COMPARATIVE STUDY OF RISK MANAGEMENT IN CENTRALIZED AND DISTRIBUTED SOFTWARE DEVELOPMENT ENVIRONMENT

Muhammad Perbat Baloch[1], Salman Qadri[1], Shafiq Hussain[2], Shabir Ahmad[3], Abu Buker Siddique[1], Farooq Azam[1]

[1]Department of Computer Science, The Islamia University of Bahwalpur, Pakistan
perbatali@gmail.com, salmanbzu@gmail.com, ab_siddique@yahoo.com, farooq_baryer@hotmail.com
[2]Department of Computer Science, Bahauddin Zakariya University, Sub-Campus Sahiwal, Pakistan
Email:shafiq.hussain57@gmail.com
[3] Department of Computer Science, Government College of Commerce, Multan, Pakistan
Email: mian_shabbir@hotmail.com (Corresponding Author)

ABSTRACT-- *Risk management is used to increase the chance of success of any future project by exploring its uncertainties. It will meet all the remedies to make the software development project successful by keeping in view all the future problems that may occur during the project process. It includes the identification of risk and their assessment in the project course and tries to make improvement to make project constructive. Risk management goals are to overcome project task risks those are identified before starting of the project and during the implementation. This paper describes the phases in the risk management process and provided methods to analysis and safety of management. The paper focuses on a study risk management in centralized and distributed software development projects. This study recognizes valuable, constant and free communication as the basics for victorious risk management. Therefore, it registers all incoming information memorize much in the same pattern as the "black box" device during an aircraft flight. The description and evaluation tools are also included, may used during the risk management study in the software development environment.*

KEYWORD: Risk Management, Software Development, Centralized and Distributed Environment, Risk Management, Software Risk, Risk Model

1. INTRODUCTION

Risk management is not new tool and a lot of standards and it is an integral component of good management and decision-making at all levels of an organization. At present, a further generic standard on risk management is in preparation as a common ISO/IEC standard [1] describing a systemic top down as well as a functional bottom up approach. This standard is intended to support existing industry or sector specific standards.

1.1 Software Risk

This growth along with application associated with software exposes the community to several threats. Initial, the malfunction of any software challenge to be a business starting brings about money along with time spend and an overlooked income opportunity. Raise the risk associated with like malfunction is referred to as it challenge risk. An additional risk relates to the security from the residents and the atmosphere. Failing of any software technique can lead to a car accident that, from the most detrimental scenario, could lead to loosing individual lifestyle. Here is the software security risk.

Regardless of the progress within technologies, it jobs nevertheless confront exactly the same troubles because 30 in years past [2]. Nonetheless, certain requirements from the customers usually are not profoundly realized, that brings about continual growth from the technique range or perhaps within sexual rejection from the remaining technique. This participation of people is actually non-stop incorporating the element associated with individual brain along with character on the techie problems from the jobs. Eventually, it is error-prone, the cooperation one of many challenge customers is frequently weak. Because of this, the anticipations from the purchaser usually are not pleased. Entirely, it calls for a number of major upgrades on the software growth along with exchange process. Certainly one of like influential solutions recognized by all of the software engineering along with challenge operations guidebooks [3] may be the danger operations.

1.2 Risk Management

Risk management is used to increase the chance of success of any future project by exploring its uncertainties. It includes the analysis of possible drawbacks (risk) in the project course and the alleviation of their negative potential. Boehm [4], argues in his research that by reducing risk in the project it will lead to reduce around 40% of software costs.

Risk analysis is a project wise approach for the identification of software development project risk. It is commonly considered that better risk assessment involve good communication on risk and proper documentation that may be collected on the basis of experiences and project risk knowledge which help to avoid the risks.

1.3 Notational conventions

Some models included in this paper are built with the Unified Modeling Language (UML) from Object Management Group. The specification of UML notation and some guidance on the practical use of UML may be found in [5, 6, 7] respectively.

2. REVIEW OF LITERATURE

Quality and success of a research is often a reflection of the time and effort invested in developing research ideas and concepts. The immediate goal of a literature survey is to determine whether the idea is worth pursuing or not.

2.1 SOFTWARE DEVELOPMENT PROJECT RISK

A simple definition of project risk states that it is a problem

that has not yet occurred but which could cause loss to one's project if it did [8]. The concept of risk is associated with a number of human endeavors ranging from space exploration and company acquisition to information systems development [9].

Empirical studies on how managers deal with risks show that the managers are not necessarily rational in reacting to risks. They look at a risky choice as one that contains a threat of a very poor performance [10]. Also, risk is not a probability concept; it deals with the magnitude of the bad outcome. Accordingly, managers act in a loss-aversive manner rather than a rational manner as predicted by the traditional theory.

The extensive literature review resulted in the identification of over 100 risk factors. The next step was to try to group similar factors together in order to get a clearer picture of the general types of software project risk factors. This resulted in the creation of 12 general types of software project risk categories.

Team related factors

- Effectiveness of task Communication
- Project Manager Characteristics
- Organizational Climate and Support
- External Factors
- Role of the user
- Formalization of project charter
- Project estimation and planning
- Tools and technology
- Requirement stability and accuracy
- Effectiveness of Project Monitoring
- Cross cultural and gender issues

2.2 Risk Management Practices

Risk management is concerned with a phased and systematic approach to analyze and control the risks occurring in a specific context. Software project risk management is risk management applied to the development and/or deployment of software-intensive systems. A typical risk management framework involves implementing and monitoring measures to reduce risk. Project risk management encompasses both hard skills such as estimating and scheduling tasks, and soft skills, which include motivating and managing team members [11].

In addition, risk management approaches feature a repertoire of risk resolution techniques. These are derived from local causal theories on how risky incidents affect software development and how interventions affect development trajectories. A thorough review of literature on risk management strategies for software projects, helped to identify a range of risk resolutions techniques which are discussed under following categories:

- Leadership Strategies
- HR Policies
- Training
- Project Coordination
- User Coordination
- Requirement Management
- Estimation Techniques
- Appropriate Methodology
- Project Control

2.3 Check Lists on Software Project Risk and Risk Management

One of the most common methods for identifying the presence of risk factors and risk management strategies in a particular project are the checklists. One of the pioneering studies in this regard is the top 10 risk list of Boehm [12]. His list has been compiled by probing several large software projects and their common risks and is thus empirically grounded.

One of the most quoted international studies on software project risk factors was conducted by Schmidt et al. [13]. In an attempt to compensate for some of the previous shortcomings in checklists of risk factors, Schmidt et al. (1996) conducted a survey of project managers and developed an extensive list of risk factors in software development. The particular research was conducted by three simultaneous Delphi surveys in three different settings: Hong Kong, Finland and the America. In each country, a panel of project managers was formed and a "ranking-type" Delphi survey was used to solicit risk items from the panel.

2.4 Review of Studies on Project Linking Risk, Management and Its Outcome

The studies referred above consider software risks along several dimensions and have provided some empirically founded insights of typical software risks and risk management strategies to mitigate them. Overall, these studies provide insights into risk management deliberations, but are weak in explaining the true impact of risk and risk management practices on the project outcome. A few studies have gone further to establish how risk management efforts reduce the exposure to software risk and can thereby increase software quality and improve software development.

A number of system performance criteria have been developed and empirically tested. Saarinen [14] proposed a system success measure with four dimensions: system development process, system use, system quality, and organizational impacts. Process outcome measures refer to the "successfulness" of the development process of the project. The focus is on completing the project within budget, within schedule and the on the overall quality of the development process. Both aspects are important as the software delivered by the project may be of high quality but the project itself may have exceeded the time and cost projections. On the other hand, well managed projects which come in below cost and time budgets may deliver poor products.

3. CENTRALIZED SOFTWARE DEVELOPMENT AND RISK MANAGEMENT

Today's software development has moved away from the "single team single location single management structure" paradigm to distributed, collaborating teams with flexible management relationships. In addition, recent experience with complex projects has shown that older development practices, with fully specified requirements and sign-offs and completely predetermined interfaces between major components, have substantial problems and are especially vulnerable both to schedule pressure and to unexpected changes and events. Finally, economic factors have

encouraged inter-organizational development practices such as outsourcing and off-shoring.

For these reasons, less centralized approaches to development have been pursued.

In *multi-organizational development*, participating teams work for different organizations. Multi-organizational development can be either:

Contractual, with one central authority (either one of the developer organizations or, less frequently, a customer) and other teams working on specific components with carefully specified predefined inter-faces and behavior, or

Cooperative, with teams working on sub-systems or low-coupled components with iteratively specified interfaces and behavior, often without a clear, universally accepted central authority for resolving differences and conflicts.

Both distributed development [15] and Centralized software development [16] introduce a number of new risk management concerns and modify or intensify others. Centralized soft-ware development entails a comprehensive change in the software engineering practices, from business case and product vision through development processes to management policies. Cooperation and communication concerns are significantly different, not only in level but also in kind. Software development requires a common product vision and architecture, extensive idea and design exchange, continuous communication, and active use of consultation, approval, and consensus con-strained only by intellectual property, privacy, and security considerations.

3.1 Principles of Centralized Risk Management

Successful collaboration requires collaboration-aware management, intra- and inter-organizationally. This entails *collaboration-aware risk management*, which is an extension of traditional risk management as well as team-based risk management [17, 18].

In the continuing application of the risk management process to large software development programs, the most dramatic effect has been in opening the communication channels for dialogues within organizations relating to risk and risk management.

In addition to the usual benefits of a rigorous approach to risk management, collaborative risk management may itself be an important early step in establishing trust and handling cultural and language problems. Cultural familiarity and trust have consistently been identified among the top four important success factors in collaboration [19].

3.2 A Framework for Effective Risk Management for CSD: A Layered Approach

An effective risk management plan should be based on Centralized-risk management principles and should provide clear definition of decisions, actions, and responsibilities related to the risk management functions defined in a collaboration-aware risk management plan must:

- Address traditional intra-organization risk identification and management in collaborating agencies.
- Handle risks identified as introduced or intensified by CSD, including risks within a single organization, resulting from interfaces, communication, and collaboration.
- Handle Centralized risks not well managed intra-organizationally.
- Drive incremental modification of policies, processes, and activities as needed.
- Support negotiation to resolve conflicts and to assign responsibilities for risk management.

Three alternative strategies for collaborative risk management include:

- Distribute responsibility for management of Centralized risks in modified, individual organization risk management plans.
- Handle new risks in a monolithic risk management plan.
- Pursue a layered strategy.

3.3 Three Critical Risk Factors: Trust, Culture And Communication

Successful identification, categorization, and evaluation of risk factors that arise in the collaborative software development domain are key challenges to software projects. Even though the majority of the traditional risk factors apply to CSD and some additional factors have been identified in the literature, there is a further need to systematically identify, characterize, and classify them and to support their effective treatment in RMMM plans for large-scale, high-risk Centralized development.

Differences in culture are primarily a *risk source* (an origin for problems), whereas trust is primarily a *risk driver* (a manifestation of an existing problem). Communication can be a source (e.g., mistranslation of requirements) or a driver (manifesting lack of management sup-port), or both. Each of these three factors is de-scribed in some detail below.

4. RISK MANAGEMENT FOR DISTRIBUTED ENVIRONMENT

The importance of risk management has been well recognized by the project management community. In risk management is listed among nine key knowledge areas related to project management. In relation to software project risks, much work has been done at Software Engineering Institute (SEI) [20, 21, 22, 23, 24, 25].

Software projects are exposed to various risks and risk management in such projects is still inadequate as is shown by the percentage of failed, delayed or too expensive projects [26, 27, 28]. The goal of a project is to deliver, in time and within the budget constraints, a product that meets stakeholders' needs and expectations. The essential factors of the project success are the quality, the time and the budget. Present software projects are often facing expanding and changing client demands and are put under schedule pressure. The systems are growing in size and become increasingly complex. To shorten the development time, the systems are built out of reused (but often not reusable) components.

The idea of having a constantly open and highly available channel for communicating and memorizing risk-related information is shown in Figure1. As the project advances, risks can be identified either during scheduled project activities or informally, e.g. when people talk to each other

at lunchtime, travel or during their leisure time. The idea of *risk black box* comes from the fact that memorizing this risk-related in-formation should be effective and as complete as possible (much like it is done during the aircraft flight). The difference to the aircraft black box is that we want to use this information with the proactive attitude, although we do not exclude its use for retrospection (e.g. to analyze the risk history after the project success/failure).

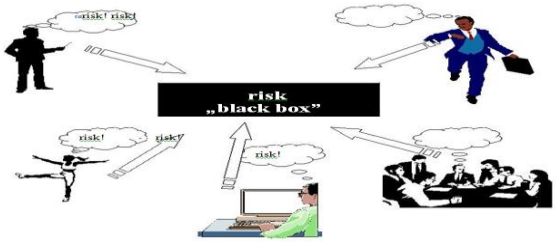

Figure1. Constantly open risk memorizing channel

5. RISK MODEL IN CENTRALIZED AND DISTRIBUTED SOFTWARE ENVIRONMENT

A risk scenario in the context of a given process can be expressed in more detail by investigating the constituent sub-processes of the contextual process (the *super-process*). The internal error propagation within a given process can be mapped to the external error propagation among that process' sub-processes. An error in a sub-process is also an error in its super-process. When an error in a sub-process causes this sub-process to fail, the failure remains internal to the super-process, unless that sub-process' external state is part of the super-process' external state (i.e. the sub-process delivers part of the super-process' service). In the opposite case, the error reaches the super-process' service interface and leads to the process failure. The inner structure of a risk scenario mapped to a process' sub-processes is shown in Figure 2.

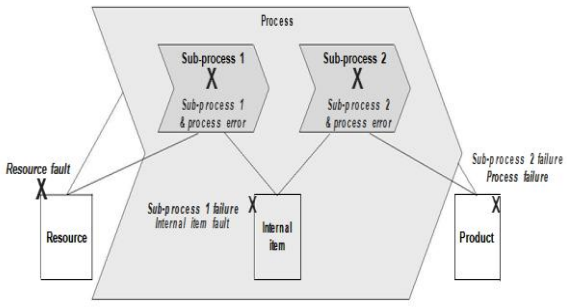

Figure2. Inner structure of a risk scenario within a process with sub-processes

6. RISK ANALYSIS IN CENTRALIZED AND DISTRIBUTED SOFTWARE DEVELOPMENT

The new techniques of risk analysis proposed for the Process Model-based Risk Assessment method. The techniques cover the initial processing of the information on identified risk with risk snapshots, relative ranking of risks, and the analysis of the overall process risk. The proposed techniques are described in detail in the following sections, preceded by a short overview in the introduction.

Risk analysis aims at providing the decision makers with the information on which of the identified risks should be mitigated and which could be accepted as well as which risks to mitigate first. To achieve this objective, the risk analysis needs some indicators that allow differentiating the identified risk scenarios according to the level of posed risk. Two new risk indicators are proposed:

Risk indications – the information on how the risk was identified in the risk identification phase,

Risk rating – the rating points explicitly assigned to the analyzed risk scenarios by the invited participants of a risk analysis session. The next sections detail the concept of the risk snapshot, explain risk ranking with the proposed indicators as well as discuss the estimation of the overall process risk.

6.1 Estimation of Overall Process Risk

Overall process risk is defined as the global level of risk present in the entire process.

It is very difficult (if not impossible) to estimate the overall process risk accurately. It is proposed that the overall risk associated with the particular classes of model elements is used as an indicator towards the estimate of the overall process risk. In the following sections, the overall risk metrics for the classes of model elements and the indicator of the overall process risk are defined.

Overall risk of activities – R_A

Let R_A denote the overall risk associated with the activities in a process model. R_A is estimated as the risk of activity R(A) summed up for all activities of the model, as given by equation 1.

$$R_A = \sum R(A), \quad R_A \in R_+ \cup \{0\} \qquad (1)$$

7. DISCUSSION AND RESULTS

The scope of the proposed method covers all the activities involved in the risk assessment:

- Risk identification
- Risk analysis
- Risk documentation
- Risk communication

The risk assessment process defined within the method follows the key principles of risk management indicated in the literature:

- Team participation
- Continuous process
- Open communication with provisions for information security
- Learning from experience

8. CONCLUSIONS AND FUTURE WORK

Many approaches have been already proposed under a common flag of the risk management to increase the projects' chance of success. However, the evidence shows that there is still a big gap between what we currently have in arms against the project risk and what we would wish to have. The investigation of the methods for the risk

assessment seems particularly worthwhile. Application of methodical support to risk identification and analysis (through explicit software process modeling and dedicated techniques) with dedicated software tools provides for early identification of project risks and increases the effectiveness of risk mitigation.

In this paper, we have provided a comprehensive, if preliminary, approach to collaborative risk management. We highlighted the differences between traditional and collaborative software development (CSD) that involves multiple organizational units and identified risk management principles for CSD that extend traditional and team-based risk classifications. On the basis of prior literature and our own field study, we then present a framework for CSD risk management and a layered approach for its implementation. Practitioners can use these ideas to develop an effective risk management plan for their particular kind of collaborative software environment.

Finally, an indicator of the overall process risk was proposed to assess the combined level of risk from the process activities, artifacts and roles. This indicator further allows for the advanced process analyses such as the simulation of risk resolution by process improvements or the assessment of process' risk tolerance.

We emphasized the essential role of communication in the risk management process and proposed a concept of a risk "black box" memorizing all the risk-related information arising in the project. We distinguished three hierarchical layers of risk assessment and explained how they interact. Finally, we presented a process of continuous risk assessment taking benefit from all the above ideas.

The proposed method may be further improved and extended in the areas like new risk patterns related to other classes of risk events, new metrics of process model structure providing more information on process risk, wider scope of tool support through further development of the RiskGuide tool.

REFERENCES

1. Sahibudin, Shamsul, Mohammad Sharifi, and Masarat Ayat. "Combining ITIL, COBIT and ISO/IEC 27002 in order to design a comprehensive IT framework in organizations." In *Modeling & Simulation, 2008. AICMS 08. Second Asia International Conference on*, pp. 749-753. IEEE, 2008.
2. Kerr, Eve A., et al. "Managed care and capitation in California: how do physicians at financial risk control their own utilization?." *Annals of Internal Medicine* **123**(7): 500-504 (1995).
3. Turner, Richard, and Apurva Jain. "Agile meets CMMI: Culture clash or common cause?." *Extreme Programming and Agile Methods—XP/Agile Universe 2002*. Springer Berlin Heidelberg, 2002. 153-165.
4. Keil, Mark, et al. "A framework for identifying software project risks." *Communications of the ACM* **41**(11): 76-83 (1998).
5. Booch, Grady, James Rumbaugh, and Ivar Jacobson. "The Unified Modeling Language For Object-Oriented Development, Documentation Set Version 1.0." (1997).
6. Fensel, Dieter, et al. "The unified problem-solving method development language UPML." *Knowledge and Information Systems* **5**(1): 83-131 (2003).
7. Azam, Farooq, et al. "Framework Of Software Cost Estimation By Using Object Orientated Design Approach." *IJSTR* **3**(8): 97-100 2014.
8. Wiegers, Karl. "Know your enemy: software risk management." *SOFTWARE DEVELOPMENT-SAN FRANCISCO*- **6**: 38-44 (1998).
9. Barki, Henri, Suzanne Rivard, and Jean Talbot. "Toward an assessment of software development risk." *Journal of management information systems* (1993): 203-225.
10. March, James G., and Zur Shapira. "Managerial perspectives on risk and risk taking." *Management science* **33**(11): 1404-1418 (1987).
11. Kirsch, Laurie J., et al. "Controlling information systems development projects: The view from the client." *Management Science* **48**(4): 484-498 (2002).
12. Boehm, Barry W. "A spiral model of software development and enhancement." *Computer* **21**(5): 61-72 (1988).
13. Carstensen, Peter H., and Kjeld Schmidt. "Computer supported cooperative work: New challenges to systems design." *In K. Itoh (Ed.), Handbook of Human Factors*. 1999.
14. Saarinen, Timo, and Timo Saarinen. "System development methodology and project success: an assessment of situational approaches." *Information & Management* **19**(3): 183-193 (1990).
15. Beranek, P. M.; Broder, J.; Romano, N.; Reinig, B.; (2005). Management of virtual project teams: Guidelines for team leaders. *Communications of the Association for Information Systems*, **s** 247−259.
16. Deek, F. P.; McHugh, J. (2003). Computer-supported collaboration with applications to software development. Kluwer Academic Publishers
17. Higuera R. P.; Gluch. D. P.; Dorofee A. J.; Murphy R. L.; Walker J. A.; Williams R.C. (1994). Introduction to team risk management, special report CMU/SEI-94-SR-1, May 1994. Retrieved Fall 2005, from http://www.sei.cmu.edu/pub/ documents/94.reports/pdf/sr01.1994.pdf
18. Higuera, R. P.; Dorofee A. J.; Walker J. A.; Williams R.C. (1994a). Team risk management: A new model for customer-supplier relationship, special report CMU/SEI-94-SR-5, July 1994. Retrieved Fall 2005, from http://www.sei.cmu. edu/publications/documents/94.reports/94.sr. 005.html Infante, D. A.; Rancer, A. S.; Womack, D. F. (1993).
19. Powell, A.; Piccoli, G.; Ives, B. (2004). Virtual Teams: A review of Current Literature and Directions for Future Research. *The DATA BASE for Advances in Information Systems, 35 (1).*
20. Galagher B. P., Software Acquisition Risk Management Key Process Area (KPA) A Guidebook Version 1.02, SEI report CMU/SEI-99-HB-001, Carnegie Mellon University, Pittsburgh PA, October 1999.
21. Higuera R. P., Gluch D. P., Dorofee A. J., Murphy R. L., Walker J. A., Williams R. C., An Introduction to Team Risk Management, SEI report CMU/SEI--94-SR-01, Carnegie Mellon University, Pittsburgh PA,

May 1994.
22. Higuera R. P., Haimes Y. Y., Software Risk Management, SEI report CMU/SEI--96-TR-012, Carnegie Mellon University, Pittsburgh PA, June 1996.
23. Jones C., Assessment and Control of Software Risks, Prentice Hall, 1994.
24. McConnell S., Code Complete, Microsoft Press, 1993.
25. Miler J., Górski J., Implementing risk management in software projects, Proc. of 3rd National Software Engineering Conference, Poland, 2001.
26. ACT Insurance Authority 2004. *Risk Management Toolkit.* February 2004.
27. Ahmad, Shabir, and Bilal Ehsan. "The Cloud Computing Security Secure User Authentication Technique (Multi Level Authentication)." *IJSER* **4**(12): 2166-2171 (2013).
28. Khan, Kamran, et al. "Evaluation of PMI's Risk Management Framework and Major Causes of Software Development Failure in Software Industry." *IJSTR* **3**(11): 2014