JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 33, 1547-1560 (2017)
DOI: 10.6688/JISE.2017.33.6.10

Risk Management Approaches for Large Scale Software Development

SHEIKH TAHIR BAKHSH[1], BASIT SHAHZAD[2] AND SABEEN TAHIR[3]
[1]*Department of Computer Science*
Faculty of Computing and Information Technology
King Abdulaziz University
Jeddah, 21589 Saudi Arabia
[2]*Department of Engineering*
Faculty of Engineering and Computer Science
National University of Modern Language
Islamabad, 44000 Pakistan
[3]*Department of Information Technology*
Faculty of Computing and Information Technology
King Abdulaziz University
Jeddah, 21589 Saudi Arabia

Software risks management is a prominent area of research and has emerged over time with great significance. The identification and management of risks are important to complete the software successfully. In order to identify and handle the risks, different approaches are presented in the literature. Some approaches are feasible for small scale application, like mobile applications, while some approaches have their relevance with the large scale applications. In this paper, we present the available approaches to identify and manage the risk factor during the software development life cycle. A discussion is also presented that elaborates the strengths of the approaches discussed.

Keywords: wireless sensor networks, localization, mobile beacon, mobile anchor, RSSI

1. INTRODUCTION

Software projects can fail if the activity to develop them is not mature. Each software fails due to reasons that are generally different from the existing historical failures. Software failures not only dent the reputation of the firm but also cause the financial losses. Depending on the diversity of the failures, it becomes increasingly difficult for the firms to continue earning profits after such failures. The failed and challenged projects not only earn a bad name for organizations but the future business prospects are also compromised. The Standish Group, in its report published in 2009, has identified that the percentage of the challenged projects has increased to 44%, 32% of the projects failed while 24% got succeeded [1]. While in the report published by Standish group in 2013 [2] mentions that 39% projects remained successful, 18% failed or canceled while 43% of the projects got challenged. A project is declared challenged if it is somehow completed by overrunning the time and cost. Some other researchers like Emam [3], Cerpa [4] and Varajao [5] have also identified the similar trends in project failure.

Along with the improved emphasis on the risk management and resource allocation,

Received June 21, 2017; accepted August 18, 2017.
Communicated by Kashif Saleem.

software failures can also be reduced by maturing the software process [6, 7]. After CMM, which uses a five-level scale to measure the maturity of the process consisting of initial, repeatable, defined, managed and optimized levels, an advanced version of CMM is CMMI (Capability Maturity Model Integration) has been introduced. The CMMI also comprises of five stages to govern the process maturity. CMMI was developed by a joint collaborative work by Software Engineering Institute (SEI), Carnegie Melon University (CMU) and the US Department of Defense. CMMI orients many improvements from CMM and largely emphasis on the business needs, integration, and institutionalization [8].

CMMI Product Team [9, 10] has identified that the CMMI has a two-fold view of the capability: one view can be segmented while the other is continuous in nature. In the segmented view, there are five stages while the continuous view is composed of six stages. The levels are Incomplete, Performed, Managed, Defined, quantitatively managed and optimizing. CMM and CMMI when used at their best, can provide multiple benefits to the organization. The obvious gain is that the objectives are easy to achieve by tying the business and engineering activities: which involves the system level consideration of the software requirements and the possible way to meet them. Also, the software life cycle is mature enough to yield the software that is suitable enough to meet the customer's need and to develop the software within schedule by complying with the relevant international standards such as ISO 9000 [11].

Software risk management, over time, has evolved as a demanding discipline with respect to its influence on the software project's success. Along with the initiative to address the risks a right approach/model is to be chosen accordingly. The word 'approach' is used to describe a way of handling an issue while a model provides a top level view of addressing the issue by using some approach. Sometimes these words are used interchangeably, *e.g.* C. Misra [12] has used the words interchangeably for the software risk models. Some approaches are mentioned here in Section 2.

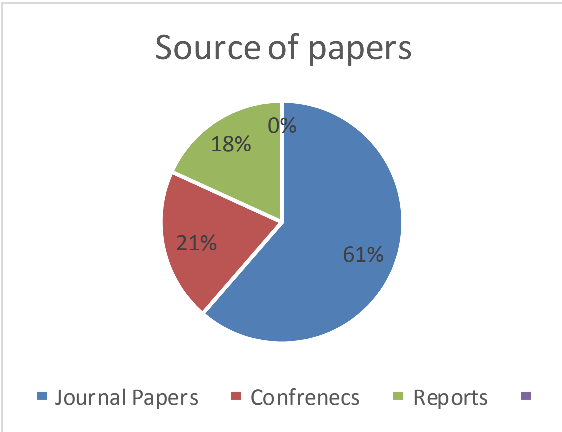
2. LITERATURE REVIEW

In this section, several approaches are presented for managing the risks for large scale software projects. The selection criteria for the selection of the related model and approaches went through a systematic review process. In this regard, following statistics were used. The search on Google scholar was initiated with the term ("software risk management" approaches for large scale projects), this returned 2150 result, while the time span was left open. In the next iteration, we limited the time span from 2000-2017, which resulted in a reduction of 300 papers. In the third iteration, we used the term ("software risk management" approaches for "large scale projects") to obtain the relevant papers discussing the risk management approaches for the large scale projects only. This resulted in significant reduction of the papers which was further purified by eliminating the books, which resulted in 56 articles. When these 56 articles were searched individually, some, 18 were found either not available or not relevant. The study, therefore, is based on the 38 articles, and their information is given in Table 1, and in Fig. 1.

The paper is based on 38 core references and 6 supportive references. The information graph about the selected papers is given here.

Table 1. Literature selection with respect to relevance.

Search Term	Results Returned	Results Discarded	Relevant	Time
"software risk management" approaches for large scale projects	2150	–	Partially	Open
"software risk management" approaches for large scale projects	1850	300	Partially	2000-2017
"software risk management" approaches for "large scale projects"	83	1767	Partially	2000-2017
"software risk management" approaches for "large scale projects"-books	56	27	Partially	2000-2017
Manual search on each article for availability, relevance, and readability	38	18	Fully	2000-2017



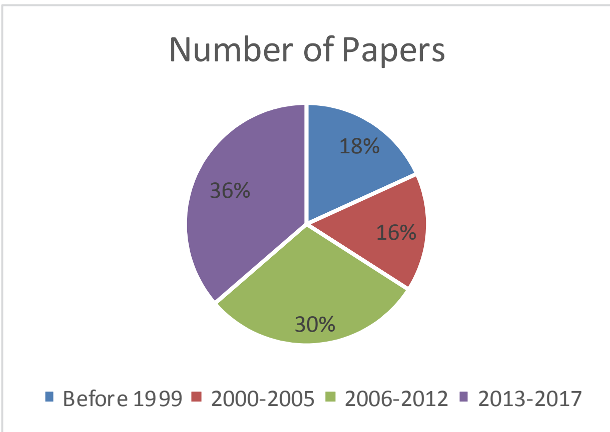


Fig. 1. Source and number of selected papers.

2.1 Halls P^2I^2 Approach

Hall [13, 14] identified that four factors people, process, infrastructure, and implementation are the core to define and implement the risk management strategy. The people factor is important as any plan, matters not how good it have to be implemented by the people working in a team. The process factor identifies the process that should be considered to reduce the possible threats of the risk orientation. The infrastructure determines the requirements, resources, and results to undertake the risk management in an organization. The implementation factor covers activities like identification, prioritizing, handling and mitigating the risks.

2.2 Karolak's Approach

It is a just-in-time approach and has [15] focus on managing the risks in early phases of software development activity to reduce time and cost required to manage risks. Karolak has identified some basic risk categories and associates risk factors with them which in turn are linked to the risk metrics and the questions that belong to the said risk factor and ultimately to the risk category. Miorando [16] has stated that the model is oriented to certain limitations. The questionnaire used in the formation of the model is isolated and not verified by the independent confirmatory studies. The model also lacks the adequate involvement of the users and tries to predict the technology risks directly without the user involvement, which may provide inconsistent results.

2.3 Kontio Riskit Approach

The implementation of the Riskit framework [17] helps the individuals to identify and communicate the information, opportunities, and risks to the concerned people to take necessary and immediate actions for the purpose of rectification. Riskit is also responsible for effectively managing the risk portion of the project, starting from the identification to the monitoring of the risk factors. Talet [18] has mentioned that Konito's idea is to use the experience and information gained from the previous project to effectively manage risks in the current projects under consideration. Felderer [19] has mentioned that apart from the meaningful usage the model has certain limitations. The model is dependent on the stakeholder's intentions and the development activity is not free from their influence. The model also requires that the 'experience repository' be well understood before applying this model.

2.4 Deursen and Kuiper's Approach

Duerson and Kuiper [20] proposed an innovative risk identification approach by considering the project based on the primary and secondary facts. Primary facts are relevant to the system where the system is studied while secondary facts are relevant to the stakeholders where the requirement documents, contracts, and communications are evaluated to identify any possible risk. After the completion of this phase, the primary and secondary lists are matched to see if they are consistent with each other.

2.5 Roy's Approach

Roy [21, 22] developed a framework that is considered to have pro-risk application and is focused on two aspects namely business domain and operational domain. Business domain measures the financial and other business points of the environment where the project has to be developed. The operational domain considers the formal modeling of diversified aspects of risk management in the projects. The activities include but are not limited to calculation of risk values, risk assessment, mitigation plan, implementation of mitigation plan and continuous monitoring. The model is very resource intensive and observes considerable complexity.

2.6 Tiwana and Keil's Approach

Tiwana and Keil [12, 23] have proposed a swift technique that can help the risk managers to quickly assess the project risks and their impact. The tool was developed as a result from the data gathered from 60 companies based on the questionnaire that was sent to them. This application has an advantage that instead of having the full-fledged detailed application this is cost effective and quick in identifying the threats to the system. Along with some features, the model has certain limitations. The model does not observe the absolute scaling and has been designed to provide the simple and quick results. At the cost of simplicity, the model may provide inconsistent results when the measuring scales are not effectively used.

2.7 Misra *et al.*'s Approach

Misra [12] proposed an innovative idea for risk management focusing on the fact that how the risk origin can be identified very early while the software development starts. The models discussed as yet consider 'what' question about the risk management process while this approach is inclined towards finding answers for 'how' the process can be detailed enough to find the origin of risks and assess and mitigate them in the very beginning while they are identified to reduce cost of the project and also the cost of mitigating risks by exploring the strategic dependencies. The application of the model is limited by the fact that it is too centric to the activities and does not really focus on the people and process that make these activities work. Along with that, the mitigation support provided in this model is minimal and not adequate in case of absolute disasters.

2.8 Foo and Muruganathan's Approach

Foo [24] believes that the impact of any risk factor can be quantified and thus he uses an approach that is based on a survey to seek opinion and impact of the risks. This way an ultimate threat level to the system can be defined. Their proposed model is called Software Risk Assessment Model (SRAM) that takes into consideration the risk of any specific situation to predict the risks. By considering the nine risk factors, they provide a list of three relevant categories and the assessor chooses the most relevant in their opinion.

2.9 SEI's Software Risk Management Approach

SEI's (Software Engineering Institute) at Carnegie Melon University has elaborated risk management framework that comprises of three modules

(a) *Software risk evaluation:* deals with the identification, dealing and mitigation of the risk factors under consideration for the given projects [25, 26]. The module uses a structure called the risk taxonomy. The taxonomy is a defined structure that helps in organizing and classifying the risks into their categories and has been extremely helpful is classifying the risks into different phases like Requirements Engineering &Analysis, Design, Software Development, Software Testing, Contract risks *etc.*
(b) *Continuous risk management:* it's a principle based approach that deals with the process, methods, and tools to help in the continuous management of risks throughout all phases of the software development lifecycle.
(c) *Team risk management:* is a team based approach that focuses to develop the process, methods and tools to determine the relationship of the customer and the teams and also the inter-team communication.

SEI's model has certain features and limitations. The model is limited in its application as it follows the waterfall approach and consequent phases cannot be started until the initial phase is completed, hence causes a lot of resources to be wasted. The model also over-emphasises the need for risk taxonomy and classification of the risks instead of establishing cohesive ways of addressing the risks in its early stages. Table 2 summarizes the features and shortcomings of all the models discussed in sections 2.1-2.9.

4. LIMITATIONS AND FEATURES OF THE APPROACHES

The limitations and features of the approaches discussed in section 2.1-2.9 are summarised in Table 2. Each approach has some factors and in Table 2, we are discussing the number of factors and have identified the relevant factors. The features and the limitations of the models are given along with the relevant reference from the literature.

Table 2. Features and Limitations of the existing risk management model.

Approach	Features	Limitations	References
SEI SRM	SEI SRM is a risk management model that advocates the development team to deal with the software risks. The model covers the risk management paradigm in all the phases of software development sequentially. The cohesion is gained by the continuous iterations.	The model is dependent on risk taxonomy, Phase dependency as it follows waterfall approach.	[25-29]
Hall $P^2 I^2$	Considers people's communication vital, Implementation phases to carry all the complexity	Duplication of tasks in process and infrastructure, Implementation is dependent on three factors; People's communication is subjective in nature and is difficult to model in precise terms.	[13, 14]

Karolak	Just-in-time approach, Applicable to early phases of the projects. The approach is simple and deals with most of the risks in early stages when the development is not complex.	Questionnaire is an isolated activity; Technology risks *etc.* may not be questioned directly, Only useful when people are involved. Which also generates a bottleneck of inconsistent communication standards, which can be modeled later.	[15, 16, 30, 31]
Kontio RISKIT	RISKIT provides a complete conceptual framework using goal-stakeholder approach. The timely dissemination of information provides the opportunity to stakeholders to understand the risk properly, and define a strategy to deal with the risks. The Riskit analysis graph counts utility loss by applying different functions.	Based on stakeholders intentions, Requires understanding of experience repository.	[17-19, 32]
Foo	Foo's model is based on the Questionnaire and Risk quantification and overall threat level is calculated based on the user feedback against the factors mentioned here.	Foo's model does not use any validation on the user feedback and uses the results as it is. In addition, the nine factors based on which the model is vague and less descriptive.	[24]
Deursen	Primary factors from system, Secondary factors from interviewing stakeholders, Use advantages of both	System viewpoint may be different from user, Qualitative data is relative and non-confirmatory, Inconsistent user responses	[20, 33, 34]
Roy	Based on AS/NZS 4350 standard, Business domain based on the environmental parameters, Operational domain covers aspects of risk management in projects.	Time and resource intensive, Complex	[21, 22, 35, 36]
Tiwana	Quickly accesses some risk factors, Based on questionnaires, Simple	Absolute scales are lacking, Results may be inconsistent in some circumstances	[12, 23]
Misra	Identifies the structural origin of the risk, Explores structural dependencies between actors of a project and analyses the motivation, intention and rationale behind activities in the project.	Too centric to activities and linkage to actors, Mitigation support minimal	[12, 37, 38]

4. DISCUSSION ON RISK MANAGEMENT APPROACHES

Several prominent risk management approaches have been presented in this section. Hall's model is developed by considering four factors including people, Process, Infrastructure, and Implementation. The model considers that the communication among the individuals is vital in the sense that they determine the execution circumstances. The model gains its momentum in the implementation phase and the other three factors don't get adequate weightage. As a limitation of the model, it can be identified that the activities performed in the 'Process' factor and in 'Infrastructure' factor somehow duplicate each other and at times one of them becomes redundant. Other limitation of the model includes the dependence of 'Implementation' phase on the outcome of the three factors. The communication of 'People' is subjective and neither documented not analysed, thus confirming it to be an informal activity. Fentron has also found that [39] the model demonstrates weakness in the coherence of its components and does not demonstrate sufficient strength to perform risk management in all phases of software development.

Karlok's model contains three categories including 'High level risk categories', 'Risk metrics' and 'Questionnaire'. This is a just-in-time approach, *i.e.* its works when it is applied and does not have much computational cost and back logs involved. Donovan [30] has experimented the model and reached a conclusion that the model is applicable from the earlier phases of software development and is simple in nature. While Banneman [31] has identified that the model is very suitable for deployment in the early phases and as the development activity moves in next phases the application of model reduces. The model is specifically designed to be deployed at the beginning of the development activity. Among some critiques on the model, one is that the questionnaire is isolated from the prime focus of the model. This model structure is such that it questions from the stakeholders about the risk they originated. In this case, different risks, *e.g.* Technology risks, Environmental risks *etc.* can't be questioned directly to add into the risk repository, which constructs major shortcomings for the model.

Federer [19] has mentioned that Konito's RISKIT model works on the historical data from the previous projects to be effectively applied. In absence of the historical data, the results of the model are compromised. The model presents a complete conceptual framework using the goal-stakeholder approach. The mechanism in the model allows the timely dissemination of information, opportunity, and risks to all stakeholders. The model in response establishes a graph that monitors the worth of each risk and decides that whether it should be handled or avoided. The model also calculates the possible utility loss due to the occurrence of the risks. Biffl [32] however has identified that the model is based on the intentions of the stakeholders and may be affected by the attitude of the individuals interacting with this. The model also demands the understanding of 'Experience Repository'. As this model has a high inclination of the stakeholder's opinion and the behaviour of the model can be affected by the stakeholder's intention, it may not be a good choice in large scale projects.

Deursen's model [20] introduces the primary and secondary risk factors and emphasizes on the concurrent matching of the primary and secondary risks factors. The model takes the primary risk factors from the system and secondary factors are derived from interviewing the stakeholders. The strength of the model is that it uses the advantages of both methods when they triangulate to converge. Beuwers [34] has identified that the model's working is limited by that fact that the system and stakeholder's viewpoint may differ in nature due to the understanding of the issue due to the availability of quantitative and qualitative facts respectively. The system viewpoint many not be as dynamic and contextual as the viewpoint of stakeholder's can be. If the number of originating risk factors is multiple in such project which requires change management, the working of the model is compromised.

SEI's [27] model is based on the three processes, namely software risk evaluation, continuous risk management and team risk management. The model is only suitable for large scale projects and requires the structural and massive resource commitments during the software development activity. It is suitable only for the development environment where there is established team structure with defined roles and responsibilities. The model possesses cohesiveness among the activities and is capable of covering the risks comprehensively. Hu [28] has identified that among the shortcomings the model is highly dependent on the risk taxonomy and its up-dation requires a clear definition of the category. Sundararajan [29] has mentioned another limitation of the model is its in-built waterfall structure where next activity can't be performed without the results from the following activity and increasing the dependency to proceed. Although the limitations are not big in nature and are model is mature yet it is more relevant to be applied to the project that has ample resources and high probability for the risk orientation.

The implementation of Roy's model come from the 'Business domain' and 'Operational domain'. The business domain is based on the business parameters of the systems while the operational domain covers different aspects of risk management in software projects. The model is based on the AS/NZS 4350 standard. Marquos has identified that [35] the model is resource intensive and complex as the approach is implemented by identifying staff, risk factors, constructing a risk free model, estimating probabilities, calculating the combined values for the project, developing action plan and monitoring program. If the limitations are over-come at some point, the model may have some usage to the large scale projects. Tiwana [36] has developed a model based on the survey that was conducted on 60 software firms. This model has been found suitable for small tasks and its efficiency for the large scale projects has not demonstrated positive results. The features of the model are that it is quick in assessing risks and does not need historical information.

Misra's model focuses on the identification of the risks from its origin and the model demonstrates that if the risk is not identified in the beginning its identification in the later stages becomes difficult. Koufaris [37] has mentioned that the model explores structural dependencies between actors of a project and analyses the motivation, intention, and rationale behind activities in the project. Li [38] has identified that the model possesses few limitations, one of them is that it is too centric to activities and linkage to actors. The model does not provide and guideline or activism to provide a support in mitigating the software risks.

Foo's model is implemented with the help of nine core categories including Software complexity, Project staff, Targeted reliability, Product requirement, Estimation method, Monitoring risks, and Development process adaptation, Usability of development software and Tools. The strength of the model comes from the fact that it is based on the questionnaires that are answered by the domain experts. The model gains wisdom from the questionnaires that help in identification of the risk factors in a project. The model is capable of indenting an overall threat level to the system based on the impact of the each risk factor identified, which gives a fair idea that hoe much risk management is to be done to bring the software out of the catastrophic range. The limitation of the model comes from the fact that it is based on the assumptions of the user and there is no mechanism in the system to validate the user response even though the quantitative data is easy to validate. This model is not highly resourced intensive and hence can be applied to medium and large scale projects with expectations of reasonable results.

5. RESULTS AND DISCUSSIONS

This research study identifies prominent risks, software project factors, probabilities and impact of the risk factors and identifies the relationship among the project factors and risk factors by using a mix of the quantitative and qualitative measures. Quantitative studies observe dominance as compared to qualitative studies as for as the number of studies is concerned. This research has some similarities with the Foo's model, *i.e.* both use the quantitative measure establishing the grounds of the studies. One similarity among the approaches is that Foo's model calculates the system threat level, which this study identifies as well. Keeping into consideration the examples of project failures presented in Table 2, it is important to give ample consideration to the project planning and managing the risks that are expected in the entire software development process. It is important to note that most of the failures are because of poor risk management and identification, therefore, considerable emphasis is being given to the identification of the risk so that a safeguard strategy can be prepared for that purpose [40]. Benaroch [41] provides some guidelines about the handling of the risks based on their impact. Caper Jones [42] has presented a list of 60 risk factors while Steve McConnel's [43] has presented a comprehensive list of the project scheduled risk factors that include checklist catalogue of 109 risk factors in 12 areas of software projects including schedule, product, personnel or customers. SEI presents 194 questions that can help in identifying the risk factors, to exhibit the level of commitment in identifying, managing [44], handling or avoiding the risk factors that can possibly be present in software development lifecycle.

6. CONCLUSION

The intention of this study is to propose a Risk Reduction Model that minimizes/ eliminates risk and reduces the cost of development for large-scale projects without spending too many resources on this exercise. Foo's model is also considered suitable for medium and large-scale projects, as it is not resourced intensive. Both studies contrast in a number of ways. Foo's model does not validate the user responses while the results are validated in this study at each level. Another contrast comes from the fact that Foo's model does not use qualitative data in the model while this research study not only uses the qualitative data but also triangulates that to converge to similar findings as of the quantitative results. The components on which the research studies are based are also largely different. The similarities and differences of both approaches are summarized in Table 2 which compares similarity and difference among the Foo's model and the model proposed in this research (Risk Reduction Model). It can be concluded that there is no vital similarity in the compared approaches apart from the methodological similarity which too is partial and the parameters, methods, processes and the results of both models are significantly different and are neither overlapping nor identical by any means.

ACKNOWLEDGEMENTS

This project was funded by the Deanship of Scientific Research (DSR) at King Abdulaziz University, Jeddah, under Grant No. (G-595-611-38). The authors, therefore, acknowledge with thanks DSR for technical and financial support.

REFERENCES

1. S. Group, 2013, "CHOAS Manifesto, think big act small," http://www.versionone.com /assets/img/files/CHAOSManifesto2013.pdf, 2016.
2. B. Shahzad, A. Md Said, and R. Aziz, "Using wage rate analysis to determine software project scale," *Research Journal of Applied Sciences*, *Engineering and Technology*, Vol. 8, 2014, pp. 99-104.
3. V. Benson, S. Morgan, and H. Tennakoon, "A framework for knowledge management in higher education using social neworking," *International Journal of Knowledge Society Research*, Vol. 3, 2012, pp. 44-54.
4. N. Cerpa and J. M. Verner, "Why did your project fail?" *Communications of the ACM*, Vol. 52, 2009, pp. 130-134.
5. J. Varajão, C. Dominguez, P. Ribeiro, and A. Paiva, "Critical success aspects in project management: similarities and differences between the construction and software industry," *Tehnički vjesnik*, Vol. 21, 2014, pp. 583-589.
6. M. Paulk, "Capability maturity model for software," *Encyclopedia of Software Engineering*, 1993.
7. J. Herbsleb, D. Zubrow, D. Goldenson, W. Hayes, and M. Paulk, "Software quality and the capability maturity model," *Communications of the ACM*, Vol. 40, 1997, pp. 30-40.
8. B. Curtis, B. Hefley, and S. Miller, "People capability maturity model (P-CMM) version 2.0," DTIC Document 2009.
9. C. P. Team, "CMMI for development, version 1.2," 2006.
10. M. B. Chrissis, M. Konrad, and S. Shrum, *CMMI for Development: Guidelines for Process Integration and Product Improvement*, Pearson Education, Boston, 2011.
11. F. Buttle, "ISO 9000: marketing motivations and benefits," *International Journal of Quality and Reliability Management*, Vol. 14, 1997, pp. 936-947.
12. S. C. Misra, U. Kumar, V. Kumar, and M. A. Shareef, "Risk management models in software engineering," *International Journal of Process Management and Benchmarking*, Vol. 2, 2007, pp. 59-70.
13. W. W. Royce, "Managing the development of large software systems," in *Proceedings of IEEE WESCON*, 1970, pp. 1-9.
14. E. M. Hall, *Managing Risk: Methods for Software Systems Development*, Pearson Education, Boston, 1998.
15. D. W. Karolak and N. Karolak, *Software Engineering Risk Management: A Just-in-Time Approach*, IEEE Computer Society Press, 1995.
16. R. F. Miorando, J. L. D. Ribeiro, and M. N. Cortimiglia, "An economic-probabilistic model for risk analysis in technological innovation projects," *Technovation*, Vol. 34, 2014, pp. 485-498.

17. J. Kontio, "Software engineering risk management: a method, improvement framework, and empirical evaluation," Department of Computer Science and Engineering, Helsinki University of Technology, 2001.
18. A. N. Talet, R. Mat-Zin, and M. Houari, "Risk management and information technology projects," *International Journal of Digital Information and Wireless Communications*, Vol. 4, 2014, pp. 1-9.
19. M. Felderer and R. Ramler, "Integrating risk-based testing in industrial test processes," *Software Quality Journal*, Vol. 22, 2014, pp. 543-575.
20. A. van Deursen and T. Kuipers, "Source-based software risk assessment," in *Proceedings of International Conference on Software Maintenance*, 2003, pp. 385-388.
21. M. Keil, P. E. Cule, K. Lyytinen, and R. C. Schmidt, "A framework for identifying software project risks," *Communications of the ACM*, Vol. 41, 1998, pp. 76-83.
22. R. Schmidt, K. Lyytinen, M. Keil, and P. Cule, "Identifying software project risks: an international Delphi study," *Journal of Management Information Systems*, Vol. 17, 2001, pp. 5-36.
23. A. T. M. Keil, "Reconciling user and project manager perceptions of IT project risk: A delphi study," *Information Systems Journal*, Vol. 12, 2002, pp. 113-119.
24. S.-W. Foo and A. Muruganantham, "Software risk assessment model," in *Proceedings of IEEE International Conference on Management of Innovation and Technology*, 2000, pp. 536-544.
25. A. Christopher and D. Audrey, "A framework for categorizing key drivers of risk," Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, 2009.
26. A. Christopher and D. Audrey, "Risk management framework," Technical Report CMU/SEI-2010-TR-017, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, 2010.
27. P. Chawan, J. Patil, and R. Naik, "Software risk management," *International Journal of Advances in Engineering Sciences*, Vol. 3, 2013, pp. 17-21.
28. Y. Hu, X. Zhang, E. Ngai, R. Cai, and M. Liu, "Software project risk analysis using Bayesian networks with causality constraints," *Decision Support Systems*, Vol. 56, 2013, pp. 439-449.
29. S. Sundararajan, M. Bhasi, and P. K. Vijayaraghavan, "Case study on risk management practice in large offshore-outsourced Agile software projects," *IET Software*, Vol. 8, pp. 245-257, 2014.
30. A. R. Donovan and C. Oppenheimer, "Modelling risk and risking models: The diffusive boundary between science and policy in volcanic risk management," *Geoforum*, Vol. 58, 2015, pp. 153-165.
31. P. L. Bannerman, "A reassessment of risk management in software projects," in *Handbook on Project Management and Scheduling*, Vol. 2, ed., Springer, 2015, pp. 1119-1134.
32. S. Biffl, D. Winkler, R. Mordinyi, S. Scheiber, and G. Holl, "Efficient monitoring of multi-disciplinary engineering constraints with semantic data integration in the multimodel Dashboard process," in *Proceedings of IEEE Emerging Technology and Factory Automation*, 2014, pp. 1-10.
33. T. A. Ghaleb, A. A. Alsri, L. Shabaneh, and M. Niazi, "A survey of project risk assessment and estimation models," in *Proceedings of the World Congress on En-*

gineering, 2014, pp. 2-4.
34. E. Bouwers, A. van Deursen, and J. Visser, "Software metrics: pitfalls and best practices," in *Proceedings of the 35th International Conference on Software Engineering*, 2013, pp. 1491-1492.
35. P. Marques, P. F. Cunha, F. Valente, and A. Leitão, "A methodology for product-service systems development," *Procedia CIRP*, Vol. 7, 2013, pp. 371-376.
36. A. Tiwana and M. Keil, "The one-minute risk assessment tool," *Communications of the ACM*, Vol. 47, 2004, pp. 73-77.
37. M. Koufaris, "Applying the technology acceptance model and flow theory to online consumer behavior," *Information Systems Research*, Vol. 13, 2002, pp. 205-223.
38. Y.-H. Li and J.-W. Huang, "Applying theory of perceived risk and technology acceptance model in the online shopping channel," *World Academy of Science*, *Engineering and Technology*, Vol. 53, 2009, pp. 919-925.
39. N. Fenton and J. Bieman, *Software Metrics: a Rigorous and Practical Approach*, CRC Press, 2014.
40. B. Shahzad and A. S. Al-Mudimigh, "Risk identification, mitigation and avoidance model for handling software risk," in *Proceedings of the 2nd International Conference on Computational Intelligence*, *Communication Systems and Networks*, 2010, pp. 191-196.
41. M. Benaroch, M. Jeffery, R. J. Kauffman, and S. Shah, "Option-based risk management: A field study of sequential information technology investment decisions," *Journal of Management Information Systems*, Vol. 24, 2007, pp. 103-140.
42. C. Jones, "Our worst current development practices," *IEEE Software*, Vol. 13, 1996, pp. 102-104.
43. H. Lin, A. Lai, R. Ullrich, M. Kuca, K. McClelland, J. Shaffer-Gant, S. Pacheco, K. Dalton, and W. Watkins, "Cots software selection process," in *Proceedings of the 6th International IEEE Conference on Commercial-off-the-Shelf (COTS)-Based Software Systems*, 2007, pp. 114-122.
44. S. Thomas and M. Bhasi, "Software development project risk: a second order factor model validated in the Indian context," *International Journal of Information Technology Project Management*, Vol. 3, 2012, pp. 41-55.

Sheikh Tahir Bakhsh has been bestowed upon by Gold Medal by the Rector COMSATS Institute of Information Technology, Abbottabad, Pakistan for securing 1st position in MCS in August 2006. Dr. Tahir has received the Ph.D. degree in Computer and Information Sciences from Universiti Teknologi Petronas, Malaysia in 2012. He joined the faculty of Computing and Information Technology, King Abdul Aziz University, Saudi Arabia as an Assistant professor in 2013. In the recent he has completed LTE HICI project with the collaboration of Stanford. He has also directed graduate and undergrad graduate projects. His areas of reach interests include bluetooth network, wireless sensor network (WSN), software, engineering, mobile ad hoc network (MANET), and computer networks. He works mainly on wireless network protocol designs optimizing the performance of networks. Recently, he has been involved in project related physical protocol design for bluetooth scatternet. He has published more than 40 journal articles and referred conference papers in these areas.

Basit Shahzad earned his Ph.D. from University Technology Petronas, Malaysia, and his MSc degree from National University of Science and Technology, Islamabad, Pakistan. Dr Shahzad, is a Visiting Scientist at University of Cambridge, UK and Visiting Fellow of Macquarie University, Australia. Dr Shahzad is a Collaborating Researcher with the Hagenberg Centre for Software Competence, Austria. His research and teaching career span over 16 years. Dr. Shahzad is currently at Faculty of Engineering and Computer Science, NUML, Islamabad as an Assistant Professor. Before this he has served King Saud University, Riyadh and has worked as an Assistant Professor at COMSATS Institute of Information Technology, Islamabad. Dr. Shahzad has numerous publications in journals and conferences of international repute and has a very active research profile. He has editorial role in several conferences and journals of high repute and has edited numbers of special issues in significant journals in the areas of software engineering, social networks, and mobile healthcare. Dr. Shahzad is reviewer of several high impact journals and is in teh program committee of several distinguished conferences. Dr. Shahzad values 'research' a lot and his career objectives revolve around conducting and supervising research in Information Systems (Enterprise Architecture, Software cost and risk modeling, Mitigating technological risks in modern banking), Advancements in Research Methodologies (Quantitative, Qualitative, Mixed method), mobile healthcare, and Social Networks (Theory, Empirical Social Network Analysis, Micro-blogging based analysis) *etc*.

Sabeen Tahir received the Ph.D. from Universiti Teknologi Petronas, Malaysia, in 2013. She started her career in academic in 2014, which prior to that she has worked in the IT field. She joined the faculty of Computing and Information Technology, King Abdul Aziz University, Saudi Arabia as an Assistant professor in 2013. In the recent years, she has involved in 4 research projects, inclusive LTE and wireless sensor networks. Her current research interests include wireless sensor networks, bluetooth radio networks, ad hoc wireless networks, mobile computing and traffic analysis.