

Offshore-Outsourced Software Development Risk Management Model

Shareeful Islam, Siv Hilde Houmb, Daniel Mendez-Fernandez and Md. Mahbubul Alam Joarder

Institut für Informatik, Technische Universität München, Germany

Connected Objects Laboratory, Telenor R&I Service Platforms, Norway

Institut für Informatik, Technische Universität München, Germany

Institute of Information Technology, University of Dhaka, Bangladesh

islam@in.tum.de, siv-hilde.houmb@telenor.com, mendezfe@in.tum.de, joarder@univdhaka.edu

Abstract

Offshore-outsourced software development is gaining popularity because companies are continuously forced to reduce production costs while keeping sustainable competitive strength. However, this trend of software development increases projects' complexity and brings up risks to the overall project environment. Therefore, risks of offshore software development require to be managed as early as possible for a successful project. This paper considers a risk management model from a holistic perspective to manage offshore software development risk, integrated into early stages of development. The approach effectively identifies and specifies the goals of a project and the related risk factors. This is done at the basis of selected software development components within the running project. We show how to trace and control these risks already during early requirements engineering activities. The model at hand is implemented into an on-going offshore software development project to (1) identify goals and risk factors from the local context and finally (2) to determine its applicability of the approach in offshore software development projects from a vendor's perspective.

Keywords: Software development risk, goal modeling language, offshore outsourced software development, requirement engineering.

I. INTRODUCTION

Offshore-outsourced software development (O-OSD) has become a highly favored topic for companies aiming at cost savings while achieving final product delivery within estimated time schedules. Still, this type of development has several challenges due to its inherent nature. For instance, decreased degrees of communication, lack of knowledge about customers' business domains, disputes on legal issues [1, 6] may pose any potential risks to the project. A recent report [1] suggested that outsourcing magnifies existing risks and creates additional threats to

the offshore projects. These risk factors are not only given by technical issues, but also by non-technical issues. There is in general an observable tendency to over-manage the technical issues and underestimate the non-technical ones. Consequently, O-OSD has to emphasize particular goals, such as an effective co-ordination of project works between offshore customers/users with local development teams, building trust, attain security besides generic software development goals like schedule, cost and quality.

This paper evaluates a goal-driven risk management model (GSRM) that is integrated into Requirement Engineering (RE) activities in order to manage risks of O-OSD. The approach explicitly defines the relations between the goals relating to project success from offshore environment and the risk factors that obstruct the goals respecting technical as well as non-technical development components. In addition, it defines the control actions that enable the satisfaction of the goals. Therefore, GSRM assesses and manages risk that relate to the challenges of the offshore context right from the beginning of a project. We claim that this integration contributes to a reduction of errors that arise from elicited user and / or detailed system requirements. This is in particular important to the offshore environment because our result showed that requirements errors are a common problem in offshore development projects [8, 12, 13].

We performed a field study within an on-going offshore software project in Bangladesh. The field study evaluates applicability of the model and compares the identified risk factors with our previously published survey results from the same local context [8] and with the other published risk factors [6, 14] of the offshore context. The study context is from a developing country where the offshore market is rapidly expanding by significantly increasing investments in the recent years [5].

The remainder of the work is as follows. We first give in Sect. 2 an overview on risk management approaches and related survey. In Sect. 3 we introduce the basic concepts of goal-based risk management and in Sect. 4 the O-OSD specific approach. It is evaluated in Sect. 4 before giving

The work is partly supported by the German Academic Exchange Service (DAAD), Germany and the Institute of Information Technology (IIT), University of Dhaka, Bangladesh.

in Sect. 5 finally some concluding remarks.

II. RELATED WORK

There are several valuable contributions in the research area of software risk management, including models, process descriptions and techniques. Still, only few focus on the integration with RE. Boehm proposed a risk-driven spiral model in [4] including an iterative approach to manage risks in software project. Karolak proposed the Software Engineering Risk Management (SERIM) framework based on four interconnected risk trees including 81 risk factors, in turn, categorized by technology, cost and schedule [9]. Kontio proposed the Riskit methodology [10] by initially identifying stakeholder goals and risks that threaten the goals. Risks are analyzed and prioritized by deriving scenarios which is a non-trivial task when a scenario depends on more than one probabilistic element. A recent contribution that focuses on integrating risk management into RE is the one of Ansar et al [2]. It contributes by using Tropos a goal-driven approach to identify and manage risks that obstruct the goals within RE. Procaccino et al. [15] identifies seven early development factors and discusses how these contribute to the success or failure of a software project. Other directions of work empirically analyze risk factors. Iacovou et al. summarize for example in [6] the top ten risk factors for offshore-outsourced development projects. Aspray et al. consider in the ACM task force report [1] risks from both technical and non-technical issues. Tsuji et al. [14] propose questionnaires assessment schemes based on software, vendor, and project properties to quantify risks in offshore software outsourcing. The survey shows that vendor properties including communication and project management abilities affect more the result to development in comparison to software (technical) properties like requirement volatility. However, our risk management model is goal-driven and extends the basic concepts of the KAOS approach. We focus on a holistic view of technical and non-technical issues.

III. GOAL & RISK FACTORS

We analyze early software development components and project success factors from the existence literature as a background foundation to develop the model. This lays the foundation for GSRM considering a holistic view from a both technical and a non-technical. We consider technical issues in software development as those aspects that directly relate to hardware and software while non-technical issues relate to human, managerial, organizational and environmental factors. Based on a literature survey, we categorized development components according to project constraints, development

process, product, human and finally (internal & external) environment. These components are described through the essential elements that are required for software development. The elements may be described by single or by multiple factors. Therefore, elements and factors together represent the activities, the activities' results (the artefacts) and the general characteristics of the individual components. For instance, project planning and control is an element under the component project constraints that represents project schedule, budget, staffing and other project planning and management related issues. This hierarchy eases the identification of goals to be satisfied and risk factors that obstruct these goals like maintaining a realistic project schedule. Based on this hierarchical framework, we consider a conceptual view for the risk management model within GSRM, as shown by Fig. 1. Goals are derived from the development components by considering the factors relating to project success. Project stakeholders are responsible to these goals. Risk factors certainly obstruct these goals and support casual relationships to the risk event. Likelihood of risk events along with the risk impact supports to prioritize the risks. Finally, control actions are implemented to reduce the risk event and contribute for the goal satisfaction.

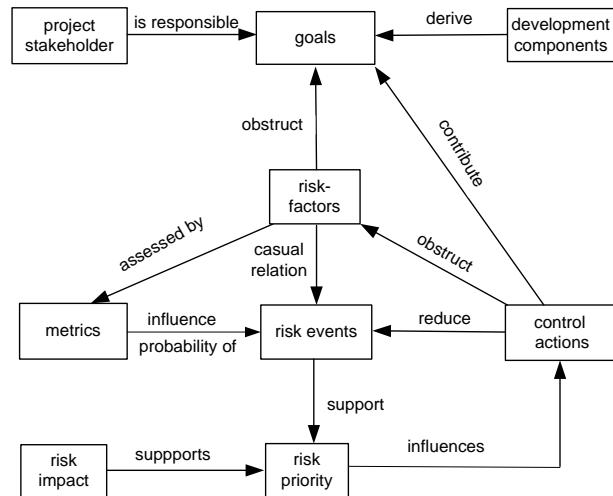


Fig. 1. Conceptual view for the risk management model

IV. O-OSD RISK MANAGEMENT MODEL

GSRM is based on existing goal modeling techniques to accommodate risk management activities. We extend the KAOS goal modeling approach [11] to support risk management activities and integrate it with RE respecting O-OSD. Goals in general provide an anchor to analyse the risks in software development. Therefore, risk management requires the identification, the analysis and the refinement of the goals and the risk factors that obstruct the goals [11]. KAOS defines an obstacle as an undesirable behavior against stakeholders' strategic

interests [11]. GSRM adopts this concept of KAOS and defines software risks as obstacles that contribute negatively to the fulfillment of specific goals. In GSRM we extend KAOS furthermore with risk assessment and treatment techniques as shown in [7]. This is done by using four layers of abstraction within the modeling structure of GSRM. Fig. 2 depicts the four different layers that are subsequently described.

Goal Layer. Goals are the objectives, constraints and expectations that have to be achieved by a software development project through the cooperation of system agents. These agents represent the development components and the project stakeholders. Therefore, the model initiates with the goals by following the development component, the elements and the factors hierarchy besides the project stakeholders' expectations. Goals can be stated at different levels of abstraction from higher level coarsely grained to lower level finely-grained goal assertions. This goal hierarchy enables developers to model all system agents, even though these often are somewhat fuzzy. GSRM follows informal temporal pattern as stated in KAOS [11] to represent each goal. The pattern structures an assertion into a prefix and a condition/property. For instance, a statement could be "reduce [erroneous requirement]", whereby the prefix "reduce" represents a goal that demands a reduction of defected requirement.

Risk Obstacle Layer. The risk obstacle layer encompasses the potential obstacles and specifies which goals they obstruct, i.e. incurred problems within the development environment. The layer allows the practitioner to directly link all types of obstacles to the goals. Same and similar risk obstacles can be relevant to more than one goal. This is important in order to consider effective treatment options. Risk factors that cross-cut several goals are in general more effective to counter since the treatment effect often then propagate also to goals that are not directly linked with the particular risk factor. In GSRM, we follow a set of questionnaires based on the early development components and brainstorming techniques to identify these risk obstacles. All identified obstacles are then analyzed further within the assessment layer.

Assessment Layer. The assessment layer mainly analyses the risk events that influence single or multiple risk factors. Each risk event is characterized by two properties: *likelihood* and *impact*. Likelihood specifies the rate of occurrences of a risk event and is modeled as a property of the risk event itself. The impact is a measure over the negative outcome of the risk event occurrence. All risk

events and goal relationships are modeled by adding an obstruction link from the risk event to the specific goal that it obstructs, and in cases where several goals are affected, an obstruction link is established between the risk event and each of these goals. GSRM supports the use of risk metric values to identify the likelihood of the risk event by estimating the casual relationship of the related risk factors. GSRM uses for this purpose the Bayesian interpretation and in particular Bayes theorem to estimate the risk events based on their casual relationship with the risk factors. Risk events likelihood and impact in particular with high-highs, high-mediums and medium-mediums while ignoring low-lows give us certain beliefs about the dissatisfaction (DSAT) and the satisfaction (SAT) of the goal fulfillment. Finally, risks are prioritized based on their likelihood and impact.

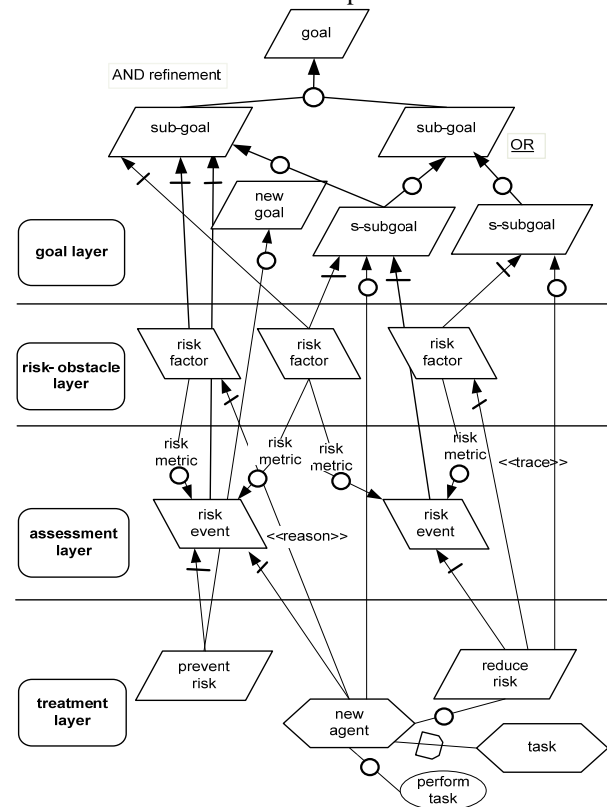


Fig. 2. GSRM framework

Treatment Layer. Once the goals, risk factors and risk events are identified and analyzed, it is crucial to identify and then implement suitable and cost effective countermeasures. Therefore, the aim of the treatment layer is to gain control of the risks as early as possible and preferable during RE activities by assigning appropriate countermeasures. To visualize the relationship between treatment, risk obstacle and goal, we establish a contribution link from chosen control actions to the goal by specifying the ability of the treatment to support the goal and by reducing the effect or likelihood of removing

the associated risk factors. Additionally it is also necessary to analyze the cost-benefits before implementing a suitable control action.

V. FIELD STUDY

We have conducted a field study to evaluate the model through an on-going software development project by a vendor company in Bangladesh. A customer that is located in Australia triggers the project. Due to space limitation we present a brief overview of the study and its results. We formulated two research questions and derived the study design according to these questions.

(RQ1) “How valid are the identified risk factors?”

We analyze the validity of the identified risk factors by comparing the identified risk factors from the field study with previously published ones [6, 8, 14]. Although the identified risk factors are suitable, the process can still be inefficient. For this reason we analyze in a second step the applicability of GSRM.

(RQ2) “Is GSRM applicable to offshore software project?”

We analyze the GSRM according to its applicability as part of structured interviews respecting particular advantages and limitations (lessons learned) in an active software development project.

A. Study Design

Our case study is conducted in two different phases. The first one addresses RQ 1, the second RQ 2.

We have performed the approach in collaboration with a Bangladeshi vendor, implementing the approach to an ongoing project. In the first phase, we conducted a brainstorming session to identify the goals of the early development components as expectation for the project to success besides obtaining background information of the company and the selected practitioners involved in the project. We provided a short tutorial for understanding the basic terminology and features of risk management in particular about GSRM. Finally we derived by following the approach the risk factors and evaluated them for their validity. In the second phase, a structured interview was conducted to the individual participants of the project in order to evaluate the applicability. The interview template contained 108 questions including 80 close questions and remaining open questions. The questionnaires were prepared by reviewing checklists, similar questionnaires format from literature and by considering the goals and sub-goals for the project’s success. Each close question has three possible answers. Open questions obtain the information relating to the on-going project and existing practice of risk management throughout their projects and about GSRM. A total of nine interviewers participated in the second phase.

B. Results

The project concerns the development of a system to support the core business processes of a sales department. It contains two modules “account” and “reporting” with several features such as bar code readable sales system, inventory and purchase. The project size was nine man-months with a total duration of eleven months. The vendor only listed generic risks once in the overall development period mostly focusing on time and budget. No other risk factors such as human, environment, technology related issues are considered. As a preparation for applying GSRM, we start to identify the main goals from the project context. After approving the goals, participants were interviewed to identify the state of the development components, elements and factors so that risks relating to these components are identified.

B.1 Research Question 1

We have reviewed and agreed a set of goals with the project participants by following the development components, elements and factors concept of GSRM. The identified goals are rather high level but their refinements are not included in the paper due to confidentiality reason. Most of the goals require a prefix of different types such as maintain, attain, reduce and improve, depending on the nature of the goal. For instance:

- When a goal concerns maintenance, it implies that certain conditions of the development elements or factors should always be kept at the same level throughout the software development.
- When a goal concerns achievement it implies that certain conditions of the development elements or factors should be achieved beyond the current state.
- When a goal concerns reduction, it implies that a certain state of the elements or factors should be reduced, respectively minimized.
- When a goal concerns improvement, it implies that certain conditions, states of the elements or factors should be increased.

Some of the agreed high level goals under the development components are:

- Project constraints
 - Maintain [actual estimated budget]
 - Maintain [actual estimated schedule]
 - Reduce [project complexity]
- Process
 - Improve [development activities]
 - Improve [formal risk management practice]
- Product
 - Attain [clear business needs]
 - Reduce [error from requirements]
 - Improve [product quality]
- Human
 - Improve [competency of team members]

Improve [communication & coordination]
 Improve [proper management direction & support]
 Improve [customer/user involvement]

- Internal & external environment
 - Improve [adequate development facilities]
 - Maintain [implementation of policies & procedure]

We have identified major risk factors based on the responses to the interview. Interview questions mainly focus on the identification of the state of the development components that obstruct the goals. A total of thirty eight risk factors are identified that directly obstructs the goals while creating problems to the development. Table I shows the major risk obstacles that influence several risk event for the project such as erroneous requirements, budget and schedule over-runs, poor communication, incompetence practitioner, all having likelihood and impact between high and medium for the risk event.

Table I Major risk factors

Risk-obstacles
<ul style="list-style-type: none"> • Requirements faults: incorrect, unstable, incomplete and ambiguous/underspecified • passive customer/user involvement • lack of project specific domain knowledge of the software practitioners • unstable organization structure • hidden factors(variation of bank fee, strike, interrupt power and internet supply) for schedule & budget overruns • incomplete requirement specification document • ineffective RE process to elicit, analyze , validate, and document requirements • employee absence • inadequate support to handle change

Most of the identified risk factors of table I are from the product and human components and mainly influence the non-technical issues. In order to answer finally RQ1 we made a cross study of our finding with the published risk factors from our previously study [8] as well as with the other published risk factors from the offshore environment [6, 14]. This comparison shows similarities with our previously published risk factors in [8]. Several common risk factors from the local context are requirements faults, lack of domain knowledge, employee absence, hidden factors that influence for schedule or budget overruns, erroneous requirements and ineffective communication. We have noticed that some risk factors take influence on other risk factors as well as several risk events to occur. We treated these as important risk factors (see Tab. I) for the project that required evaluation and control actions as early as possible. Requirements problems are the most important ones from the local context. This is due to the passive involvement of the customer/ user, lack of practitioner project specific domain knowledge or

ineffective RE process. An unstable organizational structure and incomplete requirement specifications are newly identified risk factors for this project context as an outcome of the study. In comparison to other published risk factors by Iacovou et al. [6], we observed both similarities and dissimilarities. For instance, inadequate user involvements, poor change control and lack of business know-how to by offshore team from Iacovou et al are also identified by our case study. But the participants confirmed that they faced no problems arising from the lack of top management commitment, cross-cultural and time-zone difference, technical knowledge difference, threats to the security of information resources, and project contract as stated in [6]. Requirements volatility is the top prioritized risk factor by our field study which shows not as much severe by Tsuji et al. in [14]. Therefore, our result regarding risk factors in offshore environment concludes that local environmental context certainly plays an important role for the success of offshore project. Technology differences between the vendor and the offshore client, differences in the time zone, cultural difference, security problems and finally disputes in contractual agreement are no longer important problems in the local context. This study also increases our confidence about the validity of the recorded result by comparing it with other published literature.

B.2 Research Question 2

Due to tight schedule pressure in the project we were not able to perform a comprehensive risk management approach under GSRM to the on-going project. In particular, this affected the probability estimation of the risk event occurrence. The participants were more interested to consider the control actions in order to mitigate risk factors rather than continuing detailed risk analysis through assessment layer. By analysing the participant's response to the open questions we observed that they are more concern about simple and straightforward techniques for software risk management. They remarked GSRM as quite straight forward goal – driven process for risk management. The underlying principles of GSRM are relatively easy to communicate in general to the software practitioners and in particular to the project manager and requirements engineers. The participants appreciate the combination of questionnaires and brainstorming sessions concerning the identification of risk obstacles. The effort of developing the risk artefacts such as detailed risk factors, goal-risk-treatment models and risk treatment plans require some extra resource with a minimum level of domain knowledge. This is especially important within tight time schedules. Based on our observation several lessons are learned from the study. In conclusion, they are:

- Risk management should be considered from early stages of development accompanying the software development activities.
- The identification and elaboration of goals certainly ease the risk management process as well as the communication with the project practitioners.
- Risk identification through questionnaires and brainstorming session allow including experiences along with checklists for systematically identifying software development risks.
- Many risk factors of offshore software development are influenced by local environmental context. Traditional offshore software risks, such as time difference, mismatch in technology or legal disputes are not always important to our context.
- An organization should have a repository for reusing risks across several projects. This is important, because project members are always under constant time pressure and without having explicit list of risk it is hard for them to spend sufficient time. Besides this aspect it is not economically reasonable to analyze risks from the beginning for each new project.

VI. CONCLUSIONS & FUTURE DIRECTION

Risk is a fuzzy concept term and it can mean different things to many people. Therefore, a systematic simplified approach is required to handle risk management in hectic on-going project environments. We believe that GSRM contributes within this context by structuring goals and risk factors and choosing suitable control actions. The identified goals and risk factors from our study and previous study result would be effective for developing the offshore market in Bangladesh.

However, we neither fully implemented the GSRM to the running project nor did we perform detailed elaboration of the goals. Therefore, a detailed analysis of GSRM in terms of its usefulness is not fully completed through the study. Still, we have learned some lessons. We plan to use the information gained from the field study to improve the GSRM in particular the tasks and methods used for the risk management activities. We also did not cover the cost/benefit analysis of the model during the evaluation since it is hard to quantitatively analyze costs and benefits within the risk management context. Hence, we are planning to conduct the field study to evaluate the model further in terms of its usefulness and feasibility regarding integration into RE.

REFERENCES

- [1] Astray, W., Maydays, F., and Verdi, M.Y. Globalization and of shoring of Software: A Report of the ACM Job Migration Task Force. ACM, NY, 2006.
- [2] Ansar, Y. and Georgina, P. Modeling Risk and Identifying Countermeasure in Organizations. In: Proceedings of the 1st International Workshop on Critical Information Infrastructures Security, pp. 55–66. Springer, Heidelberg, 2006.
- [3] Boehm, B.W. Software Engineering Economics. Englewood Cliffs, N.J., 1981.
- [4] Boehm, B.W. Software Risk Management: Principles and Practices, Piscataway: IEEE Software, v. 8, p. 32–41, Jan. 1991.
- [5] Bangladesh Associate of Software & Information Services (BASIS), <http://www.basis.org.bd>.
- [6] Iacovou, C. L. and Nakatsu, R. A Risk Profile of Offshore-outsourced development projects. Communication of ACM 51, June 2008.
- [7] Islam, S. 2009. Software development risk management model: a goal driven approach. In Proceedings of the Doctoral Symposium for ESEC/FSE on Doctoral Symposium (Amsterdam, The Netherlands, 2009). ACM, New York, NY, 5-8. DOI=<http://doi.acm.org/10.1145/1595782.1595785>
- [8] Islam, S. Joarder, M. A., Houmb, S.H. Goal and Risk Factors in Offshore Outsourced Software Development from Vendor's Viewpoint, pp. 347-352, In proceedings of the Fourth IEEE International Conference on Global Software Engineering, 2009, Ireland. <http://dx.doi.org/10.1109/ICGSE.2009.54>
- [9] Karolak, D. Software Engineering Risk Management, IEEE Computer Society Press, Los Alamitos, CA, USA, 1996.
- [10] Kontio, J. The Riskit Method for Software Risk Management. Version 1.00, Technical Report, CS-TR-3782 /UMIACS-TR-97-38, University of Maryland, Computer Science, College Park, USA, 1997.
- [11] Lamsweerde, Van A. Requirements Engineering: From System Goals to UML Models to Software Specifications, Wiley, January 2009.
- [12] Linberg, R., Software Developer Perceptions about Software Project Failure: A Case Study, the Journal of Systems and Software, Vol. 49, Issue 2/3, 1999.
- [13] McConnell, S. Rapid Development. Microsoft Press, 1996.
- [14] Tsuji, H., Sakurai, A., Yoshida, K., Tiwana, A. and Bush, A. Questionnaire-Based Risk Assessment Scheme for Japanese Offshore Software Outsourcing, B. Meyer and M. Joseph (Eds.): SEAFOOD 2007, LNCS 4716, pp. 114–127, 2007.
- [15] Procaccino, J. D. and Verner, J. M. Case Study: Factors for Early prediction of software development success; Information and Software Technology; Vol. 44, 2002.