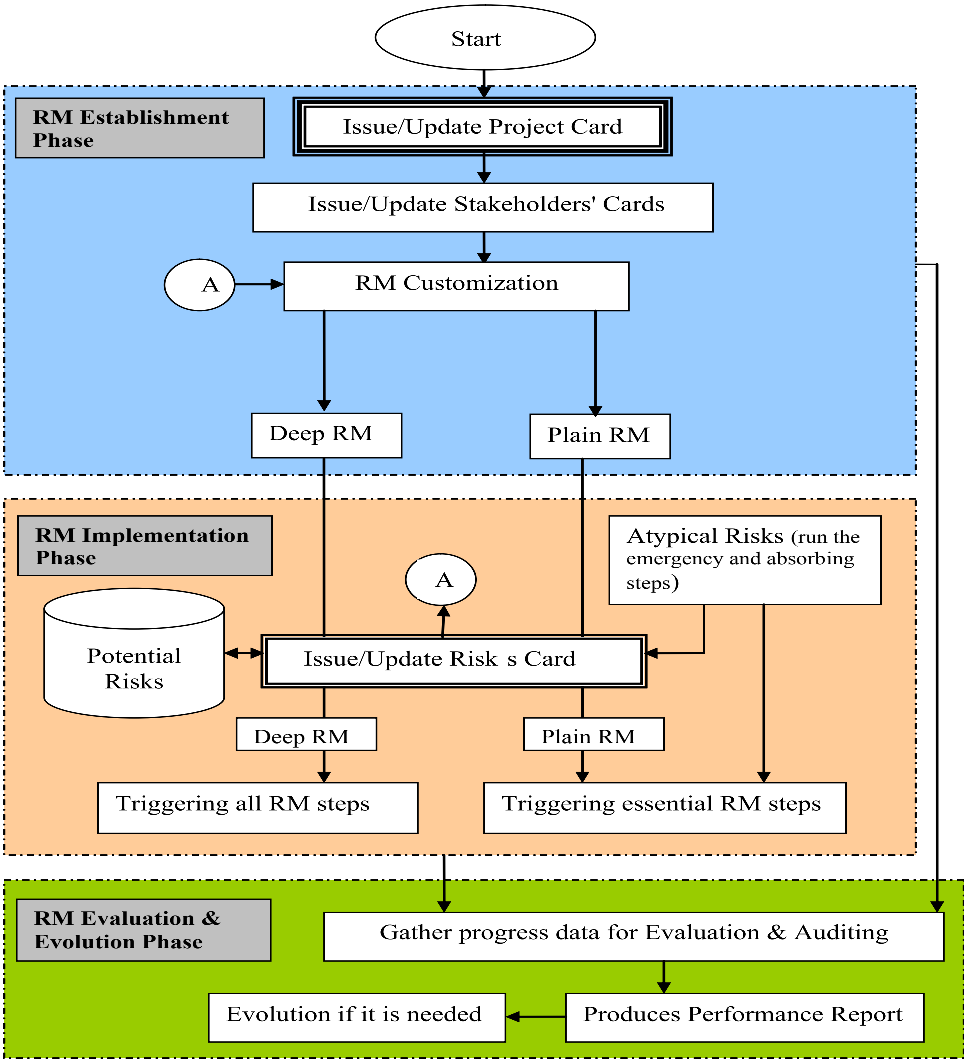

Start
RM Establishment Phase
Issue/Update Project Card
Issue/Update Stakeholders' Cards
A
RM Customization
Deep RM
Plain RM
RM Implementation Phase
Atypical Risks (run the emergency and absorbing steps)
A
Potential Risks
Issue/Update Risk s Card
Deep RM
Plain RM
Triggering all RM steps
Triggering essential RM steps
RM Evaluation & Evolution Phase
Gather progress data for Evaluation & Auditing
Evolution if it is needed
Produces Performance Report

Figure 4.14: WeDRisk Running Phases

Table 4.22: WDF Estimation Matrix - Case Study Example

	Factors Levels				
WD Factors	***1**	***2**	***3**	***4**	***5**
Sites Dependency Level n=1	NO Dependency (D)	Low D. Affects One Node	Medium D. Affects One Node + It is Cross D.	High D. Affects Multi Nodes	✓ Very High D.Affects Multi Nodes + it is Cross D.
Sites Distribution n=2	1 site	more than one 1 site but in the same city	Sites are in different cities but in the same country	Sites are in different countries but on the same continent	✓Sites are in different cities, countries and different continents
Communication Availability n=3	Excellent 24/7/12 available, excellent history and infrastructure	Good Good history and infrastructure. Very rare to face problems	Acceptable The history and infrastructure are fine but there is a very small chance of problems	Bad Faces problems from time to time and either the history or infrastructure are bad	Totally Unavailable Currently not available and both the history and infrastructure are very bad
Sub-Totals = (No. of Ticks * Factor Level)					2*5=10
WDF = $\sum Sub-Totals$	WDF= 0+0+0+0+10 =10				

TREV=RE*$\sum WDF$

TREV $= 1.5 * 10 = 15$

Table 4.23: Example of Prioritized Estimated Risks - Case Study Example

	Prioritized based on RE		Prioritized based on TREV	
RM Cycle	Risk ID	RE	Risk ID	TREV
LiP-RM-Cy01	-	-	R311	15
	-	-	R213	13.5
Atypical risk: Suddenly losing all highly qualified development staff all together and then an essential development site becoming out of work				

Table 4.24: RM Performance Evaluation Report- Case Study Example

Project No.	RM Cycle No.	Risk No.	RE/TREV Before controlling	RE/TREV After Controlling	Expected R. Mag. before RM	R. Mag. after RM	RM Cost
–	–	–	–	–	–	–	–
LlP-12-04-002	LiP-RM-Cy01	R113	13.5	0.5	3	1	One hour work
–	–	–	–	–	–	–	–

4.4 Summary

In this chapter, a new approach called WeDRisk to manage WD development risks is introduced. Many of the specified requirements for the risk management for WD development are addressed by the WeDRisk approach, which is designed to be evolvable, flexible, simple to use and which considers WD development factors. Simple notations were designed and used to represent modules, phases, control flow and components of the WeDRisk approach and to make them simple and self-descriptive. The modular design of the WeDRisk approach allows the approach to produce quicker results, since a minimum number of steps are triggered for each process and they can work semi individually. The WeDRisk approach can easily accommodate any necessary improvements as it is an evolvable approach and has a special module to handle the evolution requests. The approach is provided with a risk repository, which is initiated with the preliminary list of the potential WD development risk cards. WeDRisk uses an improved risk estimation equation to consider the WD development factors that could have a significant effect on the risks. The risk management in the WeDRisk approach is a customisable process as the approach offers two management types (Plain and Deep). This is to encourage the developer/manager to practise risk management under all conditions (i.e. even with time or resource limitations). WeDRisk is ready to deal with atypical risk types. It is supported with a special module to absorb the atypical risks. For more flexibility, the WeDRisk approach is designed to use the minimum available information during the RM customization and RM estimation.

The approach considers and clusters the risks from the three perspectives (project, process and product) and is supported with clustering criteria for this purpose, which aims to save developers time and effort by focusing on particular perspective risks in each management cycle. The auditing end evaluation module in the approach is to evaluate the efficiency of RM operations and monitor the risk situations and suggest the necessary corrective actions. Finally, the WeDRisk approach version which is presented in this chapter came after conducting the evaluation cycles on the approach. In the forthcoming chapters, a description and report on how the novel aspects of the WeDRisk approach are evaluated is given. Different evaluation methods were used, including peer review, two controlled experiments, expert evaluation and a case study.

Chapter 5

Evaluation

5.1 Introduction

This chapter is an introduction to a number of empirical evaluation methods that were used to evaluate the novel contributions of the WeDRisk approach. This chapter describes evaluation through Peer Review; subsequent chapters report in detail on the other evaluation methods (two controlled experiments, expert evaluation and case study). The evaluation results are linked to the research questions and discussed in the discussion and conclusion chapter. Evaluation dimensions include usability, effectiveness, ease of use, coverage of WD factors and simplicity of the approach.

5.2 Evaluation Strategy

Generally, the evaluation of the WeDRisk approach aims to answer the following two questions:
Q1: *Is the WeDRisk approach able to satisfy the research aim (see Section 1.3)?*
Q2: *What sort of improvements can be made to the WeDRisk approach?*

In fact, it is difficult to find one evaluation method that fits all the evaluation needs without constraints or limitations. This is due to many factors such as cost, result generalization, availability of participants and time limitation. Therefore, a number of evaluation methods were involved in the evaluation of the WeDRisk approach, as mentioned previously. These evaluation methods were conducted based on suitability to evaluation needs (e.g. at an early stage, peer review was used), and availability of related resources. Hereafter, the methods used are described briefly.

5.2.1 *Peer Review*

The Peer Review [99] method was used in the early stages of building the WeDRisk approach to collect suggestions and ideas to improve the design of the approach, and to explore the sorts of modifications that could be made to the approach. This method was used because it is easy to conduct, not costly and gives direct and quick results. However, its result is subject to participants' knowledge limitations and it is also difficult to find the relevant people to participate in the evaluation cycles. The idea behind this method is simple and depends on presenting the WeDRisk approach to the participants and then having an open discussion to collect their feedback and suggestions. This is done in the shape of informal interviews with the participants. Most of the peer reviews were conducted in the early stages of building the WedRisk approach. This was helpful for the incremental improvement strategy, which was used for developing and improving the WeDRisk approach.

All of the peer review participants had knowledge or experience in software development, software risk management or software engineering. The peer review cycles were arranged at one-to-one meetings, university presentations, and research committee presentations or international software conference presentations. Table 5.1 summarizes the contributions of the cycles of peer review in evaluating the WeDRisk approach and its modules.

Table 5.1: Peer Review Cycle Summary

Peer Review Cycle	**Contribution to WeDRisk approach**
Informal interview and discussion with researchers and colleagues in the school	Revising and discussing the identified challenges and risks of WD development; refining the initial ideas of developing and evaluating the WeDRisk approach.
Presentation at ICIMP 2010 conferences	Discussing the identified weaknesses of the existing risk management approaches; extracting ideas to build the WeDRisk approach.
Presentation of the work at SRG group meeting in the school	Revising the initial structure of the WeDRisk approach.
Presentation for Software Dependability Group at the best paper ceremony	Reviewing the WeDRisk approach, specifically the structure of the WeDRisk approach (e.g. suggestions regarding the unsuitability of using the word "Layer" and using other names to group the modules of WeDRisk.
Presentation at ICIMP 2011 conferences	Presenting and discussing the result of the first evaluation experiment.
Presentations for the thesis committee	Updating the thesis committee members about the research progress in general and WeDRisk modules in particular and getting comments and suggestions.

5.2.2 *Experiment One*

The aim of this experiment was to evaluate the significance of a list of potential WD development risks and the usefulness of clustering them from the 3P perspectives. The experiment was also used to examine the vulnerability of WD development to atypical risks type and the usefulness of atypical risk absorbing mechanisms. The experiment was designed to test four hypotheses (H1-H4). H1 evaluates the importance of potential risks to WD development. H2, and H3 evaluate the consideration and clustering of the 3P perspectives, and H4 evaluates the atypical risk absorbing strategy. Questions, hypotheses, design, result and other related aspects of this experiment are reported in Chapter 6.

5.2.3 *Expert evaluation*

This evaluation method aims to evaluate the WeDRisk approach using a number of experts in software development from both academia and industry. This evaluation was conducted during the Sixth International Conference on Internet Monitoring and Protection (ICIMP 2011) at St. Maarten in 2011, where a number of software scientists, experts and researchers gathered from around the world to attend the conference. About 10 of them were selected to participate in the evaluation (i.e. their experience and related knowledge met the evaluation aim which is explained to them during the conference time in informal interviews). Seven of them accepted the invitation and participated in this evaluation. The experts evaluated the modules of the WeDRisk individually first and then evaluated WeDRisk overall. The evaluation dimensions included aspects like how WeDRisk is useful, understandable, flexible, easy to use, and how it considers WD development risk factors. The questions, results, findings and related aspects of this evaluation are presented in Chapter 7.

5.2.4 *Experiment Two*

This experiment aimed to evaluate some aspects of the WeDRisk approach, mainly consideration and estimation of WD factors, risk management customization and the atypical risk concept, and the atypical absorbing mechanism. The idea behind this experiment was to present some risky distribution and none distribution situations and then ask the subjects to deal with these situations using the evaluated modules. At some stages of the experiment, and based on the need and nature of the test, the subjects were divided into control and experimental groups. Five hypotheses (H1-H5) were tested in this experiment. The hypotheses were distributed between the modules (i.e. H1, H2, H3 to evaluate the Estimation module, H4 to evaluate the Customization module and H5 to evaluate the Atypical module). Questions, hypotheses, design, results and other related aspects of this experiment are reported in Chapter 8.

5.2.5 *Case Study*

Three software development distributed projects were involved in this case study. The case study aimed to evaluate the WeDRisk initiated potential WD risk list and the significance of risks, and to identify what sort of improvements could be made to the WeDRisk approach.

WeD-RM is a prototype tool that was built based on the WeDRisk approach and used by one of the developers who participated in the case study. WeD- RM supports some major components in estimation, RM customization and risk repository modules. The case study design and its result, as well as a description of the WeD-RM prototype are presented in Chapter 9.

5.3 Evaluation Limitation

Different empirical evaluation methods were involved in the evaluation of the WeDRisk approach. This is due to suitability/unsuitability or limitations of the evaluation methods used in covering some evaluation aspects. However, there are still some limitations with these methods such as:

- Participants' knowledge and background differences, which required extra training in order for them to be equal in terms of the required level of knowledge to evaluate the WeDRisk approach.

- The amount of preparation required to avoid biases and to validate the results.

- Limitation related to the cost and research time.

- The fact that none of the evaluation cycles was able alone to evaluate all the aspects.

- The lack of availability of WD development RM historical data that match the evaluation requirements; thus, extra effort was required in order to emulate them.

5.4 Summary

This chapter is an introduction to evaluation studies that were conducted to evaluate the WeDRisk approach. Five evaluation methods were introduced in this chapter (peer review, two controlled experiments, expert evaluation and a case study). The evaluation methods are reported in detail in separate chapters following this one. Generally, the evaluated aspects of WeDRisk have gained significant support in the empirical evaluation studies. Nevertheless, some weaknesses have been addressed in the approach, which have resulted in improvements to the WeDRisk approach or recommendations for future work. In fact, the WeDRisk approach showed a unique ability to manage atypical risks and consider WD factors and risk management customization. All the evaluation results, which are presented in the forthcoming chapters, are discussed and linked to the research aims, research questions and RM requirements for WD development in the discussion and conclusion chapter.

Chapter 6

Experiment One

6.1 Introduction

In this chapter, the first controlled experiment used to evaluate some aspects of the WeDRisk approach is reported. In this experiment, the importance of a list of WD development potential risks, the usefulness of clustering of WD risks from 3P perspectives, the vulnerability of WD development to atypical risks and the atypical risk absorbing concept are evaluated. The choice of the controlled experiment to evaluate these aspects was due to:

- Suitability and flexibility of the controlled experiment to evaluate the specified evaluation aspects.
- Ability to emulate some conditions and observe some parameters such as the time used.
- Ability to avoid any outside influence on the subjects which could lead to bias.
- Ability to ensure that any clarifications are provided to the participants equally.
- Ability to ensure the exact task implementation sequence during all experiment stages and observe their implementation.

The experiment design, results and analysis, findings and conclusion and limitations are presented in this chapter and an early version of them was published in [23].

6.2 Experiment Scope

This experiment covers the evaluation of the WeDRisk approach, specifically the clustering of risks from the 3P perspectives, the importance of some risks to WD development, and the atypical risks concept.

6.3 Questions and Hypotheses

The experiment aims to answer five questions, which help to evaluate the WeDRisk approach. The questions (Q1-Q5) are:

Question Q1: *What sort of risks should WD developers focus on?*
Question Q2: *Does clustering of risks from 3P perspectives save time and effort in WD risk identification?*
Question Q3: *Do the proposed clustering criteria help developers to cluster the risks with less time and effort?*
Question Q4: *To what extent are WD developments vulnerable to atypical risks?*
Question Q5: *How feasible is the atypical absorbing mechanism?*

Four hypotheses (H1-H4) are designed and tested in this experiment in order to answer these questions. Hypothesis H1 evaluates the importance of potential risks to WD development. Hypotheses H2, and H3 evaluate the consideration of the 3P perspectives and the last hypothesis H4 evaluates the atypical risk absorbing strategy. These hypotheses (H1-H4) are listed below:
Hypothesis H1: *All of the proposed risks are important to WD development and have the same level of importance.*
Hypothesis H2: *If developers use the proposed clustering criteria, then the clustering time of WD risks from the 3P perspectives will be shorter than without the criteria and effort will be saved.*
Hypothesis H3: *Clustering the risks from the 3P perspectives saves time and effort.*
Hypothesis H4: *WD development is highly vulnerable to atypical risks, and absorbing their effects is an effective way to deal with them.*

6.4 Method

The experiment design was inspired by works in [100; 101; 102], especially in terms of the structure of the experiment, preparation of the hypotheses, avoidance of bias, collection, analysis of the data, discussion of the result and description of the experiment. Before conducting the experiment, the design was discussed with an expert from Carnegie Mellon University and other researchers at Newcastle University who provided valuable comments on how to improve the experiment design.

In order to gather more information to improve the design of the experiment, a pilot study was conducted. Three PhD students from the School of Computing Science/Newcastle University participated in this study. The pilot study emulated a similar environment, conditions and measures of the planned experiment and its running scenario, and gathered related information in the shape of errors, problems, comments, observations, suggestions, required time and task implementation flow. Based on this information, modifications and improvements were made to the experiment design and the support material. The modifications included changing the method of recording the time during the experiment, and giving more freedom to subjects in order to reduce the time pressure on them. Other modifications were related to the arrangement and sequence of handling the experiment material and tasks. The pilot study also helped in estimating the required time for each subject to perform the experiment: 30-35 minutes was found to be suitable. The required improvements to the experiment material and measurement were made and then the real experiment was started.

It is made clear to the subjects that the data collected in the experiment are strictly confidential to the experimenter and his supervisor. They are only used for research purposes and not for any other intention. The subjects' contact details were only used for providing free Amazon vouchers as compensation for the subjects' time through the school administration and were discarded afterwards.

6.4.1 *Apparatus*

The apparatus used in the experiment include computer for data entry, ordinary office environment, normal stationery, hard copies of the experiment material, forms (see Appendix A), and a stopwatch.

6.4.2 *Subjects*

30 subjects (male and female) were recruited for this experiment. They were PhD students, researchers and MSc students at the School of Computing Science, Newcastle University-UK. The majority were PhD students or researchers. The subjects were recruited by email; emails were sent to all MSc students, PhD students and researchers at the school and this returned a positive response from about 35 subjects, of whom 30 were chosen. This set of subjects was selected as it was expected that they had enough knowledge or experience with software development and that many of them had participated in software development projects at least as part of their courses. The subjects who performed the experiment were compensated with £10 Amazon vouchers for their time. Instead of using the participants' real names

or numbers, a special reference number was assigned to each so that it could be used anonymously for future research after this experiment.

6.4.3 *Variables*

In this experiment, the dependent variable was the potential risks, whereas the independent variable was the subjects' clustering/searching.

6.4.4 *Measurement Units*

Table 6.1 defines the measurement units (e.g. used time, effort), which are used in this experiment.

Table 6.1: Experiment One Measurement Units

The Unit	Definition	How it is measured
Used Time	The duration of time that is spent to implement a specific task	Difference between starting time and ending time of the task implementation
Effort	The exertion spent to implement a specific task or achieve the goal	Observations, used time comparison, asking questions and number of tries to implement the task

6.4.5 *Generalization and Threats Validity*

Generalization

A number of measures were taken in order to make the sample reflect the real population of real WD development. Mainly, the selected sample was concentrated on a set of subjects who work or have worked in the field of software development, or have attended software engineering courses. The selected sample of subjects is intended to reflect real software projects. However, it would be costly (and impossible) to cover all software development populations in this experiment. Different evaluation techniques were used to evaluate the WeDRisk approach, including case studies, other experiments and expert evaluations in order generalize as much as possible.

Validation

The experiment validity is an important issue to ensure the quality and generalization of findings. Two types of validity are involved in this experiment: internal validity, which is concerned with how the study supports to the findings; and the external, which is concerned with generalization of the results [103; 104]. The threats to internal and external validity are addressed and taken into account as follows:

Internal Validity

Selection: The subjects were randomly assigned to the control and experimental groups.
History: The subjects were selected from the same place; therefore, in order to reduce the influence effect, they were recruited and contacted individually and performed the tasks individually at different times. Moreover, it is difficult to memorise the experiment data.
Motivation: Since the subjects were volunteers and the performed tasks did not take a long time, there was not much concern about boredom or loss of enthusiasm during the experiment.
Time: The time required for the experiment was estimated after conducting a pilot study and the subjects were informed of this when they were recruited. During the experiment, they were told to take enough time to perform the tasks and that they could stop if they were not willing to continue.
Training: A brief description was given to all of the subjects and the necessary clarification and training was provided before each task. Moreover, the subjects were told that they had the right to ask any questions.

External Validity

Subjects: The difficulties of generalizing from students to professionals was taken into account. Therefore, the subjects who work in software projects were mixed (students and researchers). Indeed, the use of students as subjects in this experiment may not threaten the validity of the research since most of the students were PhD students who had experience with software projects and had some professional abilities.

6.4.6 *Subjects and Experimenter Tasks*

Subjects Tasks

In the experiment, the subjects' tasks (see Appendix A) were numbered (from T1 to T11) to make it easy to refer to them during implementation, data analysis and results discussion. All the tasks are described in conjunction with the results in the result and analysis section. The subjects' tasks can be summarized as follows:

- Understanding and performing the assigned roles.
- Specifying the importance of a list of potential WD development risks.

- Clustering the WD risks from the 3P perspectives. For this task, the subjects were divided into control and experimental groups. Control group members perform the clustering operation based on their own knowledge and experience, whereas the experimental group used the WeDRisk clustering criteria to cluster the risks from the 3P perspectives.
- Searching twice for certain perspective risks before and after the clustering.
- Evaluating the vulnerability of WD development to atypical risks and the proposed absorbing concept.

All the subjects were told that they had the right to ask for clarification during the experiment and that they could stop at any stage of the experiment.

Experimenter Tasks

The experimenter tasks can be summarized as follows:

- Distributing the subjects into control and experimental groups (on a random basis).
- Providing the necessary training, experiment related material and required support to all subjects.
- Managing the tasks' sequence during the experiment.
- Collecting the data, observing experiment progress, assigning the tasks and recording used time.

6.4.7 *Avoiding Bias and Control Measures*

Experiments are very sensitive to errors. Many errors could arise due to bias in the experiment. The following measures were taken to avoid and reduce any bias in this experiment:

- Hard copies of the experiment material, essential information, instructions, training and support were provided for all subjects equally.
- The subjects were randomly divided into control and experimental groups.
- On all the data documents, only the subjects' reference numbers were used, rather than the names. This anonymity makes the data analysis more reliable and maintains the subjects' privacy.

- Bias was also avoided in the result analysis. This was achieved by participation of a third party in the data analysis without giving him any information about the subjects.

- The environment was controlled to record time consumption individually, avoid the subjects affecting each other or any other meaning, provide the same level of support, observe subjects' progress during the task implementation, and give the same amount of time and support.

- The sequence of tasks was maintained by the experimenter during the experiment, but if any subject refused to continue at any stage of the experiment then there would not be any effect on other stages, and the data from that implemented stage could be used as part of the experiment result.

6.4.8 *Procedure*

The experiment was begun with a brief description about the experiment aim, phases and the assigned tasks to make the subjects ready for the task implementation. Hard copies of the experiment material were also provided for the subjects' use to avoid bias and conflict and to reduce the need for support during the experiment. Before the subjects started the experiment, they were asked to read and sign the consent form and they were told that they had the right to stop at any time. The subjects were asked to be as accurate as they could during the task implementation with less time pressure. The subjects participated in the experiment individually so that support could be provided easily, parameters monitored individually, and any influence between the subjects during the experiment avoided; therefore, the experiment procedure was repeated according to the number of subjects. In some stages of the experiment and tasks, the subjects were divided randomly into control and experimental groups.

In this experiment, the subjects clustered 36 potential WD risks from the 3P perspectives (with and without using the clustering criteria provided), specified risk perspectives before and after the clustering, and provided feedback about the atypical risk concept and its absorbing mechanism. How these tasks and others were implemented is described in combination with related hypotheses in the results and analysis section (based on the tested hypotheses order). This is to avoid unnecessary repetition.

6.5 Results and Analysis

The data collected from the experiment were in the form of tables and answers to questions. Several tasks (see Appendix A) were designed to test each hypothesis. For this reason, the results and analysis of the data are arranged in order of the hypotheses. In this section, each hypothesis is stated and followed by its related result and analysis.

Hypothesis H1: *All of the proposed risks are important to WD development and have the same level of importance.*

Task T1 in the experiment was designed to test hypothesis H1. In this task, the subjects were asked to evaluate and specify the importance of a list of 36 potential risks to WD development. For this purpose, the Likert scale [105] was used and was scaled from one to five as shown below.

1	**2**	**3**	**4**	**5**
Not Important (**NI**)	Somewhat Important(**SI**)	Important(**I**)	Very Important(**VI**)	Extremely Important(**EI**)

This scale was used because it easy to use and helps to assign weights to the ranks on the scale and then make it easy to calculate the points for each risk.

At the beginning of the experiment, the experimenter explained the task and gave the instructions on how to perform the task. Then, the subjects were asked to specify the importance of each listed risk to WD development, using the Likert scale. The experimenter also answered all related questions raised by the subjects during the task implementation. The subjects voted for the importance of each risk based on their knowledge and experience.

A weighting technique was used to analyze the data. Based on this technique, all the categories (NI, SI, I, VI, EI) were given weights from 1 to 5 respectively. Then, for each risk, the number of votes under each category was multiplied by its matched weight. This gave the number of points for each risk. An example is shown in Table 6.3, where risk number 1 (*Unfamiliarity with international and foreign contract law*) has 110 points, calculated as follows:

***1**	***2**	***3**	***4**	***5**	$\sum Points$
0	4*2	7*3	14*4	5*5	= **110**

As shown in Table 6.2, the minimum number of points could be zero if the number of votes in all the categories is zero. On the other hand, the maximum number of points could be 150 if all 30 subjects placed the risk in the EI category (see Table 6.2).

Table 6.2: Categories Maximum Weights

Category	**NI**	**SI**	**I**	**VI**	**EI**
Weights	***1**	***2**	***3**	***4**	***5**
Maximum No of points	30	60	90	120	150
Minimum No of points	0	0	0	0	0

Table 6.3: Risks Weights and Total of Points

Risk	**Risk Name**	**NI**	**SI**	**I**	**VI**	**EI**	$\sum Points$
No.		**1***	**2***	**3***	**4***	**5***	
1	Unfamiliarity with international and foreign contract law	0	4	7	14	5	110
2	Inadequate customer requirement (see and change strategy)	0	2	7	9	12	121
3	Poor documentation	0	3	12	8	7	109
4	Low visibility of project process	0	7	13	8	2	95
5	Inadequate process development	0	4	13	8	5	104
6	Insufficient measurement and estimations	1	7	8	8	6	101
7	Lack of security precautions	0	1	7	8	14	125
8	Weaknesses in protection procedures for Intellectual Property rights	2	2	12	7	6	100
9	Vendor feasibility	2	6	14	5	3	91
10	Insufficient competence	1	3	6	14	6	111
11	Communication failures	0	1	7	8	14	125
12	Poor sites management control	0	3	10	10	7	111
13	Failure to manage user expectations	0	4	5	10	10	113
14	Insufficient project stakeholder involvement	2	11	4	6	6	90
15	Process instability	1	4	12	10	2	95
16	Poor performance	0	5	7	9	9	112
17	Poor UI	1	6	7	7	9	107
18	Insecure communication channels	3	2	6	8	11	112
19	Lack of requirement specification	0	1	4	10	15	129
20	Inadequate user involvement	0	4	16	8	2	98
21	Difficulties in ongoing support and maintenance	0	3	9	12	6	111
22	Unrealistic estimation of the number of users	1	2	8	13	6	111
23	Differences in the development methodologies and processes	2	11	6	9	2	88
24	Weak or inadequate contracts	0	7	12	6	5	99
25	Complicated development dependencies between project sites	1	7	8	6	7	98
26	Cross-cultural differences / influence	1	13	11	3	1	77
27	Poor product functionality	1	4	4	11	10	107
28	Market fluctuations	2	9	11	3	4	85
29	Scalability limitations	1	1	12	9	7	110
30	Poor availability	0	2	4	10	14	126
31	Lack of top management commitment	0	7	5	10	8	109
32	Instability in other project sites	1	7	15	5	2	90
33	Lack of face-to-face meetings	6	9	10	5	0	74
34	Lack of Management availability and efficiency	0	4	14	7	5	103
35	Unfamiliarity with customer type	0	9	5	9	7	104
36	Constraints due to time zone differences	8	9	8	4	1	71

The above technique is applied to calculate the number of points for all 36 risks. Table 6.3 illustrates the number of votes in each category, along with the total points for each risk.

Generally, Figure 6.1 demonstrates the risks votes distribution for all categories. As can be seen in the figure, most of the votes fall into Important, Very Important or Extremely Important categories. This distribution gives an indication of the importance of risks to WD development in general.

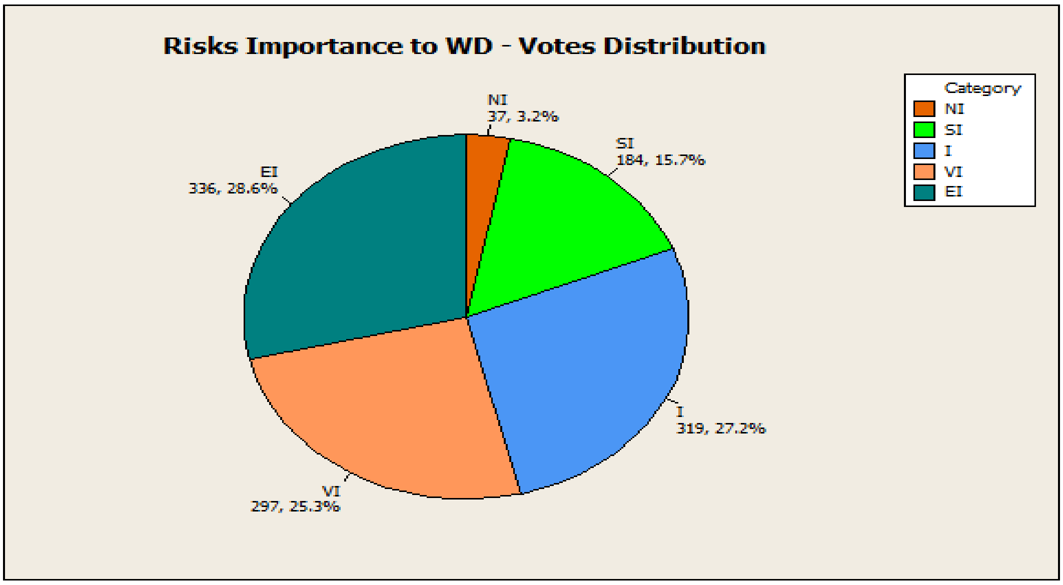


Figure 6.1: Risks Importance to WD - Votes Distribution

The results in Table 6.3 can be read and interpreted form different perspectives (e.g. the most important risks, and the degree of importance). In the risk world, if there is any potential for a risk it means there is a risk and it should be taken into account. Thus, the only risks that could be ignored are those which have 30 points in the Not Important category. The judgement of risk importance is very sensitive as there is no clear border between important or not important. Therefore, rather than saying this risk is important or not important, it is better to rate its level of importance to WD development (using the above weighting and points technique).

Table 6.3 illustrates the number of points for each risk. As can be understood from the table, the majority of risks had a high number of points when they were tested against the importance to WD development (see also Figure 6.2).

Returning to hypothesis H1, it can be seen that the result of this experiment shows that the list of potential risks is important to WD development, but with a different degree of importance for each risk (e.g. Risk number 19 has the highest number of points which is 129 points, whereas risk number 36 has the lowest points, at 71 points).

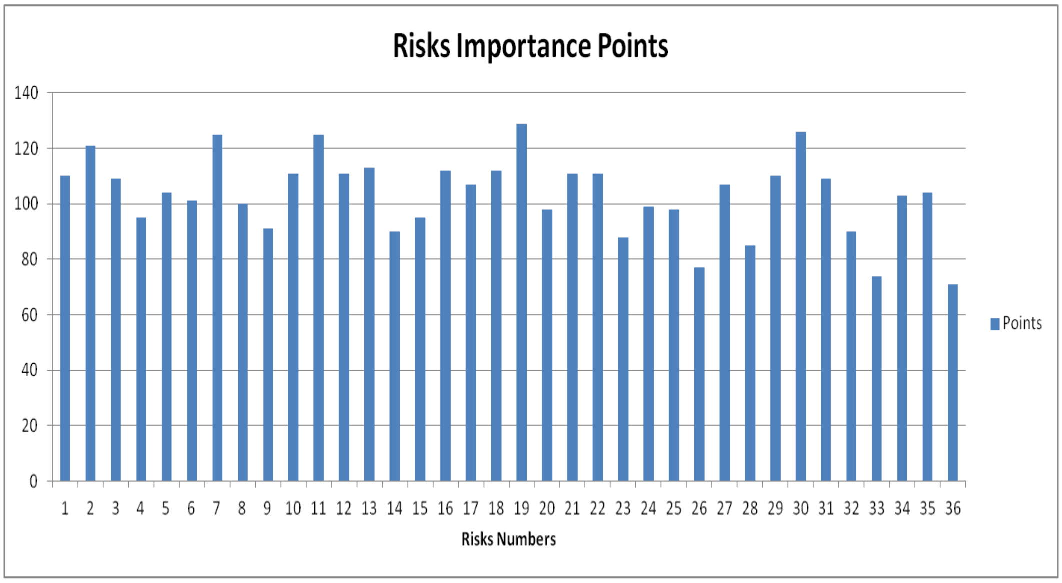


Figure 6.2: Risk Importance Points (Risk Numbers refer to risk names in Table 2.2)

Hypothesis H2: *If developers use the proposed clustering criteria, then the clustering time of WD risks from the 3P perspectives will be shorter than without the criteria and effort will be saved.*

In order to test hypothesis H2, the subjects were randomly divided into "control" and "experimental" groups (15 in each). The subjects of both control and experimental groups were asked to cluster the potential WD risks (Please see Table 2.2 for the potential WD risk list) from the 3P perspectives. The control group completed their task without any clustering criteria, whereas the experimental group used the WeDRisk 3P clustering criteria (see Figure 4.6). During the implementation of this task, the experimenter recorded the time used. The experiment result showed that there was a significant difference between the time taken by the two groups, shown by the Mann-Whitney U statistical test at **(p-value = 0.0079, U = 168.0)**. This test was used, because it is suitable for small samples [106]. The total time used by control group subjects for the clustering (**56 minutes**) of the risks was less than that used by the experimental group (**108 minutes**).

Hypothesis H3: *Clustering the risks from the 3P perspectives saves time and effort.*

The difference in time and effort used between the groups using clustered and non-clustered risks was used to test this hypothesis. The data were collected during different tasks (T2, T3, T7, T8, T3, T4 and T9), which are described below. The used time data were obtained from tasks T2, T3, T7 and T8, whereas answers to some questions in task T4 (Q2 and Q3) and task T9 (Q1) were used to evaluate the effort. In order to avoid any bias influence, tasks T7 and T8 were performed separately (in both time and sequence) from tasks T2 and T3. The following sections describe how the data on time and effort used were obtained and evaluated:

I) Used Time: Used time is a preliminary indication of whether the task is easy, difficult or complicated. Task T2 and task T7 were the same, except that task T2 was on non-clustered risks and task T7 was on pre-clustered risks. In these two tasks, the subjects were asked to **specify two risks for each of the three perspectives**.

Task T3 and task T8 were also the same but task T3 was on non-clustered risks and task T8 was on pre-clustered risks. In these two tasks, the subjects were asked to **specify the perspectives for three pre-ticked risks by the experimenter**. While the subject was implementing the tasks (T2, T3, T7 and T8) they were monitored, and the time taken was recorded. Table 6.4 shows the total used time during tasks T2, T3, T7 and T8. The illustrated values in Table 6.4 are for the time that was used by the subjects for both non-clustered and pre-clustered risks. As shown in Table 6.4, the subjects spent less time with the pre-clustered risks compared with non-clustered risks for the above tasks, suggesting that clustering from three perspectives reduces the required time for dealing with the risks.

Table 6.4: Total Used Time for Tasks T2,T3,T7 and T8

Non Clustered Risks		Pre-Clustered Risks	
Task No.	$\sum UsedTime$	**Task No.**	$\sum UsedTime$
T 2	129.32m	T 7	42.13m
T 3	27.16m	T 8	8.29m

II) Effort: It is not easy to evaluate effort; therefore, a set of questions was used which was designed and distributed among the tasks in a specific order to collect the subjects' feedback. This included opinions about the usefulness of clustering the risks from the three perspectives.

For this purpose, and after performing tasks T2 and T3 on non-clustered risks, the subjects were asked "**Was it easy for you to specify the risks or perspectives?**" This question was answered by **29** participants. **16** of them, which is more than **(55%)**, answered **No** to this question and **13** answered **Yes (44.8 %)**.

The subjects were asked another question after performing these two tasks, "**Do you agree with the idea that the above tasks would be much easier and that time and effort could be saved if risks were clustered from the three perspectives?**" with 5 options (**Strongly Agree, Agree, Neutral, Disagree** and **Strongly Disagree**) to select from them. As shown in Figure 6.3, the number of subjects who voted **Strongly Agree, Agree, Neutral, Disagree and Strongly**

Disagree was **4, 22, 3, 0, 0** respectively. This means that the majority of the subjects were in agreement (or strong agreement) in their answers to this question.

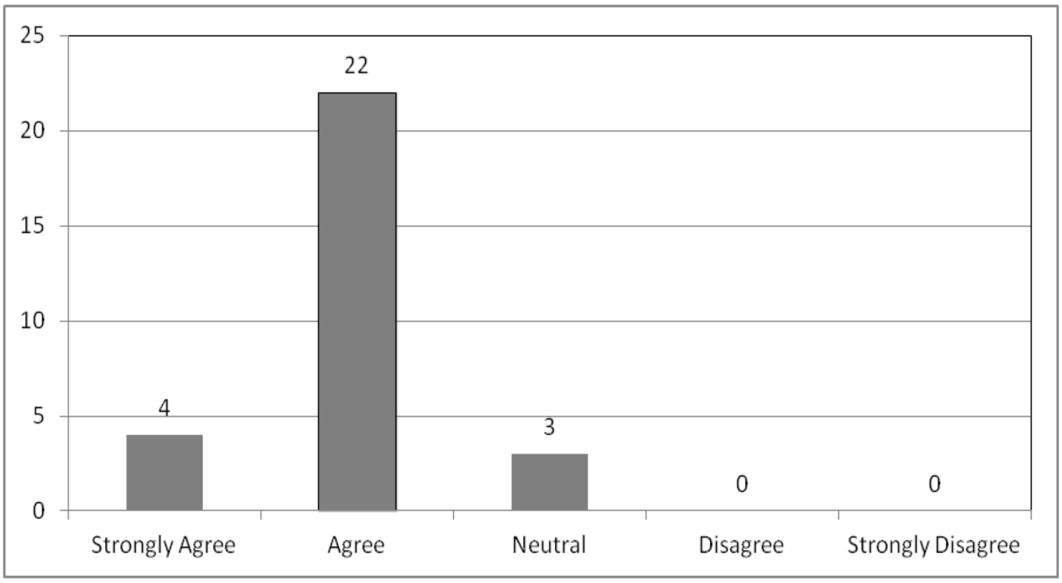


Figure 6.3: Voting for Used Effort (Clustered and Non-Clustered)

On the other hand, after performing tasks T7 and T8, the subjects were asked the same questions, but in a different way: "**To what extent do you agree with the idea statement that concentrating only on the risks of the appointed perspective saves time and effort?**" with the same options to answer this question (**Strongly Agree, Agree, Neutral, Disagree and Strongly Disagree**). Only two subjects disagreed. The rest of subjects agreed (including those who strongly agreed), (see Figure 6.4).

As shown in the figure, the number of answers on the "agree" side (**Strongly Agree and Agree = 20**) is higher than the "disagree" side (**Disagree and Strongly Disagree = 2**), with **6** subjects answering **Neutral**. This means that the idea of "concentrating only on the risks of the appointed perspective to save time and effort" has strong support from the subjects in the experiment.

Support for the Pre-Clustered list:
As a part of the experiment, the pre-clustered list of the WD development potential risks was compared with the clustering of the two groups (control and experimental). The results, which are summarized in Table 6.5, will be used to improve the pre-clustered risks. It was found that some had categorized some risks into more than one perspective.

As can be seen in Table 6.5, the clustering of risks has higher support from both the control and experimental groups. In total, 66.6% of the clustering of risks from the 3P perspectives were supported, and only 27.7% were not supported. This is

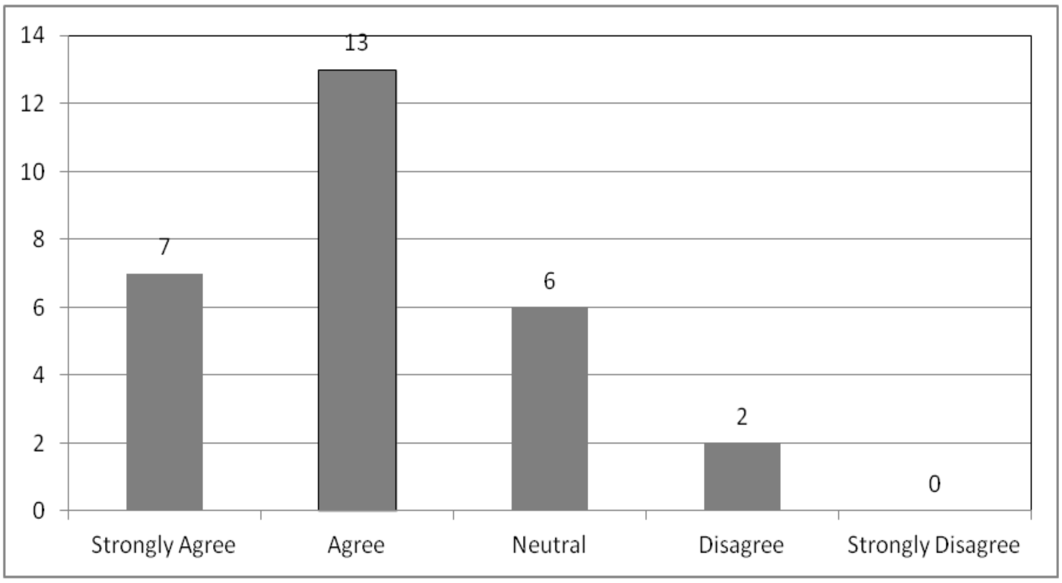


Figure 6.4: Concentrating on appointed perspective risks saves time and effort

Table 6.5: Support to Pre-Clustered risks list

	Supported by			
Risk perspective	**Both Groups**	**Control Group only**	**Experimental Group only**	**No One**
Project	10/13	-	1/13	2/13
Process	4/9	-	1/9	4/9
Product	10/14	-	0/14	4/14
Totals	24/36	0	2/36	10/36

understandable from the numbers shown in Table 6.5. The clustering for the project and product risks has stronger support from the groups. For the clustering project risks, only 2 risks out of 36 were not supported by the groups. The pre-clustered list for the product risks was also strongly supported by the groups as only the clustering of 4 risks out 14 was not supported. By contrast, the support of process risks was medium as the clustering of 4 risks out of 9 was not supported.

Hypothesis H4: *WD development is highly vulnerable to atypical risks, and absorbing their effects is an effective way to deal with them.*

Firstly, the experimenter briefly described the atypical risks' nature and clarified it with some examples; then the subjects were asked to implement tasks T10 and T11, which are designed to test hypothesis H4. In task T10 (below), the subjects were asked to rate the vulnerability of WD development to atypical types of risk. The subjects implemented the task based on their experience with WD development and understanding of the atypical risk concept.

Task T10: Atypical risks are new risks and can not be predicted or imagined before they occur. To what extent do you agree that WD development is vulnerable to *such types of risk?*

Strongly Agree	**Agree**	**Neutral**	**Disagree**	**StronglyDisagree**
1	**2**	**3**	**4**	**5**

The subjects rated the vulnerability of WD development to atypical risks. As can be understood from Figure 6.5, the majority of subjects chose either Strongly Agree 33.3% (10/30) or Agree 43.3% (13/30) in answer to this question. In contrast, very few subjects chose Neutral 6.6% (2/30) or Disagree 10% (3/30).

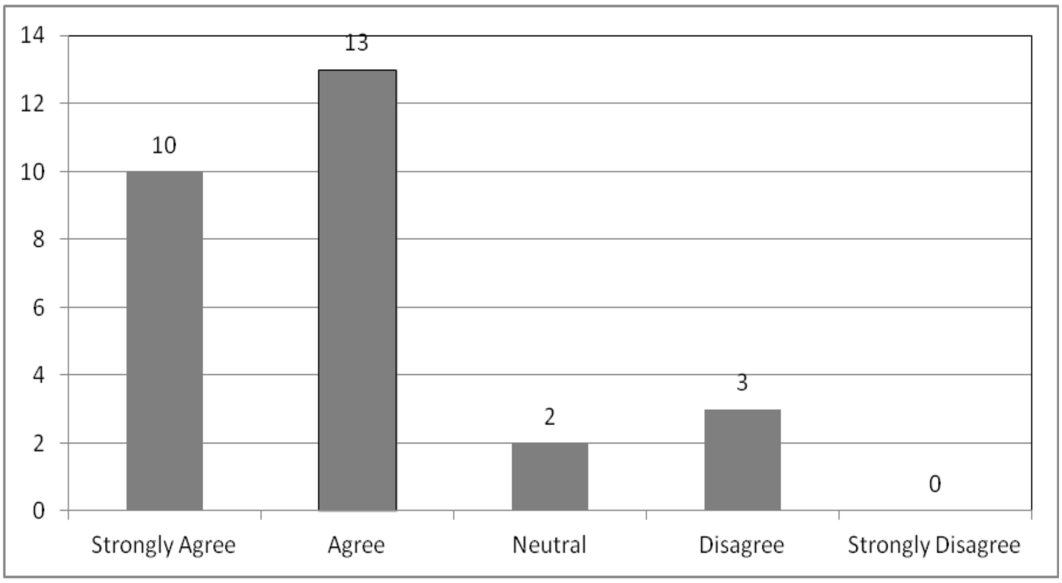


Figure 6.5: Vulnerability to Atypical Risks in WD Development

In task T11, the experimenter introduced the concept and components of the WeDRisk module to absorb the atypical risks, and then the subjects were asked to rate the usefulness of the absorbing strategy to deal with the effects of atypical risks.

Task T11: Due to the lack of information, it is difficult to expect atypical risks or even to be ready for them. Therefore, the WeDRisk approach attempts to absorb their effects and then consider them in future risk management cycles. How do you rate the usefulness of this strategy?

Not Useful	**Somewhat Useful**	**Useful**	**Very Useful**	**Strongly Useful**
1	**2**	**3**	**4**	**5**

Do you suggest any other useful strategies to deal with atypical risk types? Please give details: ——————————————————————————

Figure 6.6 illustrates the results of task T11. As can be seen in the figure, only 3.3% (1/30) of the subjects selected **Not Useful**, and 10% (3/30) chose **Somewhat Useful**, whereas the remaining subjects voted them **Useful** 36.6% (11/30), **Very**

Useful 33.3% (10/30) or **Strongly Useful** 13.3% (4/30). This means the concept of absorbing atypical risks received strong support from the majority of the subjects. Regarding the second part of the question, there were no significant suggestions to deal with atypical risks other than absorbing their effects.

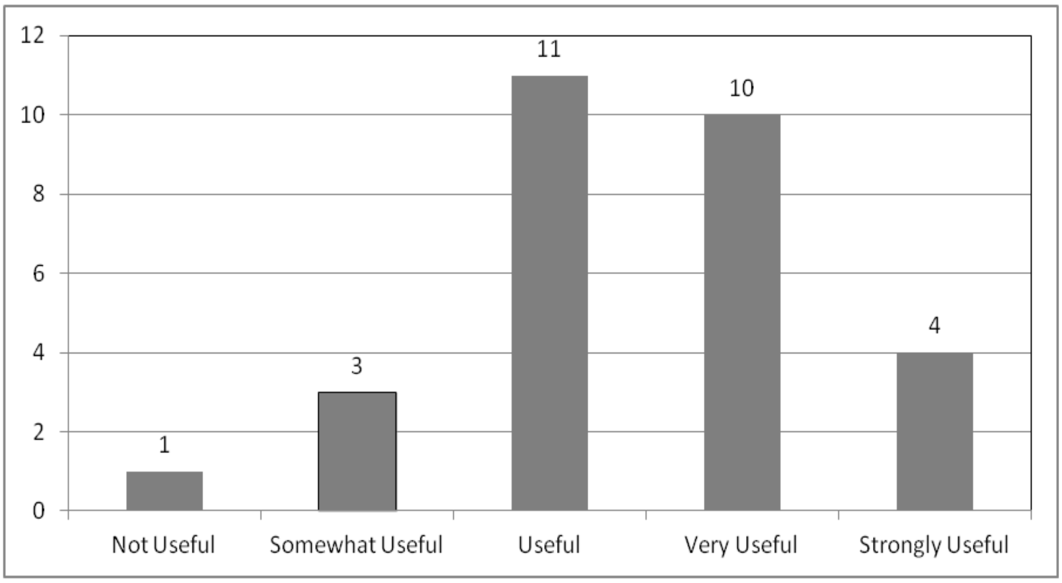


Figure 6.6: Usefulness Of Absorbing Strategy With Atypical Risks

From the experiment results, it can be understood that WD development is vulnerable to atypical types of risk and that the absorbing mechanism could be a useful strategy to deal with them, which supports hypothesis H4.

6.6 Study Limitation

It would have been preferred if the experiment had been undertaken at one of the software development houses, but this was not feasible as most software companies have restrictions concerning data security. Indeed, several local web development companies were contacted, but were unable to participate.

6.7 Discussion

For simplicity and consistency purposes, the hypotheses are discussed one by one hereafter, in the same order that they were introduced in the previous sections.

Hypothesis H1: *All of the proposed risks are important to WD development and have the same level of importance.*

The Likert scale was used to evaluate the importance of 36 risks (see Table 2.2) to WD development. The subjects specified the importance of each risk to WD development, ranging from Not Important to Extremely Important. A weighting technique was used to categorize and analyze the resulting data. The number of points for each risk relates to the importance of all of the listed risks for WD development, supporting the first part of hypothesis H1; however, the hypothesis assumption that each risk has the same degree of importance to WD development is not supported. This means that none of the risks can be ignored, but more attention could be given to some of them based on development perspectives and surrounding conditions (i.e. the importance of risks could change from one situation to another).

Hypothesis H2: *If developers use the proposed clustering criteria, then the clustering time of WD risks from the 3P perspectives will be shorter than without the criteria and effort will be saved.*

The result of the experiment does not support hypothesis H2. This is clear since the time used by the experimental group was higher than that of the control group. This could be due to the following reasons:

- Actual time for reading the criteria (it is zero in the case of the control group).
- Poor criteria design or difficulty in understanding the criteria.
- Lack of standard criteria, meaning questions could be raised about the control group clustering and answers are subject to the participants' knowledge and experience.

More training and improvement to the criteria will make them much easier to understand and use.

Hypothesis H3: *Clustering the risks from the 3P perspectives saves time and effort.*

Trying to manage all perspectives' risks together wastes developers' time and effort or could lead them to locate more resources for one perspective's risks and ignore others. The WeDRisk concept considers the risks from the 3P perspectives and clusters them from the 3P perspectives to focus on the risks of a particular perspective each time to save time and effort. Therefore, proposed clustering criteria to cluster the risks from the 3P perspective are evaluated in terms of their ability to save time and effort. The results support the concept of managing the WD risks from the 3P perspectives since they overlap more in WD development. The experiment results show that the clustering has a significant impact by reducing the search time and effort. This results in the difference of time as the subjects have spent less time and effort with the pre-clustered risks compared with non-clustered risks.

As a by-product of this experiment, the pre-clustered risks list, which is clustered using the proposed criteria and used in this experiment, was tested for precision. The test results demonstrated that the list gained significant support from the control and experimental groups in general. However, the clustering of some risks was not supported by the groups and some subjects argued that some risks could affect more than one perspective (shared risks).

Hypothesis H4: *WD development is highly vulnerable to atypical risks, and absorbing their effects is an effective way to deal with them.*

In order to test this hypothesis, the experimenter firstly introduced the atypical risk concept and its nature and provided some examples; then, the subjects expressed their level of agreement (from Strongly Agree, Agree, Neutral, Disagree and Strongly Disagree) with the statement that WD development is vulnerable to atypical risks. The resulting data demonstrate that most of the subjects chose either Strongly Agree or Agree for this statement. This reflects how WD development is vulnerable to atypical risks and implies support for hypothesis H4.

6.8 Summary

In this chapter, the first controlled experiment to evaluate some WeDRisk aspects was described and reported. It was designed to evaluate the importance of a list of potential WD development risks and their degree of importance, as well as to consider the risks from the 3P perspectives (i.e. proposed 3P perspective clustering criteria). The experiment was also used to evaluate the vulnerability of WD development to atypical risk types and the proposed absorbing concept. This experiment was undertaken at the School of Computing Science/Newcastle University, UK. There were 30 recruited subjects (MSc, PhD and Post-doctoral researchers) who either had experience and had worked in software development projects, or at least had an appropriate knowledge of software engineering and software development. During some stages, the subjects were divided into experimental and control groups (based on the evaluation needs). The results of the experiment highlighted the following points:

- The listed risks to WD development are all important to varying degrees.
- The clustering of risks from the 3P perspectives gained a high degree of support from the subjects. It was effective in saving time and effort, since the subjects spent less time and effort searching in the pre-clustered risks compared with non-clustered risks.
- It seems that using the proposed 3P perspectives clustering criteria takes more time than clustering without the criteria. This could be due to the time taken to read and understand them. Nevertheless using the clustering criteria could be more useful because:

- they help to avoid subjective and uncertain decisions;
- in the long run, using the criteria is expected to take a shorter time as the developers become more familiar with them; and
- clustering without the criteria needs enough experience and it is not expected that all the developers have that.

- The subjects were in agreement with the vulnerability of WD development to atypical risk types.

Generally, the results of the experiment confirmed the importance of the list of potential WD risks, but with different degrees of importance. Thus, all the potential risks should be monitored. The result also confirmed the importance and usefulness of the clustering and considered the risks from the 3P perspectives as a way to save time and effort, thus increasing the efficiency of risk management. However, it highlighted the need for updating and improving our proposed 3P perspectives clustering criteria to make them more understandable and less time-consuming (more training on the criteria could also help to reduce time consumption). On other hand, the experiment result was used to revise and rectify the pre-clustered risks list, since the result illustrated that some risks could affect more than one perspective. The subjects were in agreement with the vulnerability of WD development to the atypical risk types and the importance of the absorbing concept in dealing with them. Finally, the experiment was designed to be ready for replication in the future if necessary. The collected results help to improve different aspects of WeDRisk, which were implemented and re-evaluated in other empirical studies and are reported in the following chapters.

Chapter 7

Expert Evaluation

7.1 Introduction

This chapter reports the expert evaluation [107], which is used to evaluate the main modules of the WeDRisk approach and the approach overall. This evaluation was conducted after making the improvements which resulted from the first controlled experiment. This expert evaluation for the WeDRisk approach took place on the sidelines of the Sixth International Conference on Internet Monitoring and Protection (ICIMP2011) at St. Maarten 2011. The researcher presented a paper and attended the conference; therefore, this chance was exploited to evaluate the WeDRisk approach. Seven international experts from academia and industry were involved in this evaluation. In addition to the evaluation of the WeDRisk modules, the evaluation aimed to assess different aspects such as the usefulness of the approach in managing WD development risks. It was also used to discover what sort of improvements could be made to the approach before moving to the next development stages. The results of this evaluation are important, because they reflect academic and industrial experience, the time this evaluation was conducted which was in the middle of building the WeDRisk approach and the type of experts involved.

7.2 Evaluation Aim

This expert evaluation for the WeDRisk approach aims to criticize the approach based on real experience in order to detect strengths and weaknesses of the approach, and to extract some suggestions and ideas to improve its design.

7.3 Evaluation Aspects

Because of the expert and conference time limitations, the period of this evaluation was short; therefore, the researcher tried to identify the main weaknesses of the WeDRisk approach and its main modules. The evaluation concentrates on understanding, usefulness and helpfulness of the approach and its coverage of WD development risk management aspects.

7.4 Evaluation Question

This evaluation section aims to answer the following question:
Question Q1: *What sort of improvements need to be made to the WeDRisk approach modules?*

7.5 The Methodology

The methodology used in this evaluation can be summarized as follows:

- Recruiting suitable experts and signing the consent form
- Presenting the WeDRisk approach modules followed by their related specific questions (one by one) and asking the expert to evaluate the modules and give feedback and suggestions
- Asking the expert to evaluate WeDRisk overall
- Discussing with the researcher the improvement issues of the WeDRisk approach at the end of the evaluation, and investigating whether there are any other aspects of WD development that need to be considered.

7.6 Questionnaire Design

In order to save experts' time, it was decided to collect the evaluation data via a questionnaire (see Appendix B). Although most of the questions are guided questions, this does not prevent the expert from writing any comments, criticisms or suggestions, or even from using other ways to give an evaluation. Furthermore, almost all of the questions have free space for writing opinions and ideas. The questions were divided into two groups; module specific questions and approach overall evaluation questions. The module specific questions are those which are asked immediately after presenting each module and these aim to explore the weaknesses and strengths of the desired module, as well as to get comments and suggestions that could be used to improve the module. On the other hand, the second group of questions (overall questions) are those which are about the WeDRisk approach in general and these were asked after completion of all modules evaluations. The questions were designed carefully and were revised and tested with some colleagues before the real use; they cover evaluation aspects like usefulness and intelligibility (understandability) levels of the approach.

7.7 Experts Selection

Although participation in this evaluation was voluntary, in order to satisfy the evaluation aim, the following criteria were used to select the experts [107; 108]:

- The expert should have enough experience in related aspects (i.e. work experience in more than one project, two years or more experience in WD development, or has a number of publications in related software development).
- It is preferable if the expert has a background in industry; nevertheless, an academic background is also acceptable.
- Experience in WD development is preferred.

This evaluation was conducted during the Sixth International Conference on Internet Monitoring and Protection (ICIMP2011) at St. Maarten from 20- 25/3/2011. Factors that helped to conduct the evaluation at the ICIMP2011 conference are:

- A number of software scientists from both academia and industry had gathered from around the world with different experiences and backgrounds.
- The conference covered the research scope (e.g. web development and risk assessment).
- Arranging this evaluation cycle was not a complicated task since the experts and the researcher participated in the same conference.
- The timing of the conference was opportune as it coincided with the middle of the research period.
- It was expected that there would be little or no bias, because the experts were not previously known to the researcher and the evaluation was a voluntary process.

- The experts' explicit criticism was needed as it reflects their real experiences in software development and management.

The researcher contacted and interviewed a number of experts who participated in the ICIMP2011 conference in order to explore their experiences and backgrounds and the extent to which they were willing to participate. The researcher briefly described the WeDRisk approach, evaluation aim, evaluation scope, estimated time required and evaluation procedure to them. Of the experts contacted, seven met the evaluation criteria and accepted the invitation. However, due to their interests or time limitations some agreed to participate in some aspects only. Six of the experts have 10 years' experience in software development. Only one expert has a small amount of experience, but he has a relevant background and knowledge. Their experience ranged from software engineering, project and IT managers, reader and IT and software development. The selected experts have worked in universities, polytechnic institutes or software development companies.

7.8 The procedure

The evaluation sessions were performed individually. This is due to three reasons: the expert time constraints, and researcher availability, due to the need to provide clarification for the experts and also to avoid any influence or bias. At the beginning of each evaluation session, the researcher briefed the expert involved with general information (e.g. evaluation aim, aspects, procedure and approach description) and then presented the modules (one at a time) followed by a related evaluation questionnaire to collect the expert evaluation feedback, comments and suggestions. There were no constraints on the experts' answers. For instance, he/she might not have followed the questionnaire style and just criticized, discussed or given comments and feedback in different ways. The expert could refuse to continue or skip some sections at any time. Once the modules evaluation was completed, the experts started the overall evaluation of the WeDRisk approach as a complete idea about the approach was formed. Finally, the researcher and the expert could have an open discussion about WeDRisk and WD development risk management challenges in general, which could lead to making some improvements to the WeDRisk approach.

7.9 Result

The result of this evaluation is summarized in two tables; Table 7.1 summarizes the experts' evaluation for the WeDRisk modules' specific evaluation, and Table 7.2 summarizes the evaluation for the overall WeDRisk approach. The results are discussed in the analysis and discussion section.

7.10 Analysis and Discussion

This evaluation was on two levels: module specific evaluation and WeDRisk overall evaluation. The experts were asked to explicitly reflect their experience without any restrictions. Thus, they expressed their criticism of the WeDRisk approach in different ways, including answering the questionnaires, and providing comments and open discussions.

7.10.1 *Individual Evaluation*

The evaluation results, which are summarized in Table 7.1, can be discussed as follows:

Project Card

Most of the experts rated the usefulness of the project card concept as Very or Strongly Useful and asserted that it covered the most needed aspects. Some experts suggested including other aspects (e.g. quality, budget, time, stakeholder data and objectives).

Some of these suggestions were accepted and included in the card (e.g. budget and using the term *Actual Time* instead of *Current Time*), because they are useful somehow to the risk management process and enhance the card. However, some other suggestions were rejected and they were not included in the card for different reasons (e.g. the quality management issue is rejected because this project card focuses only on risk management issues not general project management issues; including the stakeholder data was rejected because WeDRisk has a special module to handle this).

Table 7.1: Experts - Modules Specific Evaluation Result Summary

Rate the usefulness of the project card concept				
Not Useful	Somewhat Useful	Useful	Very Useful	Strongly Useful
0	1	0	4	2
Evaluate the project card coverage of RM data aspects				
It covers most needed aspects	It covers all aspects	Missed fields	Comments	
6	1	Some experts urged to include quality, budget, time, stakeholder data and objectives in the card	- It needs more testing ; Changing word *Current Finishing Date* with *Actual Finiting Date*	
Evaluate the helpfulness of the RM customization concept				
Almost half of the experts argued that the customizing, as it was presented during the evaluation, was not useful and they gave some suggestions to improve it, such as: the RM customization could be different from risk to risk; factors like budget, history and environment could be considered; and using mathematics could help to avoid any subjective result judgements regarding the type of RM. The customization should be much easier and less time consuming and it needs more testing.				
How accurate is the proposed pre-clustered WD risk list? - It is a good classification - It should include a ”no result” risk under project risks; it is an initial classification - it is acceptable and would be useful if there were several groupings (e.g. technical)				
Do you agree with the statement that consideration of risks from the 3P perspectives saves time and effort?				
Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
0	1	1	4	1
One expert commented that it was useful and another one suggested looking for other classifications.				

Table 7.1 Experts - Modules Specific Modules Result Summary - Continued

Rate the helpfulness of clustering criteria in clustering risks from the 3P perspectives				
Not Helpful	Somewhat Helpful	Helpful	Very Helpful	Strongly Helpful
0	1	2	4	0
The main comment is that some factors are valid for two perspectives.				
To what extent do you agree that WD development is vulnerable to atypical Risks?				
Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
0	0	0	1	6
To what extent do you agree that the absorbing mechanism as an effective way to deal with WD atypical risks?				
Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
0	0	2	3	2
One expert suggested testing the instruction list more				
Are there any other factors which can be included in the estimation matrix?				
No other factors, but the factor importance is different from site to site, risk dependency varies from one risk type to another, which could be considered, and factors level options could be improved.				
Evaluate the usefulness of the RM customization concept				
Saves time and effort	Gives more flexibility	Comments		
3	3	One expert suggested providing help in choosing various processes to solve the problem which according to him, will save time and effort.		
Evaluate the evolution module				
Toward concept and covers evolving needs	Comments			
5	They suggested differentiating “ evolution” from “evaluation”.			

Table 7.2: Experts - WeDRisk Overall Evaluation Result Summary

<table>
<tr><th colspan="5">Evaluate how simple, understandable or complicated the WeDRisk architecture is</th></tr>
<tr><td>Understandable</td><td>Simple and easy to follow</td><td>Complicated</td><td colspan="2">Comments</td></tr>
<tr><td>4</td><td>3</td><td>0</td><td colspan="2"></td></tr>
<tr><th colspan="5">Evaluate the WeDRisk coverage of RM aspects of WD development</th></tr>
<tr><td colspan="2">They cover all aspects of managing WD risks</td><td colspan="3">Not enough coverage due to the following</td></tr>
<tr><td colspan="2">5</td><td colspan="3">Due the managerial aspects (a real case can help)</td></tr>
<tr><th colspan="5">Rate the expected usefulness of the WeDRisk approach for managing WD Risks</th></tr>
<tr><td>Not useful</td><td>Useful</td><td>Very useful</td><td colspan="2">Comments</td></tr>
<tr><td>0</td><td>3</td><td>3</td><td colspan="2">Automate the filling process</td></tr>
<tr><th colspan="5">Rate the understandability level of WeDRisk Modules in general</th></tr>
<tr><td>Very Low</td><td>Low</td><td>Medium</td><td>High</td><td>Very High</td></tr>
<tr><td>0</td><td>0</td><td>2</td><td>3</td><td>1</td></tr>
<tr><td colspan="5">There are no comments</td></tr>
</table>

RM Customization

The experts advised making some improvements to the RM customization matrix before it became ready for use. They suggested treating the situations individually, considering the budget factor and using mathematic equations for making the RM type decisions. In fact, treating the situations individually and considering the budget factor are accepted issues and they are already considered in the following version of WeDRisk. However, using mathematical equations for making the decisions was avoided because the WeDRisk approach is designed to be available for use by developers from different backgrounds and managers who usually prefer simple procedures rather than using mathematical equations. Regarding the extra time consumption issue, it is expected that experience of using the customization process will decrease the time needed. Furthermore, the overall time used for the customized RM will be shorter than that wasted on the wrong RM type.

3P Perspective Consideration and Clustering

The concept of consideration and clustering of risks from the 3P perspectives received great support from the experts and they rated it as Helpful or Very Helpful, emphasizing the importance of the consideration and clustering concept. The evaluation results illustrate that the most of the pre-clustered risks are clustered correctly, which indirectly supports the WeDRisk clustering criteria which were used to cluster the risks. One expert suggested deeper classification of risks from the 3P perspectives. In this regard, it is expected that deeper means more complicated.

Atypical Risks and Absorbing Concept

Almost all of the experts strongly agreed that WD development is vulnerable to atypical risks. This reflects the fact that atypical risks should be considered and preparation is an important issue. Experts also agreed with the effectiveness of the absorbing mechanism to deal with atypical risks, but they recommended further evaluation.

Estimation

Regarding the WDF estimation matrix, the experts made an important suggestion, which was not to use only three levels (Low, Medium and High) to rank the factors, but to elaborate more. This is excellent advice to improve the matrix as it clarifies it more without adding too much options and information. This was considered in the following version of WeDRisk.

Finally, the evolution concept has been supported by the experts.

7.10.2 *Overall Evaluation*

The overall evaluation of the WeDRisk approach was done after the module specific evaluation was completed. The overall evaluation aspects include: the WeDRisk architecture, understandability, usefulness and coverage of WD risk management aspects. According to the results (see Table 7.2), it is obvious that WeDRisk in general has successfully recorded acceptable levels in these evaluated aspects. However, the experts recommended conducting more evaluation studies, as well as making some necessary improvements to the approach, which resulted from suggestions, comments and ideas which arose from discussions.

7.10.3 *Findings*

The expert evaluation results for the WeDRisk approach showed that the approach is understandable, useful, and helpful and covers important risk management needs for WD development. In fact, the result of this evaluation cycle has helped to make many improvements to the WeDRisk approach, which are outlined below:

- The main architecture of the WeDRisk approach, including the process flow and notations, were redesigned and improved.
- The project card was modified (e.g. *Current Date* became *Actual Date*).
- The WDF estimation matrix factor levels were changed from Low Medium and High into more detailed levels.

- The customization matrix became more detailed, self explanation and also new factors like RM budget were added.
- More evaluation studies are planned including another experiment and a case study.
- Clustering criteria factors have been revised.

Thus, the evaluation question (*Question Q1*) is answered and a set of possible improvements to the approach have been extracted.

7.11 Summary

The experts' evaluation was very important to this research as it was conducted during the middle stage and was a reflection of the experience and knowledge of a number of international experts who work in related fields. Indeed, it has had a significant impact on improving the WeDRisk approach, either via the evaluation results or open discussions. Ultimately, according to the experts' feedback, the WeDRisk approach is a promising, helpful, useful and understandable approach. The experts gave some valuable suggestions and comments, which resulted in a number of improvements to the following version of WeDRisk. The next chapter reports another empirical study to evaluate the WeDRisk approach.

Chapter 8

Experiment Two

8.1 Introduction

In this chapter the second controlled experiment in this research is reported, which was conducted to evaluate some novel aspects of the three main modules in the WeDRisk approach (customization, estimation and atypical module). This empirical study was conducted after the necessary improvements that resulted from the experts' evaluation of WeDRisk were performed.

In particular, the novel aspects of the WeDRisk approach which were evaluated, include: consideration and estimation of WD factors, a risk management customization matrix, and atypical risks concept and its module. This experiment was divided into three sections and each section tested one or more hypotheses to evaluate specific module. The idea behind this experiment was to present (inject) some "distributed" and "non-distributed" risk situations and then ask the subjects to deal with these situations using the evaluated modules. At some experiment stages, the subjects were divided into control and experimental groups (based on the nature of the test). Experimental group subjects used WeDRisk modules, but the control group subjects had to rely on their knowledge and experience. The design of this experiment (i.e. hypothesis, subject selection, data collection, avoidance of bias, validation, procedures, tasks and instructions) was inspired by the design of some software engineering experiments and empirical studies in [101; 103; 109; 110; 111]. The controlled experiment methodology was chosen to evaluate these WeDRisk modules for the following reasons [110; 112]:

- It is necessary to emulate the same working environment for all the subjects.
- It is not feasible to generate an atypical risk situation for evaluation purposes. It happens randomly, but can be emulated in the experiment.

- Many observations, measures and support works need to be done or provided during the evaluation of the modules, and the experiment is the best option for that.

- It is good to focus on specific variables, measures, and the relationships between them with extra flexibility in asking questions.

- The evaluated modules were new; thus, it was not possible to find suitable data from pervious approaches that could be used to evaluate these modules. This was also due to the restrictions on data imposed by developers.

8.1.1 *Experiment Scope*

This experiment is mainly designed to evaluate three WeDRisk approach modules (Estimation, Customization and Atypical).

8.2 Questions and Hypotheses

The experiment is mainly designed to answer the following three questions:
Q1: What is the coverage and consideration of WD factors by the WeDRisk approach?
Q2: How easy is it to understand and use the WeDRisk approach?
Q3: How usable and helpful are the evaluated modules?

In order to answer these questions, five hypotheses were tested in this experiment. Hypotheses H1 and H2 evaluate the Estimation module; Hypotheses H3 and H4 evaluate the Customization module; and hypothesis H5 evaluates the Atypical module; these hypotheses are:

Hypothesis H1: *The TREV equation is an ideal option to estimate WD development risks and to consider the WD factors compared with the RE equation.*
Hypothesis H2: *The WDF estimation matrix is useful, understandable and helpful to estimate and consider WD factors.*
Hypothesis H3: *The Customization matrix is easy to use and it helps to specify what type of RM (Plain or Deep) is suitable to manage the risk involved.*
Hypothesis H4: *Risk management customization by providing two risk management types (Plain and Deep) is useful.*
Hypothesis H5: *Providing an absorbing mechanism to deal with atypical risks is a useful tactic and helps developers to deal with such risk types with less confusion.*

8.3 Method

As with the first experiment in Chapter 6, the method design and necessary measures were taken in order to obtain the best results and avoid bias [101; 103; 109; 110; 111].

8.3.1 *Apparatus*

The apparatus that were used in this experiment include normal stationery, hard copies of the experiment material, data collection forms (see Appendix C) and a sportwatch. A computer was used for saving data and a Minitab tool was used for statistical analysis purposes.

8.3.2 *Materials*

Hard copies of all required information and forms were prepared and provided for all the subjects to use equally.

8.3.3 *Subjects*

About 35 subjects were recruited by email to participate in this experiment. Of these, the researcher selected 24 subjects to participate in this experiment as they met the criteria for participation (experience and knowledge in software development). The subjects were researchers, PhD students at the School of Computing Science/Newcastle University, and some visiting students. The subjects had either been involved in software projects or had at least attended software engineering courses.

In order to improve the design of the experiment, a pilot study was carried out before starting the real experiment sessions. The pilot study was conducted using another group of participants. A number of issues were addressed and improved for the real experiment, including the task's required time estimation, sequence, instructions to subjects, data collection procedures and risk situation design.

8.3.4 *Data Confidentiality*

The subjects were told that all subjects' bibliographic data and the collected data are confidential to the experimenter and his supervisor. The data are used for research purposes only. All subjects were given reference names and their actual data were used only to provide them with the £10 Amazon vouchers as compensation for their time.

8.3.5 *Variables*

The dependent variable of this experiment was the presented (injected) situation, whereas the independent variable was the subjects' reactions.

8.3.6 *Measurement Units*

Table 8.1 defines the measurement units (e.g. time and effort used), which are used in this experiment.

Table 8.1: Experiment Two Measurement Units

The Unit	Definition	How it is measured
Used Time	The length of time spent to implement a specific task	Difference between starting time and ending time of the task implementation
Effort	The exertion made to implement a specific task or achieve a goal	Observations, used time comparison, asking specific questions, feedback and attempts to implement the task

8.3.7 *Generalization and Threats Validity*

Generalization

A number of measures have been taken in order to make the experiment sample reflect the real population of WD development. Mainly, the selected sample was concentrated on a set of subjects who are working or have worked in the field of software development, or have attended software engineering courses. Indeed, all of the recruited subjects were either software researchers who work in WD software development projects, or PhD students who were also involved in software research and had experience of related projects. The selected sample of subjects was intended to reflect real software projects. However, it is costly and almost impossible to cover all software development populations in this experiment. Different evaluation techniques were used to evaluate the WeDRisk approach including case studies, other experiments and expert evaluations in order to be able to generalize as much as possible.

Validation

The experiment validity is important to ensure the quality and generalizability of findings. Two types of validity are involved in this experiment: internal validity, which concerns whether the study supports the findings; and external validity, which is connected to the generalizability of the results [103; 104]. The threats to internal and external validity are taken into account and addressed as follows:

Internal Validity

Selection: The assigning of the subjects to the experimental and controlled groups was done on a random basis; in addition, the risk situations were also applied randomly.
History: The subjects were selected from the same place; therefore, in order to reduce any influence they might have had on one another, they were recruited and contacted individually, and they performed the tasks individually at different times.
Motivation: As the performed tasks did not take a long time, there was little concern about boredom or loss of enthusiasm during the experiment.
Time: It was expected that the subjects might perform the tasks in a hurry, due to anxiety that there was insufficient time to perform the tasks, which could have affected the decisions that were made and, consequently, the results. Thus, in the pilot study, the required time for the experiment was estimated and the subjects informed of the estimated time when they were recruited. Moreover, during the experiment, they were told to take enough time to perform tasks and given the option of stopping if they were unwilling to continue.
Training: Before performing any tasks in the experiment, a brief description was given to the subjects and enough clarification and training was provided before each section of the experiment; they also had the right to ask any questions.

External Validity

Subjects: The difficulties of generalizing from students to professionals were taken into account. Therefore, the subjects were mixed (students and researchers who work in software projects). Indeed, the use of students as subjects in this experiment may not threaten the results, since most of the students were PhD students who had experience of software projects and some professional abilities.
Environment: The experiment environment tried to emulate a real development environment (developer, place, project and risks).

8.3.8 *Control Measures*

The control measures in the experiment were taken in order to reduce any bias and to keep everything the same except the tested variable. They were also to ensure the experiment could be replicated. The control measures include the following aspects:
Environment: The environment was controlled in order to enable the recording of time consumption individually, avoid the subjects' influence upon one another or by any other means, provide help on an equal basis, observe subjects' progress during the task implementation, and give the subjects the same time and support.
Injected Situations: The same injected situations were used by all of the subjects but in random order and using reference numbers. The injection of the situations was controlled by the experimenter to avoid any biases.
Tasks Sequence: The sequence of tasks was maintained by the experimenter during the experiment, but if any subject refused to continue at any stage of the experiment then there would not have been any effect on the other stages and the data from the implemented stages could have been used as a part of the experiment result.
Provided Support: All of the essential information, support and training material was provided on an equal basis.
Control Group: The control group was used in this experiment as a part of the control measures and for results comparison.

8.3.9 *Experimenter and Subject Tasks*

At the beginning of the experiment, the experimenter gave a brief description about the experiment and the assigned tasks, and then subjects read, filled and signed the consent form. After that, the experimenter requested the subjects understand and perform the assigned tasks (e.g. reading the injected risk situation and using the provided modules) and also asked them to be as accurate as possible. The subjects were also told that they had the right to ask any related questions and receive the required clarification, and that they had the right to stop at any time if they were not comfortable. At the end of each section, the subjects were asked to give their feedback and provide any suggestions or comments that could improve the desired module. Generally, during the experiment, the experimenter had other tasks which included:

- Providing the training, related material and support required on an equal basis
- Injecting the risk situation on a random basis
- Managing the task sequence during the experiment implementation

- Collecting the data, observing experiment progress, assigning the tasks and recording the time
- Distributing the subjects into control and experimental groups (on a random basis)

In addition to the common subject tasks (e.g. reading and signing the consent forms), the subjects had to perform some specific tasks related to specific evaluated modules, which included the:

Estimation Module: Specifying the suitable estimation equation (RE or TREV) for each injected risk situation and then estimating the risk involved using that equation.
Customization Module: Based on the assigned group (control or experimental), the subjects had to specify a suitable RM type (Plain or Deep) for each risk. For this purpose, the experimental group subjects used the support matrix provided, whereas the control group subjects used their experience and knowledge.
Atypical Module: According to the assigned group, the subjects had to deal with the injected atypical risk situation. The experimental group used the atypical module, whereas the control used their experience or knowledge.

8.3.10 *Avoiding Bias*

Bias is always expected in any experiment; however, the following are the measures that were taken to avoid potential bias during this experiment:

- Randomizing the subject groupings and the risk situation injection.
- Providing the required information, material and support on an equal basis.
- Removing time pressure during task implementation in order to avoid any subjective answers.
- Allowing subjects to request clarification and ask any related questions to garner the necessary information.
- Giving reference numbers to the subjects and injected situations to maintain the privacy.
- Allowing the subjects to freely express their comments and suggestions at the end of each module evaluation.

8.3.11 *Injected Risk Situations*

Emulating the management of WD development risks in this controlled experiment was not an easy task. A number of risk situations were designed and injected during the running of the experiment phases with consideration to the fact that:

- The situation should cover the evaluation aspects of the modules.
- It could be reused to evaluate more than one module.
- It could work independently (standalone).
- It should reflect real risk situations as closely as possible.
- It should be simple, self explanatory, consistent and understandable.
- It must be short to avoid boredom and wasting time.

For these purposes, the designed situations were tested many times and improved based on the comments and notifications received, taking into the account the reading time. Generally, the evaluation of the modules was an independent operation. Only the injected situations were shared (to reduce the reading and understanding time) without any considerable effect on the evaluation operation. Therefore, in order to avoid any bias, the situations were injected randomly with a time gap between them. The injected situations included collocated development and WD development situations (see Appendix C) and were given reference numbers for ease of use during the experiment.

8.4 Procedures

This section describes the experiment design and the procedures to evaluate the three WeDRisk modules. All the tasks were implemented individually by the subjects (one subject in each session).

8.4.1 *Estimation Module*

Three main tasks were performed by the subject in this stage (specifying a suitable estimation equation, estimating using the specified equation and giving feedback). These tasks are described in detail below:

I) *Specifying a Suitable Estimation Equation*

Four prepared risk situations (see Appendix C) were randomly and individually injected. The subjects were asked to specify which estimation equation (RE or TREV) was suitable to deal with each situation. The subjects read the injected situation and, based on their understanding and experience, indicated a suitable equation to estimate the risks involved. Regardless of the selected equations, the subjects were also requested to justify their decisions by writing down the reasons behind their selection. The experimenter monitored the implementation progress of this task and recorded the time used to make each decision. After the completion of this task, the subjects were asked to make any comments or suggestions about the equations.

II) *Estimating Using the Specified Equation*

Since the estimation equation is specified for each risk, the next step is the estimation of risks using the specified equation (RE or TREV). If the decision is the RE equation, then the subject needs firstly to estimate the probability and magnitude of the risk and then apply the ordinary RE equation to estimate the risk. If the decision is the TREV equation, then the subject estimates the RE value and the total WDF value for the risk before using the TREV equation to estimate the risk. The implementation details for the both two equations (RE and TREV) are described below:

- **RE Equation:** In order to estimate the RE value (see equation 2.1), the subject reads the injected risk situation first and then uses the line estimation technique (see Figure 4.8) to estimate the probability and magnitude of the risk. Based on this ranking, he/she can obtain the associated values for the probability and magnitude from the line. Subsequently, RE is obtained by multiplying the probability and magnitude values. The minimum value of RE is zero, which happens only when there is no chance of the risk occurring, and the maximum is five when the risk is certain and its magnitude is severe.

- **TREV Equation:** Before applying the TREV equation (see equation 4.1) the subject needs to estimate the RE value using the same technique as above and then estimates the second part of the TREV equation, which is the $\sum WDF$ (see equation 4.2) using the WDF estimation matrix (See Table 4.11). Subsequently, the subject applies the TREV equation to get the TREV estimation for the desired risk.

III) *General Feedback*

Once the subject completes the specifying and estimation tasks for the four injected risk situations, he is requested to give his feedback about the estimation operation in general. Four guided questions were designed for this purpose (see Table 8.4). The subject can also write any comments or suggestions that might help to improve the estimation module.

8.4.2 *Customization Module*

The subjects involved in this stage were randomly divided into control and experimental groups. Both groups were asked to decide the suitable RM type (Plain or Deep) for each injected situation. Control group subjects implemented the task based only on their experience and knowledge, whereas the experimental group subjects used the WeDRisk customization matrix provided (see Table 4.5). Regardless of group, the subjects were provided with general information about the risk management customization concept and the definitions of plain and deep RM types. The only extra information and material that were provided to the experimental group subjects were about the customization support matrix.

In order to save subjects time and avoid boredom, the same risk situations that were used to evaluate the estimation module are re-injected to evaluate this module, though on a random basis.

8.4.3 *Atypical Module*

In order to test the evaluation hypothesis for the atypical module, the subjects were also divided into control and experimental groups. Experiment group subjects used the atypical module (see Figure 4.10), whereas the control group subjects depended on their experience and knowledge. At the beginning, and in order to explore the level of knowledge about the atypical risks concept, the experimenter briefed the subjects about the atypical risks and then asked them some questions about the atypical risk concept, such as *"What do you know about atypical risks?" "Did you face any atypical risks?"* and *"How do you deal with atypical risks?"*. Another purpose of asking these questions was an attempt to extract any valuable ideas that might be used to deal with such types of risk and to see the weaknesses in dealing with atypical risks. Subsequently, the experimenter introduced the atypical risk situation and requested the subjects to deal with it based on their assigned groups. Special forms were designed to gather the data during the task implementation. The experimenter observed the task progress, provided the required support and recorded the time used. At the end, the subjects were asked to provide their comments and suggestions regarding the module.

8.5 Results and Analysis

For simplicity purposes, the results and analysis of the experiment are presented based on the modules and presented along with the related hypotheses.

8.5.1 *Estimation Module*

Hypothesis H1: *The TREV equation is an ideal option to estimate WD development risks and to consider the WD factors compared with the RE equation.*

Three tasks were implemented by the subjects to test hypothesis H1, which are: specifying the suitable type of estimation equation (RE or TREV); and justifying their decision and giving feedback regarding the suitability of TREV or RE equation for WD development. As can be seen in Table 8.2, the results demonstrated that there is a consensus in the decisions taken (e.g. for situation number 6004, 17 subjects agreed to use the RE, while only 6 did not concur, and for situation reference number 0901, 22 subjects agreed to use TREV against only 1 for RE). Chi-square is a test used to determine whether there is a significant difference between two or more frequencies in one or more categories. The Chi-Square (X2) test was used in this experiment and showed that the proportion of subjects who selected RE equations in situations 6004 and 8033 is significantly higher than those who selected the TREV equation in the same situation with Chi-Square (X2) of 23 with p-value of 0.001 (<0.05). In contrast, in situations 0901 and 1072, the number of subjects who selected TREV was significantly higher than the number who selected RE. This result complies with expectations regarding the suitability of RE or TREV equations for each situation. Generally, the time that was used to decide which type of estimation was suitable for the situations was slightly higher in some situations due to differences in text length. The justifications for the subjects' decisions (see Table 8.3) were based on the number of sites, involvement of WD factors or complexity of the situation. They justified the selection of RE as due to a single site, less involvement of WD factors, or simplicity of the situation, and the opposite is correct for the TREV equation.

The subjects were asked (*"To what extent do you agree that TREV is more suitable than RE for WD risks estimation?"*). As can be seen in Table 8.4, most of the subjects Agreed or Strongly Agreed with the suitability of the TREV equation for WD development risks. Meanwhile, the subjects are also agreed that WD factors should be considered in the estimation of the WD development risks.

Hypothesis H2: *The WDF estimation matrix is useful, understandable and helpful to estimate and consider WD factors.*

Consensus from the subjects' feedback on the estimated WDF values regarding the

Table 8.2: Summary of Estimation Decision (RE/TREV)

Situation No.:	**6004**	**0901**	**1072**	**8033**
RE	17	1	5	19
TREV	6	22	18	4
Avg. Used Time	22.0	27.9	32.8	28.0

Table 8.3: RE/TREV Decision Justifications Summary

Justifications for RE	**Justifications for TREV**
Single site and no distribution	**M**ulti Sites and WD factors are involved
No involvement of WD factors	**C**omplexity of the situation and involved factors
Simple and RE is enough	**T**REV gives more abilities

usefulness, helpfulness and ease of use of the matrix, any confusion or need for support and the experimenter observation parameters were used to help test this hypothesis. The statistical calculations for the WDF estimated values for situation number 0901 illustrated that the mean is 12.455 and the median is 12.00; the standard deviation is 1.057 and the P-value is less than 0.05. This indicates that the data are very close together and clustered around the mean. Meanwhile, looking at the distribution of WDF for situation number 1072, as illustrated by the statistical graphic summary (see Figure 8.1), it can be seen that the mean is 10.611 and the median is 10.50, indicating that the data are very close together; in addition, the standard deviation is 1.145 and P-value is less than 0.05, indicating that the data are clustered around the mean. Therefore, the WDF estimation for both situations (0901 and 1072) demonstrate a strong consensus, taking into consideration that the values of WDF in the matrix range from 3 to 15 (minimum WDF is (1*3 = 3) when all of the selected options are on the first level, and the maximum WDF is (3*5 = 15) when all of them are on the fifth level).

The estimated values for WDF (sites dependency, sites distribution and communication availability) for the two situations numbered 0901 and 1072 have also shown a strong consensus, as shown by the statistical calculations. Figure 8.2 illustrates an example of a statistical graphic summary for the sites distribution factor estimation for situation number 1072.

Generally, the average time used for the estimation using the TREV equation was 3.069 minutes which is acceptable and was not high compared with the average time taken using the RE equation, which was 1.375, taking into account that in the TREV

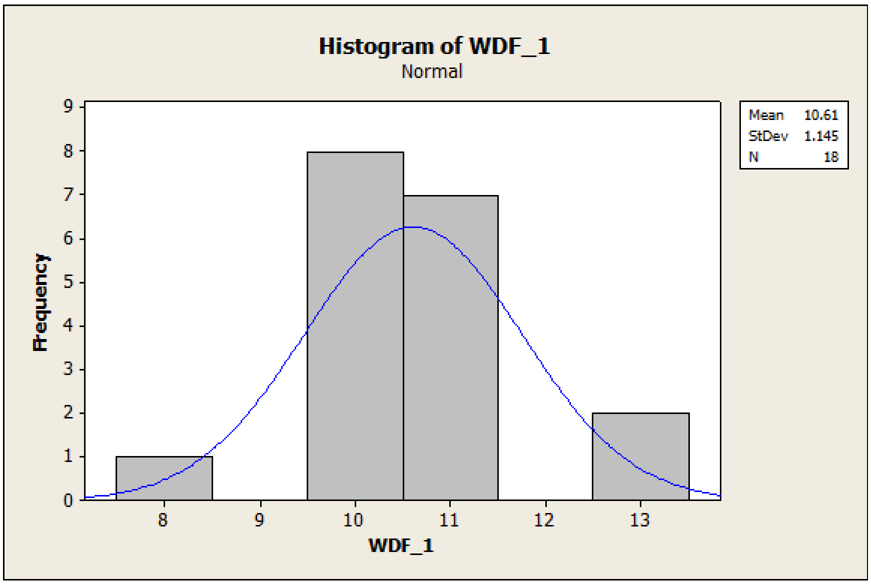


Figure 8.1: Statistics Graphical Summary for (WDF situation 1072)

case, the estimation includes additional WD factor estimation and calculation.

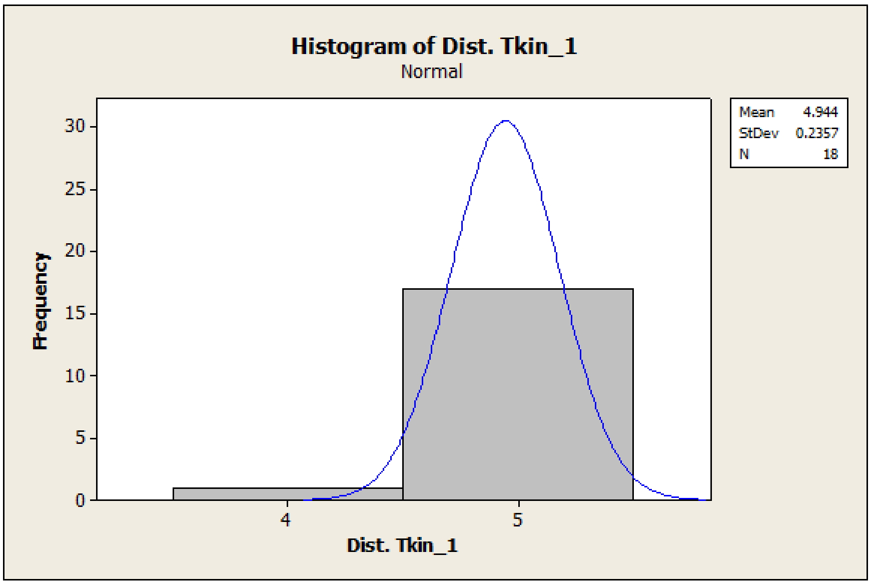


Figure 8.2: Statistics Graphical Summary for (Sites Distribution 1072)

At the end of this evaluation stage, the subjects were asked to give general feedback about the estimation module by answering some guided questions and also to provide their suggestions and comments to improve the module. Table 8.4 summarizes the answers and the feedback regarding the WD factors, TREV equation and the support matrix. The subjects were asked to rate the usefulness of the WDF estimation matrix, and 17 subjects responded. Their answers, as shown in Table 8.4, ranged from Useful to Very Useful with a percentage rate of 29.4% for Useful, 52.9% for Very Useful, 17.6% for Strongly Useful; the other options (Not Useful and Somewhat Useful) were not selected. The next question was about the ease of use of the matrix to estimate the WDF, to which 23 subjects responded. As Table 8.4 illustrates, the answers confirmed the ease of use of the matrix. The values, as shown in the table, were 0% for Difficult and Somewhat Easy, 13% for Moderate, 52.1% for Easy and 34.7% for Very Easy. The subjects agreed with including the WDF in the estimation of WD development risks with a percentage rate of 59.0% for Strongly Agree, 40.9% for the Agree option, and 0% for the remaining options (Neutral, Disagree and Strongly Disagree). The subjects showed their support for the factors included as they did not criticise the three factors; however, some of them suggested including other factors, as can be seen in Table 8.4.

8.5.2 *Customization Module*

Hypothesis H3: *The Customization matrix is easy to use and it helps to specify what type of RM (Plain or Deep) is suitable to manage the risk involved.*

The subjects in the control and experimental groups made decisions on the suitability of RM types for each injected risk situation. Table 8.5 summarizes the decisions taken and the average time used by the control group subjects.

As can be seen in the table, there is a high degree of inconsistency in the decisions taken and in some cases the subjects were unable to decide. The subjects justified their "*Plain*" decisions thus: time is critical; few factors are involved; the situation is simple and clear; or there is enough RM experience. In turn, they justified their "*Deep*" decisions thus: there are enough resources for risk management; there is complex interdependency; the risk is high; or there are many sites.

Table 8.4: Used Estimation Feedback

To what extent do you rate the usefulness of the WDF estimation matrix?				
Not Useful	Somewhat Useful	Useful	Very Useful	Strongly Useful
0	0	5	9	3
How easy was it to use the WDF estimation matrix?				
Difficult	Somewhat Easy	Moderate	Easy	Very Easy
0	0	3	12	8
To what extant do you agree that TREV is more suitable than RE for WD risk estimation?				
Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
8	12	2	0	0
To what extant do you agree that WD factors should be included in WD risks estimation?				
Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
13	9	0	0	0
Any other WD factors that should be considered: - Reliability of the sites - Sites' local communications - Methods used and technology compatibility across the sites - Geographical or environmental phenomena				
Any comments / suggestions: - It is easy to learn and use, clear and gives a quick results - Lack of finance information consideration - Estimate sites risks individually and then combine the overall sites estimations - The formula (WD/TREV) works well depending on how much information is provided for each scenario -Use different guidance for TREV (big company risks different from small company risks)				

Table 8.5: Summary of Customization-Control Group Decisions

Situation No.:	**6004**	**0901**	**1072**	**8033**
Plain	6	5	3	3
Deep	5	5	7	8
No Decision	0	1	1	0
Avg. Used Time	0.5	0.3	0.8	3.6

On the other hand, as can be understood form Table 8.6, there is consensus in the experimental group decisions (e.g. 75% of the subjects selected the plain RM type for situation number 6004; in contrast, only 25% of them selected the deep RM type for the same situation). All of the experimental group subjects were able to decide, as the number of people unable to decide was always zero.

Table 8.6: Summary of Customization-Experimental Group Decisions

Situation No.:	**6004**	**0901**	**1072**	**8033**
Plain	9	3	11	9
Deep	3	9	1	3
No Decision	0	0	0	0
Avg. Used Time	1.0	0.9	0.8	0.8

Table 8.7 shows the number of ticks under resources and experience options for each injected situation in the customization matrix by the experimental group subjects. In view of the results in the table and the task implementation flow, it was found that the subjects used the matrix conditions perfectly to produce their decisions. Concisely, the matrix, which was used by the experimental group, reveals the following points:

- The time to decide was shorter compared with the control group
- There were no subjective decisions and the available information was used perfectly.
- There was no confusion with the use of the matrix and it was easily and correctly used by the subjects.

The experimental group subjects were asked to rate the usefulness of the customization matrix. 17 of them responded (see Table 8.8) and, as can be seen in the table, 88.2% of them rated the matrix as Useful, Very Useful or Strongly Useful. Meanwhile, 11.7% rated it as Somewhat Useful and 0% as Not Useful. The subjects also provided some comments and suggestions as follows:

Table 8.7: Customization-Matrix Experimental Group RM Resources Ticks

Situations					
		6004	0901	1072	8033
Time	**L**imited	5	3	9	7
	Enough	1	5	0	2
Staff	**L**imited	3	0	4	1
	Enough	2	0	0	0
Budget	**L**imited	0	0	5	3
	Enough	0	0	0	0
Experience	**L**imited	0	0	1	0
	Enough	6	0	0	0

Table 8.8: Usefulness of the RM Customization Decision Support Matrix

Usefulness of the Customization Support Matrix				
Not Useful	Somewhat Useful	Useful	Very Useful	Strongly Useful
0	2	5	8	2

- It would be better if the matrix considered the proportion of plain in deep, as well as the cost of applying plain and deep and additional risks by using plain risk management.
- Not all risks are equal some need more consideration than others.

Hypothesis H4: *Risk management customization by providing two risk management types (Plain and Deep) is useful.*

In order to test this hypothesis, the subjects were asked to evaluate the usefulness of the RM customization concept as follows:

Evaluate the usefulness of the RM customization concept (Please tick all boxes that apply)

- *It saves time and effort*
- *It is needed when there is criticality of the situation*
- *It is not helpful*
- *Comments to improve it: - - - - - - - - - -*

The subjects' responses are summarized in Table 8.9. From the numbers in the table, it can be seen that 4.3% of the subjects rated it as not helpful, whereas 73.9% of the subjects agreed that the customization of RM saves time and effort, and 59.5% agreed that it is needed when there is criticality of the situation. The subjects provided two important comments, which are:

- RM customization increases the implementation of risk management and helps the developer not to ignore it, as there are more options that can fit the needs.

- The developers and managers should know what Plain does and what deep does.

Table 8.9: Risk Management Customization Usefulness

Risk Management Customization Usefulness		
Saves Time and Effort	It is Needed	Not Helpful
17/23	16/23	1/23

8.5.3 *Atypical Module*

Hypothesis H5: *Providing an absorbing mechanism to deal with atypical risks is a useful tactic and helps developers to deal with such risk types with less confusion.*

The following four points test this hypothesis:

I) ***Level of Knowledge of Atypical Risk***

When the subjects were asked what they knew about atypical risks, the answers showed that the subjects from both groups had a very low level of awareness and knowledge about them. Most of them 60.8% had a very limited or no idea at all about the atypical risks, and 26% had based their ideas on what the experimenter had told them. Only about 13% of the subjects had previous, limited knowledge about atypical risks. Based on this knowledge, it is not expected that they can avoid or deal with atypical risk threats.

II) ***Experience of Atypical Risks***

From the two groups, there was only one subject who had experience of atypical risks as a company for which he had previously worked had faced a real atypical risk.

III) ***Preparedness for Atypical Risks***

The subjects were asked what to do if they faced an atypical risk; most of the subjects (69.5%) answered either 'do nothing' or 'do not know what to do'; other subjects (30.4%) suggested different or random and unsystematic steps to deal with atypical risks. Some of them suggested giving up and stopping the work totally or hiring some experts, and others suggested the same risk management steps but in a different order. Thus, it can be understood that there is no clear and standard procedure that can be followed in the case of atypical risks.

IV) *Reactions and Time Used*

Atypical risk situations were injected and dealt with by the subjects. Table 8.10 summarizes the actions taken and the average time used from the control and experimental groups against the injected atypical risk situation. As can be seen in the table, the subjects from the control group could not manage the risk situation, or they recommended hiring experts, whereas the subjects of the experimental group were able to follow the mechanism provided and suggested systematic steps. However, the average time used to manage the atypical risk situation was slightly lower in the case of the control group compared with the experimental group. This is could be because the control group subjects had done nothing in general.

Table 8.10: Reactions Summary Against Injected Atypical Risky Situation

<table>
<tr><td>Group</td><td>Reactions Summary and Type</td><td>Used Time Avg</td></tr>
<tr><td>Control</td><td>Hire Expert
Do nothing
Search for info.</td><td>3.574 m.</td></tr>
<tr><td>Experimental</td><td>Following the absorbing mechanism</td><td>4.013 m.</td></tr>
<tr><td colspan="3">**Subjects Comments and Suggestions**
- Pre-evaluation should be done before other actions, which means, it is the first step
- Feedback regarding the atypical risk and management processes should be circulated to all people concerned (e.g. managers, developers and stakeholders)
- Absorbing actions may cause new risks which should be taken into account</td></tr>
<tr><td colspan="3">**Experimenter Observations**
- It is was not easy to give more examples about atypical risks
- Some subjects asked for clarification on the sequence of the absorbing mechanism steps</td></tr>
</table>

8.6 Discussion

Three modules were evaluated in this controlled experiment, namely estimation, risk management customization and the atypical risks module. For simplicity purposes, the experiment was designed with a minimum of dependency so that the evaluation of each module could be conducted individually and without any effect on the evaluation of other modules. The evaluation aspects include; ease of understandability and use, usability and usefulness. The result of the experiment was used to explore what sort of improvements need to be made to the WeDRisk modules. Hereafter follows a discussion on how the hypotheses were tested for each module.

8.6.1 *Estimation Module*

Hypotheses H1 and H2 were tested to evaluate the estimation module. Experiment results regarding these hypotheses are discussed below:

Hypothesis H1: *The TREV equation is an ideal option to estimate WD development risks and to consider the WD factors compared with the RE equation.*

The result of the experiment regarding hypothesis H1 implies the following two points:

I) There was consensus in the subjects' decisions regarding the suitability of RE and TREV for each injected situation and this was as the experimenter had expected. The subjects justified their decisions for RE (see Table 8.3) according to simplicity, not the involvement of WD factors or a single site, and they justified the TREV selections according to complexity, existence of a multisite or involvement of WD factors. From the results, it can be understood that the subjects decided to use the RE equation for non-WD development situations and the TREV equation for situations involving WD development.

II) The subjects agreed or strongly agreed that the TREV equation is more suitable than the RE equation for estimating the WD risks.

Therefore, based on the above results, it can be concluded that hypothesis H1 has very strong support from the experiment result.

Hypothesis H2: *The WDF estimation matrix is useful, understandable and helpful to estimate and consider WD factors.*

The evaluated matrix involves three WD factors (Sites Dependency, Sites distributions and Communication), which could have an effect on the WD development risks. These factors are not final and one of the evaluation aims is to explore whether they need to be modified or there are any other factors which need to be considered. In this regard, as the results showed in the previous section, the matrix helped the subjects to estimate the risks with consideration to the WD factors involved. The matrix was easily used with negligible confusion, a high degree of understanding and gave a consensual result. Some subjects suggested adding some other factors to the matrix (e.g. reliability of the sites; sites' local communications; methods used and technology compatibility across the sites; geographical or environmental phenomena). However, the factors that should be included in this matrix are those which have an effect on the risks but are not risks themselves. Therefore, the above suggested factors and others could be revised and added to the matrix if they comply with the WDF factor definition. Changing and modifying the matrix factors is an easy task and the matrix could be used to estimate other types of software development with some changes.

Based on the consensuses in the estimated WDF values, as it is demonstrated with situations 0901 and 1072, the positive feedback regarding the usefulness of the matrix, ease of use of the matrix, suggestions and comments regarding the matrix as described above, and also the experimenter observations it can be concluded that hypothesis H2 has strong support in this experiment. Meanwhile, observation, suggestions and discussions regarding the TREV equation during the experiment exhibit that there is an ability to use the TREV equation for other types of software development risks, if the factors which affect the risks are identified and considered. A valuable aspect of this matrix is that if any of the WD factors is not included in the desired development, a default value of "1" will be given to that factor, which has no effect on the result. Finally, from the experiment results, it is clear that there is a chance to generalize the TREV equation and upgrade the matrix by including different factors when needed.

8.6.2 *Customization Module*

Many software developers and managers avoid or run away from the practice of risk management because it is costly, needs extra effort or they think it is time-consuming. One of the hidden reasons behind this is that the existing approaches are not flexible enough to match the implementation of risk management and availability of resources. Usually, only one type of risk management is offered, which is the deep type. In this type, the developers and managers need to implement all RM steps, even when time is critical or there are insufficient resources. Thus, in order to increase the level of risk management practice, and encourage developers to practise it under all conditions, the WeDRisk approach offers two types of RM (plain and deep). Two hypotheses (H3 and H4) were tested in this experiment in order to evaluate the customization module. Hereafter, the results of the experiment and the two hypotheses are discussed:

Hypothesis H3: *The customization matrix is easy to use and it helps to specify what type of RM (Plain or Deep) is suitable to manage the risk involved.*

As the experiment results show in Tables 8.5 and 8.6, it is clear that the customization matrix helped the experimental group subjects to decide what type of RM was suitable for each injected risk situation. This can be understood from the total time used by each group, as the control group subjects used more time to decide compared with the experimental group. Moreover, there were inconsistencies in the control group decisions compared with the experimental group decisions, which is clear from the numbers in each decision (see Tables 8.5 and 8.6). In some situations, the control group subjects were not able to decide what RM type was suitable, whereas the experimental group subjects were able to decide for all of the injected situations. Moreover, during observation of the experiment, the experimenter noticed that the control group subjects were confused and asked for support many times, whereas the experimental group subjects used the matrix smoothly and perfectly, with minimal support from the experimenter. Indeed, the matrix helped a lot and reduced the number of subjective decisions. The experiment results show that the matrix was easy to use, simple and gave quick results, which supports hypothesis H3.

Hypothesis H4: *Risk management customization by providing two risk management types (Plain and Deep) is useful.*

The usefulness of RM customization was evaluated from different aspects (e.g. time and effort saving). For this purpose, the subjects from the two groups expressed their opinions regarding the RM customization concept. The result, as shown in Table 8.9, demonstrates that the subjects agreed with the idea that the RM customization is useful, saves time and effort and is needed when there is criticality in the situations. Indeed, this supports hypothesis H4. Some subjects suggested specifying and describing the tasks that need to be implemented under the plain and deep risk management types. Accordingly, it is expected that the customization of risk management could increase the chances of practicing risk management, since it had strong support from both groups.

8.6.3 *Atypical Module*

Hypothesis H5 was tested to evaluate the atypical module. The result of the experiment in this regard is discussed below:

Hypothesis H5: *Providing an absorbing mechanism to deal with atypical risks is a useful tactic and helps developers to deal with such risk types with less confusion.*

At the beginning of this stage, the level of knowledge and awareness of atypical types of risk was explored. In hindsight, the experiment result showed that the level of knowledge with atypical risk types was very low. The experiment result also demonstrated that there was very little experience of and preparedness for atypical risks. Two reasons could be behind this: the low frequency of atypical risks or lack of approaches that can deal with atypical risks. In fact, as exhibited in Chapter 3, the existing software risk management approaches are designed to deal with ordinary risk types when there is at least minimum information about the risk (e.g. probability and magnitude), which is usually not available in the case of atypical risks.

Ultimately, the experiment result exhibited that the experimental group subjects were able to act systematically and absorb the injected atypical risk by following the module steps, whereas the control group subjects acted randomly without any harmony in their actions. However, the average amount of time used by the control group subjects was slightly lower compared with the experimental group. This is most likely because the majority of the control group subjects did nothing to manage the injected atypical risk. In consequence, from the above listed points, it can be understood that hypothesis H5 is supported by this experiment. Being proactive and absorbing the atypical risks can be done and it is better than doing nothing or acting randomly when this type of risk is faced.

8.7 Limitations

Although an attempt was made to emulate real risk situations in order to evaluate the WeDRisk approach in this controlled experiment, there are a number of limitations associated with this experiment. Some of these limitations are general and related to the use of the experiment methods, while others are specific to this experiment. The limitations are as follows:

- The running costs of the experiment are high and thus limit the number of subjects; therefore, the experimenter tried to tackle this by focusing on the type of subjects involved.
- Sometimes the samples in the experiments did not reflect the real population, but in this experiment most of the subjects have experience of software projects via their work, study or research backgrounds.
- Usually, experiments are affected by bias. In this experiment, some measures were taken to avoid and reduce bias.
- It would be preferable if the subjects were from the distributed software industry, but due to time limitations and imposed data restrictions, this was not possible.
- Due to the experimenter's availability limitations during the experiment running time, the subjects participated individually (one by none) in order to ensure provision of necessary support, observation of task implementation and recording of the time used for task implementation.

8.8 Summary

This chapter reported the second controlled experiment in this research that was used to evaluate some central parts of the approach (i.e. estimation, customization and atypical risk modules), in terms of their usefulness, ease of understandability and usability. The evaluated WeDRisk modules showed a good ability to deal with the injected risk situations, as they dealt with them successfully and effectively. In particular, the experiment results illustrated how these modules are useful, understandable and easy to use and are able to consider WD factors. Nevertheless, the results also show that there are still some improvements that can be made to the modules, as seen in the suggestions, observations or findings. Examples of the suggested improvements are the generalization of the TREV equation to estimate other types of software development risks by adding other risks factors to the WDF estimation matrix, and simplifying the clustering criteria to reduce the reading time. Finally, the experiment is described and validated in order to make it available for any replication and all original related documents have been kept for any further analysis or evaluation. More evaluation work is recommended to evaluate WeDRisk overall.

Chapter 9

Evaluation Case Study and WeD-RM Prototype

9.1 Introduction

This chapter describes a case study that was conducted to evaluate the modules of the WeDRisk approach and the approach overall. Three distributed development software projects were involved in the case study. A prototype tool called WeD-RM was built and used by one of the developers involved (based on the WeDRisk approach). This is also presented and described in this chapter. This case is an extension of the previous empirical evaluation studies and cycles and was designed after reviewing some related case studies [2; 25]. The advantage of this evaluation study is that it is conducted with real distributed projects which reflect real experience.

9.2 Aims

There are two aims of this case study, which are:
Exploration Aim: WeDRisk is initiated with a list of potential WD development risks (exploring their significance and also exploring other potential risks).
Improvement Aim: Identify the weaknesses of the WeDRisk approach in order to tackle them and then improve the approach.

9.3 Case study Questions

The questions of this case study were prepared in order to help answer some of the main research questions, specifically those which are related to the evaluation of the WeDRisk approach and its list of initiated WD development potential risks.

These case study questions were extracted after specifying the aims of this case study and reviewing some software engineering evaluation methodologies and case studies [102; 104; 111; 113; 114]. The following are the questions of this case study:

Question Q1: *Which WD potential risks are practically more significant to WD development than others?*

Question Q2: *What sort of improvements need to be made to the WeDRisk approach?*

9.4 Involved Projects

Discussions were held with several WD developers in order to find suitable WD development project(s) to run in this case study. Four developers were invited to participate in the case study and they replied positively and accepted the invitation. However, only three of them stayed with this case study and signed the study consent form. All three were sited at Newcastle University and were involved in distributed software development projects. Some factors which support the selection of the case study projects are:

- Suitability of the projects (they are distributed software development projects).
- The fact that the projects' developers agreed to help and participate in the case study (such help could not be garnered from others).
- Flexibility (the development sites involved were at the same university, so there was direct contacts with them).

Due to imposed data confidentiality restrictions, the involved projects are referred to with reference names (**C1-01-2011**; **C1-02-2011**; **C1-03-2011**) to make them anonymous for others. Two of them are Europe projects and one is a UK project. The identification data of the projects involved in this case study are summarized in Table 9.1.

Table 9.1: Case Study - Involved Projects Summary

	C1-01-2011	**C1-02-2011**	**C1-03-2011**
No. of Staff	2-50	50-100	> 100
No. of Sites	6	16	8
Developer Experience with WD	4 Years	10 Years	> 30 years, Some in collocated software
Current Perspective (s)	Project and Product	Project and Process	Project and Process

9.5 Evaluation Aspects

This case study is concerned with the evaluation of three aspects related to the WeDRisk approach, which are:

- The significance of the WeDRisk initiated list of potential WD risks based on a real WD development environment. Although there are always differences in risk importance from project to project and from time to time, it is expected that some risks could have the same degree of significance for the majority of the WD development projects. This study is expected to identify them, so that they can be highlighted for WD developers and managers.
- Developer's feedback and suggestions about the WeDRisk modules and the approach overall.
- Evaluating learning simplicity and understandability level of the WeDRisk approach.

9.6 Methodology

The methodology of this case study is simple, since it depends on a number of semi-structured interviews between the researcher and the developers. During the interviews the researcher provided the developers with all necessary material including a list of WD potential risks, WeDRisk approach modules and necessary training materials and evaluation forms; then the researcher asked the developers to utilize the list of risks and the WeDRisk approach modules and evaluate them, providing feedback based on their project data. The methodology steps can be summarized as follows:

- Introducing the case study aims, objectives and stages to the developers and signing the consent form.
- Introducing the WeDRisk approach structure and modules and evaluation aspects to the developers and providing them with feedback forms.
- Providing the developers with a list of WD development potential risks at the first stage of the case study and asking them to utilize the list and specify which risks they have faced before, often they occurred and the possibility of facing others. The developers can also add other faced risks to the list.
- Training the developers on how to utilize the WeDRisk approach at the second stage and then asking them to evaluate their level of understanding and comment on any difficulties.

- Providing the developers with all WeDRisk related material and asking them to utilize the approach and evaluate its modules based on their WD development projects (under real work load), thereby providing their feedback and suggestions to improve the WeDRisk approach.

9.7 Avoiding bias

It is expected that any empirical study could involve some bias, which needs to be considered and avoided as much as possible. In this study, the following measures were taken to avoid the expected bias:

- No influence or restriction was placed on developers after providing the necessary material and the explanations.
- All case study projects and developers data were confidential. Reference numbers were used to refer to them and they were told that no identification data would be shared and there would not be any security threat to their project data.
- The developers were given the required time to evaluate the approach and provide their feedback.
- In order to reduce any researcher bias, as he might desire to see only the positive results, the original interviews, discussions and feedback documentation notes have been kept and discussed carefully.

9.8 Validity

The validity issue was considered during all the case study phases. The validity in this case study was covered as follows [104; 111; 113; 114]:

Construct validity: This means that the operational measures reflect and represent the research aims, questions and investigations exactly as the researcher intended. This is ensured by revising and simplifying the case study procedure and material (e.g. the questionnaire was reviewed with colleagues to ensure that the participants would understand the questionnaire questions exactly in the way the researcher intended).

Internal validity: This is to ensure that none of the factors affect other factors. In this regard, and to avoid the effect of time pressure, the developers were given enough time to study and respond. They were provided with the required clarification when it was needed. Moreover, there was no change in the case study environment during all the case study phases.

External validation: This concerns the generalization of the case study findings.

The case study is intended to reflect real work in three real distributed projects. Nevertheless, the researcher is not completely sure about the generalizability of this case study result.

Reliability: This concerns the replication of the study and data analysis. This case study depended on the researcher but it can be replicated by any other researcher after a short period of training to ensure familiarity with the approach, contact with the developers, conducting of interviews and collection of data. Thus, it is easy to replicate any number of times when there is a chance to conduct it. Moreover, the same findings can be obtained if the data analysis is replicated by any other researcher.

9.9 Data Source

The projects involved are the data source for this case study. Special forms were designed for this purpose and the developers were asked to be as accurate as possible. The collected data are qualitative and have been collected in the shape of numbers represented by ticks on the forms, as well as comments and suggestions. The developers were also able to express their opinions about the WeDRisk approach and its initial potential WD development risk list. The method of data collection can be considered as a direct method of collection as the researcher had direct contact with the case study subjects (developers). This direct method has many advantages since the researcher can have a high degree of control on the data collection (e.g. how it is collected, collection form, and the context).

9.10 Procedure

The procedure of this case study can be summarized as follows:

- A number of informal meetings and exchanged emails between the researcher and the developers took place before running the case study. This was to arrange how and when to run the case study and how the data would be collected.
- The case study materials were emailed or submitted personally to the developers. The material provided included the list of WD potential risks, evaluation form, WeDRisk presentation and guidelines. Meetings were arranged to provide explanations about the case study procedure and assigned tasks.
- The researcher arranged a number of interviews with the developers in order to submit material, discuss related issues, clarify some aspects or to collect evaluation results. The meetings and interviews were conducted individually to avoid any influence leading to bias and to maintain data security.

- The procedure of this case study was implemented during real development work time, so that the evaluation results reflected the reality of the WD development environment, risks and challenges.

In summary, the collection of the data in this case study depended on semi structured interviews (i.e. question order was not important) since this was more flexible for both the researcher and developers. Hence, at least two interview sessions were conducted individually with the developers of each project. Usually, in the first interview, the researcher introduced the case study objectives, explained its stages and how the data were to be collected, and provided the data forms. Then at the next interviews, the researcher collected the data, as the subjects (developers) had either answered the questions or prepared to answer them during the interviews.

9.11 Result

9.11.1 *Potential List of WD Risks Evaluation*

The significance of WD development potential risks (see Table 2.2) was evaluated in this case study. Three developers evaluated the risks during the course of their projects. Only the project and process development perspective risks were involved in this evaluation. This is because the development progress had not yet reached the product perspective at the time the case study was conducted. The evaluation aspects included the risks faced, the frequency of these risks, and the possibility of facing other risks.

As can be understood from Table 9.2, which summarizes the result of the risk evaluation, 63.8% of the risks had been faced before at different times in one or more of the projects involved. In more detail, 8.3% of the risks were faced by the three projects, 16.6% were faced by two projects and 38.8% of the risks were faced by only one project. On the other hand, 36.1% of the risks had not been faced before by any of the projects involved in the case study.

Table 9.2: Case Study - Faced Risks

	Risks Faced by 3 Projects	**Risks Faced by 2 Projects**	**Risks Faced by 1 Project**	**Risks Faced by 0 Project**
Risks' Numbers	2-19-33	6-11-14-20-23-29	3-4-6-9-13-17-21-24-25-26-27-31-34-36	1-5-7-8-10-12-15-18-22-28-30-32-35
Total	3	6	14	13

For simplicity purposes, in Tables 9.2 and 9.3, numbers are used to refer to the risks rather than names. For instance, in Table 9.2, the numbers 2,19 and 33 refer to risks *Inadequate customer requirement (see and change strategy)*, *Lack of requirement specification* and *Lack of face-to-face meetings* respectively. These numbers are the same numbers which are used to refer to the risks in the main list of potential WD risks (see Table 2.2).

The frequency of the risks faced is illustrated in Table 9.3. As can be understood from the table, some of the risks fell into more than one category, such as risk number 19, which is categorised as having Low, Medium and High frequency. In the same table, the risks which are flagged with * symbol are those which have the same level of frequency (Low, Medium or High) in more than one project.

Table 9.3: Case Study - Faced Risks Frequencies

	Low Frequency Risks	**Medium Frequency Risks**	**High Frequency Risks**
Risks' Numbers	11*-36-13-14-19-4-23-16-27-17-20*	26-33*-24-31-34-2*-19-23-25-6-9-29*	33-2-14-3-19-6-16-21
Total	11	12	8

Finally, almost all the developers predicted the possibility of facing risks which had not been previously encountered as low.

9.11.2 *WeDRisk Approach Overall Evaluation*

Developers C1-02-2011 and C1-03-2011 evaluated the overall WeDRisk approach. One of them sent his feedback via ordinary mail and the other one presented it during a meeting with the researcher. Comments and suggestions provided by the two developers are presented in Table 9.4.

9.12 Discussion

The case study results illustrate that there is a group of risks which have higher importance than others. This group consists of risks numbered 33, 2, and 19 (see Table 2.2 for risk names). These risks were the most common risks for the projects and they were faced by the three projects; moreover, they recorded medium or high frequencies. The second important group of risks are those which are faced by two projects (e.g. risks numbered 11, 14, 23,6,20 and 29). In general, by looking at Tables 9.2 and 9.3, it can be noticed that:

- The risks which were common to the three projects also had higher frequency.

Table 9.4: Case Study - WeDRisk General Evaluation Summary

	C1-02-2011	**C1-03-2011**
WeDRisk Main Architecture	It is Easy, but the word layer is not a suitable word (change it for another term); Implementation Phase can be in the shape of a cycle and the communication channel could be in a horizontal direction.	The architecture is acceptable and it is very easy to understand. Nevertheless, it can be improved by considering the Information flow between modules and fundamental I/O data for each module (how modules can communicate).
Clustering Criteria	What is the primary goal for splitting into 3P?	Consider compatibility with the product risks
WDF Estimation Matrix	Add something about cultural difference Side Discussion: The ranks could be better if you use 1, 1.25, 1.5, 1.75, 2 rather than 1,2 ,3 4,5 (this will reduce the difference)	Good pragmatic approach would like to see it used in example; Leave it open for other factors as it could be different from one project to another.
RM Customization Matrix	Agree	Agree with it but it needs guidance. Other suggested factors that can be added include: Strike requirement, Complexity of development, Dependency, Deep analysis and Customer and Stakeholder requirements
Sites Dependency Mapping	It should include weights. Might be difficult to estimate. I think that the WDF matrix is easier to use	Agree with all advantages- (note: it can be justified by examples)
Atypical	Dependes on urgency for urgent things you'd apply a simple emergency plan	Prepare a good example and try to work through with it.

- Risks numbered 33, 2 and 29 recorded medium frequency in more than one project, and risks numbered 11 and 20 recoded low frequency in more than one project.
- The developers ranked the risks which had not been previously encountered as low possibility risks. This could be due to lack of experience of these risks.

The second objective of this case study is the evaluation of the WeDRisk approach overall. To this end, the developers were taught how to use the WeDRisk approach and were provided with all support material and evaluation forms. Then, the developers were asked to evaluate the WeDRisk approach during the course of their projects and reflect this in the evaluation results. The developers were able to write whatever they wanted regarding the evaluation of the WeDRisk approach without any restrictions.

Generally, the results of this case study demonstrate that the WeDRisk approach is capable of managing WD development risks. However, as in Table 9.4, the results also establish that there are a number of suggested improvements which can be made to WeDRisk to make it more effective. Many of the suggested improvements were valuable and convincing and had thus already been considered in the later version of the WeDRisk approach (e.g. improving information flow and modules I/O, replacing the word "Layer" with another suitable term, leaving the WDF estimation matrix open for other factors, adding guidance into the customization matrix). On the other hand, some other suggestions have not yet been included and are discussed as follows:

- One of the developers suggested using a horizontal design for the WeDRisk approach architecture rather than a vertical one. In fact, whether the design is vertical or horizontal does not make any difference since there is no change in the architecture of the approach.
- Changing the ranking system used in the WDF estimation matrix from "**1, 2, 3, 4 ,5**" into "**1, 1.25, 1.5, 1.75, 2**" would be a good idea if the evaluation list mixed the RE and TREV values, but the risk evaluation in WeDRisk separates the RE from the TREV values, so there would be no effect. The advantage of this separation strategy is the ability to use two different evaluation equations at the same time.
- Regarding the suggestion of adding cultural differences into the WDF estimation matrix as a factor, it is indirectly considered by the sites distribution factor; moreover, it is also one of the WD risks and is not a factor.
- Some developers suggested including some other factors into the customization matrix (see Table 9.4). In fact, the design of the matrix allows including other factors; however, including a large number of factors makes the RM customization a time-consuming and more complicated process.

Finally, from the results, it can be understood that the case study has reached its aim by answering the following case study questions:

Question Q1: *Which WD potential risks are practically more significant to WD development than others?*

The case study involved three different distributed software development projects, which reflected three different experiences. According to the case study results, 63.8% of the risks were faced by the case study projects. This is a significant indication of the importance of these risks to WD development. On the other hand, the same risks were recorded with different frequency occurrence and their frequency differed from one project to another. Thus, the chance of facing the majority of these risks is high, but their frequency could be different from one situation or project to another. The fact that some risks were not faced in this study could be because: they belong to the product perspective, which is not involved in this case study; they were not faced by the projects sites involved; or because of the short duration of this case study. To sum up, although the study showed that there is a high vulnerability to the majority of the risks to be faced, it is worth giving more attention to the higher frequency and more significant ones, without ignoring the others.

Question Q2: *What sort of improvements need to be made to the WeDRisk approach?*

The case study result revealed that there are still some improvements which can be made to the WeDRisk approach. Consequently, these improvements have resulted in the newer version of the WeDRisk approach. The improvements included: WeDRIsk architecture (i.e. enhancing information flow, specifying modules I/O, replacing 'layer' with a better term), leaving the WDF estimation matrix open to include other factors and adding guidance to the customization matrix.

Finally, the approach was accepted by the developers. As evidence of this, two developers have shown an interest in utilizing the WeDRisk approach and have requested copies of its related material. As a result, the researcher had a number of discussions with one of the developers, who has built a prototype tool called WeD-RM based on the WeDRisk approach, regarding the utilization of WeDRisk in his project. The WeD-RM prototype is described in more detail below.

9.13 Case Study Limitations

The case study was limited by the number and the site distribution of the projects involved. Indeed it was difficult for the researcher to expand the number and the distribution of the projects involved due to the imposed data restrictions, time limitations and interests of the developers. The evaluation aspects were also limited due to the same reasons (three developers were able to evaluate the significance of WD list of potential risks, but the overall evaluation of the WeDRisk was evaluated by only two of them).

9.14 WeD-RM Prototype

WeD-RM is a simple prototype tool designed based on the WeDRisk approach. It is designed, implemented and used by developer number C1-02-2011 in this case study to identify and manage his distributed project risks. The early version of the prototype was initiated for some potential WD risks. Then the developer and researcher revised the prototype several times based on needs. The developer used the tool to manage his distributed project risks and, although the WeD-RM prototype does not cover all aspects of the WeDRisk approach, it covers several important aspects.

WeD-RM was developed using the Fossil system. The Fossil system was used to develop the WeD-RM prototype for the following reasons [115]:

- The developer already used Fossil for tracking and managing his distributed project activities so it was easy for him to embed the prototype into his management system.
- Fossil is already a distributed system and is easy to embed into distributed projects (it works as a server or client).

9.14.1 *WeD-RM Functionality overview*

The prototype mainly covers the clustering of risks from the 3P perspectives, WDF factor estimation, and estimation of the risks using the TREV equation. Specifically, the user can add new WD clustered risk cards, build project cards, estimate RE values, estimate WDF factors' values, and estimate TREV values. In the prototype, all processes' histories are tracked through a tickets system. WeD-RM is also supported by a risk repository and Wiki . It also supports ordinary system functions, such as adding new risks, editing risk cards and managing users' accounts. Samples of screenshots from the WeD-RM prototype are shown in Appendix D.

9.14.2 *WeD-RM Limitation*

Although the Fossil system was successfully used to develop the WeD-RM prototype, it is not suitable for building a complete application based on the WeDRisk approach. This is due to its limitations in terms of the design of the user interface and graphics, and mathematics function support. Other limitations are relevant to the WeD-RM prototype scalability and testing. WeD-RM was used at one site and has not been tested in other development areas; furthermore, it does not cover all WeDRisk aspects (e.g. atypical module and RM evaluation and evolution phase).

9.14.3 *WeD-RM Utilization Findings*

The adaptation of the WeDRisk approach to build the WeD-RM prototype, and the use of the WeD-RM prototype as part of a configuration management system for real distributed software development projects is evidence of the applicability of the WeDRisk approach, and this reveals the following points:

- The WeDRisk approach can be easily automated.

- WeDRisk is flexible enough to be embedded into other management approaches.
- Building and using this prototype provides an opportunity to identify new risks and make improvements to the WeDRisk approach.
- The success with the building of this embedded prototype has opened the door for building a complete, functional tool based on the WeDRisk approach.

The WeD-RM prototype can be considered another form of evaluation for the WeDRisk approach in real WD development. It shows that the WeDRisk approach is an implemental approach and is accepted by some WD developers. Nevertheless, the WeD-RM is limited in terms of the number of developers involved and the functions provided, but it has opened the door for the adoption of the WeDRisk approach by developing a robust tool based upon it.

9.15 Summary

This chapter reported a case study that was conducted to evaluate the WeDRisk approach. The importance of this case study is that it depended on three real distributed software development projects. In this case study, the significance of a list of potential WD development risks was evaluated. As a result, it was found that the majority of the evaluated risks (see Table 2.2) had been previously encountered by those involved in the projects and that these risks had different degrees of significance to the projects. Thus, the risks which occur most frequently are expected to be faced more often than others, although other risks should not be ignored. According to the results, the order in the WD potential risks list can be rearranged. For instance, the most significant risks could be at the top of the list. This is an indicative order to help developers to understand where to pay more attention and it is not related to risk priorities during RM evaluation, which depends only on TREV or RE values. In common with the previous evaluation methods, the comments, suggestions and observations of this case study have helped the researcher to improve the WeDRisk approach. Specifically, the case results were used to improve WeDRisk architecture (e.g. enhancing information flow, specifying module input and outputs, changing some terms used), leaving the WDF estimation matrix open to include other factors and add guidance to the customization matrix.

Finally, as a result of this case study, one of the developers adopted the WeDRisk approach and built a prototype tool called WeD-RM, based on WeDRisk. He embedded the WeD-RM into his project management system and used it for managing his distributed project risks. The main benefit of the development and utilization of the WeD-RM prototype is that it proved that WeDRisk is an applicable approach.

Chapter 10

Discussion and Conclusions

10.1 Introduction

This chapter concludes and discusses contributions, implications, limitations and future work directions of this research. The chapter also relates research findings to research questions and discusses how each question is answered in this thesis.

10.2 Summary of the Study

This research was motivated by the special challenges and risks that WD development faces and the shortcomings and weaknesses of the existing RM approaches to deal with them. Three main questions were asked in this research:

Question R-Q1: *What are the challenges of WD development for risk management?*

Question R-Q2: *Can existing software risk management address the identified challenges?*

Question R-Q3: *How can risk management approaches be adapted to tackle the weaknesses of the existing approaches?*

In order to answer these three questions and achieve the research aim, a research methodology was prepared and followed. Based on the proposed methodology, the WD development literature was reviewed and some WD software researchers and developers were interviewed. This helped to identify challenges, risks and RM needs for WD development. Existing RM approaches were also reviewed to investigate their ability to accommodate these challenges and RM needs. As a result, a number of weaknesses associated with these approaches were identified (see Chapter 3). The identified weaknesses were on two levels; management style level (e.g. management types offered) and coverage level (e.g. consideration of WD factors). Utilization of inappropriate or weak approaches to manage WD development risks could lead to:

- extra effort and time,
- partial or total lack of implementation of risk management because of inability, limitation or unsuitability of the RM approaches used, or
- zero consideration of some aspects, such as WD development factors.

Requirements for an approach to deal with risk management challenges and needs for WD development are specified in Table 3.5. Accordingly, the weaknesses of the existing approaches illustrate that none of the existing approaches is able alone to manage the WD development risks and it is difficult to make improvements to any of them. This shows the necessity of developing a new approach to manage WD development risks and be able to consider their challenges and factors. Furthermore, the following two reasons support the decision to build a new approach rather than make improvements to the existing ones:

- The weaknesses and strengths of the existing RM approaches are distributed among all of the reviewed approaches.
- Trying to improve one of the approaches means enlarging it and adding more unnecessary complications.

Therefore, in order to address the risk management needs for WD development, a new approach called WeDRisk is introduced and evaluated in this research. WeDRisk is an attempt to tackle the weaknesses of the existing approaches and to cover some risk management aspects that had not yet been covered. A modular structure strategy was used to design and develop the WeDRisk approach. This type of structure was adopted to ensure flexibility and an ability to evolve and to reduce the dependencies between the approach components. A number of evaluation cycles and methods were used to evaluate the approach, including peer reviews, experiments, expert evaluation and a case study which is supported with a prototype tool. Based on the evaluation results, the necessary improvements were made to the approach (after each evaluation cycle). The WeDRisk approach is intended to encourage WD developers and managers to practise risk management, even when the resources and time are limited.

10.3 Discussion

The research hypothesis has been broken down into research questions. Each research question has been answered in the research as follows:

Question R-Q1: *What are the challenges of WD development for risk management?*

In collocated software development, the development challenges are confined to specific team members, who are located at one development site and share the same development environment and time. However, in WD development this is different (i.e. development across different sites, different time zones, diffrent backgrounds, different cultures and different development environments). The differences between collocated and WD developments pose new challenges and risks to WD development, which are different from those of ordinary collocated software development. Identifying these challenges and risks helps to identify the challenges of WD development for RM. Therefore, in order to answer this question, the available WD development literature was reviewed. As a result, the WD development challenges and risks were identified. Some of these challenges and risks are unique to WD development, while others are common to all sections of the software industry. The challenges are presented in Section 2.4.2 and Section 2.4.4. The potential risks of WD development were also presented and listed in the same chapter (see Table 2.2). Thus, the needs and challenges for RM in WD development, which answer this question, were identified and highlighted in Section 2.6, and can be summarized as follows:

- ability to evolve to accommodate the continuous and rapid evolution of WD development,
- consideration of the WD development environment, challenges, characteristics, risks and factors,
- simplicity and the ability to cope with the different backgrounds of WD stakeholders and developers.

Accordingly, answering the question R-Q1 shed light on the need for approaches that are able to handle the specific WD developments risk management challenges and needs, which are different from the collocated ones. This need led finding an answer to the second research question R-Q2.

Question R-Q2: *Can existing software risk management address the identified challenges?*

Although the answer of the first research question R-Q1 has illustrated the RM challenges and needs for WD development, it does not confirm whether the existing RM approaches are able to accommodate these challenges. The current research question R-Q2 is intended to examine the ability of the existing approaches in this regard. Therefore, based on the specified needs of RM for WD development, which are identified by answering the first question R-Q1, a review of the existing software RM approaches became necessary to answer the second question R-Q2. For this purpose, and based on the specified WD development needs, investigation criteria were prepared to evaluate the existing approaches and explore their ability to fulfil the RM needs for WD development (see Section 3.2). Of the existing software risk management approaches, 12 were elaborated upon for detailed investigation, using the criteria mentioned. The approaches were selected either because they are dedicated to managing web or distributed development risks, or are believed to have some potential in this regard.

In fact, the R-Q2 question answered in Chapter 3 was important to specify whether there is still a need to build a new approach to manage WD risks based on the specified criteria (see Table 3.1), a systematic review was made on the selected approaches (see Table 3.4). The results of the review illustrated that the existing approaches show many weaknesses in their abilities to manage the WD development risks and also weaknesses related to software risk management (see page 40). Moreover, the approaches' strengths are distributed between them and none of the approaches is able alone to manage the WD development risks effectively. Generally, answering the question R-Q2 has specified the approaches' weaknesses, and specified the requirements for any approach that intends to manage WD risks (see Table 3.5). In this regard, the weaknesses of the existing approaches are:

- The existing approaches concentrate on the project perspective of software development and do not pay enough attention to the process and product perspectives.
- They are not ready to accommodate the continuous evolution of WD development.
- They do not consider the WD development risk factors (e.g. site distribution and dependencies).
- They lack preparedness for atypical risks.
- They are not flexible enough and offer only the deep type of risk management. Plain risk management is not offered.

To sum up, these weaknesses and the requirements specified have demonstrated the need for building a new approach to manage the WD development risks effectively.

Question R-Q3: *How can risk management approaches be adapted to tackle the weaknesses of the existing approaches?*

As a result of answering the second research question R-Q2, a requirement table for a new approach to manage WD development risks has been produced (see Table 3.5). This table draws the shape of any intended approach to manage WD development risks. Thus, it was found that there is a necessity to introduce a new approach to manage WD development risks rather than modify the existing ones because of the following reasons:

- None of the existing approaches is ready to be adopted as it is to manage WD risks (many aspects still need to be covered and considered).
- None of the existing RM approaches is able alone to fulfil all the needs of RM for WD development.
- Due to the weaknesses involved, any plan to improve any of the approaches means making the approach more complicated or too large.
- Features such as the ability to evolve, flexibility and customization of RM are difficult to add to any of the approaches as this would entail a total restructure of the approach as a whole.

Therefore, in order to bridge the gap in managing the WD development risks, a new approach has been built, called WeDRisk, which aims to tackle the weaknesses of the existing approaches and improve software risk management in general. The WeDRisk approach has been designed based on the RM requirement specification table, which resulted from answering the R-Q2 question (see Table 3.5).

The WeDRisk approach was described in detailed in Chapter 4. In brief, it consists of three phases (i.e. the RM Establishment, RM Implementation, RM Evaluation and Evolution phases), which are linked with a communication channel for data exchanges. The phases consist of modules, which contain components, steps, techniques and guidelines (see Section 4.2 for more details about the WeDRisk structure and its notations).

The WeDRisk approach is designed to be evolvable, flexible, simple to use and able to consider WD factors. It is provided with a risk repository, which is initiated with a list of WD development potential risks and their cards. In fact, the approach is designed to consider the risks from the 3P perspectives; therefore, it has a special module to cluster the risks from these 3P perspectives. This is to save time and effort and concentrate on a certain number of risks for the particular perspective which is under management. Moreover, compared with other software risk management approaches, WeDRisk is the only approach which has a module to deal with atypical risks to ensure continuous preparedness for atypical risks. In order to make risk management more flexible and perform in different conditions, as well as to encourage developers to practise risk management in all conditions even when resources or time are limited, the WeDRisk approach has been designed to be customizable. In other words, it offers two types of risk management (Plain and Deep). The plain type (essential RM) is intended for use when resources are limited or time is critical, whereas the deep type (full RM) is intended for ordinary times when there are sufficient resources and enough time, or when there is no experience with RM.

The modular design of the WeDRisk approach allows it to produce quicker results, since a minimum number of steps are triggered each RM cycle, with minimum dependency between the components. This modular design of the approach makes it more flexible and ready for any evolution. In addition, the approach is provided with an evolving module to accommodate any new RM needs for WD development.

WeDRisk modules are designed to be self-descriptive and use the minimum available RM data (e.g. customization and estimation matrixes). The approach contains an evaluation and auditing module to evaluate the efficiency of RM operations, monitor the risk situations and suggest any necessary corrective actions.

The research question R-Q3 has been answered in Chapters 4, 5, 6, 7, 8 and 9. In Chapter 4, the approach is introduced and described and is evaluated in the remaining chapters (5-9). As shown in Table 10.1, the RM requirements for WD development are revisited and related to the WeDRisk approach. As can be understood from the table, most of the requirements and weaknesses are addressed by the WeDRisk approach. In fact, since these requirements were revealed from a review of the existing risk management approaches (i.e. strengths and weaknesses were identified) and the RM needs for WD development (i.e. the gap was identified), Table 10.1 is indirectly a benchmark to show the ability of the WeDRisk approach to manage the WD development risks compared with the existing reviewed approaches.

Table 10.1: WD-RM Requirements Addressed By WeDRisk

Requirement	**Addressing by WeDRisk**
WD-R1: Usability (easy to learn, simple).	Simplicity is ensured via the modular structure and clear notations of WeDRisk. Evaluation results showed how, after a short training period, the subjects managed to understand and utilize WeDRisk and implement the assigned tasks smoothly with minimal support. Their feedback have also confirmed the simplicity and ease of use of WeDRisk.
WD-R2: It should produce quick results.	The estimation and customization matrixes depend on simple ticks, simple calculations and a minimum number of steps. Moreover the modules have short management cycles with minimum component dependencies to give quick results.
WD-R3: It should have the ability to evolve.	The WeDRisk modular structure and its evolving module ensure the evolvability of WeDRisk (i.e. modular structure makes it easy to add, modify, or plug-in components to the approach).
WD-R4: It should consider the characteristics of WD development.	WeDRisk is provided with a communication channel, initiated with the potential WD list of risks, and it considers the evolution of WD development and its factors.
WD-R5: It should consider the WD factors that might affect the risks (e.g. dependencies).	The WD factors are considered and estimated in the WDF estimation matrix and also included by the TREV estimation equation.
WD-R6: It should be a customisable approach.	In WeDRisk offers two RM types (Plain and Deep) via the RM Customization Module.
WD-R7: Ability to use the minimum information to manage the risks.	The estimation and customization matrixes and atypical module use the minimum available information.
WD-R8: Ability to deal with atypical risks.	The atypical module is dedicated to dealing with atypical risks type.
WD-R9: It should be initiated with a list of potential WD development risks.	The approach is initiated with a list of WD potential risks and it is ready for any updates.
WD-R10: Consideration of risks from 3P perspectives.	The approach considers the risks from the 3P perspectives and it is supported by a clustering module for this purpose.

10.3.1 *Evaluation of the WeDRisk Approach*

The WeDRisk approach is the answer to the first part of the third research question R-Q3, whereas the evaluation of WeDRisk is the answer to the second part of the same question. In addition to the WeDRisk module evaluation, the evaluation also included a list of WD development potential risks, which is used by the WeDRisk approach. In particular, the WeDRisk approach evaluation focused on the following perspectives:

- The importance of the potential WD list of risks
- The usability of the WeDRisk approach
- The usefulness of the WeDRisk approach
- The ease of use and understanding of the WeDRisk approach
- The coverage of the WD development factors

These evaluation perspectives were considered in different forms during the evaluation cycles. The focus on these perspectives is to see how WeDRisk is able to satisfy the research aim and manage the WD development risks easily while maintaining the specified RM requirements for WD development. In fact, only the novel aspects of the WeDRisk approach to RM were targeted for the evaluation. This is due to being new to the RM field, the need to avoid any replications, and to limited research time. Various evaluation methods, including peer reviews, experiments, case study and expert evaluation, have been used to evaluate the WeDRisk approach modules and aspects. The selection and use of the evaluation methods was subject to the availability and suitability of the method and resources available. Indeed, the evaluation cycles were implemented with time gaps between them. These time gaps were used to analyze the evaluation results and make improvements to the WeDRisk approach before conducting the next evaluation cycle. The following are summaries and contributions of the evaluation methods to the improvement of the WeDRisk approach.

Peer Reviews

A number of informal interviews and presentations with software researchers were conducted at the beginning and the medial stage of this research to explore needs and evaluate some aspects of WeDRisk, or to discuss some related issues. As a result, the peer reviews helped to identify the RM requirements for WD development and make early improvements to the proposed modules. In particular, this involved identifying the WD potential risks list, reviewing the early structural drafts of the WeDRisk approach, and extracting ideas to improve the atypical and estimation modules.

Experiment One

The results achieved by this experiment (see Chapter 6) showed that, although the results asserted the importance of the potential risk list in general, there was a difference in their perceived importance. This would help the developers and made them focus more on the most important ones. The consideration and clustering of the risks from the 3P perspectives gained significant support from the subjects. However, the utilization of the proposed clustering criteria took longer than was expected by the researcher, which could be due to factors such as the time needed for reading and understanding the criteria. In addition, the results illustrated that there are some risks which could affect more than one perspective, which needs to be considered during risk identification. Regarding the atypical risks, the subjects agreed that these pose a threat to WD development and therefore supported the absorbing concept to deal with them.

The results of this experiment have been used to improve many essential aspects of the early version of the WeDRisk approach, especially the clustering criteria and the building of an atypical risk module. It was also used to simplify and improve the training and utilization aspects of the approach. The differences in the risks' importance helps to reorder the risk list so that developers could focus more on those of higher importance.

Expert Evaluation

This evaluation is a reflection of the experience and knowledge of a number of international experts who work or have worked in software development and related fields and participated in this evaluation (see Chapter 7). A number of significant improvements to the WeDRisk approach were made based on the findings of this evaluation. The WeDRisk modules and components that were targeted for individual evaluation included the project card, RM customization, atypical risk absorbing concept, and WDF estimation matrix. On the other hand, the overall evaluation of WeDRisk included aspects such as WeDRisk architecture, understandability, usefulness and coverage of WD risk management aspects. The findings and improvements are presented in Section 7.10.3, which briefly includes the following points: the main architecture of WeDRisk has been improved; project card fields have been updated; WDF estimation matrix factor levels have been improved; the customization matrix has become more detailed and the self description matrix and includes new factors; the 3P clustering criteria have been revised; and the atypical module has been improved. All these improvements were undertaken before conducting the next evaluation cycle.

Experiment Two

This was the second controlled experiment (see Chapter 8) used to evaluate some of WeDRisk's novel aspects (i.e. estimation, customization and atypical risk modules, usefulness and usability of the approach). The main idea of this experiment was to inject a number of software risk situations and ask the subjects to utilize the evaluated modules to deal with these situations. The situations were a mixture of WD development and ordinary collocated situations. The subjects in this experiment were divided into experimental and control groups.

Generally, the results of this experiment exhibited how the WeDRisk modules successfully and effectively dealt with the injected risk situations. Specifically, the results showed that the modules were useful, understandable and easy to use to manage the situations. The results also showed that WeDRisk covers and considers the WD factors. On the other hand, the experiment results, suggestions and observations demonstrated that there are some improvements which can be made to the WeDRisk approach to make it better (i.e. the TREV equation can be generalized to estimate other types of software development risks by updating the WDF estimation matrix factors, and the clustering criteria could be improved to reduce the reading time).

The Case study

As reported in Chapter 9, a case study was conducted to evaluate some specific aspects of the WeDRisk approach, as well as the approach overall. The evaluation of the potential WD risks in this case study illustrated that WD developers and managers should pay more attention to the more important risks without ignoring the others, as the importance of risks could vary from one situation or development environment to another. The case study results showed that WeDRisk is a promising approach for managing WD development risks. Some improvements to the approach to make it more effective (e.g. information flow, modules I/O, replacing some of the terms used, such as the word ”layer” with other suitable ones, making the WDF estimation matrix ready for other factors, and adding guidance into the customization matrix), had already been considered and made in the later version of the WeDRisk approach. Other suggested improvements (e.g. changing the vertical design of WeDRisk into a horizontal design, and changing the ranking of the WDF estimation matrix) were discarded. The reasons for this are discussed in discussion Section 9.12.

Finally, as a result of this case study, two developers have shown an interest in applying WeDRisk and have requested copies of all its related material. Furthermore, one of them has already adopted the WeDRisk approach to manage his distributed development risks and for this purpose he built a prototype tool called WeD-RM, which is based on WeDRisk, and started using it (see Section 9.14 for more details). In fact, this provides evidence of the applicability of the WeDRisk approach. However, more work to build a complete tool is recommended.

As a result of the evaluation cycles, the WeDRisk approach has been improved. After each evaluation cycle, the results were analyzed and the necessary improvements to the approach were made. Nevertheless, some improvement suggestions have not yet been implemented (i.e. either they do not affect the current version of WeDRisk or they are out of this research scope and could be part of any future work). As an example of the non-implemented improvements, it was suggested that the ranking system of the WDF estimation matrix, which is 1, 2,3, 4, 5, be changed to 1, 1.25, 1.5, 1.75, 2. However, this modification does not need just extra time, but also needs extra evaluation effort. In the current version of WeDRisk , this may not have the desired effect because, as discussed in Chapter 4, the evaluation of the risks splits RE values from the TREV values. In general, the modifications and improvements were usually made after each evaluation cycle and are mentioned in the evaluation Chapters 5, 6, 7, 8 and 9.

In summary, the WeDRisk evaluation methods have illustrated that WeDRisk is an applicable approach, is useful, easy to use and understandable, and which also considers the WD development factors. This is also supported by the fact that developers have shown their interest in the WeDRisk approach and asked for copies of its related material. One of the developers has already built a prototype tool, embedded it into his distributed management system and utilized it. Moreover, the case study evaluation results have supported and reflected the results of other evaluation cycles (in particular the expert evaluation). All this ensures that the WeDRisk approach has met the planned aims and satisfied the requirements of RM for WD development, which was one of the research goals. Nevertheless, there are some limitations that are associated with the WeDRisk and future work chances are identified and described in followed sections.

10.4 Research Implications

10.4.1 *Implication of Theory*

The research findings have extended software risk management knowledge in terms of exploring new areas and concepts (i.e. considering and clustering the risks from 3P perspectives, atypical risk concept and preparedness for atypical risks, risk management customization, consideration of WD factors, and introduction of the TREV improved risk estimation equation). The research has gone beyond introducing and outlining these issues and concepts and has presented a new approach to managing WD development risks. Moreover, a clustered list of potential risks to WD development is presented and the significance of risks to WD development has been specified. Finally, the risk management requirements for WD development have been identified.

10.4.2 *Implications for Practice*

The significance of the research has resulted in a clearer understanding of the nature of WD development and its RM needs, specifying the weaknesses of the existing RM approaches, identifying the potential WD development risks and their level of significance, and introducing the WeDRisk management approach. This approach was designed to be easily applied, simple, flexible and evolvable and the evaluation findings show that this is the case. Moreover, two developers have showed their interest to get copies of the approach related material. One of them has built a prototype tool based on the WeDRisk approach (called WeD-RM) which is described in Section 9.14. RM customization is intended to increase the practice of RM in WD development, and software in general, as it allows the developers to practise plain or deep RM types rather than ignoring RM when resources or time are limited.

10.5 Research Contributions

As a result of this research, a number of contributions have been made to the body of knowledge (software risk management). Indeed, the major contribution is the WeDRisk approach, which is intended to manage WD development risks. Other contributions include identifying a list of potential risks to WD development and their significance levels, identifying the weaknesses of the existing risk management approaches in managing WD development risks, and specifying the requirements of RM for WD development. These contributions are described in more detail below.

10.5.1 *The WeDRisk Approach*

The WeDRisk approach is considered the major contribution of this research. It is intended to manage WD development risks, but can also be used to manage software development risks in general. The approach makes a number of contributions to the research field, which include:

- Introducing a new RM style to manage the risks by customizing the RM and offering two RM options (Plain and Deep). Many software developers and managers ignore the practice of RM due to resource or time limitations; therefore, introducing this new management style is to encourage developers and managers to practise RM under any conditions and with available resources.

- Highlighting the preparedness of the atypical risk concept and introducing a module to absorb them. This is to help developers/managers to act systematically to absorb atypical risks with minimum time and effort and avoid any confusion.
- Considering the risks from the 3P perspectives and introducing the 3P clustering criteria. In general, the existing risk management approaches focus on project perspective risks. WeDRisk clusters the risks from the 3P perspectives, which helps the developers to consider risks from all 3P perspectives while focusing on the particular perspective associated with the risk at hand.
- Considering WD factors via the WDF estimation matrix and TREV estimation equation. Estimation of the risk without consideration of the WD factors does not reflect the real severity and associated effects of the WD development risks. For example, the level of site dependency could have a significant effect on the importance of any risk if the dependency level is high; thus, without including this factor, the estimation is still correct but does not reflect the real impact of the risk.
- Providing WeDRisk with a WD updatable risk repository, which is initiated with the potential list of WD development risks and their cards. This list is to help the developer/manager to start the risk management process easily and early and become aware of these risks before they become a real threat to their developments.
- Evolvability and flexibility: WeDRisk has the ability to evolve and accommodate changes or new RM needs. Its modular structure, minimum dependencies between the modules and the evolution module support this feature. This evolution ability is to accommodate the rapid evolution of the WD development.
- WeDRisk's risk evaluation module being able to deal with two risk estimation equations (RE and TREV). For this purpose, WeDRisk splits the RE values from the TREV values, which allows managers to use either of these equations based on the risk management type used.

Finally, and as an example of WeDRisk's applicability, the WeDRisk approach has been adopted by one of the developers involved in the evaluation case study and a prototype tool has been built based on the approach and embedded in the distributed management system. The evaluation result illustrated that WeDRisk is a flexible, easy to use and understandable approach. However, there are some related limitations are discussed in the following section.

10.5.2 *Risk Significance*

In addition to the research's main contribution, the available literature of WD software development was reviewed to identify the challenges and potential risks to WD development. Moreover, the significance of the identified risks was specified via different evaluation cycles. This was in order to initialize the risk repository and help the developers/managers to give more attention to the most important risks so that they could save time and effort. Meanwhile, the specific RM needs for WD development were identified and used to explore the abilities of the existing RM approaches to manage WD development risks.

10.5.3 *Weaknesses of Existing Approaches*

The research results provide evidence of how the existing RM approaches are still suffering from many weaknesses. The abilities of the existing software risk management approaches to satisfy RM needs of WD development were reviewed. The results of this review illustrated that the previous approaches were not able to properly manage WD development risks in particular, or software risks in general (see Section 3.5). This review was important to identify the gap in risk management for WD development, and to specify the requirements for a new approach to manage WD development risks, which were used to build and evaluate the WeDRisk approach.

10.6 Limitations

Although, the research has introduced the WeDRisk approach, which shows significant abilities to satisfy the RM requirements for WD development by tackling many weaknesses of the existing approaches and developing valuable, new concepts (e.g. consideration of risks from the 3P perspectives, RM customization and preparedness for atypical risks, introducing the TREV equation with consideration to WD factors, and flexibility and evolvability), but there are still some limitations, namely:

- In the current WeDRisk version, the WDF estimation matrix is limited to three WD development factors. These factors can be updated or changed according to the type of the development (e.g. cloud development might have different factors).

- The prototype tool developed is limited in terms of its functionality and in the coverage of the WeDRisk approach modules and aspects.
- The evaluation case study was limited in terms of the number and type of developers involved. It would be better if it reflected a wider range of WD developers.
- The subjects involved in the experiments were from the same university. This was due to the imposed constraints, resources and time limitations, as well as the availability of other subjects. Therefore, other evaluation methods were used to avoid bias and make the results more generalizationble.
- Waiting for atypical risks to occur would have been beyond this research time (i.e. it could have taken one day, months or more to occur); thus, a predesigned situation was used to emulate an atypical risk situation and evaluate its module.

10.7 Implications for Further Research

We believe that the research presented in this thesis can significantly improve the risk management practicing for WD development and software risk management in general as it has introduced the new WeDRisk approach, which developed new concepts and aspects in the software risk management field. Thus, the door is open for more future research work to extend and improve this work, in order to make generalization, cover other development areas, and to tackle the limitations of this research. The following are some opportunities that stem from this research and can be addressed by future research:

- WeDRisk contributions like customization, using two estimation equations, evolution ability, preparedness to atypical risks and scalability of WDF estimation matrix could make the approach applicable for managing different modern software development risks if it is updated with their factors and risks. Thus, in order to expand this work to cover other risk management areas, such as “mobile” or “cloud” developments, more future research is required to study and specify the actual RM needs for these developments. For example, the relevant challenges, risks and factors of “cloud” need to be identified. Information security, disaster recovery, third-party/service level management and history of cloud service providers are examples of cloud risk areas [116; 117] that could be studied.
- As a result of the evaluation cycles of WeDRisk approach, there is a suggestion to

modify (change) the ranking system of the WDF estimation matrix from 1, 2, 3, 4, 5 into 1, 1.25, 1.5, 1.75, 2. The justification for that is to reduce the variance between the estimated RE and TREV values. In fact, with the current version of WeDRisk approach there is no problem with the existing ranking system, since the WeDRisk evaluation module separates the RE values from TREV values. However, this issue could be investigated more by empirical studies.

- The WD factors (i.e. sits distributions and dependencies) which are used by the WDF estimation matrix could be mapped and visualized using notations to represent the dependencies and relations. This issue could be studied and notations could be designed for this purpose.
- The evaluation result showed some promised issues regarding the applicability of WeDRisk approach. One of case study developers has built a prototype tool called WeD-RM based on the approach and already utilized it. Therefore, in order to increase the practice level of risk management in WD development, it is recommended to build a complete tool based on the WeDRisk approach and make it available to WD developers. This could also help to evaluate and improve the WeDRisk.

Finally, since the WeDRisk approach is intended to be an applicable approach, the evaluation of WeDRisk could be replicated in industrial WD development environment, which could add more improvements on the WeDRisk approach.

References

[1] B. Khoshroo and H. Rashidi, "Distributed development - A literature analysis on challenges and solutions," in *Proceedings of the 5th IEEE-GCC Conference and Exhibition*, pp. 1–5, March 2009. x, 15, 16

[2] T. Al-Rousan, S. Sulaiman, and R. Salam, "Project management using risk identification architecture pattern RIAP model: A case study on a web-based application," in *Proceedings of the Asia-Pacific Software Engineering Conference*, APSEC09, pp. 449–456, IEEE Computer Society, Dec 1-3 2009. x, 26, 27, 28, 142

[3] J. Kontio, G. Getto, and D. Landes, "Experiences in improving risk management processes using the concepts of the Riskit method," in *Proceedings of the 6th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, SIGSOFT '98/FSE-6, pp. 163–174, ACM, 1998. x, xii, 31, 32

[4] B. Boehm, "Software risk management: Principles and practices," *IEEE Software*, vol. 8, no. 1, pp. 32–41, 1991. xii, 9, 10, 11, 12, 59, 60

[5] A. Lamersdorf, J. Munch, A. Fernandez-del Viso Torre, and C. Rebate Sanchez, "A Risk-Driven model for work allocation in global software development projects," in *Proceedings of the 6th IEEE International Conference on Global Software Engineering*, ICGSE11, pp. 15–24, IEEE Computer Society, Aug. 15-18 2011. xii, 15, 23, 24

[6] A. Lamersdorf, J. Munch, A. Viso Torre, S. C., M. Heinz, and D. Rombach, "A Rule-Based model for customized risk identification in distributed software development projects," in *Proceedings of the 5th IEEE International Conference on Global Software Engineering*, ICGSE10, pp. 209–218, IEEE Computer Society, Aug. 23-26 2010. xii, 15, 24, 25

[7] S. Betz, S. Hickl, and A. Oberweis, "Risk management in global software development process planning," in *Proceedings of the 37th EUROMICRO Conference*

on Software Engineering and Advanced Applications, SEAA11, pp. 357–361, IEEE Computer Society, Aug 30, - Sep 2 2011. xii, 2, 26

[8] T. Al-Rousan, S. Sulaiman, and R. Salam, "WPRiMA tool: Managing risks in web projects," *World Academy of Science, Engineering and Technology*, vol. 50, pp. 619–625, 2009. xii, 27

[9] J. Gorski and J. Miler, "Towards an integrated environment for risk management in distributed software projects," in *Proceedings of the 7th European Conference on Software Quality*, ECSQ2002, pp. 1–12, June 10-12 2002. xii, 28, 29

[10] G. Roy, "A risk management framework for software engineering practice," in *Proceedings of the 2004 Australian Software Engineering Conference*, ASWEC '04, pp. 60– 67, IEEE Computer Society, April 13 - 16 2004. xii, 29, 30

[11] A. Keshlaf and K. Hashim, "A model and prototype tool to manage software risks," in *Proceedings of the The First Asia-Pacific Conference on Quality Software*, APAQS '00, pp. 297–305, IEEE Computer Society, 2000. xii, 10, 11, 31, 32

[12] R. Williams, "The CMMI-RSKM process area as a risk management standard," in *Proceedings of the 16th Annual International Symposium of the International Council on Systems Engineering*, INCOSE, July 8 - 14 2006. xii, 33

[13] IEEE-Std, "IEEE guide-adoption of the project management institute PMI(R) standard a guide to the project management body of knowledge PMBOK(R) guide - fourth edition," *IEEE Std 1490-2011*, pp. 1 –508, 2011. xii, 34, 35

[14] J. Presson, L. Mathiassen, B. Jesper, T. Madsen, and F. Steinson, "Managing risks in distributed software projects: An integrative framework," *IEEE Transaction on Software Engineering*, vol. 56, no. 3, pp. 508–532, 2009. xii, 14, 18, 22, 34, 36

[15] L. Wallace, M. Keill, and A. Rai, "How software project risk affects project performance: An investigation of the dimensions of risk and an exploratory model," *Decision Sciences*, vol. 35, no. 2, pp. 289–321, 2004. xii, 36, 37

[16] A. Keshlaf and S. Riddle, "Risk management for web and distributed software development projects," in *Proceedings of the 5th International Conference on Internet Monitoring and Protection*, ICIMP'10, pp. 22–28, IEEE Computer Society, May 09 - 15 2010. 1, 6, 11, 21

[17] H. Yong, C. Juhua, R. Zhenbang, M. Liu, and X. Kang, "A neural networks approach for software risk analysis," in *Proceedings of the 6th IEEE International Conference on Data Mining - Workshops*, ICDMW '06, pp. 722–725, IEEE, Dec. 18 -22 2006. 1, 11

[18] P. Bannerman, "Risk and risk management in software projects: A reassessment," *The Journal of Systems and Software - Elsevier*, vol. 81, no. 12, pp. 2118 – 2133, 2008. 1, 11

[19] A. Tiwana and M. Keil, "Functionality risk in information systems development: An empirical investigation," *IEEE Transactions on Engineering Management*, vol. 53, no. 3, pp. 412–425, 2006. 1

[20] K. T. Hijazi, H. and A. Alarabeyyat, "A review of risk management in different software development methodologies," *International Journal of Computer Applications*, vol. 45, no. 7, pp. 8–12, 2012. 1, 20

[21] I. de Farias Junior, R. de Azevedo, H. de Moura, and D. Martins da Silva, "Elicitation of communication inherent risks in distributed software development," in *Proceedings of the IEEE 7th International Conference on Global Software Engineering Workshops*, ICGSEW '12, pp. 37–42, IEEE, Aug. 2012. 1, 17

[22] L. Hai, "The four ps in an undergraduate software engineering course," in *Proceedings of the 37th Annual Frontiers in Education Conference - Global Engineering: Knowledge Without Borders, Opportunities Without Passports*, FIE'07, pp. S4E–7–S4E–12, IEEE, Oct. 2007. 1

[23] A. Keshlaf and S. Riddle, "An empirical study of web and distributed software risks from three perspectives: Project, process and product," in *Proceedings of the 6th International Conference on Internet Monitoring and Protection*, ICIMP11, pp. 38–44, IARIA, March 20-25 2011. 1, 6, 55, 89

[24] S. Misra, U. Kumar, V. Kumar, and M. Shareef, "Risk management models in software engineering," *International Journal of Process Management and Benchmarking (IJPMB)*, vol. 2, no. 1, pp. 59–70, 2007. 1, 22

[25] B. Boehm, J. Kwan, D. Port, A. Shah, and R. Madachy, "Using the WinWin Spiral Model: A case study," *IEEE Computer*, vol. 31, no. 7, pp. 33–44, 1998. 1, 22, 142

[26] A. Keshlaf and S. Riddle, "Web and distributed software development risks management: WeDRisk approach," *International Journal On Advances in Software*, vol. 3, no. 3-4, pp. 447–460, 2010. 2, 6, 19

[27] D. Smite and J. Borzovs, “Managing uncertainty in globally distributed software development projects.” http://www.lu.lv/materiali/apgads/raksti/733_pp_9-23.pdf, 2008. [Online; accessed 13-Aug-2012]. 2, 16

[28] T. Al-Rousan, S. Sulaiman, and R. Abdul Salam, “Risk analysis and web project management,” *Journal of software*, vol. 4, no. 6, pp. 614–621, 2009. 2, 13

[29] R. Williams, G. Pandelios, and S. Behrens, “Software risk evaluation SRE method description (version 2.0),” tech. rep., Software Engineering Institute (SEI), December 1999. 9

[30] S. Sherer, “The three dimensions of software risk: technical, organizational, and environmental,” in *Proceedings of the 28th Hawaii International Conference on System Sciences*, pp. 369 –378, IEEE Computer, Jan 3-6 1995. 9

[31] K. Hashim and A. Keshlaf, “An approach to sharing solutions to software project management problems,” in *Proceedings of the International Conference on Information Management and Engineering*, ICIME ’09, pp. 694–697, IEEE Computer Society, April 3-5 2009. 10

[32] W. Han and S. Huang, “An empirical analysis of risk components and performance on software projects,” *Journal of Systems and Software*, vol. 80, no. 1, pp. 42–50, 2007. 11

[33] Y. Kwak and J. Stoddard, “Project risk management: Lessons learned from software development environment,” *Technovation*, vol. 24, no. 11, pp. 915–920, 2004. 11

[34] S. Islam, “Software development risk management model: A goal driven approach,” in *Proceedings of the ESEC/FSE on Doctoral Symposium*, ESEC/FSE Doctoral Symposium ’09, pp. 5–8, ACM, 2009. 11

[35] J. Esteves, J. Pastor, N. Rodriguez, and R. Roy, “Implementing and improving the SEI risk management method in a university software project,” *Latin America Transactions, IEEE (Revista IEEE America Latina)*, vol. 3, no. 1, pp. 90–97, 2005. 11

[36] M. Kajko-Mattsson and J. Nyfjord, “State of software risk management practice,” *IAENG International Journal of Computer Science - On-line Issue*, vol. 35, no. 4, 2008. 11, 14

[37] T. Berners-Lee, “The World Wide Web initiative.” http://assets.cs.ncl.ac.uk/seminars/92.pdf, 1993. [Online; accessed 12-Nov-2012]. 12

[38] Wikipedia, "Web application." http://en.wikipedia.org/wiki/Web_application, 2012. [Online; accessed 18-Sep-2012]. 12

[39] H. Gellersen and M. Gaedke, "Object-oriented web application development," *Internet Computing, IEEE*, vol. 3, pp. 60 –68, Jan/Feb 1999. 12

[40] Y. D. Wang and N. Zahadat, "Teaching web development in the web 2.0 era," in *Proceedings of the 10th ACM Conference on SIG-Information Technology Education*, SIGITE '09, pp. 80–86, ACM, Oct. 22 - 24 2009. 12

[41] V. Beal, "The difference between the internet and world wide web." http://www.webopedia.com/DidYouKnow/Internet/2002/Web_vs_Internet.asp, 2011. [Online; accessed 21-01-2013]. 12

[42] L. Baresi and S. Morasca, "Three empirical studies on estimating the design effort of web applications," *ACM Trans. Softw. Eng. Methodol.*, vol. 16, no. 4, pp. 15–40, 2007. 12, 13

[43] A. Taivalsaari, "Mashware: The future of web applications," tech. rep., Sun Microsystems Laboratories, February 2009. 13, 14, 18, 22

[44] G. Kappel, B. Proll, S. Reich, and W. Retschitzegger, *Web Engineering the Discipline of Systematic Development of Web Application*. John Wiley & Sons, Ltd, 2006. 13, 14, 22

[45] J. Offut, "Quality attributes of web software applications," *IEEE Software*, vol. 19, no. 2, pp. 25–32, 2002. 13, 18, 22

[46] F. Donini, M. Mongiello, M. Ruta, and M. Totaro, "A model checking-based method for verifying web application design," *Electronic Notes in Theoretical Computer Science*, vol. 151, no. 2, pp. 19–32, 2006. 13, 14

[47] F. Ahmad, B. F., and M. Husni, "Investigating the awareness of applying the important web application development and measurement practices in small software firms," *International Journal of Computer Science & Information Technology (IJCSIT)*, vol. 3, no. 6, pp. 147–158, 2011. 13

[48] V. Bruno, A. Tam, and J. Thom, "Characteristics of web applications that affect usability: a review," in *Proceedings of the 17th Australia conference on Computer-Human Interaction: Citizens Online: Considerations for Today and the Future*, OZCHI '05, pp. 1–4, 2005. 13, 14, 18, 22

[49] D. Lowe and B. Henderson-Sellers, "Characteristics of web development processes," in *Proceedings of the International Conference on Advances in Infrastructure for Electronic Business, Science and Education on the Internet*, SSGRR01, pp. 1–12, 2001. 13

[50] S. Maurya1 and G. Shankar, "Maintainability assessment of web based application," *Journal of Global Research in Computer Science*, vol. 3, no. 7, pp. 30–33, 2012. 13

[51] D. Reifer, "Ten deadly risks in internet and intranet software development," *IEEE Software*, vol. 19, no. 2, pp. 12–14, 2002. 13

[52] C. M. Beise, "IT project management and virtual teams," in *Proceedings of the 2004 SIGMIS Conference on Computer Personnel Research: Careers, Culture, and Ethics in a Networked Environment*, SIGMIS CPR '04, pp. 129–133, ACM, 2004. 13, 14, 18

[53] B. Boehm, "A view of 20th and 21st century software engineering," in *Proceedings of the 28th International Conference on Software Engineering*, ICSE '06, pp. 12–29, ACM, 2006. 13

[54] T. Berners-Lee, "White paper: Effectively managing high-performing business-critical web application." http://zones.computerworld.com/ca/, 2008. [Online; accessed 16-Jan-2011]. 13, 14, 22

[55] C. West, "Five common project management challenges." http://www.compaid.com/caiinternet/ezine/west-fivechallenges.pdf, 2009. [Online; accessed 12-Nov-2012]. 13

[56] S. Murugesan, *Web Application Development: Challenges And The Role Of Web Engineering*. Web Engineering, Springer, 2008. 13, 15

[57] S. Murugesan and Y. Deshpande, "Meeting the challenges of web application development: the web engineering approach," in *Proceedings of the 24th International Conference on Software Engineering*, ICSE '02, pp. 687–688, ACM, 2002. 13

[58] E. Mendes and N. Mosley, *Web Engineering*. Berlin Heidelberg: Springer-Verlag, 2006. 13, 18, 22

[59] T. Al-Rousan, *Web Engineering: Awareness on the Impotance of Early Riks Management in Web Project*. Saarbrucken, Germany: LAP LAMBERT Academic Publishing, 2011. 13, 17, 18

[60] B. Behkamal, M. Kahani, and M. Akbari, "Customizing ISO 9126 quality model for evaluation of B2B applications," *Information and Software Technology*, vol. 51, no. 3, pp. 599–609, 2009. 14, 22

[61] J. Tian, S. Rudrarjuand, and Z. Li, "Evaluating web software reliability based on workload and failure data extracted from server logs," *IEEE Transaction on Software Engineering*, vol. 30, no. 11, pp. 754–769, 2004. 14, 22

[62] J. Kontio, M. Hoglund, J. Ryden, and P. Abrahamsson, "Managing commitments and risks: Challenges in distributed agile development," in *Proceedings of the 26th International Conference on Software Engineering*, ICSE '04, pp. 732–733, IEEE Computer Society, May 23-28 2004. 14, 22

[63] M. F. Rabbi and K. O. B. Mannan, "A review of software risk management for selection of best tools and techniques," in *Proceedings of the 9th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, SNPD '08, pp. 773–778, IEEE Computer Society, Aug. 6-8 2008. 14, 22

[64] W. B. Glisson and R. Welland, "Web development evolution: The assimilation of web engineering security," in *Proceedings of the 3th Latin American Web Congress*, LA-WEB '05, pp. 49–, IEEE Computer Society, oct.-2 nov. 2005. 14, 18, 22

[65] H. Romero, J. Haddad, and Molero, "A methodological tool for asset identification in web applications: Security risk assessment," in *Proceedings of the 4th International Conference on Software Engineering Advances*, ICSEA '09, pp. 413–418, IEEE Computer Society, 2009. 14, 22

[66] Y. Huang, C. Tsai, D. Lee, and S. Kuo, "Non-detrimental web application security scanning," in *Proceedings of the 15th International Symposium on Software Reliability Engineering*, ISSRE '04, pp. 219–230, IEEE Computer Society, 2004. 14, 22

[67] X. Ge, R. F. Paige, F. A. Polack, H. Chivers, and P. J. Brooke, "Agile development of secure web applications," in *Proceedings of the 6th International Conference on Web Engineering*, ICWE '06, pp. 305–312, ACM, 2006. 14, 22

[68] S. Willis, *Software quality and software testing in internet times*. New York, NY, USA: Springer-Verlag New York, Inc., 2002. 14, 18, 22

[69] C. Iacovou and R. Nakatsu, "A risk profile of offshore-outsourced development projects," *Communications of the ACM*, vol. 51, no. 6, pp. 89–94, 2008. 14, 18

[70] T. Mikkonen and A. Taivalsaari, "The mashware challenge: bridging the gap between web development and software engineering," in *Proceedings of the Workshop on Future of Software Engineering Research at the 18th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, FoSER10, pp. 245–250, Nov. 2010. 14

[71] S. Jeary, K. Phalp, and J. Vincent, "An evaluation of the utility of web development methods," *Software Quality Control*, vol. 17, no. 2, pp. 125–150, 2009. 14, 15

[72] D. Herbsleb and D. Moitra, "Guest editors' introduction: Global software development," *IEEE Softw.*, vol. 18, no. 2, pp. 16–20, 2001. 15

[73] S. Krishna, S. Sahay, and G. Walsham, "Managing cross-cultural issues in global software outsourcing," *Commun. ACM*, vol. 47, no. 4, pp. 62–66, 2004. 15

[74] M. Jimenez, M. Piattini, and A. Vizcaino, "Challenges and improvements in distributed software development: A systematic review," *Advances in Software Engineering*, vol. January, no. 3, pp. 1–14, 2009. 15, 18

[75] M. Malarvannan, "Managing offshore software teams." `http://outsourceportfolio.com/wp-content/themes/revol/pdfs/rrs/156.pdf`, 2009. [Online; accessed 15-Nov-2012]. 15, 22

[76] R. Prikladnicki, J. Audy, and R. Evaristo, "Global software development in practice lessons learned," *Software Process Improvement and Practice*, vol. 8, no. 4, pp. 267–281, 2003. 15, 18

[77] C. Ebert, B. Murthy, and N. Jha, "Managing risks in global software engineering: Principles and practices," in *Proceedings of the 2008 IEEE International Conference on Global Software Engineering*, ICGSE '08, pp. 131–140, IEEE Computer Society, Aug. 17-20 2008. 15, 18

[78] A. Marques, R. Rodrigues, and T. Conte, "Systematic literature reviews in distributed software development: A tertiary study," in *Proceedings of the 7th IEEE International Conference on Global Software Engineering*, ICGSE'12, pp. 134 –143, IEEE, Aug. 2012. 15

[79] D. Damian and D. Moitra, "Guest editors' introduction: Global software development: How far have we come?," *IEEE Software*, vol. 23, no. 5, pp. 17–19, 2006. 15

[80] B. Sengupta, S. Chandra, and V. Sinha, “A research agenda for distributed software development,” in *Proceedings of the 28th International Conference on Software Engineering*, ICSE ’06, pp. 731–740, ACM, May 20 28 2006. 16, 18, 22

[81] K. Nidiffer and D. Dolan, “Evolving distributed project management,” *IEEE Software*, vol. 22, no. 5, pp. 63–72, 2005. 16, 18, 22

[82] A. Reed and L. Knight, “Project risk differences between virtual and collocated teams,” *Journal of Computer Information Systems*, vol. 51, no. 1, pp. 19–30, 2010. 17, 18

[83] P. Pressman and D. Low, *Web Engineering A Practitioners Approach*. International Edition: Mc Graw Hill, 2009. 18, 22

[84] D. Meixia and Jiping, “Experience and inspiration of risk management of indias software and service outsourcing,” *Management Science and Engineering*, vol. 6, no. 3, pp. 34–38, 2012. 20

[85] IEEE-Std-1540-2001, “IEEE standard for software life cycle processes - risk management,” tech. rep., IEEE, 2001. 22

[86] CambridgeDictionariesOnline. `http://dictionary.cambridge.org/dictionary/business-english/`, 2012. [Online; accessed 31-March-2012]. 23

[87] M. Carr, S. Konda, I. Monarch, C. Walker, and F. Ulrich, “Taxonomy-based risk identification,” tech. rep., Software Engineering Institute (SEI), June 1993. 30

[88] J. Kontio, “The Riskit method for software risk management, version 1.00 cs-tr-3782 / umiacs-tr- 97-38,” tech. rep., University of Maryland, 1997. 31

[89] J. Smith, S. Bohner, and D. McCricard, “Project management for the 21st century: supporting collaborative design through risk analysis,” in *Proceedings of the 43rd Annual Southeast Regional Conference - Volume 2*, ACM-SE 43, pp. 300–305, ACM, March 18-20 2005. 31

[90] SEI-Team, “Cmmi for development Version 1.3,” tech. rep., Software Engineering Institute (SEI) - Carnegie Mellon University, November 2010. 33

[91] J. Dhlamini, I. Nhamu, and A. Kaihepa, “Intelligent risk management tools for software development,” in *Proceedings of the 2009 Annual Conference of*

the Southern African Computer Lecturers' Association, SACLA '09, pp. 33–40, ACM, July 2-11 2009. 33, 34

[92] C. Pan and Y. Chen, "An optimization model of CMMI-based software project risk response planning," *International Journal of Applied Mathematics and Computer Sciences*, vol. 1, no. 3, pp. 155–159, 2005. 33

[93] D. Callegari and R. Bastos, "Project management and software development processes: Integrating RUP and PMBOK," in *Proceedings of the 2007 International Conference on Systems Engineering and Modeling*, ICSEM '07, pp. 1–8, IEEE, March 2007. 34

[94] W. R. Duncan, "A guide to the project management body of knowledge PMBOK-PMI," tech. rep., Project management Institute, 1996. 34

[95] E. Odzaly, D. Greer, and P. Sage, "Software risk management barriers: An empirical study," in *Proceedings of the 3rd International Symposium on Empirical Software Engineering and Measurement*, ESEM '09, pp. 418 – 421, IEEE Computer Society, Oct. 2009. 52

[96] K. Lyytinen, L. Mathiassen, and J. Ropponen, "Attention shaping and software risk? a categorical analysis of four classical risk management approaches," *Journal of Information Systems Research*, vol. 9, no. 3, pp. 233–255, 1998. 59, 60

[97] G. A. Stoneburner G. and F. A., "Risk management guide for information technology systems." `http://csrc.nist.gov/publications/nistpubs/800-30/sp800-30.pdf`, July 2002. [Online; accessed 16-April-2012]. 60

[98] Commonwealth-Of-Australia, "Implementation planning - cabinet submissions and new policy proposals - qualitative risk analysis." `http://www.dpmc.gov.au/implementation/docs/qualitative_risk_analysis.pdf`, 22 September 2010. [Online; accessed 16-April-2012]. 61

[99] G. J. K. R. Chisholm, G.H. and J. McHugh, "Peer review of the trusted software methodology," tech. rep., Decision and Information Sciences Division, Argonne National Laboratory, Argonne, February 1994. 85

[100] P. Dunphy, A. Heiner, and N. Asokan, "A closer look at recognition-based graphical passwords on mobile devices," in *Proceedings of the 6th Symposium on Usable Privacy and Security*, SOUPS '10, pp. 3:1 – 3:12, July 14-16 2010. 90

[101] R. Maxion and K. Killourhy, “Keystroke biometrics with number-pad input,” in *Proceedings of the IEEE/IFIP International Conference on Dependable Systems and Networks*, DSN’10, pp. 201–210, IEEE, July 2010. 90, 118, 120

[102] B. Kitchenham, S. Pfleeger, L. Pickard, P. Jones, D. Hoaglin, K. Emam, and J. Rosenberg, “Preliminary guidelines for empirical research in software engineering,” *IEEE Transactions on Software Engineering*, vol. 28, no. 8, pp. 721–734, 2002. 90, 143

[103] M. Nordio, H. Estler, B. Meyer, J. Tschannen, C. Ghezzi, and E. Di Nitto, “How do distribution and time zones affect software development? a case study on communication,” in *Proceedings of the 6th IEEE International Conference on Global Software Engineering*, ICGSE, pp. 176–184, Aug. 2011. 92, 118, 120, 122

[104] D. Sjoberg, T. Dyba, and M. Jorgensen, “The future of empirical methods in software engineering research,” in *Proceedings of the Future of Software Engineering*, FOSE ’07, pp. 358–378, IEEE Computer Society, 2007. 92, 122, 143, 145

[105] D. Bertram, “Likert scales.” `http://poincare.matf.bg.ac.rs/~kristina/topic-dane-likert.pdf`, 2012. [Online; accessed 31-July-2012]. 96

[106] P. Billiet, “The Mann-Whitney U-Test.” `http://www.saburchill.com/IBbiology/downloads/001.pdf`, 2003. [Online; accessed 01-Aug-2012]. 99

[107] M. Li and C. Smidts, “A ranking of software engineering measures based on expert opinion,” *Software Engineering, IEEE Transactions on*, vol. 29, no. 9, pp. 811–824, 2003. 108, 110

[108] A. Karoulis, S. Demetriadis, and A. Pombortsis, “Comparison of expert-based and empirical evaluation methodologies in the case a CBL environment: the ”orestis” experience,” *Computers and Education*, vol. 47, pp. 172–185, Sep 2006. 110

[109] M. Zelkowitz, D. Wallace, and D. Binkley, “Lecture notes on empirical software engineering,” ch. Experimental validation of new software technology, pp. 229–263, World Scientific Publishing Co., Inc., 2003. 118, 120

[110] B. Kitchenham, “DESMET: A method for evaluating software engineering methods and tools,” tech. rep., University of Keele UK, 1999. 118, 120

[111] B. Kitchenham, M. Al-Khilidar, H.and Babar, M. Berry, K. Cox, J. Keung, F. Kurniawati, M. Staples, H. Zhang, and L. Zhu, "Evaluating guidelines for reporting empirical software engineering studies," *Empirical Software Engineering*, vol. 13, no. 1, pp. 97–121, 2008. 118, 120, 143, 145

[112] V. Basili, "The role of controlled experiments in software engineering research," in *Proceedings of the 2006 International Conference on Empirical Software Engineering Issues: Critical Assessment and Future Directions*, pp. 33–37, Springer-Verlag, 2007. 118

[113] P. Runeson and M. Host, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Softw. Engg.- Springer*, vol. 14, no. 2, pp. 131–164, 2009. 143, 145

[114] B. Freimut, S. Hartkopf, P. Kaiser, J. Kontio, and W. Kobitzsch, "An industrial case study of implementing software risk management," *SIGSOFT Softw. Eng. Notes*, vol. 26, no. 5. 143, 145

[115] Fossil, "Fossil version [3613fe5666] 2012-05-12." `http://www.fossil-scm.org/xfer/doc/trunk/www/concepts.wiki`, May 2012. [Online; accessed 30-May-2012]. 152

[116] M. Carroll, A. van der Merwe, and P. Kotze, "Secure cloud computing: Benefits, risks and controls," in *Proceedings of the 2011 Information Security South Africa*, ISSA'11, pp. 1 –9, IEEE, Aug. 2011. 168

[117] J. Morin, J. Aubert, and B. Gateau, "Towards cloud computing sla risk management: Issues and challenges," in *Proceedings of the 45th Hawaii International Conference on System Science*, HICSS '12', pp. 5509 –5514, IEEE, Jan. 2012. 168

Appendix A: Experiment One Materials

Contents

Subjects References Numbers

Student Name	Student No.	Education Background & Experience	Reference No.

Experiment Execution Checklist

Before Execution:

☐ Make sure that the all required handouts are ready.

☐ Make sure that the all of the subjects are informed about the experiment aim and all related information.

☐ Make sure that the subject are signed the experiment participant conformation and they are understood their roles.

☐ Give a secret codes to participant to ensure the privacy.

☐ Ask the participant to select the suitable time for them to participate in the experiment

☐ Send them a conformation about the experiment schedule (contains the their scheduled time for the experiment)

During Execution:

☐ Participant should fill in a form about himself/herself contains data about the proficiently and experience and bibliographic information and inform him that all the data is under the act no of privacy and will not be used for any other thing just the experiment. Meanwhile a secret code should be given to this form rather that the names.

☐ During all experiment stages ask the participant to take his/her time while they perform the tasks and carefully read the material, evaluate and try to give the accurate answers or taking the right decisions.

☐ submit the related handouts to the participant and ask him to read and ask any questions.

☐ Tell the participant that he/she has the right to stop at any time from completing the experiment as well as he/she has the right to ask any related questions.

☐ Every participant has to sign a clarification document which clarifies his rights and his roles ate the experiment.

☐ Time should be monitored and measured for all the experiment sessions, stages and for all participants.

☐ Data forms should be filled

☐ Prepare for the next participant.

After Execution:

☐ Make sure that experiment data has been collected well at the execution

☐ Classify the gathered data.

☐ Prepare for the next participant

Implementation Time, Observation and Comments Sheet

Participant Ref. No.			**Date: / /**
Section No	**Task No.**	**Used Time**	**Observation**
1 *Risks importance to Web and Distributed development and specifying risks and perspectives (All subjects)*	T1		
	T2		
	T3		
	T4		
2 *Clustering from the three perspectives* Subject Group : ☐ *Control Group* ☐ *Experimental Group*	T5		
	T6		
3 *Searching in Pre-Clustered Risks (All subjects)*	T7		
	T8		
	T9		
4 *Atypical Risks and WD development (All subjects)*	T10		
	T11		
Participant Comment			
Experimenter Comment			

Instructions and Material

The experiment consists of four sections. Each section is started with subject's tasks and then it is followed with the related experiment material and forms.

General Instruction To All of the Participant

During all experiment stages please take your time and carefully read the material, evaluate and try to give the accurate answers and take the right decisions.

Perspectives (Project Process Product):

Project perspective is concerning with project aspects such as budgets, plans, goals, responsibilities and schedules.

Process perspective is concerning with the methods, tasks and process of producing the software.

Product perspective is concerning with the final product aspects such as its functionality, maintenance, market competence and security.

Subject Ref. No. :	**Experiment Date :** / /
Experiment Section 1 Subjects instructions	***Risks importance to Web and Distributed development and specifying risks and perspectives (All subjects)***
Task (1)	Specify the importance of the risks in the followed table to Web and Distributed development (WD). Tick where appropriate.
Task (2)	Please specify two risks for each one of the three perspectives. Tick where appropriate.
Task (3)	There are three risks were ticked by the experimenter. Please specify to which perspectives they are related to.
Task (4)	Please answer the following questions after finishing the above three tasks: Q1: Are there any other risks that need to be added to the WD development potential risks list? Please state if any. ------------------------ ---------------------------------- Q2: Was it easy for you to specify the risks in task 2 and the perspectives in task 3? Yes No Q3: Do you agree with the idea that the above tasks will be much easier and the time and effort can be saved if risks were clustered from the three perspectives?

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
1	2	3	4	5

Experiment Section 1: Risks importance to Web and Distributed development and specifying risks and perspectives		**Task 1**					**Tasks 2, 3**		
							ST : ET:		
		Level of Importance to WD*					**Perspective**		
R. .No	**Risk Name**	**NI**	**SI**	**I**	**VI**	**EI**	**Proj.**	**Proc.**	**Prod.**
	Unfamiliarity with international and foreign contract law								
	Inadequate customer requirement (see and change strategy)								
	Poor documentation								
	Low visibility of project process								
	Inadequate process development								
	Not enough measurement and estimations								
	Lack of security precautions								
	Weaknesses in protection procedures for Intellectual Property rights								
	Vendor feasibility								
	Insufficient competence								
	Communication failures								
	Poor sites management control								
	Failure to manage user expectations								
	Insufficient project stakeholder involvement								
	Process instability								
	Poor performance								
	Poor UI								
	Insecure of communication channels								
	Lack of requirement specification								
	Inadequate user involvement								
	Difficulties in ongoing support and maintenance								
	Unrealistic estimation of the number of users								
	Differences in the development methodologies and processes								
	Weak or inadequate contracts								
	Complicated development dependencies between project sites								
	A Cross cultural differences / influence								
	Poor product functionality								
	Market fluctuations								
	Scalability limitations								
	Poor availability								
	Lack of top management commitment								
	Instability in other project sites								
	Lack of Face-To-Face meetings								
	Lack of Management availability and efficiency								
	Unfamiliarity with customer type								
	Constraints due to time zone differences								

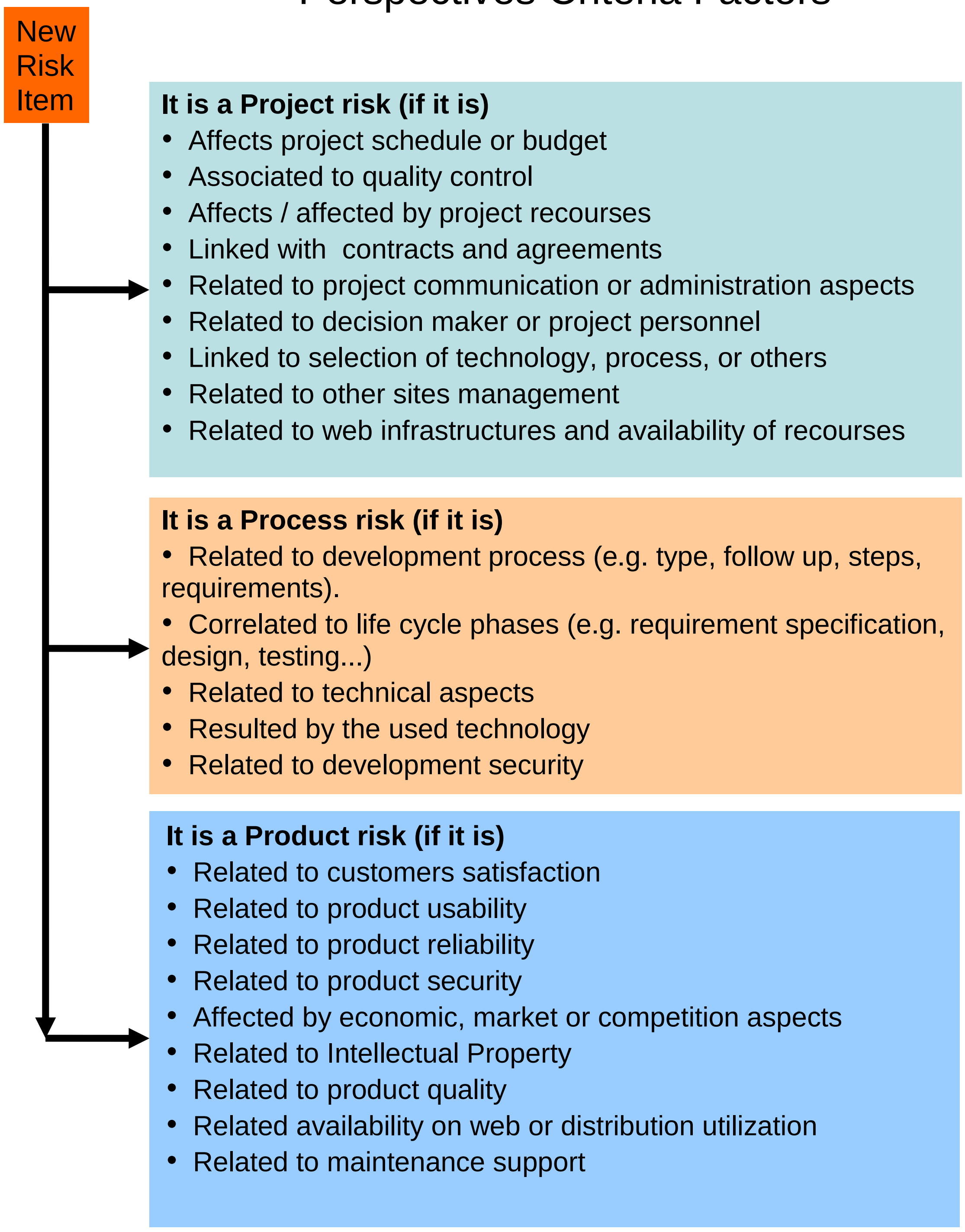
*__NI__ = Not Important **SI**= Somewhat Important **I**= Important **VI**=Very Important **EI**= Extremely Important

Subject Ref. No. : **Date:** / / **Group:** □ *Control* □ *Experimental*

Experiment Section 2 ***Subjects instructions***	***Clustering from the three perspectives (subjects tasks)***
Task (5) Control group	Cluster the risks from the three perspectives
Task (6) Experimental group	Cluster the risks using the provided criteria factors

ST: ET:	**Perspectives**		
Risk Name	**Project**	**Process**	**Product**
Unfamiliarity with international and foreign contract law			
Inadequate customer requirement (see and change strategy)			
Poor documentation			
Low visibility of project process			
Inadequate process development			
Not enough measurement and estimations			
Lack of security precautions			
Weaknesses in protection procedures for Intellectual Property rights			
Vendor feasibility			
Insufficient competence			
Communication failures			
Poor sites management control			
Failure to manage user expectations			
Insufficient project stakeholder involvement			
Process instability			
Poor performance			
Poor UI			
Insecure of communication channels			
Lack of requirement specification			
Inadequate user involvement			
Difficulties in ongoing support and maintenance			
Unrealistic estimation of the number of users			
Differences in the development methodologies and processes			
Weak or inadequate contracts			
Complicated development dependencies between project sites			
A Cross cultural differences / influence			
Poor product functionality			
Market fluctuations			
Scalability limitations			
Poor availability			
Lack of top management commitment			
Instability in other project sites			
Lack of Face-To-Face meetings			
Lack of Management availability and efficiency			
Unfamiliarity with customer type			
Constraints due to time zone differences			

Perspectives Criteria Factors



New Risk Item

It is a Project risk (if it is)

- Affects project schedule or budget
- Associated to quality control
- Affects / affected by project recourses
- Linked with contracts and agreements
- Related to project communication or administration aspects
- Related to decision maker or project personnel
- Linked to selection of technology, process, or others
- Related to other sites management
- Related to web infrastructures and availability of recourses

It is a Process risk (if it is)

- Related to development process (e.g. type, follow up, steps, requirements).
- Correlated to life cycle phases (e.g. requirement specification, design, testing...)
- Related to technical aspects
- Resulted by the used technology
- Related to development security

It is a Product risk (if it is)

- Related to customers satisfaction
- Related to product usability
- Related to product reliability
- Related to product security
- Affected by economic, market or competition aspects
- Related to Intellectual Property
- Related to product quality
- Related availability on web or distribution utilization
- Related to maintenance support

Subject Ref. No. : **Date: / /**

Experiment Section 3
Subjects instructions

Searching in Pre-Clustered Risks (*All subjects*)

Task (7) Please specify two risks for each one of the three perspectives. Tick where appropriate.

Task (8) There are three risks that were ticked by the experimenter. Please specify their perspectives.

Task (9) Please answer the following questions after finishing the above two tasks:

Q1: To what extent do you agree with the idea statement that "concentrating only on the risks of the appointed perspective saves time and effort"

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
1	2	3	4	5

Experiment Section 3: Searching in Pre-Clustered Risks		**Subject Reference No.** :		
		Perspectives		
St: ET:		**Project**	**Product**	**Process**
Project Perspective Potential Risks				
	Communication Failures			
	A Cross cultural differences / influence			
	Lack of Face-To-Face meetings			
	Poor sites management control			
	Weak or inadequate contracts			
	Lack of top management commitment			
	Constraints due to time zone differences			
	Instability in other project sites			
	Unfamiliarity with international and foreign contract low			
	Failure to manage user expectations			
	Lack of Management availability and efficiency			
	Inadequate customer requirement (see and change strategy)			
	Insufficient project stakeholder involvement			
Process Perspective Potential Risks				
	Poor documentation			
	Lack of requirement specification			
	Process instability			
	Low visibility of project process			
	Differences in development methodology / process			
	Complicated development dependencies between project sites			
	Inadequate process development			
	Insecure communication channel			
	Not enough measurement and estimations			
Product Perspective Potential Risks				
	Unrealistic estimation of the number of users			
	Poor performance			
	Lack of security precautions			
	Poor product functionality			
	Poor availability			
	Poor UI			
	Weaknesses in protection procedures for the intellectual property rights			
	Inadequate user involvement			
	Vendor feasibility			
	Market fluctuations			
	Difficulties in ongoing support and maintenance			
	Insufficient competence			
	Unfamiliarity with customer type			
	Scalability limitations			

Subject Ref. No. : **Date:** / /

Experiment Section 4 Subjects instructions **Atypical Risks** and WD development (*All subjects)*

Task (10)

Atypical risks are risks that could not be predicted before they occur. To what extent do you agree that Web and Distributed development is vulnerable to such type of risks

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
1	2	3	4	5

Task (11)

Due to the lack of information it is difficult to expect atypical risks or even to be ready for them. Therefore, the approach attempts to absorb their side affects and then consider them in future risk management cycles.

To what extent do you rate the usefulness of this strategy?

Not Useful	Somewhat Useful	Useful	Very Useful	Strongly Useful
1	2	3	4	5

Do you suggest any other useful strategies to deal with atypical risks type? Please state that.

---------- ---

-------------------- --

Experiment Summary

Title of Experiment: Experiment of Evaluating the Importance of A List of Potential Risks to Web and Distributed (WD) Development, the Usefulness of Perspectives Clustering and Vulnerability of WD to Atypical Risk. **Page 1/2**

Background:

This experiment is a part of PhD research at Newcastle University which aims to build a software risks management approach to manage Web and Distributed (WD) development risks. The approach is called *WeDRisk* and it is still under construction. The aim of this experiment is to test some hypotheses which are prepared in order to validate the significance of a list of proposed potential risks to WD development and the usefulness of clustering them from three perspectives (project, product and process). Moreover, the experiment will be used to examine the WD vulnerability to atypical risks and the usefulness of side effect absorbing mechanism. The result of this experiment is very important to our research as it will be used to validate some important aspects of our proposed approach.

Aim / Objectives

The experiment was designed to evaluate the importance of a list of proposed risks WD development, as well as to evaluate a criteria factors to cluster the risks from there perspectives project, process and product and the usefulness of the clustering from these perspectives. Mainly the experiment results will be used to evaluate the following aspects:

- Proposed WD risks
- Perspectives clustering
- Perspectives criteria factors
- The distribution of WD risks based on the proposed criteria factors
- The effectiveness of proposed clustering criteria on time and effort
- WD development vulnerability to atypical risks and absorber mechanism

Generally the result of this experiment will be used to improve the proposed *WeDRisk* approach.

Description:

Introductory:

Before the experiment begins, participants will be asked to fill in demographic information. We asked their 1) student number, 2) age, 3) gender, 4) educational background and 5) experiences with software development. A brief introductory session will be provided to them including explanation about necessary information (e.g. description of the experiment, software risks, proposed list of risks, WD development, software risk management, software perspectives).

Then the tasks of participants during the experiment will be explained to them. Printed information will also be supplied to support the participants' understandings. Basically, the participants are divided into two control and experimental groups. Some tasks will be performed by both of the groups. This is depends on the nature of the task and needed measurements.

Experimenter Tasks:

He will give a brief description for experiment as well as the necessary required training and he will make sure that all participants have understood their roles. The experimenter will give this information and at the same time maintain the roles of avoiding any bias. Meanwhile, during all the experiment stages and tasks the experimenter collects data which is needed for the measurement to test the hypothesis.

The experimenter provides a suitable working environment for running the experiment.

Experiment Summary (Cont.)

Title of Experiment: Experiment of Evaluating the Importance of A List of Potential Risks to Web and Distributed (WD) Development, the Usefulness of Perspectives Clustering and Vulnerability of WD to Atypical Risk. **Page 1/2**

Participants Tasks:

The participants tasks can be summarized as follows:

1. Understanding their roles in the experiment.
2. Performing the assigned roles with understanding that they have the right to ask for any clarifications and they can stop at any stage of the experiment.
3. Classifying the list of potential risks based on their importance to WD developments.
4. Clustering the WD risks from three perspectives Project, Process and Product. For this task the participants are divided into control and experimental groups: Control group members perform the clustering operation based on their own knowledge whereas, the experimental group use a specific criteria factors for clustering the risk from the three perspectives.
5. Searching twice for certain perspectives risks before and after the clustering
6. Expecting vulnerability level to atypical risks in WD development.

Time:

The performing of the experiment tasks is conducted under specific procedures to collect the measurement data and avoid the bias.

We estimated the session above will take approximately 25 minutes for each participant.

*A flow chart summarising the above procedures has been prepared.

Subjects / Participants:

The subjects of this experiment are expected to be about 35 master and PhD students from the school of computing science / Newcastle University. We plan to use email to recruit the participants for this experiment.

This set of participant has been selected as we expect that they have enough knowledge or experience with software development and many of them have participated in software development projects as part of their courses. Other reason for selecting them is the difficulties of getting participants from software companies due to their security restriction with their projects data and information.

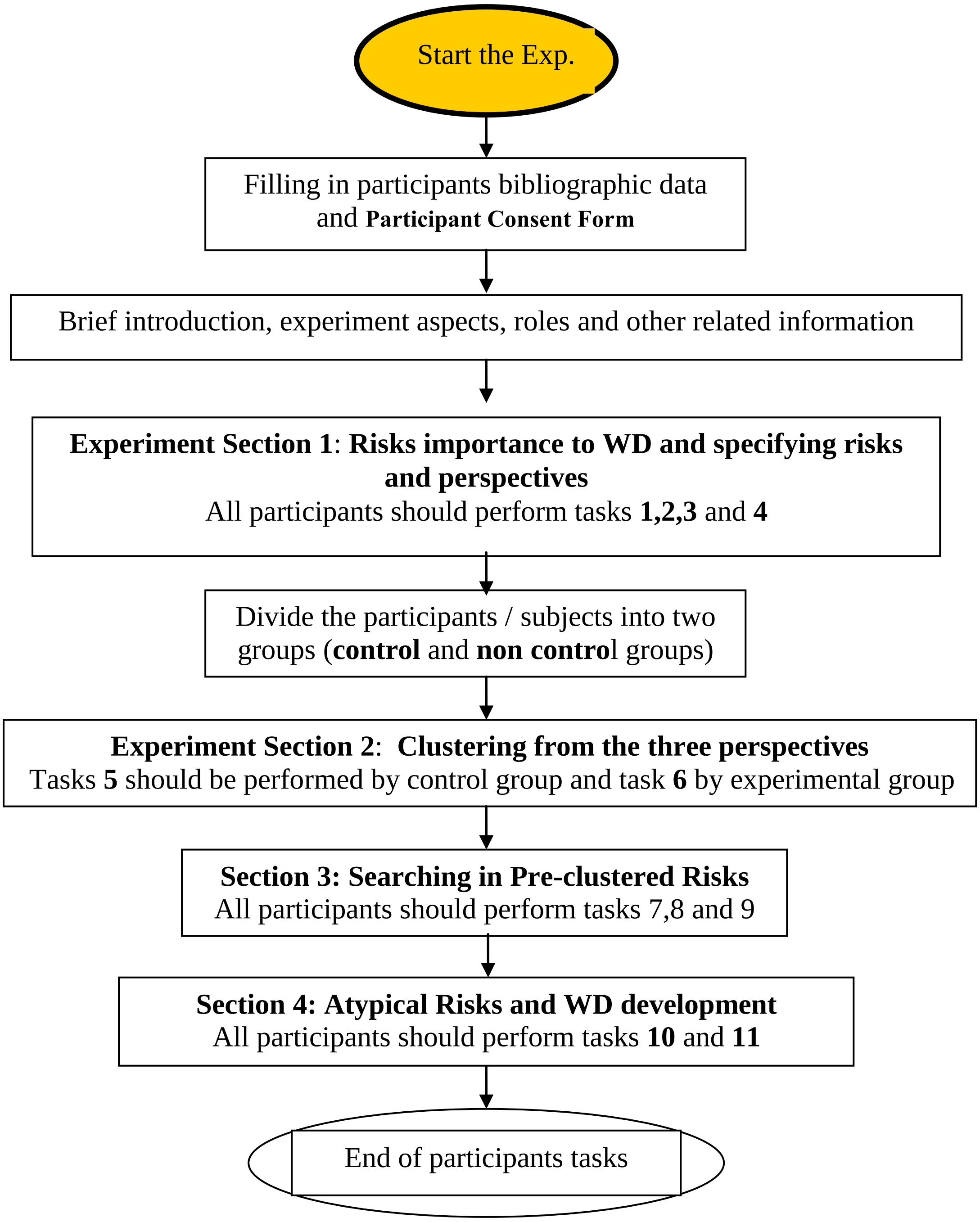
We plan to compensate each participant with £10 Amazon Voucher. We will be needing participants' student number in order to provide them with the Amazon Voucher and will use the information **strictly** for this purpose only.

*A consent form is ready.

Data:

The data that we will collect will be strictly confidential to the experimenter and his supervisor. The data gathered will only be used for the research purposes and not for other intention. We intend to only use the student number for providing the free printing credits (through ISS) and will discard the data afterwards.

Experiment Summary Map



Start the Exp.

Filling in participants bibliographic data and **Participant Consent Form**

Brief introduction, experiment aspects, roles and other related information

Experiment Section 1: **Risks importance to WD and specifying risks and perspectives**
All participants should perform tasks **1,2,3** and **4**

Divide the participants / subjects into two groups (**control** and **non control** groups)

Experiment Section 2: **Clustering from the three perspectives**
Tasks **5** should be performed by control group and task **6** by experimental group

Section 3: Searching in Pre-clustered Risks
All participants should perform tasks 7,8 and 9

Section 4: Atypical Risks and WD development
All participants should perform tasks **10** and **11**

End of participants tasks

SCHOOL OF COMPUTING SCIENCE
FACULTY OF SCIENCE, AGRICULTURAL & ENGINEERING
NEWCASTLE UNIVERSITY

Participant Consent Form

Experiment Title:

Evaluation of clustering risks from Project, Process Product Perspectives (Clustering criteria and importance of risks to Web and Distributed software development)

Purpose:

The experiment was designed to evaluate the importance of a list of proposed risks WD development, as well as to evaluate criteria to cluster the risks from three perspectives project, process and product and the usefulness of the clustering from these perspectives. Web and Distributed development vulnerability to atypical risks and treatment mechanism will be evaluated as well.

Procedure:

If you agree to be in this study, you will be asked to do the following:

1. Understanding your roles in the experiment.
2. Performing the assigned roles with understanding that they have the right to ask for any clarifications and you can stop at any stage of the experiment.
3. Classifying the list of potential risks based on their importance to WD developments.
4. Clustering the WD risks from three perspectives Project, Process and Product. For this task the participants are divided into control and experimental groups: Control group members perform the clustering operation based on their own knowledge whereas, the experimental group use a specific criteria factors for clustering the risk from the three perspectives.
5. Searching twice for certain perspectives risks before and after the clustering
6. Expecting vulnerability level to atypical risks in WD development.

The total time required to complete the study should be approximately 30 minutes. You will receive 500 pages of free printing credits for participating.

Benefits/Risks to Participant:

Participants will learn more about software risk management, the three perspectives (project, process product) web and distributed risks. Possible risks include frustration caused by not being able to cluster the risks from the three perspectives.

Voluntary Nature of the Study/Confidentiality:

Your participation in this study is entirely voluntary and you may refuse to complete the study at any point during the experiment, or refuse to complete any task which you are uncomfortable. You may also stop at any time and ask the researcher any questions you may have. Your student number will never be connected to your results instead; it will only be used for providing compensation for your participation. We will use serial numbers instead, for identification purposes. Information that would make it possible to identify you or any other participants will never be included in any sort of report. The data will be accessible only to those working on the project.

Contacts and Questions:

At this time you may ask any questions you may have regarding this study. If you have questions later, you may contact the person conducting the study, Ayad Ali Keshlaf via email at a.a.a.keshlaf@ncl.ac.uk or Dr Steve Riddle (his supervisor) via email at steve.riddle@ncl.ac.uk . Questions or concerns about institutional approval should be directed to Ms Jo Mayne, Deputy Head of Administration at the Faculty of Science, Agricultural & Engineering, Newcastle University via email joanne.mayne@ncl.ac.uk or call her at 0191 222 5923.

Statement of Consent:

I have read the above information. I have asked any questions I had regarding the experimental procedure and they have been answered to my satisfaction. I consent to participate in this study.

Name of Participant____Date: __________

Signature of Participant ____

Age: ________**(Note: You must be 18 years of age or older to participate in this study. Let the experimenter know if you are under 18 years old.)**

Thank you for your participation!

Participants Recruiting Email

Hi,

My name is Ayad and I am PhD student at School of Computing Science/ Newcastle University under Dr. Steve Riddle supervision. I am doing research in the area of software risks for Web and Distributed development. As part of my research work, I am planning to evaluate the importance of some risks to Web and Distributed development and the clustering them from Project, Process Product Perspectives.
Therefore, I need people to help me out with this test. The test takes only 30 minutes and should be interesting to do. I will be able to compensate your time with £10 Amazon voucher.
The test would take place at Room **10.04 Claremont Tower** on **28th, 29th, 30th July 2010** or on **3rd, 4th and 5th August** 2010. I will be there from 930am till 630pm and you can choose to come between these hours to try out the test at any date of the above dates (please email me on a.a.a.keshlaf@ncl.ac.uk or call me at **0191 222 5405** to arrange your participation date and time).

For any further information, please do not hesitate to contact me.

Thank you and I'm waiting for your participation in this experiment.

Ayad Ali Keshlaf
Room 10.04 Claremont Tower
School of Computing Science
Newcastle University
a.a.a.keshlaf@ncl.ac.uk
0191 222 5405

Official Personal Website: http://www.cs.ncl.ac.uk/people/a.a.a.keshlaf

Appendix B: Experts Evaluation Materials

Contents

- Modules Evaluation
- WeDRisk General Aspects (Overall) Evaluation

The Result of WeDRisk Expert Evaluation Questionnaire (at ICIMP 201)

Expert Ref. No.	Name	Institute	Profession	Experience with software development (years)	Email	Date
1						
~						
~						

1. Modules Evaluation

1.1 - Rate the usefulness of project card concept

□ Not useful □ Somewhat Useful □ Useful □ Very useful □ Strongly Useful

Comments if any

1.2 - Evaluate the project card from coverage perspective and tick where appropriate

□ It covers all aspects
□ It covers most of the aspects
□ Missed fields:

□ Unnecessary fields:

1.3 - Evaluate the helpfulness of RM customization concept (*Please tick all boxes that apply***)**

□ It saves time and effort
□ It provides more options for RM for different situations
□ It is not helpful

Comments to improve it:

1.4 –How accurate are the proposed pre-clustered W-D risks? (Please give suggestion if any changes are needed**)**

1.5 – Do you agree with the statement that consideration of risks from three perspectives (project process and product) save time and effort?

□ Strongly Disagree □ Disagree □ Somewhat Agree □ Agree □ Strongly Agree

Other Comments

1.6 - Rate the helpfulness of clustering criteria in clustering risks from the 3P perspectives

□ Not Helpful □ Somewhat Helpful □ Helpful □ Very Helpful □ Strongly Helpful

Any Comments to improve it:

1.7 - To what extent do you agree that W-D development is vulnerable to atypical Risks?
□ Strongly Disagree □ Disagree □ Somewhat Agree □ Agree □ Strongly Agree

1.8 - Rate the absorbing mechanism as an effective way to deal with W-D atypical risks
□ Strongly Disagree □ Disagree □ Somewhat Agree □ Agree □ Strongly Agree

1.9 - WeDRisk considers W-D factors in the estimation module by estimating and adding them to risk exposure.
Are there any other factors could be included?

Is the W-D factor matrix effective/useful or can you suggest some improvement?

1.10 - RM customization module in WeDRisk offers two options for RM (plain and deep) based on situation. This is reflected in other WeDRisk modules (e.g., estimation and evaluation modules).

□ This saves time and effort
□ Gives more flexibility for developers
□ Less time and effort consumption

1.11 - In additional to the top ten risks based on RE or TREV prioritized list, WeDRisk considers any risk as a top risk if it is:

- **Completely new**
- **Atypical risk**
- **Affecting more one site or perspective**
- **There is dependency on the affected components**
- **The history with the risk is bad**

To what extent do you agree with this evaluation of top risks in managing W-D risks?
□ Strongly Disagree □ Disagree □ Somewhat Agree □ Agree □ Strongly Agree

1.12 - WeDRisk offers precaution and reduction plans options for planning and controlling which can be used for one site or multisite. Please tick where you agree

□ This decentralization provides more flexibility to W-D development
□ Decentralization provides faster actions which saves time, resources and effort
□ Acting individually quarantines risks early
□ In case of multisite it means sharing experience and alerting early
□ Precautions could reduce the chance of many risks with less effort

1.13 - WeDRisk has a module for mapping the dependency and interoperability of W-D risks relations. Please tick where you are agreed:

□ It helps in avoiding of risks combination which could produce other risks of higher impact
□ Helps to deal with sourced risks by concentrating on the most important ones (higher relations)
□ It helps in the monitoring of risks which help in planning and controlling (giving priority)

1.14 - W-D developments are continuously evolving, therefore WeDRisk is evolvable approach and has a module for that. Please evaluate the evolving module and tick where appropriate:

□ It is toward concept and covers all evolving needs
□ Improvement comments:

----

2. WeDRisk General Aspects (Overall) Evaluation

2.1 - What do you think about WeDRisk main architecture?
□ Understandable
□ Simple and easy to follow
□ Complicated
□ Needs some improvements which are:
------------------- --
------------------- ------------ --

2.2 - Evaluate the WeDRisk modules coverage for RM aspect for W-D development
□ They cover all aspects managing W-D risks
□ Not enough coverage due to the following:

2.3 - Rate the expected usefulness of WeDRisk approach for managing W-D risks
□ Not useful □ Useful □ Very useful
Comments if any

2.4 - Please rate the understandably level of WeDRisk Modules in general
□ Very Low □ Low □ Medium □ High □ Very High
Modules those were difficult to understand and your suggestion:

2.5 - Please if you have any extra comments or suggestions to improve WeDRisk approach:

Appendix C: Experiment Two Materials

Contents

- Participant Consent Form
- Section 1: WD Factors Estimation (Material)
- Section 2: Risk Management Customization (Material)
- Section 3: Atypical Risks (Material)
- Risk Situations
- Experiment Execution Checklist

SCHOOL OF COMPUTING SCIENCE
FACULTY OF SCIENCE, AGRICULTURAL & ENGINEERING
NEWCASTLE UNIVERSITY
Participant Consent Form

Experiment Title:
Evaluation of novel aspects of WeDRisk approach: WD factor estimation, risk management customization and atypical risks absorbing concept

Purpose:
WeDRisk is an approach that has been designed to mange risks in Web and Distributed (WD) development and in software development in general. It is designed to tackle some weaknesses of existing software risk management approaches. This experiment aims to evaluate some novel aspects of WeDRisk approach (e.g. WD factors estimation, risk management customization and atypical risks absorbing concept). Based on the targeted aspect of the evaluation the subjects will be divided into two groups experimental and control (sometimes). The subjects will be given a description about the experiment, and necessary training and information to perform the assigned tasks.

Procedure:
If you agree to be in this study, you will be asked to do the following:

1- Understand your role in the experiment.
2- Perform the assigned roles with understanding that you have the right to ask for any clarifications and you can stop at any stage of the experiment.
3- Estimate the risks for the injected situation/scenarios (in order to see the consideration of WD factors and evaluate related matrix)
4- Customizing of risk management by specifying the suitable type of risk management for each situation
5- Dealing with atypical risk. In this case the experimental group will use the atypical risk mechanism which is supported by WeDRisk approach, whereas the control group will not provided with any mechanism.

The total time required to complete the study should be approximately 35 minutes. You will receive £10 Amazon Voucher for participating.

Benefits/Risks to Participant:
Participants will learn about the estimation of software risks, how to decide the suitable type of risks management and dealing with atypical risks. In general they will learn about how to manage risks in WD development and software development in general. Possible risks include frustration caused by not being able to perform some tasks.

Voluntary Nature of the Study/Confidentiality:
Your participation in this study is entirely voluntary and you may refuse to complete the study at any point during the experiment, or refuse to complete any task which you are uncomfortable. You may also stop at any time and ask the researcher any questions you may have. Your student number will never be connected to your results; it will only be used for providing compensation for your participation. We will use serial numbers instead, for identification purposes. Information that would make it possible to identify you or any other participants will never be included in any sort of report. The data will be accessible only to those working on the project.

Contacts and Questions:
At this time you may ask any questions you may have regarding this study. If you have questions later, you may contact the person conducting the study, Ayad Ali Keshlaf via email at a.a.a.keshlaf@ncl.ac.uk or Dr Steve Riddle (his supervisor) via email at steve.riddle@ncl.ac.uk . Questions or concerns about institutional approval should be directed to Ms Jo Mayne, Deputy Head of Administration at the Faculty of Science, Agricultural & Engineering, Newcastle University via email joanne.mayne@ncl.ac.uk or call her at 0191 222 5923.

Statement of Consent:
I have read the above information. I have asked any questions I had regarding the experimental procedure and they have been answered to my satisfaction. I consent to participate in this study.

Name of Participant____Date: ___________
Signature of Participant ____
Age: _________**(Note: You must be 18 years of age or older to participate in this study. Let the experimenter know if you are under 18 years old.)**

Thank you for your participation!

Section 1: WD Factors Estimation

The purpose

This section is designed to evaluate the estimation module of WeDRisk approach in particularly the consideration to WD factors and related matrix.

Description and Diagram

For the estimation purpose WeDRisk approach offers two options, one is based on Risk Exposure (RE) and the other is based on Total Risk Estimation Value (TREV). Both options RE and TREV can be used to prioritize the risks, but each one of them is suitable for a specific situation. RE and TREV equations are described below.

RE can be obtained based on the following equation:

RE = Risk Prob. * Risk Mag. ----------------------------(1)

Where,

Risk Prob. is the probability of an unsatisfactory outcome
Risk Mag. is the loss associated to unsatisfactory outcome

The estimation module in the WeDRisk approach offers a simple technique that can be used to estimate the probability and magnitude of risks. It is based on qualitative line system with equivalent quantitative values (see Figure 1). Probability values are ranged from **0** for negligible probability to **1** for extremely high probability; whereas, the risk magnitude values are ranged from **1** for insignificant to **5** for sever.

TREV Can be obtained by the following equation:

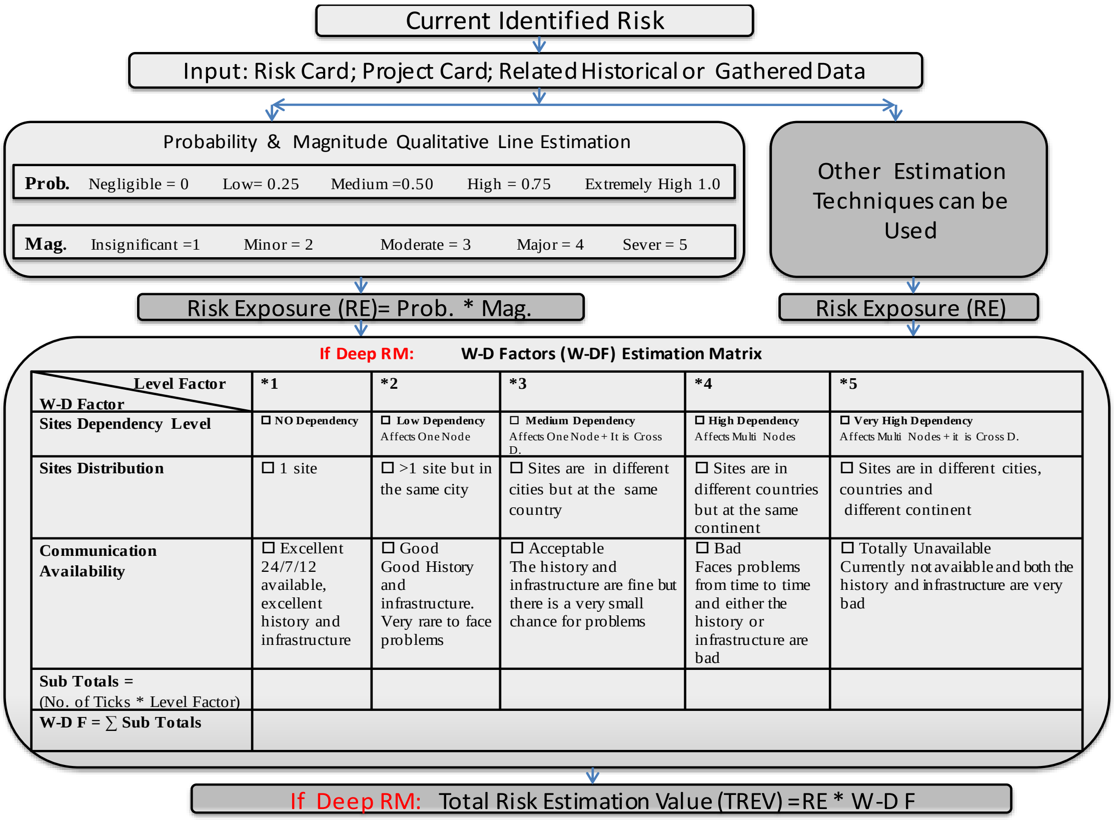
TREV= RE * WDF -------------------------------(2)

Where,

RE is the risk exposure which can be obtained by equation (1)
WDF is the Web and Distribution Factors which can be estimated with the provided matrix below (see Figure 1).

Subjects Tasks

1- The experimenter will inject randomly some risky situations.
2- You as a subject please specify a suitable type of risk estimation (RE or TREV) for each situation (use Form 1).
3- Please state the reason behind your selection for each case (use Form 1).
4- The experimenter will explain how estimate the risks using the module (Figure 1).
5- Based on the above selections of RE or TREV please estimate the risks using the WeDRisk estimation module (use Forms 2 and 3).
6- Always you can ask for a help if you really need it.
7- Finally, please fill in form 4 as a general evaluation of the estimation module.



Current Identified Risk

Input: Risk Card; Project Card; Related Historical or Gathered Data

Probability & Magnitude Qualitative Line Estimation

Prob. Negligible = 0 Low= 0.25 Medium =0.50 High = 0.75 Extremely High 1.0

Mag. Insignificant =1 Minor = 2 Moderate = 3 Major = 4 Sever = 5

Other Estimation Techniques can be Used

Risk Exposure (RE)= Prob. * Mag.

Risk Exposure (RE)

If Deep RM: **W-D Factors (W-DF) Estimation Matrix**

Level Factor **W-D Factor**	***1**	***2**	***3**	***4**	***5**
Sites Dependency Level	□ **NO Dependency**	□ **Low Dependency** Affects One Node	□ **Medium Dependency** Affects One Node + It is Cross D.	□ **High Dependency** Affects Multi Nodes	□ **Very High Dependency** Affects Multi Nodes + it is Cross D.
Sites Distribution	□ 1 site	□ >1 site but in the same city	□ Sites are in different cities but at the same country	□ Sites are in different countries but at the same continent	□ Sites are in different cities, countries and different continent
Communication Availability	□ Excellent 24/7/12 available, excellent history and infrastructure	□ Good Good History and infrastructure. Very rare to face problems	□ Acceptable The history and infrastructure are fine but there is a very small chance for problems	□ Bad Faces problems from time to time and either the history or infrastructure are bad	□ Totally Unavailable Currently not available and both the history and infrastructure are very bad
Sub Totals = (No. of Ticks * Level Factor)					
W-D F = ∑ Sub Totals					

If Deep RM: Total Risk Estimation Value (TREV) =RE * W-D F

Figure 1: WeDRisk Estimation Module

Form 1: Type of Estimation Forms:

Subject Ref.:	**Date:**	**Used Time:**
💧 **Situation No.:**	**Suitable Estimation**: □ RE □ TERV	
Selection Reasons: --		
💧 **Situation No.:**	**Suitable Estimation:** □ RE □ TERV	
Selection Reasons: --		
💧 **Situation No.:**	**Suitable Estimation:** □ RE □ TERV	
Selection Reasons: --		
💧 **Situation No.:**	**Suitable Estimation:** □ RE □ TERV	
Selection Reasons: --		
Comments / Suggestions: -- -- -- -- --		

Form 2: RE Estimation

Subject Ref.:	**Date:**	**RE Used Time:**	
Situation No.:	**Risk Probability (Prob.):**	**Risk Magnitude (Mag.):**	**Risk Exposure (RE) =** Prob. * Mag**.:**

Form 3: TREV Estimation

WD Factors Estimation Matrix

Level Factor **WD Factor**	***1**	***2**	***3**	***4**	***5**
Sites Dependency Level	□ NO Dependency	□ Low D. Affects One Node	□ Medium D. Affects One Node + It is Cross D.	□ High D. Affects Multi Nodes	□ Very High D. Affects Multi Nodes + it is Cross D.
Sites Distribution	□ 1 site	□ >1 site but in the same city	□ Sites are in different cities but at the same country	□ Sites are in different countries but at the same continent	□ Sites are in different cities, countries and different continent
Communication Availability	□ Excellent 24/7/12 available, excellent history and infrastructure	□ Good Good History and infrastructure. Very rare to face problems	□ Acceptable The history and infrastructure are fine but there is a very small chance for problems	□ Bad Faces problems from time to time and either the history or infrastructure are bad	□ Totally Unavailable Currently not available and both the history and infrastructure are very bad
Sub Totals = (No. of Ticks * Level Factor)					
WDF = ∑ Sub Totals					

Note: If you do not make any selection in the above table then WDF = 1

Total Risk Estimation Value (TREV) = RE * WDF =

Form 4: Subject Evaluation (Estimation)

Subject Ref.:	
To what extent do you rate the usefulness of WD Factors Estimation Matrix?	□Not Useful □ Somewhat Useful □Useful □Very Useful □Strongly Useful
Any other WD factors that should be considered	
How easy the use of the WD Factors Estimation Matrix?	□ Difficult □ Somewhat Easy □ Moderate □Easy □Very Easy
To what extant do you agree that TREV is more suitable than RE for WD risks estimation?	□ Strongly Agree □ Agree □ Neutral □ Disagree □ Strongly Disagree
To what extant do you agree that WD factors should be included at WD risks estimation?	□ Strongly Agree □ Agree □ Neutral □ Disagree □ Strongly Disagree
Any comments / suggestions	
No. of times the subjects has asked for support or help (where) (*filled by the experimenter*)	

Section 2: Risk Management Customization

The Purpose

This section is designed to evaluate the Risk Management (RM) Customization module in the WeDRisk approach.

Description and Diagram

In order to save time and effort and encouraging developers and managers to practices RM and to avoid ignoring it even when there is a limitation of resources which is a common practice is in such situations, WeDRisk approach offers two types of RM which are plain and deep and described below.

Plain RM is a simple type of RM and only the minimum and essential aspects of RM are to be performed under this type.

Deep RM is the ordinary type of RM.

Subject Tasks

1- As a subject you will be assigned to *Experimental Group* (Uses provided matrix) or *Control Group* without using it.
2- The experimenter will randomly inject two risky situations.
3- Based on your group please decide which type of risk management (Deep or Plain) is suitable for each situation. (use Form 5 if you are from Control Group and Form 6 if you are from Experimental Group).
4- Finally, please fill in Form 7 when you finish this section.

Note: Please ask if you need any help or clarification

RM Customization (Control Group)

Form 5: RM Type Control Group Decision

Subject Ref.:					
Situation No.	**Plain RM**	**Deep RM**	**NO Decision**	**I got Confused and I need a help**	**Decision Reasons:**

Please ask if you need a help or clarification

RM Customization (Experimental Group)

WeDRisk approach offers a module to help developers and managers to decide which suitable type of RM they can use for each risky situation. The core of this module is a decision support matrix (See Form 6).

Notes:

- Please ask if you need any help or clarification

Form 6: Experimental Group RM Customization Decision Support Matrix

Subject Ref.	
Group	**Experimental**

RM Cycle / Situation No.	**Risk No.**	**RM Time**		**RM Staff Availability**		**RM Budget Availability**		**RM Experience**		**RM Type Decision**	
		Enough	**Limited**	**Enough**	**Limited**	**Enough**	**Limited**	**Enough**	**Limited**	**Plain**	**Deep**

If there is any tick under **RED** colour Then go for Plain RM type
If all ticks are under **BLACK** then go for Deep RM
If there is a tick under “Enough”' RM Experience which is **BLUE,** (i.e. three years' or more experience in managing related risks) then decisions can be made based on experience.

Notes:

- Not necessary that all field should be ticked to take the decision. This is based on the available information.
- *Enough:* Means there is sufficient of the pointed resource (time, budget or staff) for the RM.
- *Limited:* Means there is insufficient (Shortage/ Criticality) of the pointed resource (time, budget or staffs) for the RM operation.

Form 7: RM Customization General Evaluation

Evaluate the usefulness of RM customization concept (*Please tick all boxes that apply***)**

□ It saves time and effort

□ It is needed when there is criticality of the situation

□ It is not helpful

Comments to improve it:

Rate the usefulness of the RM Customization Decision Support Matrix

□Not Useful □ Somewhat Useful □Useful □Very Useful □Strongly Useful

Comments / suggestions:

Section 3: Atypical Risks

The Purpose
This section is designed to evaluate the concept of atypical risks and absorbing mechanism
Description and Diagram
Ordinary risks are the risks that were faced before by us or by others, and they should be expected at similar situation, condition or environment but with different impact sometimes. That is why we are able to prepare ourselves, organization and projects to deal with them. We can identify them, estimate their probability and magnitude and then control them. However, there is another type of risks which does not have these properties (totally new, it is not faced before, unexpected at all, and suddenly happens) which we named it as atypical risk.
In fact it is not easy to simulate or generate an atypical risk situation because it is unexpected otherwise it will be considered as an ordinary risk. Therefore, the experimenter will try to clarify and provide some examples of real atypical risks which were happened in the real world but not in software industry. Moreover, a dummy example has been created for this purpose is provided below.

Subject Tasks

- After introducing the atypical risk concept ask the subject to fill in the Form 8
- The experimenter will divide the subjects into two groups experimental and control. The experimental group will use the atypical absorbing risk mechanism (Figure 2) which is supported by WeDRisk approach, whereas the control group will not be provided with any mechanism.
- Inject either an atypical risk example
- As a subject you will be asked to deal with the injected atypical risk based on your group and filling in Form 9.
- Monitor the time

Form 8: Before injecting Atypical Risk

Subject No.:	**Group:** □ Control □ Experimental

What do you know about atypical risks?

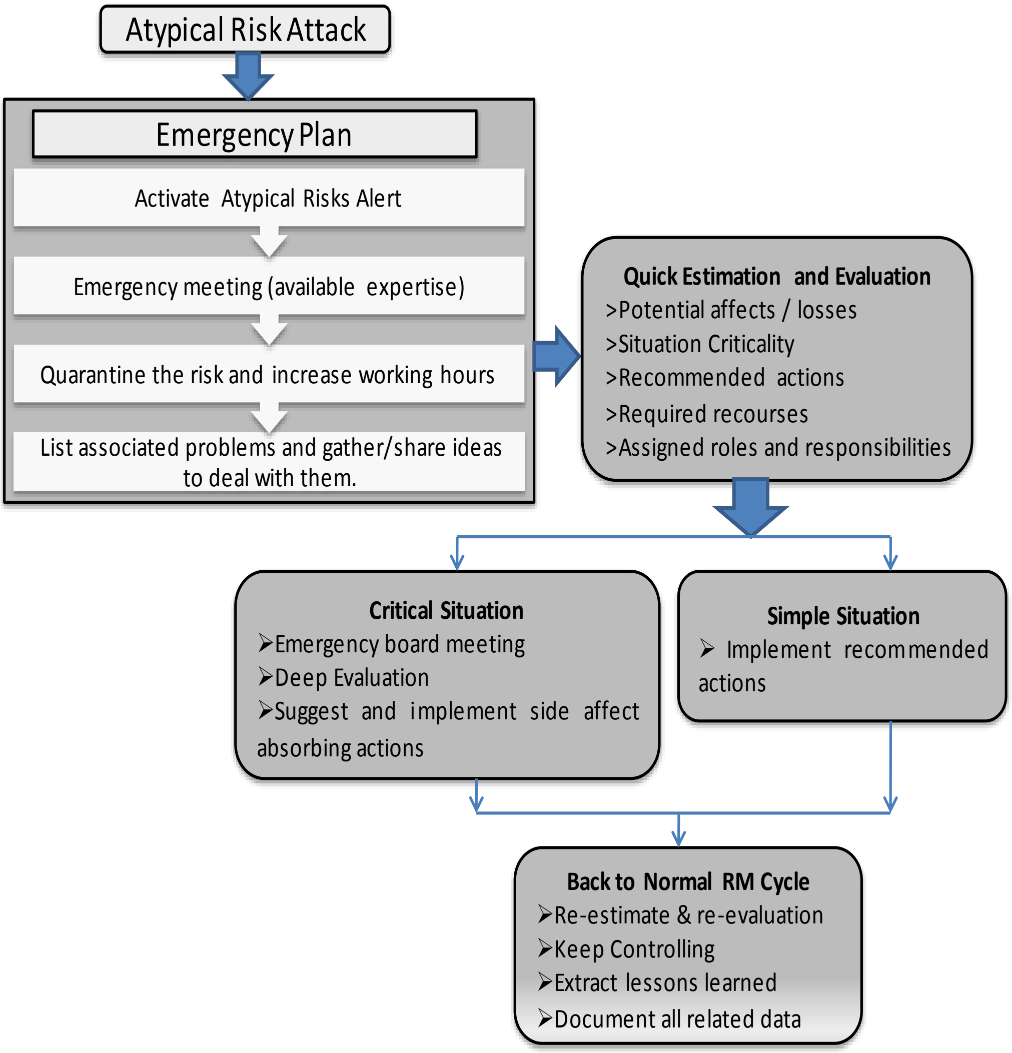
Did you face any atypical risks?

If yes, how did deal with it?

If no, how will you deal if you face it? (What will you do if you face atypical risk?)

Form 9: After Injecting Atypical Risk

Subject No.:	**Group**: ☐ Control ☐ Experimental	
Situation No.	**Identified Atypical Risk**	**Reaction**
Comments / Suggestion to Improve the Mechanism:		



Atypical Risk Attack
Emergency Plan
Activate Atypical Risks Alert
Emergency meeting (available expertise)
Quarantine the risk and increase working hours
List associated problems and gather/share ideas to deal with them.
Quick Estimation and Evaluation
>Potential affects / losses
>Situation Criticality
>Recommended actions
>Required recourses
>Assigned roles and responsibilities
Critical Situation
➢Emergency board meeting
➢Deep Evaluation
➢Suggest and implement side affect absorbing actions
Simple Situation
➢ Implement recommended actions
Back to Normal RM Cycle
➢Re-estimate & re-evaluation
➢Keep Controlling
➢Extract lessons learned
➢Document all related data

Figure 2: WedRRisk Atypical Risk Absorbing Mechanism

Situation Card 0901:

Risk : Difficulties with Communications

STWR is a globally web project under development by TKS software company. Four sites are involved in the development of this project (UK, Spain, Malaysia and India). The main site is the NCL - UK site, but all of the project sites are depending on the KL- Malaysian site which means any delay in KL site will affect all other sites. This project still needs about one month to complete.

Any difficulties with the communication will affect the data transfer and exchange between the project sites which costs a huge amount of money (about £3000 /hour). This could be happened if the main router is broken or the connection with the ISP has problem during the implementation. However, any other types of communication difficulties do not have a significant impact on this project.

The manager of this project has allocated enough fund for the risk management which costs about £100 a day, as well as the manager himself has a long experience with the risk management. In this project the warranty date of the KL site router is almost due (just 1 week) and then the router should go for annually test and maintenance. Meanwhile, the ISP infrastructure of the Indian site faces some randomly technical problems.

Situation Card 1072:

Risk: Conflicts and difficulties with process methodologies

A project has three working sites; main site A (UK), B (France) and C (India which is in a different continental). There is some sensitivity with site B because if there is any delay in this site it will affect both other sites and will cost £2400 / hour. Site C uses a different process methodology. The internet connection and communication are excellent. There is no enough resources for the risk management operations and the time is very critical as the delivery time is too close.

Adopting different methodologies in one project needs running a special training program for the project managers and developers to make them familiar with the used methodologies and to avoid any conflicts. In this project unifying the process methodologies for all sites is not easy and costly (£20000) and needs two weeks to setup, but running training program costs about £10500 and needs Five working 5 days for each site which is affordable and acceptable for this project.

Situation Card 8033:

Risk: Poor performance

Wist System is a project under the development by STN company. STN company has only one development site which is in Newcastle-UK. Usually the STN company does not perform enough testing on its final products in order to reduce the cost.

STN has been chosen to produce a weather monitoring tool (Wist System). Based on the signed contract if there is any poor performance of the produced software, it will cost the STN £10000 a day as a fine. This fine will be doubled if the poor performance happens more than twice a year.

As it is time to market there was no enough testing for the Wist System. Early evaluation versions of the Wist System had a very poor performance in the real time running.

Situation Card 6004:

Risk: Poor documentation

MSSW System is a project under the development by STN company in. STN company has only one development site which is in Newcastle-UK. The new manager of the project has enough experience with risk management.

If the developers do not pay enough time and effort to provide a good documentation of their work there will be a high chance for producing poor documentation which is very costly. In MSSW System project any reworks due to the poor documentation will cost hiring of new programmers which costs £1500 /hour because the project almost completed and the key programmers has left the company.

About half of the work has been well documented whereas, the documentation of the rest of the work was very weak as it has been developed by different programmers and there were no enough time for the documentation.

Atypical Risk Dummy Example:

TFF is controlling software for a medical system under the development a software company which has three development sites, two of them are in UK and one in India. All of the sites depend on the Indian site as it builds a main component of the software and it is used by the other sites. They cannot proceed without it and any delay with Indian site will have a direct affect on other sites.

The main component which is developed by the Indian site needs a special expertise with skills in three areas software programming, medical devises and controlling systems which is available in the Indian sites but very limited. The work was progressed according to its planed schedule and budget.

In order to avoid any risks the company has invested huge budget for the risk management any everything was fine but suddenly some un expected thing has happened (a snake hit the Indian development site and bit the two main expertises). No one has expected that and the worst of that is that nearly impossible to find such expertise to complete the job, some documentation for essential latest parts were not completed and the time is very critical.

The company was aware of all ordinary risks but no one can imagine that a snake can come and bit the two main expertises in one day. This type of unexpected risk we called it atypical risk.

Experiment Execution Checklist

Before Execution:

☐ Make sure that the all required handouts are ready.

☐ Make sure that the all of the subjects are informed about the experiment aim and all related information.

☐ Make sure that the subject are signed the experiment participant conformation and they are understood their roles.

☐ Give a secret codes to participant to ensure the privacy.

☐ Ask the participant to select the suitable time for them to participate in the experiment

☐ Send them a conformation about the experiment schedule (contains the their scheduled time for the experiment)

During Execution:

☐ Participant should fill in a form about himself/herself contains data about the proficiently and experience and bibliographic information and inform him that all the data is under the act no of privacy and will not be used for any other thing just the experiment. Meanwhile a secret code should be given to this form rather that the names.

☐ Submit the related handouts to the participant and ask him to read and ask any questions.

☐ Tell the participant that he/she has the right to ask any related questions for clarification or stop at any time if he/she does not like to continue.

☐ Every participant has to sign the consent form.

☐ Time should be monitored and measured for all the experiment sessions, stages and for all participants. .

☐ Data forms should be filled

☐ Prepare for the next participant.

After Execution:

☐ Make sure that experiment data has been collected well at the execution

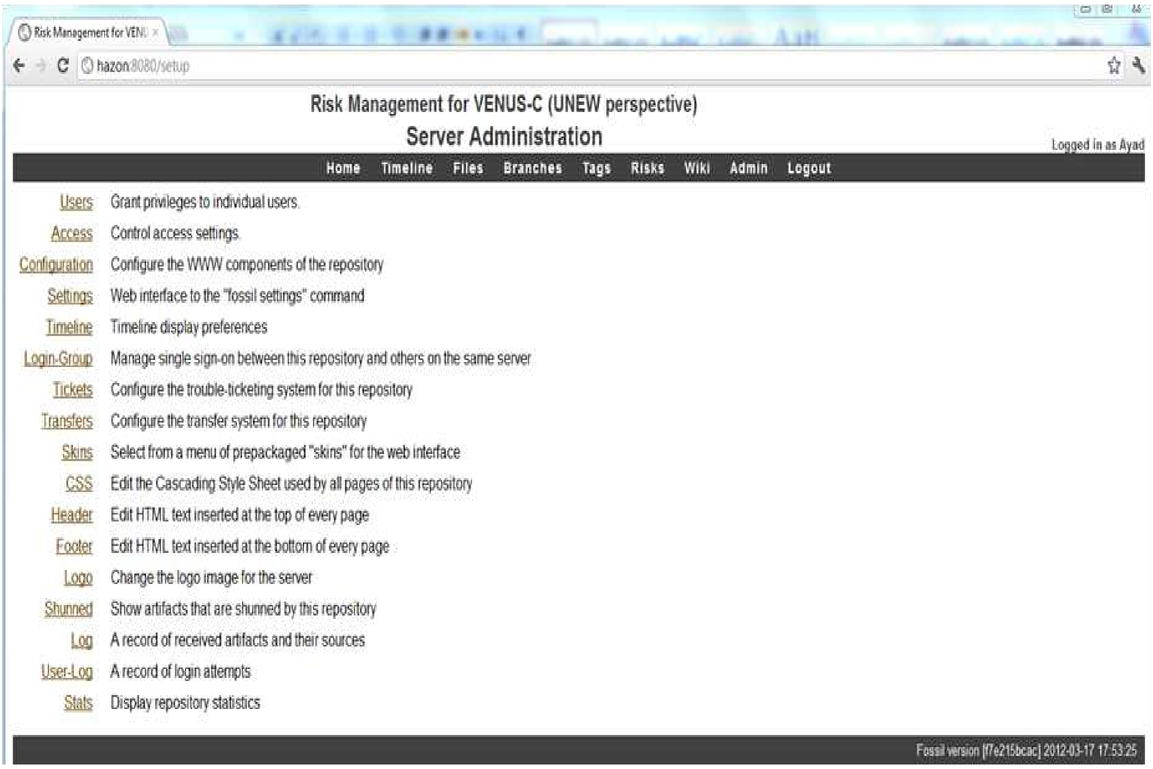
☐ Classify the gathered data.

☐ Prepare for the next participant

Subject Ref.:	**Date:**		
Task	**Used Time**	**Observation / Comment**	**Needed help (No. of times)**
Estimation Type Decisions			
S1			
S2			
S3			
S4			
RE / TREV Estimations			
S1			
S2			
S3			
S4			
Customization			
□Experimental Group □ Control Group			
S1			
S2			
S3			
S4			
Atypical			
□Experimental Group			
□Control Group			

Appendix D: WeD-RM-Prototype Screenshots

Contents



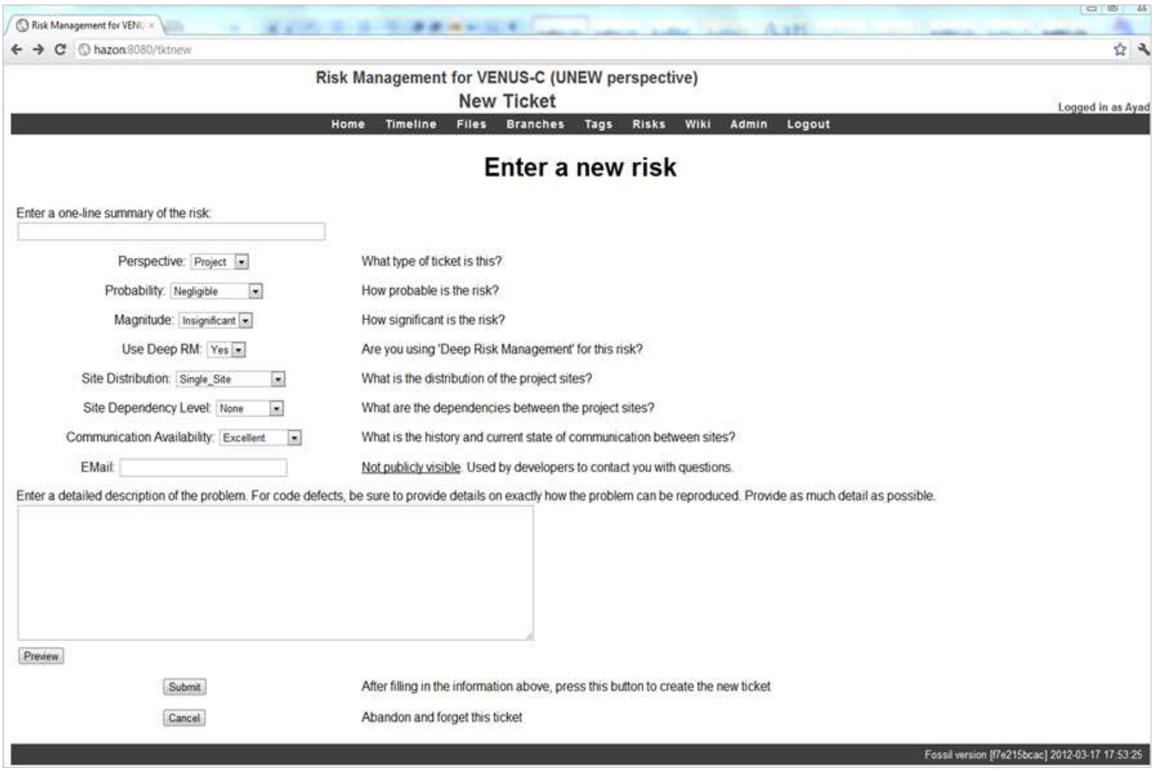
Risk Management for VENUS-C (UNEW perspective)

Server Administration

Logged in as Ayad

Users	Grant privileges to individual users.
Access	Control access settings.
Configuration	Configure the WWW components of the repository
Settings	Web interface to the "fossil settings" command
Timeline	Timeline display preferences
Login-Group	Manage single sign-on between this repository and others on the same server
Tickets	Configure the trouble-ticketing system for this repository
Transfers	Configure the transfer system for this repository
Skins	Select from a menu of prepackaged "skins" for the web interface
CSS	Edit the Cascading Style Sheet used by all pages of this repository
Header	Edit HTML text inserted at the top of every page
Footer	Edit HTML text inserted at the bottom of every page
Logo	Change the logo image for the server
Shunned	Show artifacts that are shunned by this repository
Log	A record of received artifacts and their sources
User-Log	A record of login attempts
Stats	Display repository statistics

Fossil version [f7e215bcac] 2012-03-17 17:53:25



Risk Management for VENUS-C (UNEW perspective)

New Ticket

Logged in as Ayad

Enter a new risk

Enter a one-line summary of the risk:

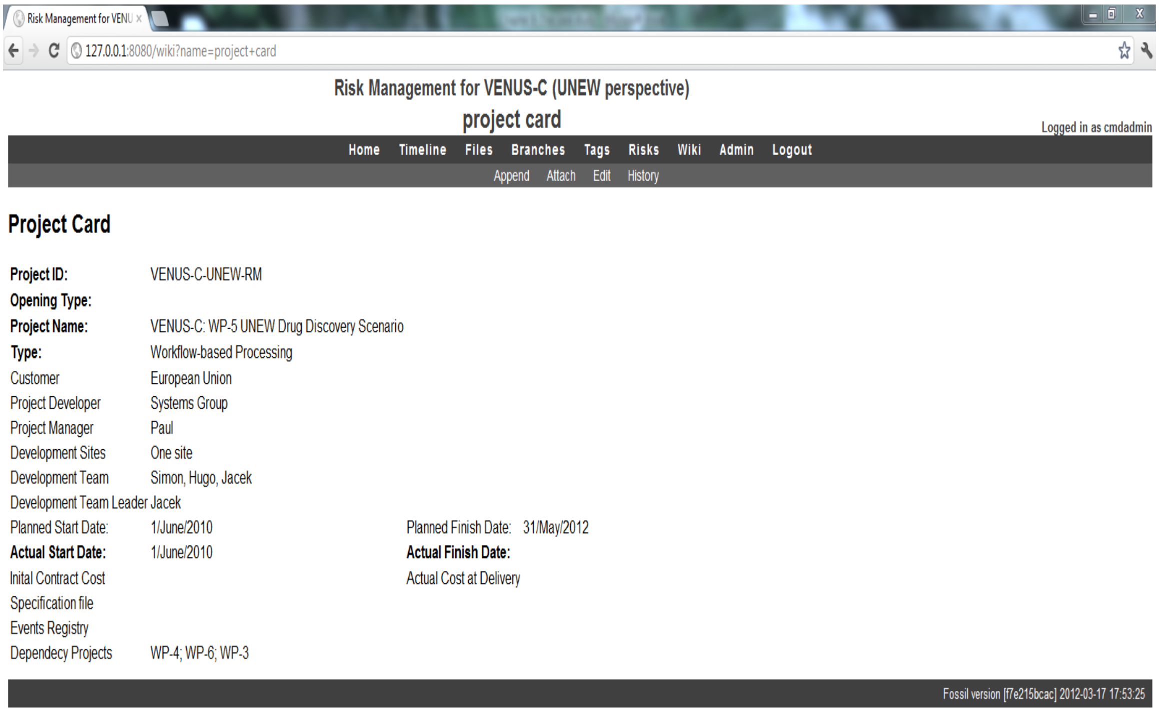
Perspective: Project	What type of ticket is this?
Probability: Negligible	How probable is the risk?
Magnitude: Insignificant	How significant is the risk?
Use Deep RM: Yes	Are you using 'Deep Risk Management' for this risk?
Site Distribution: Single_Site	What is the distribution of the project sites?
Site Dependency Level: None	What are the dependencies between the project sites?
Communication Availability: Excellent	What is the history and current state of communication between sites?
EMail:	Not publicly visible. Used by developers to contact you with questions.

Enter a detailed description of the problem. For code defects, be sure to provide details on exactly how the problem can be reproduced. Provide as much detail as possible.

After filling in the information above, press this button to create the new ticket

Abandon and forget this ticket

Fossil version [f7e215bcac] 2012-03-17 17:53:25


Risk Management for VENU ×
127.0.0.1:8080/wiki?name=project+card
Home Timeline Files Branches Tags Risks Wiki Admin Logout
Append Attach Edit History

Risk Management for VENUS-C (UNEW perspective)

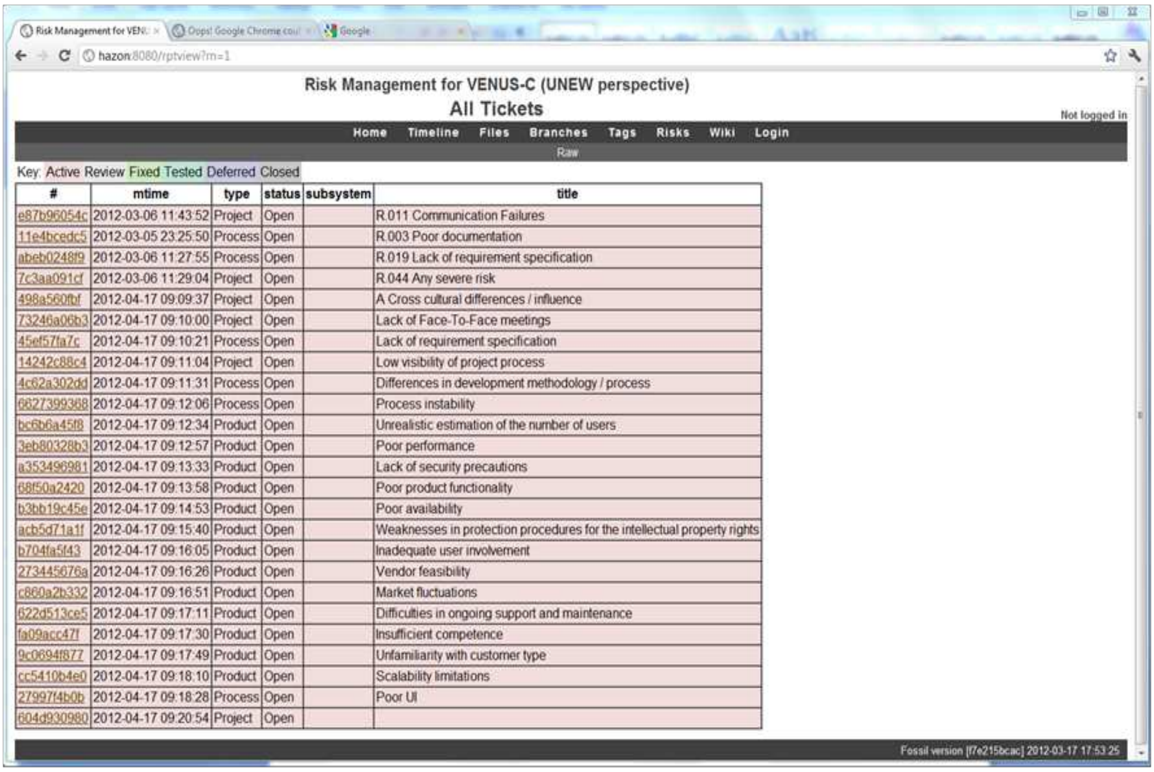
project card

Logged in as cmdadmin

Project Card

Project ID:	VENUS-C-UNEW-RM		
Opening Type:			
Project Name:	VENUS-C: WP-5 UNEW Drug Discovery Scenario		
Type:	Workflow-based Processing		
Customer	European Union		
Project Developer	Systems Group		
Project Manager	Paul		
Development Sites	One site		
Development Team	Simon, Hugo, Jacek		
Development Team Leader	Jacek		
Planned Start Date:	1/June/2010	Planned Finish Date:	31/May/2012
Actual Start Date:	1/June/2010	**Actual Finish Date:**	
Inital Contract Cost		Actual Cost at Delivery	
Specification file			
Events Registry			
Dependecy Projects	WP-4; WP-6; WP-3		

Fossil version [f7e215bcac] 2012-03-17 17:53:25


Risk Management for VENU ×
Oops! Google Chrome coul ×
Google
hazon:8080/rptview?rn=1
Home Timeline Files Branches Tags Risks Wiki Login
Raw

Risk Management for VENUS-C (UNEW perspective)

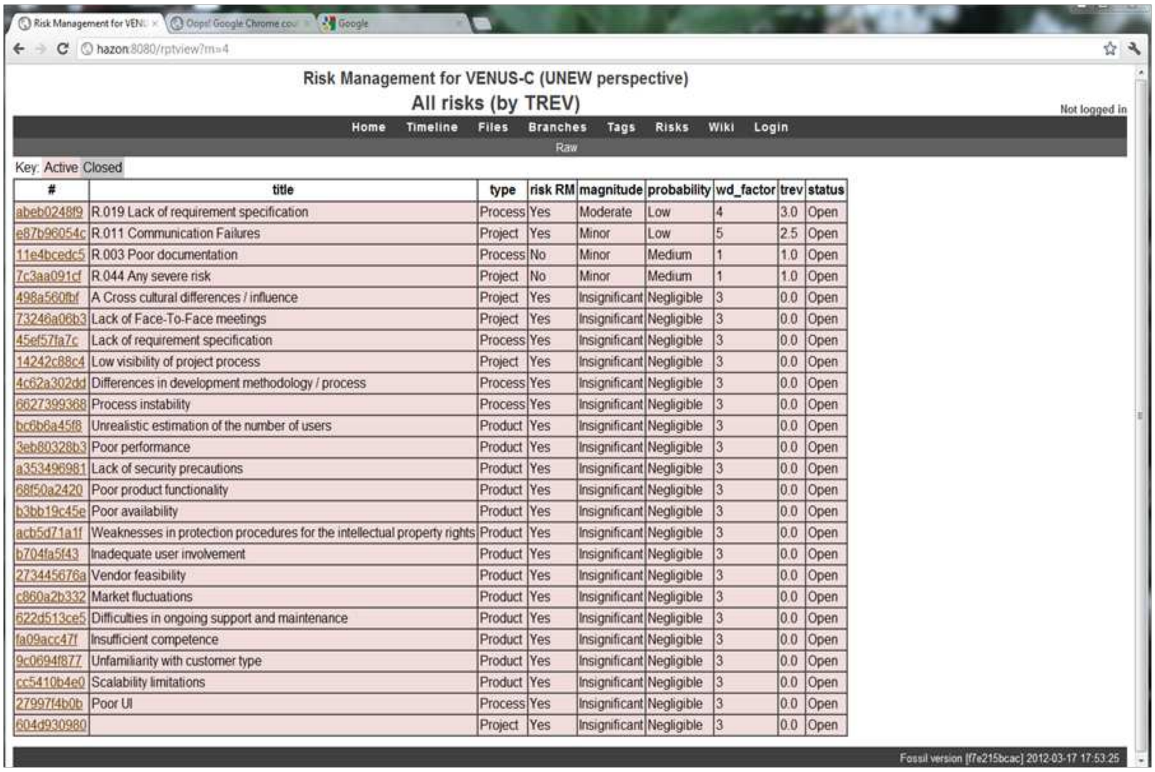
All Tickets

Not logged in

Key: Active Review Fixed Tested Deferred Closed

#	mtime	type	status	subsystem	title
e87b96054c	2012-03-06 11:43:52	Project	Open		R.011 Communication Failures
11e4bcedc5	2012-03-05 23:25:50	Process	Open		R.003 Poor documentation
abeb0248f9	2012-03-06 11:27:55	Process	Open		R.019 Lack of requirement specification
7c3aa091cf	2012-03-06 11:29:04	Project	Open		R.044 Any severe risk
498a560fbf	2012-04-17 09:09:37	Project	Open		A Cross cultural differences / influence
73246a06b3	2012-04-17 09:10:00	Project	Open		Lack of Face-To-Face meetings
45ef57fa7c	2012-04-17 09:10:21	Process	Open		Lack of requirement specification
14242c88c4	2012-04-17 09:11:04	Project	Open		Low visibility of project process
4c62a302dd	2012-04-17 09:11:31	Process	Open		Differences in development methodology / process
6627399368	2012-04-17 09:12:06	Process	Open		Process instability
bc6b6a45f8	2012-04-17 09:12:34	Product	Open		Unrealistic estimation of the number of users
3eb80328b3	2012-04-17 09:12:57	Product	Open		Poor performance
a353496981	2012-04-17 09:13:33	Product	Open		Lack of security precautions
68f50a2420	2012-04-17 09:13:58	Product	Open		Poor product functionality
b3bb19c45e	2012-04-17 09:14:53	Product	Open		Poor availability
acb5d71a1f	2012-04-17 09:15:40	Product	Open		Weaknesses in protection procedures for the intellectual property rights
b704fa5f43	2012-04-17 09:16:05	Product	Open		Inadequate user involvement
273445676a	2012-04-17 09:16:26	Product	Open		Vendor feasibility
c860a2b332	2012-04-17 09:16:51	Product	Open		Market fluctuations
622d513ce5	2012-04-17 09:17:11	Product	Open		Difficulties in ongoing support and maintenance
fa09acc47f	2012-04-17 09:17:30	Product	Open		Insufficient competence
9c0694f877	2012-04-17 09:17:49	Product	Open		Unfamiliarity with customer type
cc5410b4e0	2012-04-17 09:18:10	Product	Open		Scalability limitations
27997f4b0b	2012-04-17 09:18:28	Process	Open		Poor UI
604d930980	2012-04-17 09:20:54	Project	Open		

Fossil version [f7e215bcac] 2012-03-17 17:53:25



Risk Management for VENUS-C (UNEW perspective)

All risks (by TREV)

Key: Active Closed

#	title	type	risk RM	magnitude	probability	wd_factor	trev	status
abeb0248f9	R.019 Lack of requirement specification	Process	Yes	Moderate	Low	4	3.0	Open
e87b96054c	R.011 Communication Failures	Project	Yes	Minor	Low	5	2.5	Open
11e4bcedc5	R.003 Poor documentation	Process	No	Minor	Medium	1	1.0	Open
7c3aa091cf	R.044 Any severe risk	Project	No	Minor	Medium	1	1.0	Open
498a560fbf	A Cross cultural differences / influence	Project	Yes	Insignificant	Negligible	3	0.0	Open
73246a06b3	Lack of Face-To-Face meetings	Project	Yes	Insignificant	Negligible	3	0.0	Open
45ef57fa7c	Lack of requirement specification	Process	Yes	Insignificant	Negligible	3	0.0	Open
14242c88c4	Low visibility of project process	Project	Yes	Insignificant	Negligible	3	0.0	Open
4c62a302dd	Differences in development methodology / process	Process	Yes	Insignificant	Negligible	3	0.0	Open
6627399368	Process instability	Process	Yes	Insignificant	Negligible	3	0.0	Open
bc6b6a45f8	Unrealistic estimation of the number of users	Product	Yes	Insignificant	Negligible	3	0.0	Open
3eb80328b3	Poor performance	Product	Yes	Insignificant	Negligible	3	0.0	Open
a353496981	Lack of security precautions	Product	Yes	Insignificant	Negligible	3	0.0	Open
68f50a2420	Poor product functionality	Product	Yes	Insignificant	Negligible	3	0.0	Open
b3bb19c45e	Poor availability	Product	Yes	Insignificant	Negligible	3	0.0	Open
acb5d71a1f	Weaknesses in protection procedures for the intellectual property rights	Product	Yes	Insignificant	Negligible	3	0.0	Open
b704fa5f43	Inadequate user involvement	Product	Yes	Insignificant	Negligible	3	0.0	Open
273445676a	Vendor feasibility	Product	Yes	Insignificant	Negligible	3	0.0	Open
c860a2b332	Market fluctuations	Product	Yes	Insignificant	Negligible	3	0.0	Open
622d513ce5	Difficulties in ongoing support and maintenance	Product	Yes	Insignificant	Negligible	3	0.0	Open
fa09acc47f	Insufficient competence	Product	Yes	Insignificant	Negligible	3	0.0	Open
9c0694f877	Unfamiliarity with customer type	Product	Yes	Insignificant	Negligible	3	0.0	Open
cc5410b4e0	Scalability limitations	Product	Yes	Insignificant	Negligible	3	0.0	Open
27997f4b0b	Poor UI	Process	Yes	Insignificant	Negligible	3	0.0	Open
604d930980		Project	Yes	Insignificant	Negligible	3	0.0	Open

Fossil version [f7e215bcac] 2012-03-17 17:53:25