International Journal of Research and Development in Applied Science and Engineering (IJRDASE)

Study of Software Risk Analysis Models on Distributed Systems

Suruchi Shukla
Research Scholar,
S. V. U., Merut, U.P., India
suruchi.shukla444@gmail.com

Dr. Mohammad Husain
Research Supervisor,
S. V. U., Merut, U.P., India

***Abstract*—Risk management is an integral part of the software development process. Since risk assessment is the foundation for other risk management activities, it should drive the development process to ameliorate security issues. Developers are expected to identify, assess, rank, mitigate and manage risk throughout the software product life cycle. Methodologies used to allow risk to drive the development process have in large part been qualitative in nature. Risk analysis is a process for considering possible risks and determining which are the most significant for any particular effort. Determining which risks to address and the optimum strategy for mitigating said risks is often an intuitive and qualitative process. An objective view of the risks inherent in a development effort requires a quantitative risk model In this paper we analyze different factors and models of software risk analysis**

Keywords*—*Distributed System, Risk Analysis, Risk Analysis Models, and Risk Analysis Factors

1. INTRODUCTION:

By definition, project risks are uncertain events or conditions that, if occurs has a negative effect on project's objectives. Contrasting, positive uncertain events are called opportunities. More, risk analysis aims its practices to be tailored to the project and congruent with the organizational culture, processes and assets. Risks are unequally important, that's why it is very important to filter and prioritize risks for further attention. Risk analysis is essential for successful project management and aims to identify and prioritize risks in advance of their occurrence, and provide action-oriented information to project managers. This paper focuses on the risk analysis of distributed software projects. In developing a common frame of reference concerning management of distributed software projects, the conceptual foundations of previous research are analyzed. Additionally, practice and research-related challenges for managing distributed software projects are presented. Addressing these challenges is the primary area of concern for this Paper.

.

2 Risk Analysis:

Risk analysis aims at increasing a project's chance of success by addressing explicitly the uncertainties of the future. It involves the assessment of possible pitfalls in the project course and the mitigation of their destructive potential. Risk assessment is a project-wide systematic approach to the identification and analysis of project risk. It is commonly recognized that effective risk assessment requires communication on risk and risk documentation as well as the reuse of experiences gathered in the risk knowledge bases, which help in avoiding known dangers and learning new ones. Risk assessment and its supporting processes can also benefit from various levels of tool support. The goal of this work is to propose a method of software project risk assessment that provides for early identification of process risks and helps in their effective mitigation. One of such influential approaches recognized by all the software engineering and project management guidebooks [Pressman 2004] is the risk analysis

In a multidimensional space of the project objectives (or expectations), during project phases and based on the stakeholders evolution, the project passes a trajectory [3]. Risk may be described as the distance between the objectives (or stakeholder expectation) and the current situation (or the perceived current situation) inn the upper described multidimensional space. For this reason, target requires to be well defined, well known, and well documented for project. In the sales phase of projects, each large project crosses several phases that are relevant in terms of project Risk Management (PMR). Risk management should be applied at major project milestones and hence be included in project plans and operational documents becoming integral part of every aspect of managing the project, in every phase and in every process group.

3. Principles of Risk Management

Depending on the source, the process of risk management is defined slightly differently in terms of the detailed steps. There is however a consensus that the risk management must comprise two general phases: the first being analytical in nature and the second being synthetic. The former is typically named the risk assessment and the latter - the risk mitigation. The risk assessment phase involves the following activities, generally performed in a sequence:

- Risk Identification – identification of risk incidents threatening the project success as well as their risk factors; detection of the risk scenarios,
- Risk Analysis – transforming the raw information on risk into a decision-enabling knowledge by judging the level of risk exposure with a chosen scale (risk estimation) and ordering the risks by their relative weight (risk ranking),
- Risk Evaluation – deciding on risk acceptance by evaluating the risks against an acceptability
- scale or threshold. The risk mitigation phase includes the following activities:
- Planning Risk Control Measures planning the remedies to implement in order to proactively minimize the risk before it occurs (mitigation planning) or to reactively recover from the loss and restore the normal process state (contingency planning),
- Implementing Risk Control Measures – execution of the risk preventive tasks defined in the mitigation plans as well as preparing for the effective launching of the reactive tasks from the contingency plans,
- Monitoring – checking the effectiveness of the implemented risk control measures; involves overseeing the execution of the plans (task monitoring) and observing the current levels of mitigated risks (risk monitoring),
- Controlling – steering the risk mitigation based on the actual effectiveness of the control measures and the levels of risk, deciding on launching of the contingency plans or closing a successfully mitigated risk,
- Learning On Risk – abstracting the experiences from risk assessment and mitigation into a reusable knowledge; includes recording the specific highly context-dependent risks as well as the successfully applied risk control measures (remedies) in a risk knowledge base.

The activities of risk assessment and risk mitigation are executed within a framework for risk communication and risk documentation. These two core processes span the entire risk management and tie together its specific activities. With malfunctioning communication and documentation backbone the risk management is destined to a certain failure. It is generally agreed that, to be successful, the risk management must be run as a continuous process involving repeated risk assessment and project-wide risk mitigation.

Of the entire risk management process, the focus on risk assessment is particularly worthwhile as it is definitely the enabler of effective risk management – the risks overlooked or neglected in the risk identification and analysis will not be put under control and mitigated successfully. It is also crucial to identify the risk as early as possible to be able to apply the best risk mitigation strategy.

4 Analyses of Globally Distributed Software Development

Globally distributed software development has become a common practice in today's software industry; companies cross the barriers introduced by distance, cultural differences, and time zones, looking for the most skilled personnel and the most cost-effective solutions. Globally distributed software development may exacerbate several of the criticalities already present in traditional local software development, and it often generates its own peculiar challenges originating in the difficulty of carrying out the traditional parts of a software development project requirements elicitation, API design, project management, team communication, etc. in environments where members of the same team live and work in different countries, or even in different continents.

Given the challenges and peculiarities introduced by globally distributed software development, it is interesting to peruse the standard methods and practices that have been successful in traditional local software development, determining if they can be applied with positive results also in globally distributed settings. From the perspective of empirical research in software engineering, this general line of inquiry materializes in questions of the form What is the impact of X on the quality of globally distributed software development projects", where X" is a practice, method, or technique, and quality" may refer to different aspects such as timeliness, customer satisfaction, cost effectiveness, or the absence of problems. Examples of globally distributed software development issues investigated empirically along these lines include the usage of contracts for API design [26], the effect of time zones on various phases of development and on productivity and quality, and the impact of geographic dispersion on several quality metrics[27].

5. Risk Analysis Models

5.1 SERIM Software Engineering Risk Model (SERIM) focuses on three risk elements: (i) technical risk, (ii) cost risk, and (iii) schedule risk. The model does not take into account of the software complexity issues, which plays an important role in determining the risk for the software projects. It also does not account for issues related to requirements. The purpose of SERIM is to enable assessment of risk factors in software development from several different perspectives, and developing focused action plans to manage risks before they become realities. SERIM takes periodic “readings” on the status of software development projects so there can be a focus on high-priority risk areas. After risks are identified, SERIM helps to develop proactive plans for mitigating risk before they sabotage projects. The SERIM method is a simple and flexible way to perform software risk management. It is particularly well suited for small manufacturers that may not be able to use more expensive and complex processes. SERIM overview is as follows:

- Identifies different risks for technical implementation, cost, and schedule
- Predicts risks by software development phases
- Provides a means for corrective action to reduce risks
- Identifies the effectiveness of your software risk management activities
- Measures the risk associated with your software product and process
- Handles multiple projects for analysing software risks :Karolak, (1998).

5.2 SRAEM Measurement error, model error, and assumption error Risk exposure and mission critical requirements stability risk metrics. Software Risk Assessment and Estimation Model [23] is the latest model in the field of software risk estimation published in 2008. Initially the model estimates the sources of uncertainty using Measurement error, Model error and Assumption error. It considered the concept of functional point to explain the measurement error, Model error, and assumption error. Functional point is an important software metrics which is used to calculate the approximate LOC, Cost and effort of software. In SRAEM, there are two ways which are used to prioritize the risk. One method was based on risk exposure and the other method is based on Mission Critical Requirements Stability Risk metric (MCRSRM).

5.3 SRAEP using Model based approach Identify context using use case diagram, sequence diagram and security requirements Calculate risk exposure and compute degradation of key node safety metric The model considers the nine critical risk elements

(i) Complexity of the software;
(ii) Staff involved in the projects
(iii) Targeted reliability
(iv) Product requirement
(v) Method of estimation
(vi)Method of monitoring
(vii) Development process adopted
(viii) Usability of software
(ix) Tools.

The above existing risk assessment models does not include the sources of estimate uncertainty, i.e. measurement error, model error and assumption errors.

6.Current Practices in Risk Assessment

Identification of software project risk currently relies on two major techniques: checklists and group work, while the software project risk analysis most commonly estimates risk in qualitative scales, as opposed to quantitative ones..

6.1 Checklists

A *checklist* is a referential list of typical risk factors of project failure compiled from the experiences of some past projects. Checklists usually have the form of a *questionnaire* or a *risk list.* A questionnaire comprises a set of questions that ask for the current state of the project. Some questionnaires use only yes-no questions, some include closed test questions and some other employ open questions. The questions directly indicate the existing risk factors of project failure as well as indirectly guide towards some potential risk factors. The risk factors are identified usually by a negative answer. A risk list is directly composed of some typical risk factors i.e. the statements describing a particular state of a software project. Some risk lists express the risk factors as single statements in natural language, while the other ones provide detailed information on a risk factor through a complex data structure covering e.g. root causes, impact, methods of prevention Both, the questions of a questionnaire and the risks of a risk list are arranged in categories – *risk areas*. Typically, the risk areas refer to the disciplines or phases of the development process.

A checklist is generally used by following all its points one by one and considering them in the context of a given project. In case of a questionnaire, the questions are asked against the current state of the assessed project and the answers should indicate the risk factors sought for. In case of a risk list, each listed risk factor is compared against the current project state and decided whether it is already present in the project, it can be present in the future or it cannot occur in the assessed project at all (i.e. it is not applicable or it was made impossible to happen). It is important to notice that for a trustworthy risk identification with a checklist the user is essentially required to follow all and every one of the checklist points. Interrupting the project risk survey in the middle of the checklist puts at doubt the reliability of the findings (particularly their completeness). This impact on reliability is even stronger because checklists are not ordered so that the 'most important' issues would be positioned in the beginning, and the 'least important' ones would lay at the end. In a checklist, all the issues are 'equally (very) important'. Although checklists are intended to be precise and objective, they still involve a considerable dose of subjectivity especially where the project state is being assessed. A common technique to reduce the subjectivity in checklist application is to gather the answers from more than one viewpoint, typically represented by a role of the project participant (e.g. project manager, analyst, developer, customer, end-user).

6.2. Group work

Group work is a collaborative technique that relies on high human involvement and employs synergistically augmented human intuition and reasoning from individual experiences as a primary means of risk identification. Group work is often implemented as a *brainstorming session*. Brainstorming sessions are moderated face-to-face meetings of the (selected) project team members focused on the spontaneous collaborative identification of project risk based on the individual experiences, judgment, reasoning, and intuition of the session participants. A session is led by a moderator and usually takes about one hour. The identified risks are documented on the fly by a nominated participant – a secretary. After the session, the documented risks are distributed among the participants to be verified and supplemented. Finally, the risks are evaluated during a separate risk analysis session.

7. Quantitative Risk-Based Requirements Reasoning

The probability of satisfaction of a goal depends on the probability of occurrence of obstacles obstructing it. The severity of the consequences of an obstacle depends on the difference between the prescribed degree of satisfaction for the obstructed goals and the estimated probability of satisfaction of these goals in view of their obstruction. The quantitative risk assessment technique is model-based and anchored on an existing goal oriented framework for requirements engineering. The framework is extended with a probabilistic layer allowing behavioral goals to be characterized in terms of their estimated and required degrees of satisfaction. The specification of such goals and their obstacles has a formal semantics in terms of system behaviors, allowing probabilities to be grounded on measurable, application specific phenomena. The severity of obstacle consequences in terms of degree of goal violation is determined quantitatively and systematically by probability propagations through the obstacle and goal models. This technique was successfully applied to two non-trivial mission-critical systems for ambulance dispatching and carpooling, respectively. [24]. The study investigated the mentioned practical challenges in distributed software projects using two research efforts. The first research effort is a categorization of known challenges and resolutions related to distributed software projects, along with guidelines for how to apply these categorizations in practice. The second research effort is an in-depth investigation into practice in a distributed software project with particular emphasis on a subset of the mentioned practical challenges.

Virtual teamwork characteristics, such as geographic dispersion, electronic dependence, structural dynamism, and national diversity often hinder innovation [9]. This observation applies specifically to virtual teams developing software and implies numerous management challenges [12]; [20]. The challenges of managing distributed software projects arise because the project task is divided and distributed across several sites. The task division and distribution can make it difficult for project participants to understand the task, its purpose [20], and their own contribution to the overall task. Consideration of interfaces, subsystem influence, and workload is therefore, critical when segmenting the task in distributed software projects. An increased coupling between task segments can increase the need for inter-site communication, coordination, and integration and may thereby lead to an increased number of failures. Also, when software developers from different parts of the world collaborate, tool compatibility is a common challenge. The reason is that each site is likely to prefer different programming languages, support tools, operating systems, and development tools [21]. Selection of appropriate ICT is therefore, significant for project success.

8. Conclusion

This paper presents the issues related to the assessment of software project risk. The statistics on the failures of software projects were recalled, which constantly indicate a considerable amount of risk related to software development and the need for stronger risk management practices. The fundamental principles of risk management were outlined and the current most commonly used risk assessment practices were reviewed. The business process modeling was brought as a new opportunity in project risk assessment. The paper concluded with an overview of the current models of research in relevance to risk analysis of software projects. The remaining challenges and limitations of available approaches as well as the literature survey of related work show the research objectives to be valid and important.

REFERENCES :

[1] B. Flyvbjerg, "Eliminating Bias through Reference Class Forecasting and Good Governance", Concept Report No 17 Chapter 6, Concept-programmet, 2007.

[2] Curseu, P.L., Schalk, R., and Wessel, I. "How do virtual teams process information? A literature review and implications for management," Journal of Managerial Psychology (23:6), 2008, pp 628-652.

[3] D. Bența, M.I. Podean & C. Mircean, "On best practices for risk management in complex projects" Informatica Economica Journal, Bucharest: INFOREC Association, Vol. 15, no. 2/2011.

[4] D. Bernoulli, "Exposition of New Theory on the Measurement of Risk", Econometrica, 22, pp.23-36, 1954.

[5] D. Kahneman & A. Tversky, "Prospect theory: An analysis of decisions under risk", Econometrica, 47, pp 313-327, 1979.

[6] D. Kahneman and A. Tversky, "Intuitive Prediction: Biases and Corrective Procedures", Studies in the Management Sciences: Forecasting, 12, Amsterdam, North Holland: S. Makridakis and S. C. Wheelwright, Eds., 1979.

[8] Dibbern, J., Goles, T., Hirschheim, R., and Jayatilaka, B. "Information systems outsourcing: a survey and analysis of the literature," The DATA BASE for Advances in Information Systems (35:4), 2004, pp 6-102.

[9] Gibson, C.B., and Gibbs, J.L. "Unpacking the concept of virtuality: the effects of geographic dispersion, electronic dependence, dynamic structure, and national diversity on team innovation," Administrative Science Quarterly (51:3), 2006, pp 451-495.

[10] Gillam, C., and Oppenheim, C. "Review article: reviewing the impact of virtual teams in the information age," Journal of Information Science (32:2), 2006, pp 160-175.

[11] Hertel, G., Geister, S., and Konradt, U. "Managing Virtual teams: a review of current empirical research," Human Resource Management Review (15:1), 2005, pp 69-95.
[12] Iacovou, C.L., and Nakatsu, R. "A risk profile of offshore-outsourced development projects," Communications of the ACM (51:6), 2008, pp 89-94.
[13] Kirkman, B.L., and Mathieu, J.E. "The dimensions and antecedents of team virtuality," Journal of Management (31:5), October 1, 2005, pp 700-718.
[14] M.I. Podean, D. Benţa & C, Mircean, "Overlapping boundaries of the project time management and project risk management", SICOMAP2010, Informatica Economica Journal, Bucharest: INFOREC Association, 2010.
[15] Martins, L.L., Gilson, L.L., and Maynard, M.T. "Virtual teams: what do we know and where do we go from here?," Journal of Management (30:6), 2004, pp 805-835.
[16] Powell, A., Piccoli, G., and Ives, B. "Virtual teams: a review of current literature and directions for future research," DATA BASE for Advances in Information Systems (35:1), Win 2004, pp 6-36.
[17] PRM-PMI®, "Practice standard for Project Risk Management", Newtown Square, Pennsylvania, USA: Project Management Institute, Inc., 2009.
[18] R. M. Hogarth, "Judgement and choice: The psychology of decision", 2nd edition, Chichester, England: John Wiley & Sons, 1987.
[19] Ripeanu, M., Singh, M.P., and Vazhkudai, S.S. "Virtual organizations [guest editors' introduction]," IEEE Internet Computing (12:2), 2008, pp 10-12.
[20] Sakthivel, S. "Managing risks in offshore systems development," Communications of the ACM (50:4), 2007, pp 69-75.
[21] Sarker, S., and Sahay, S. "Implications of space and time for distributed work: an interpretive study of US-Norwegian systems development teams," European Journal of Information Systems (13:1), Mar 2004, pp 3-20.
[22] Schiller, S.Z., and Mandviwalla, M. "Virtual team research: an analysis of theory use and a framework for theory appropriation," Small Group Research (38:1), February 2007, pp 12-59.
[23] Wolf, T., Nguyen, T., and Damian, D. "Does distance still matter?," Software Process: Improvement and Practice (13:6), 2008, pp 493-510.
[24] Antoine Cailliau and Axel van Lamsweerde "A Probabilistic Framework for Goal-Oriented Risk Analysis**"**, Proc. RE'2012 : 20th IEEE International Conference on Requirements Engineering, Chicago, September 2012
[25] Estler HC, Nordio M, Furia CA, Meyer B (2013) Unifying configuration management with merge conflict detection and awareness systems. In: Dietrich J, Noble J (eds) Proceedings of the 22nd Australasian Software Engineering Conference (ASWEC), IEEE Computer Society, pp 201-210
[27] Ramasubbu N, Cataldo M, Balan RK, Herbsleb JD (2011) Configuring global software teams: a multi-company analysis of project productivity, quality, and profits. In: Proceedings of the 33rd International Conference on Software Engineering, (ICSE), ACM, pp 261{270