Risk Management for Web and Distributed Software Development Projects

Ayad Ali Keshlaf
School of Computing Science
Newcastle University
Newcastle, UK
a.a.a.keshlaf@ncl.ac.uk
ayadali2005@gmail.com

Steve Riddle
School of Computing Science
Newcastle University
Newcastle, UK
steve.riddle@ncl.ac.uk

Abstract— **The level of complexity and risks associated with software have been increasing in line with the growth of the software industry. Modern software development, with an emphasis on web and distributed development, presents specific challenges and risk areas to the software industry which need to be considered and managed. In this paper we survey a number of software risk management approaches and identify weaknesses such as the treatment of culture issues, geographical location, and process and product perspectives. These weaknesses must be tackled in order to accommodate the continuously evolving challenges to web and distributed software development and to cover some perspectives of the software industry which have not been well covered up to now. This work is a part of PhD research at Newcastle University (UK) to develop an improved approach to measure and control web and distributed development risks.**

Keywords-software risk management; web development; distributed development; software reliability

I. INTRODUCTION

Software development projects by their nature are a risky, complicated and multi-dimensional endeavor [1-3]. Software risks have been increasing for as long as the software industry has been growing [4]. 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) [5, 6]. SRM is still evolving, and many software managers have only a limited understanding of its concepts [3]. Industrial risk management practice tends to lag behind recommended risk management approaches, although there are exceptions [3, 7, 8]. This lag is clearer with Web and Distributed (W-D) software development, where the level of SRM practice is still low. This paper aims to investigate the abilities of existing SRM approaches in managing W-D software development risks, and to explore their weaknesses (the gap in the field). It gives a background on software risk management (Section 2), W-D development challenges and their source of risks (Section 3), and then it reviews the existing SRM approaches (Section 4), comparing them based on specific criteria factors (Section 5) in order to investigate their abilities to manage W-D development risks. We then present our conclusions and suggest future work in Section 6.

II. BACKGROUND

This section gives a background of SRM and its related definitions.

A. Software Risk

Software Engineering Institute (SEI) defines risk as "the possibility of suffering loss" [9] and it defines loss in a development project, as " the impact to 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." [9].

For each risk there are two aspects: risk probability and risk loss. These aspects are used to estimate the impact or Risk Exposure (RE) [10], as follows:

$$RE = P(UO) \cdot L(UO)$$ where,

RE is the Risk Exposure (or risk impact)
P(UO) is the probability of an unsatisfactory outcome
L(UO) is the loss associated to unsatisfactory outcome

Risk probability estimation is not a straightforward task and can not be 100% accurate (as otherwise there is no risk). Some probability estimation techniques use qualitative data and then convert it into its equivalent quantitative data using some equations, risk-probability table, checklists or relative scales [5, 10] where some others use subjective Bayesian approach [11] or other techniques.

The top ten software risk items (listed below), which are introduced by Boehm, are examples of sources of risk for software development projects [10].

- 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

978-0-7695-4023-8/10 $26.00 © 2010 IEEE
DOI 10.1109/ICIMP.2010.12

A further list of software risk items was introduced by the author and others [12] which includes:

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

Any list of software risk items will need to be updated from time to time, when there are new changes or challenges in software development technology and environment (e.g. social and culture issues, geographically dispersed, new technologies). The significance and type of risks and their sources will also inevitably evolve over time. As an example a recent review [13] found that different authors have identified or proposed different software risks which means that the number and items of software risks are not fixed. Therefore, new or improved methodologies, techniques and tools to identify, measure and control them are needed.

B. Software Risk Management (SRM)

Boehm [14] defined SRM as "a discipline whose objectives are to identify, address, and eliminate software risk items before they become either a threat to successful software operation or major sources of software rework". The main purpose of risk management is to identify potential problems of technical and management aspects before they occur and then take actions to decrease their impact [15]. Fig. 1 shows the basic steps of SRM [10].

C. Software Development Perspectives

Software development has three perspectives: project, process and product[16, 17]. Looking at these perspectives it is expected that each one of them includes, or could be affected by, different types of risks. For example, the "personnel shortfalls" risk item mainly affects the project perspective, "bad traceability" and "poor testing" affects process whereas "product with wrong functionality" affects product. However, one risk item may affect more than one perspective. Risk management is becoming an important issue from these three perspectives [16, 17].

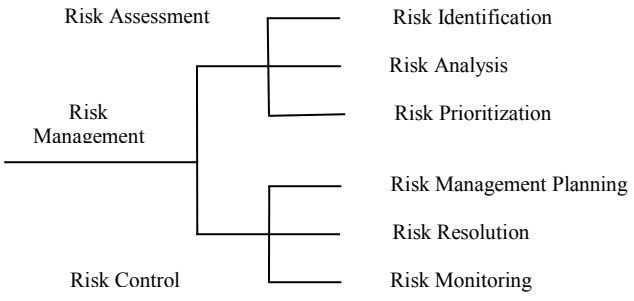

Figure 1. Software Risk Management Basic Steps [10]

III. CHALLENGES

A number of challenges to traditional software development can be seen in the fields of distributed and web development. The following section focuses on these challenges.

A. Distributed Development Challenges

Distributed Software Development as described by Jimenez and others [18] 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. Software developers have adopted distributed software development as a way of reducing the cost and increasing their projects productivity [19]. Developing software across distributed sites presents many challenges which are summarized in the following points[20, 21]:

- Inadequate informal communications
- Lack of trust
- Culture differences (e.g. different language, different corporate culture and different developers' background)
- Time-zone difference (leading to ineffective synchronous communication)
- Development process differences
- Knowledge management challenges (most of the existing management approaches are designed for co-located teams).
- Technical issue: Incompatible data formats and exchanges.
- Security issue (Ensuring electronic transmissions confidentiality and privacy).

All of these challenges could be sources of risk in a variety of development types. In the case of distributed development, they are particularly prevalent challenges and need to be considered by any proposed risk management approach.

B. Web Development Challenges

Web applications are a typical example of web developments, which have become a common type of modern software application. Mendes [22] defines a web application as "an application delivered over the Web that combines characteristics of both Web Hypermedia and Web Software application".

Web applications may be deployed instantly worldwide, without any need for installation and upgrading manuals [23]. They are growing very fast compared with the traditional software which makes them an important part of the business and software industry. High-performance web sites and applications are used widely in business-to-business ecommerce and many types of services as fully functional systems [24, 25].

The development, running and deployment environment of web development need to be considered carefully as well

as the significance of associated challenges and risks. Features of the W-D environment such as diversity and rapid change, present new challenges for the developer, manager, and to traditional project management approaches [25-28]. More effective risk management methods, models and tools should be introduced to tackle the lack of existing approaches to deal with these challenges [8, 29, 30].

The importance of web risks is different from others in a number of ways:

- Their impact and significance are different. For example the exposure to security threats is higher in the web [31-34].
- As web applications may be deployed instantly worldwide [23] their risks can affect wider range of components and applications simultaneously in very short period of time.
- Additional risk sources related to W-D environment include communication, culture, diversity and difference in geographical locations [35-38].
- Estimation of risk probability and loss is more difficult because of the involved challenges and relative lack of experience with them.

Ideally, assessment and management of web development risks should be performed during the whole life cycle of the projects [39], but unfortunately, the majority of web developers use a reactive risk strategy (they do not act until something goes wrong). This strategy is insufficient because it makes software projects vulnerable to any type of risks at any time without effective assessment and control [40].

There is no way to avoid risks in W-D development, so (as with other types of risk) the solution is to attempt to manage them. The following section gives an overview on the state of the art of existing software risk management approaches and illustrates their strengths and weaknesses.

IV. SOFTWARE RISK MANAGEMENT APPROACHES

There are many different SRM approaches. Some of these approaches are named "models" and others are named "frameworks" or "methods", but they have the same target, which is managing the software risks.

Existing SRM methods, models and tools are reviewed. Each of the approaches uses some steps, components or techniques which may be different or have some similarities with other approaches.

A. Existing Approaches

Among the existing approaches only nine of them have been selected for detailed comparison in this study. The nine approaches are the ones which are expected to satisfy the needs of risk management for software industry in W-D development environment. The approaches were selected because they are dedicated to manage W-D development risks, or related aspects. The compared approaches are described hereafter:

a) DS-RM-Concept:

Distributed Software - Risk Management Concept (DS-RM -Concept) has been designed based on the idea that communication and continuous risk assessment play a vital role in managing the risks. Risk assessment in this approach uses three concepts: reviews for risk identification, snapshots for analysis and reports for assessment [41].

b) EBIOS Methodology:

Originally the EBIOS (In French: Expression des Besoins et Identification des Objectifs de Sécurité) method has been introduced by Central Directorate of Security of Information Systems (DCSSI) in the French government. It is a risk management methodology concentrating on Information Systems Security (ISS) risks. It consists of a set of guidance steps and it is supported with a free open source software tool. The methodology has five phases: Context Study; Security Requirements Checklist; Threats Study; Identification of Security Objectives and Determination of Security Requirements [32, 42]. EBIOS could be used to manage some security risks in W-D.

C) ProRisk Framework:

ProRisk is an open system where the users can develop, calibrate a choice from published models (templates) or use different models to accommodate their project need. It is a risk management framework for small and large software projects. However, in order to provide project risk factor a detailed analysis of the project is required [43].

d) Riskit Method:

Riskit method is a software risk management method introduced by *Jyrki Kontio* [44]. Fig. 2 shows the process diagram of the method. Riskit method is designed to provide organized SRM process and to support involvement of all relevant stakeholders in risk management process [45]. The method is provided with analysis graph and it uses a specific ranking technique called Riskit Pareto Ranking Technique which uses probability and utility loss ranking [46, 47].

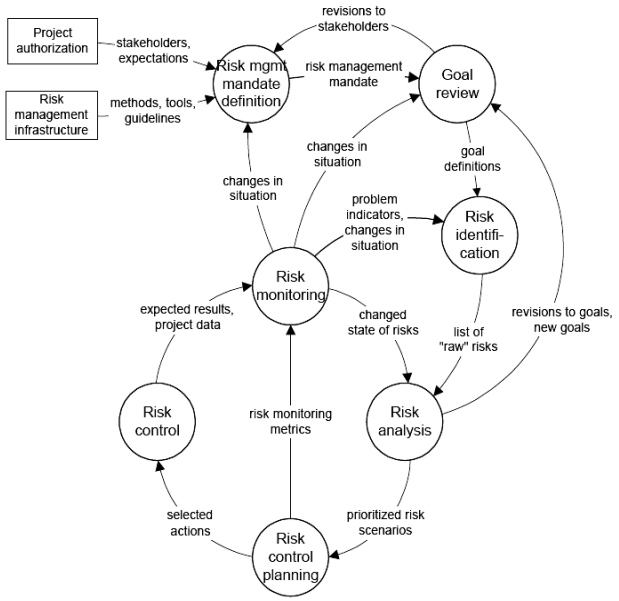

Figure 2. The Process Diagram of Riskit Method [44]

e) SoftRisk:

SoftRisk is model to manage software development risks introduced by the author and others[5]. Fig. 3 shows the main steps of SoftRisk model [5, 30, 48].

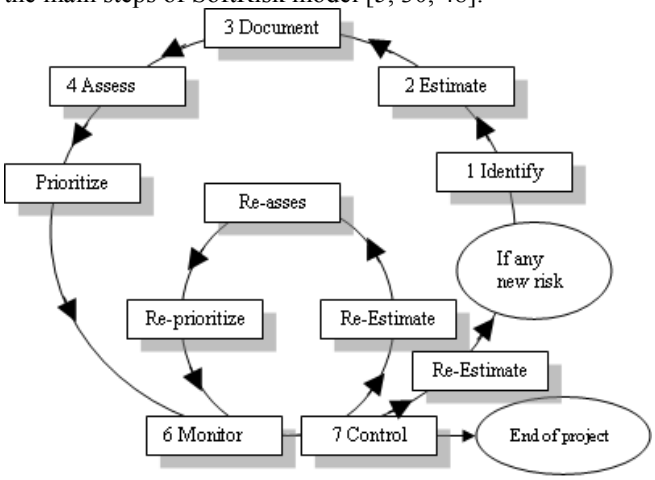

Figure 3. The Main Steps of SoftRisk Model[5]

The model is designed based on the idea of documenting and using historical risk data and focusing on top risks in order to reduce the effort and time in managing software risks. The model has been supported with a prototype tool.

f) CMMI-RSKM:

Capability Maturity Model Integration (CMMI) is an approach for improving processes within organization. The guidance which is provided by CMMI consists of a group of steps to improve development management, services, and maintenance of products. CMMI has RiSK Management (RSKM) process area and it has been adopted worldwide by many organizations. Its models cover development, acquisition, and services in projects [47, 49-51].

g) PMBOK RM Process:

Project Management Body of Knowledge (PMBOK) is a process introduced by Project Management Institute (PMI). Its third edition was published in 2004. The PMBOK combines nine areas of knowledge (Integration, scope, time, cost, quality, human resource, communications, purchase and risk). It consists of four process phases - Initiating, Planning, Executing, and Closing. It can be considered as standard for Project Management [47, 52, 53].

h) GDSP RM Framework:

Geographically Distributed Software Projects (GDSPs) is an integrated framework to manage risks in distributed software projects. It emphasizes on many aspects which are shared between GDSPs and web application developments. The idea behind this framework was based on synthesizing some known risks and risk techniques into integrated approaches. GDSPs links resolution techniques into project risk areas [38]. Elements of the framework are illustrated in Fig. 4.

i) Risk and Performance Model:

This model is designed to inspect the relationship between risk and project performance. This includes product and process performance. For this purpose six dimensions (*Organizational Environment, User, Requirements, Project Complexity, Planning & Control and Team risk*) of software risks are used by the model [54].

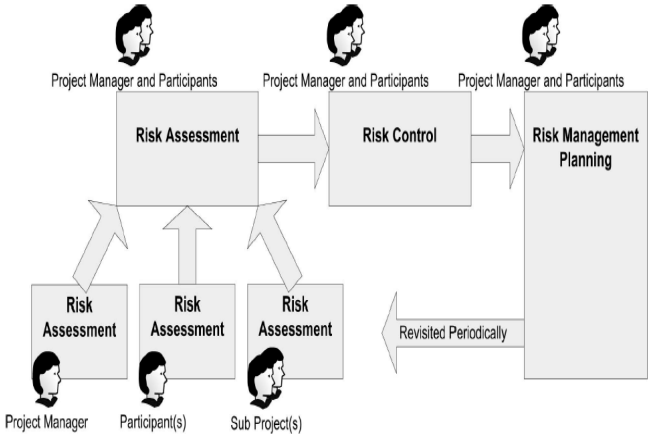

Figure 4. Elements of GDSP's Risks Management Framework [38]

V. ANALYSIS

The approaches were reviewed for their ability to manage risks of modern software development under the W-D environment and how they can deal with their challenges. In order to see their weaknesses and strengths, a comparison between them has been conducted based on our predefined criteria factors.

The criteria factors were prepared after the challenges, risk areas and characteristics of W-D development were identified, by conducting a risk management practice survey and literature search [16-18, 20-25, 28-41]. In order to get a consistent list of criteria factors initially, a list of all criteria factors has been created and then the most related ones to W-D software development were filtered. Meanwhile, some other factors are specified in order to cover aspects which we felt that were not touched before.

The factors cover important risk management aspects (e.g. Perspectives, Communications, Geographically Dispersed, Evolving Environment, Risk Management Evolution, culture issue and Interoperability tracking).

The comparison has been conducted based on available literature such as papers, reports, previous comparison, formal websites of the approaches and related technical reports (references are mentioned above in Existing Approaches Section). Table 1 shows the result of the comparison.

In Table 1 there are three options for each criteria factor:

- ✔ *when the factor is supported or agreed by the approach.*
- ✘ *if the factor is not supported or not agreed by the approach.*
- **P** *if it is partially supported or partially agreed by the approach.*

Table 1 can be read either horizontally or vertically. If it is read horizontally then the numbers on the table represent

the total of points that each criteria factor has got 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 of points each approach has got for each one of the above three options.

TABLE I. SOFTWARE RISK MANAGEMENT APPROACHES COMPARISON RESULT

Approaches / Criteria Factors	DS- RM Concept	EBIOS	ProRisk	RisKit	SoftRisk	CMMI-RSKM	PMBOK	GDSP-RM	Risk&Performance	Sub Totals: ✔	✘	P
Perspectives:												
- Project	✔	✘	✔	✔	✔	✔	✔	✔	✔	8	1	
- Process	✘	P	P	P	✘	✔	✔	✘	✔	3	3	3
- Product	✘	P	✘	✘	✘	P	✘	✘	P		6	3
Stakeholder :												
- Involved Stakeholder	✔	P	✔	✔	P	✔	✔	P	P	5		4
- Stakeholder Roles in SRM	P	P	P	P	P	✔	P	P	P	1		8
SRM & Product Quality Link	✘	✘	P	✔	✘	P	✔	✘	✔	3	4	2
Remote SRM	P	✘	P	✘	P	✘	✘	✔	✘	1	5	3
Estimating SRM Cost	✘	✘	P	P	✘	P	P	P	✘		4	5
Provided/Suggested Options :												
- Communications	✔	✔	P	✘	✘	✔	✔	✔	✘	5	3	1
- Collaboration	P	✘	✘	P	✘	✘	✘	P	✘		6	3
Consideration of:												
- Geographically dispersed	✔	✘	✘	✘	P	✘	✘	✔	✘	2	6	1
- Social and legal issues	✘	✔	✘	✘	✘	✘	✘	P	P	1	6	2
- Intellectual property	✘	✘	✘	✘	✘	✘	✘	✘	✘		9	
- Ethical issues	✘	✔	✘	✘	✘	✘	✘	✔	✘	2	7	
- Multicultural environment	✘	✘	✘	✘	✘	✘	✘	✔	✘	1	8	
- Evolving environment	✘	✘	✘	✘	✘	✘	✘	✘	✘		9	
Preparedness to Atypical Risk	✘	✘	✘	✘	✘	✘	✘	✘	✘		9	
Provided SRM Types:												
- Plain	✘	✘	✘	✘	✘	✘	✘	✘	✘		9	
- Deep / Ordinary	✔	✔	✔	✔	✔	✔	✔	✔	✔	9		
SRM Evolution Ability	✘	✘	P	✘	✘	✘	✘	✘	✘		8	1
SRM Effect Evaluation	P	✘	P	✔	P	✘	P	P	P	1	2	6
Learning from Mistakes	✔	✘	✘	✘	✔	✘	P	P	✘	2	5	2
Performance Evaluation	P	✘	P	✔	P	✔	✘	P	✔	3	2	4
Acceptable Levels	✘	✘	P	P	✔	✔	✔	✔	✘	4	3	2
Risks of SRM Exploration	✘	✘	✘	✘	✘	✘	✘	✘	✘		9	
Prediction Techniques	✔	P	P	✔	P	✘	✘	P	P	2	2	5
Side Affect Absorber	✘	✘	✘	✘	✘	✘	✘	✘	✘		9	
Interoperability Tracking	✘	✘	✘	P	✘	✘	✘	P	✔	1	6	2
Dependences Tracking	P	✘	P	P	✘	✘	P	P	P		3	6
Virtual SRM support	P	✘	✘	✘	P	✘	✘	✔	✘	1	6	2
Standard Operation Procedures	✘	✘	✘	✘	✘	✘	✘	✘	✘		9	
Risk Source Tracing	✘	✘	✘	✘	✘	P	✔	✘	✘	1	7	1
Totals :												
✔	Supported or agree	7	4	3	7	4	8	8	9	6	56	
✘	Not Supported or not agree	18	23	17	18	20	20	19	12	19	166	
P	Partially Supported or partially agree	7	5	12	7	8	4	5	11	7	66	
Total:											288	

From the numbers that appear in Table 1 it can be noticed that the total number of criteria factors that are supported or agreed by the approaches has got 56 points from the total of points which is 288 (with percentage 19%) The ones which are partially supported or partially agree have got 66 points (with percentage from the total of points 23 %) whereas the factors that have got the lowest support by the existing approaches have got the highest number of points, 166 (with percentage 58%). The criteria factors that have got the lowest support are:

- Covering of process and product perspectives
- Consideration of: Geographically dispersed, Social and legal issues, Intellectual property, Ethical issues,

Multicultural environment and Evolving environment
- Preparedness for atypical risks
- Plain risk management type
- Evolution of SRM processes
- Exploration of SRM Risks itself
- Risks side affects absorber mechanism
- Risks interoperability tracking
- Standard Operation Procedures

As can be seen in Table 1, 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 one. This is clear from the totals 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.

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

- Most likely that the existing approaches concentrate on project perspective of software development and they do not pay enough attention to other perspectives (Process and Product).
- They do not accommodate the continuous evolvement and changes issues of software industry and they do not consider aspects related to web, and distributed development environment (e.g. geographically dispersed, time zones differences, intellectual property, culture issues, evolving environment etc.).
- Lack of preparedness to atypical risks (No absorbing mechanism for side affects of atypical risks).
- They do not suggest any effective mechanisms to monitor or trace risks interoperability and dependences.
- They are not flexible enough and they offer only deep type of risk management. Plain risk management is not offered.
- Not enough monitoring to SRM performance and its associated risks.
- Most of the approaches are focused on theoretical aspects and do not provide clear guidelines for practicing.

VI. Conclusion and Future Work

In this paper we have identified W-D development challenges and shown how the importances of risks in web are different from others. Related existing software risk management approaches are reviewed and compared in order to investigate their weakness and strengths in managing W-D development risks. The comparison is based on special criteria factors which are prepared carefully in order to examine the ability of the approaches to manage the risks of W-D software development. It can be concluded that though there are many software risk management approaches the gap is still large between the existing approaches and actual practicing in software industry practice. This is due to the associated weaknesses in the approaches (e.g. not enough consideration to: difference in geographical locations, culture issues, process perspective and product perspective).

From Table 1 the following points can be concluded:

▪ There is no one approach which is able to manage software risks in W-D environments alone, unfortunately the strengths of the approaches are dispersed between them. In current situation the developers either have to use more than one approach or miss some aspects and support.

▪ Tackling the weaknesses of the approaches and combining the strengths of them in a new approach is a step toward improving risk management in W-D environment.

For effective risk management in W-D development all challenges, characteristics, risk areas, development and running environment and development perspectives (project, process and product) and other related aspects must be considered.

The reviewed approaches have added significant value to traditional software development projects, but it is clear that the web application developments are not yet well covered. As a part of PhD research an approach to manage W-D development projects risks is ongoing. The approach aims to tackle the existing approaches weaknesses and to come with new management concepts in order to improve the level of practicing of software risk management in the field. While the approach is particularly aimed towards W-D development, it should be applicable to modern software developments in general.

References

[1] Y. Kwak and J. Stoddard, "Project Risk Management: Lessons Learned from Software Development Environment," Technovation, vol. 24, November 2004, pp. 915–920.

[2] H. Yong, C. Juhua, R. Zhenbang, M. Liu, and X. Kang, "A Neural Networks Approach for Software Risk Analysis," Proc. of Sixth IEEE International Conference on Data Mining - Workshops (ICDMW'06), Hong Kong: IEEE, 2006, pp. 722-725.

[3] P. Bannerman, "Risk and Risk Management in Software Projects: A Reassessment," The Journal of Systems and Software - Elsevier, vol. 81, pp. 2118 - 2133, December 2008.

[4] A. Tiwana and M. Keil, "Functionality Risk in Information Systems Development: An Empirical Investigation," IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT, vol. 53, pp. 412- 425, AUGUST 2006.

[5] A. Keshlaf and K. Hashim, "A Model and Prototype Tool to Manage Software Risks," Proc. of First Asia–Pacific Conference on Quality Software, IEEE Computer Society, 2000, pp. 297–305.

[6] S. Islam, "Software Development Risk Management Model – A Goal Driven Approach," Proc. of ESEC/FSE Doctoral Symposium'09 Amsterdam - The Netherlands, ACM, 2009, pp. 5-8.

[7] J. Esteves, J. Pastor, N. Rodriguez, and R. Roy "Implementing and Improving the SEI Risk Management Method in a University Software Project," vol. 3, pp. 90-97, March 2005.

[8] M. Kajko-Mattsson and J. Nyfjord, "State of Software Risk Management Practice," IAENG International Journal of Computer Science - On-line Issue, vol. 35, 20 November 2008.

[9] R. Williams, G. Pandelios, and S. Behrens, "Software Risk Evaluation (SRE) Method Description (Version 2.0)," Software Engineering Institute (SEI) December 1999.

[10] B. W. Boehm, "Software Risk Management: Principles and Practices," IEEE Software, vol. 8, pp. 32-41, 1991.

[11] J. Moses, "Bayesian Probability Distributions for Assessing Measurement of Subjective Software Attributes," Information and Software Technology, vol. 42, 15 May 2000, pp. 533-546.

[12] K. Hashim and A. Keshlaf, "An Approach to Sharing Solutions to Software Project Management Problems," Proc. of International

Conference on Information Management and Engineering (ICIME '09), IEEE Computer Society, 2009, pp. 694 – 697.
[13] W. Han and S. Huang, "An Empirical Analysis of Risk Components and Performance on Software Projects," Journal of Systems and Software, vol. 80, January 2007, pp. 42-50.
[14] B. W. Boehm, Software Risk Management: IEEE Computer Society Press, 1989.
[15] IEEE Std. 1540-2001, "IEEE Standard for Software Life Cycle Processes – Risk Management," IEEE 2001.
[16] B. Boehm, J. Kwan, D. Port, A. Shah, and R. Madachy, "Using the WinWin Spiral Model: A Case Study," IEEE Computer, July 1998, pp. 33 - 44.
[17] 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, 2007, pp. 59-70.
[18] M. Jimenez, M. Piattini, and A. Vizca´ino, "Challenges and Improvements in Distributed Software Development: A Systematic Review," Advances in Software Engineering, vol. 2009, 2009, doi:10.1155/2009/710971.
[19] M. Malarvannan, "Managing Offshore Software Teams." vol. 2009: Outsource Portfolio, 2009.
[20] B. Sengupta, S. Chandra, and V. Sinha, "A Research Agenda for Distributed Software Development," Proc. of the 28th International Conference on Software Engineering (ICSE'06) Shanghai, China: ACM, 2006, pp. 731 - 740.
[21] K. Nidiffer and D. Dolan, "Evolving Distributed Project Management," IEEE Software, vol. 22, September/October 2005, pp. 63-72.
[22] E. Mendes and N. Mosley, Web Engineering, Berlin Heidelberg: Springer-Verlag, 2006.
[23] A. Taivalsaari, "Mashware: The Future of Web Applications." vol. 2009: Sun Microsystems Laboratories, 2009.
[24] J. Offut, "Quality Attributes of Web Software Applications," IEEE Software, vol. 19, March / April 2002, pp. 25-32.
[25] 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, 31 May 2006, pp. 19 - 32.
[26] C. Beise, "It Project Management and Virtual Teams " Proc. of SIGMIS'04 Conference Arizona, USA,Tucson, 2004, pp. 129-133.
[27] B. Behkamal, M. Kahani, and M. Akbari, "Customizing ISO 9126 Quality Model for Evaluation of B2B Applications," Information and Software Technology, vol. 51, pp. 599-609, March 2009.
[28] 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, November 2004, pp. 754 - 769.
[29] J. Kontio, M. Hoglund, J. Ryden, and P. Abrahamsson, " Managing Commitment and Risks: Challenging in Distributed Agile Development," in Proc. of the 26th International Conference on Software Engineering (ICSE '04), 2004, pp. 732- 733.
[30] M. Rabbi and K. Mannan, "A Review of Software Risk Management for Selection of Best Tools and Techniques," in Proc. of 9th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel / Distributed Computing, IEEE Computer Society, 2008, pp. 773 - 778.
[31] W. Glisson and R. Welland, "Web Development Evolution: The Assimilation of Web Engineering Security," in Proc. of the Third Latin American Web Congress (LA-WEB'05), IEEE Computer Scociety, 2005, pp. 5.
[32] B. Romero, M. Villegas, and M. Meza, "Simon's Intelligence Phase for Security Risk Assessment in Web Applications," Proc. of the Fifth International Conference on Information Technology, IEEE Computer Society, 2008, pp. 622-627.
[33] Y. Huang, C. Tsai, D. Lee, and S. Kuo, "Non-Detrimental Web Application Security Scanning," Proc. of the 15th International Symposium on Software Reliability Engineering (ISSRE'04): IEEE Computer Society, 2004, pp. 219 -239.
[34] X. Ge, R. Paige, F. Polack, H. Chivers, and P. Brooke, "Agile Development of Secure Web Applications," Proc. of the 6th International Conference on Web Engineering (ICWE'06), ACM, 2006, pp. 305-312.
[35] CA/ Wily, " White Paper: Application Performance Management, Effectively Managing High-Performing Business- Critical Web Application." vol. August 2008: Business-Critical Web Applications, 2008.
[36] G. Kappel, B. Proll, S. Reich, and W. Retschitzegger, Web Engineering the Discipline of Systematic Development of Web Application: John Wiley & Sons, Ltd,, 2006.
[37] V. Bruno, A. Tam, and J. Thom, "Characteristics of Web Applications that Affect Usability: A Review," Proc. of the 17th Australia conference on Computer-Human Interaction: Citizens Online: Considerations for Today and the Future (OZCHI 05), vol. 122 Canberra, Australia, 2005, pp. 1- 4.
[38] 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, 2009, pp. 1-25.
[39] S. Willis, Using QA for Risk Management in Web Projects, Software Quality and Software Testing in Internet Times. New York,USA: Springer-Veriag Inc., 2002.
[40] P. Pressman and D. Low, Web Engineering A Practitioner's Approach. International Edition: Mc Graw Hill, 2009.
[41] J. Gorski and J. Miler, "Towards an Integrated Environment for Risk Management in Distributed Software Projects," Proc. of 7th European Conference on Software Quality (ECSQ02),Helsinki, Finland, 2002.
[42] ENISA, "Ebios Product Identity Card.", ENISA, http://www.enisa.europa.eu/rmra/methods_tools/m_ebios.html, accessed on 17 June 2009.
[43] G. Roy, "A Risk Management Framework for Software Engineering Practice," Proc. of the 2004 Australian Software Engineering Conference (ASWEC'04): IEEE Computer Society, 2004, pp. 60-67.
[44] J. Kontio, "The Riskit Method for Software Risk Management, Version 1.00 CS-TR-3782 / UMIACS-TR- 97-38," University of Maryland, Maryland 1997.
[45] Kontio j. and B. V. R., "Empirical Evaluation of a Risk Management Method," in SEI Conference on Risk Management Atlantic City, Nj, USA, 1997.
[46] B. Freimut, S. Hartkopf, P. Kaiser, J. Kontio, and W. Kobitzsch, "An Industrial Case Study of Implementing Software Risk Management," Proc. of the 8th European Software Engineering Conference held jointly with 9th ACM SIGSOFT International Symposium on Foundations of Software Engineering Vienna, Austria, ACM, 2001, pp. 277- 287.
[47] J. Dhlamini, I. Nhamu, and A. Kachepa, "Intelligent Risk Management Tools for Software Development," Proc. of (SACLA 09) Mpekweni Beach Resort, South Africa: ACM, 2009, pp. 33 - 40.
[48] J. Smith, S. Bohner, and D. McCricard, "Project Management for the 21st Century Supporting Collaborative Design Through Risk Analysis," Proc. of 43rd ACM Southeast Conference Kennesaw,CA, USA: ACM, 2005.
[49] 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, 2005, pp. 155 - 159.
[50] SEI-CMMI, "What is CMMI?." vol. 2009: SEI - Carnegie Mellon University, http://www.sei.cmu.edu/cmmi/general/ accessed on 13-May-2009.
[51] R. Williams, "The CMMI RSKM Process Area as a Risk Management Standard," Proc. of Sixteenth Annual International Symposium of the International Council On Systems Engineering (INCOSE): INCOSE, 2006.
[52] D. Callegari and R. Bastos, "Project Management and Software Development Processes: Integrating RUP and PMBOK," Proc. of the 2007 International Conference on Systems Engineering and Modeling, IEEE, 2007, pp. 1- 8.
[53] W. R. Duncan, "A Guide to the Project Management Body of Knowledge PMBOK- PMI," Project management Institute, Boulevard -Newtown Square, USA 1996.
[54] 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, 2004, pp. 289 – 321.