Towards a Framework for Offshore Outsource Software Development Risk Management Model

Shareeful Islam
Institut für Informatik, Technische Universität München, Germany
Email: islam@in.tum.de
Siv Hilde Houmb
SecureNOK Ltd., Norway
Email: sivhoumb@securenok.com

Abstract— **Due to high demands on cost savings in software development projects, offshore outsource software development is becoming increasingly popular. Offshore outsourcing takes advantages of large labor pool in low-wage countries, round-the-clock development, and easy access to additional resources, in addition to the development cost savings. However, there are risks associated to this trend, as it brings new challenges into the development process, where some may even jeopardise the cost savings. The geographical and cultural distance is one part of the problem, inexperienced developers and lack of communication and a common cultural basis are other challenges. We therefore advocate the use of risk management (RM) to control risks in offshore outsource development. Here we present a RM framework tailored for such development contexts. The model; Goal-driven Software Development Risk Management modelling (GSRM) framework, assesses and manages risk during the early development phases, where risks can easier be tackled at a reasonable cost. The framework is comprised of four layers that together identify and link project goals, risks and treatments together in a goal-risk causal relationship model. The goal model makes use of an extended subset of the KAOS goal modelling language. We report on a study project focusing on the efficiency of the GSRM process model.**

Index Terms—**software development risks modelling, goal modelling language, offshore outsourced software development, requirements engineering.**

I. Introduction

Offshore Outsource Software Development (O-OSD) is an attractive business strategy in modern software development enabling the development of high quality software under low production cost by moving development activities to low wage countries. This trend has resulted in many companies from the U.S., Australia and Europe to outsource software development projects to offshore countries over the last decade. However, this one-sided focus on cost saving has during course revealed that many stakeholders ignore the investments necessary to mature their development processes to the global development environment thus resulting in an undesirable increase in coordination efforts [15,21]. O-OSD also introduces new challenges to the development process such as variable technical skills among the users and practitioners, lack of customer's business process expertise and control, and differences in perceptions which may lead to misunderstanding and inadequate project scheduling. Research suggests that half of the companies endeavouring in outsourcing have failed to address these challenges and have therefore not been able to take full advantage of the offshore outsourcing potential [6]. Dun & Bradstreet [4] found in their survey that 50% of outsourcing relationships worldwide fail within five years due to poor planning. Thus risks relating to Global Software Development (GSD) may even reduce the ability of a project to succeed and end up ewith an inferior product, cost overruns, disputes on intellectual property, eventually resulting in failure. It is necessary to understand the risks associated with offshore outsourcing to make informed decisions in particular about the important project goals like project scope, potential business benefits, budget and schedule [13].

This paper employs goal-based software development risk management modelling (GSRM) framework to assess and manage risk of O-OSD projects, focusing on the Requirements Engineering (RE) phase [19, 20]. The model explicitly establishes relations between goals necessary to achieve a successful project and risks that may obstruct these goals. This allows us to select the appropriate control actions to prevent the risks and by that increase the ability of fulfilling the project goals, including the ultimate goal which is a successful project. The framework extends the KAOS (Keep All Objective Satisfied) goal modelling language [1] to accommodate the risk management activities through a separate risk management process model.

In our earlier research, we described the conceptual framework of GSRM [19, 20]. This work extends the GRSM framework with an explicit process model consisting of activities, tasks and artefacts and its integration into the requirements engineering phase. The process model is demonstrated at the hand of a case study implementing GSRM into a running offshore software development project. The purpose of this empirical investigation was to investigate whether GSRM indeed contributed in controlling development risks and whether the process model made it easier to employ GSRM in practice. The results showed that GSRM activities are well fitted with requirements engineering activities and systematically manage development and project risk and attain project goals.

doi:10.4304/jsw.6.1.38-47

The remainder of the article is structured as following: Section II gives on overview of the GSRM framework, including a detailed description of the GSRM process model. Section III presents a walk-through of the GSRM process at the hand of the case study, while Section IV discusses lessons learnt and the validity of the observations made in the case study. Section V places GSRM into a broader context (related work) and Section VI summarises the main contribution of the article.

II. Overview of the Framework

According to Boehm [2] and McConnell [18] effective and efficient software development and ultimately project success can be framed in terms of people, process, product and technology. Procaccino et al. [11] further categorised seven factors which contribute to the success and failure of software systems and these are: management, customers and users, requirements, estimations and scheduling, project manager, and development process and development personnel. GSRM is based on the above mentioned works and categorises software development into five dimensions: (i) project execution, (ii) process, (iii) product, (iv) human and (v) environment (internal and external). These dimensions are the development components. Development components are fundamental multidimensional issues for any software development project. However individual component rather provide an abstract view which generally is comprised of single or multiple elements. Elements are the essential parts that describe a component. The elements may further be characterised by single or multiple factors, if necessary also refined into sub factors. Factors are the lowest level refinement of the development component and represent a concrete aspect of the development. Elements and factors together represent the components by following development activities, project execution issues, product quality factors, human and environmental issues and the resulting artefacts. GSRM defines this as a component-element-factor hierarchy. E.g., project execution component are described by elements such as planning & control, scope and tool support, where planning & control further are refined into factors such as budget, schedule & milestones, monitor, complexity and change management. Generally the elements are intertwined, interdependent and contribute to attain single or multiple development goals. GSRM focuses on the expectations, objectives and constraints of the development components that directly and indirectly relate to the project success. This hierarchy includes both technical and non-technical development issues which facilitate to consider a holistic view on software development risk management from the early development.

A. Layer Based Abstraction

GSRM introduces a layer based abstraction to assist in software development risk management, as shown in Figure 1. We use the same notation for goals and obstacle as that of KAOS [1]. The advantage of layer based modelling is that it allows for a diversity of techniques and methods to be used across the layers provided that the outcome can be seemingly transferred between the layers. In the following we provide a brief overview of the four layers of the GSRM model.

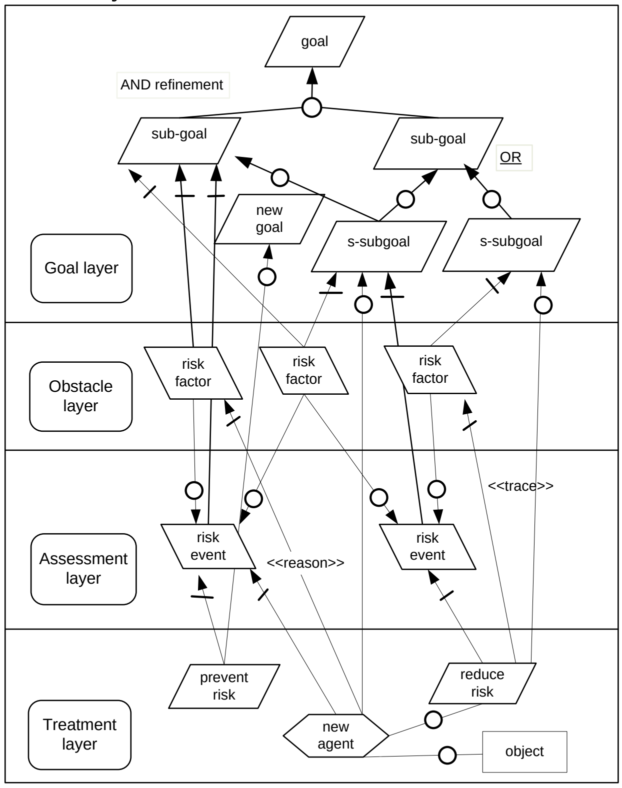

Figure 1. Overview of the GSRM Framework

Goal Layer. The concept goal is adopted from the KAOS goal modelling framework [1] and focuses on the objectives, expectations and constraints of development components. These goals are mapped with the factors contributing to project success. These goals are important as they describe what needs to be done for a project to be successful and the agents responsible to attain the goals. Agents can be development components, tools or humans who directly or indirectly are involved in the development activities. Goals can be stated at different levels of abstraction from higher level coarsely grained to lower level finely-grained goal assertions. The more the goals are refined the easier it is to identify and analyse the risk factors that obstruct the goals.

Obstacle Layer. Obstacles are events or circumstances that reduce the ability to fulfil the goals. We treat risk factors as obstacles that directly or indirectly lead to a goal negation and that create problems during the development. The layer allows the practitioner to directly link all types of obstacles to the relevant goals creating a goal-obstacle hierarchy which is later used to identify suitable treatments. Note that an obstacle can be relevant to more than one goal. Thus risk factors that cross-cut several goals are in general more effective to counter, as the treatment effect often also propagates to goals that are not directly linked to the particular risk factor.

Assessment Layer. This layer analyses the consequence of single or multiple risk factors and the extent to which they affect goal negation. It mainly quantifies the risks so that high prioritised risks can get immediate attention. It also allows for the refinement of risk factors into risk event by establishing the obstruction link from the risk factors and events to the goals.

Treatment Layer. This final layer focuses on the control actions to counter the risks so that goals can be properly attained. It also monitors the effectiveness of the control actions and identifies any new risks throughout the development. The main aim of the treatment layer is to gain control of the risks as early as possible and preferable during the requirements engineering activities by assigning appropriate countermeasures. To visualise the relationship between treatments, obstacles and goals we establish a contribution link from the chosen control actions to the affected goals and specify the ability of the treatment to support the goal and by that reduce the effect or likelihood of associated risk factors.

B. Process Model

As stated earlier GSRM is integrated into the requirements engineering phase which is accounted for in the GRSM process model. The process model integrates risk management activities directly into the requirements engineering activities. This integration allows GSRM to be an inherent component of software development enabling it to directly and efficiently support the development project decisions and ensuring that the focus is strictly kept on satisfaction the main project goals. This means that the GSRM process must systematically assess and manage software development risks and produces artefacts that provide accurate information about risks associated to the development activities. Thus GSRM support in making informed decision about the project and its goals from the early development.

Figure 2 shows the generic process model for GSRM including the activities, tasks and roles. The activities describe the aspects concerned with the risk specification artefact. Individual activity is comprised of task which in turn produces output artefacts based on the inputs. For producing the output, each task includes steps that define a concrete method for constructing specific output. A role gives an abstract description of the responsibilities of the development team or other stakeholders that directly or indirectly contribute to the artefacts. A role can take the responsibility for a specific set of artefacts and performs activities in order to produce or modify these artefacts. Individual activities should define operational character such as the use of techniques. An example of such are techniques used to identify risk factors.

Artefacts are the output products from the activities and tasks and include specification of domain specific concepts in terms of precisely defining the relevant elements and how to use these for specific description techniques [16]. The concepts define elements in terms of the artefact's attributes and their dependencies with other elements from other concepts. E.g., the artefact type risk specification consists of content items like goal-risk model and goal details. Furthermore the goal-risk model includes concept of goals and risks, where goals are comprised of the attributes: id, name and description. The artefact-oriented concept allows us to map the dependency between the risk and requirement specification artefacts. E.g., elicited system requirements are reviewed by the GSRM to identify and assess requirements risks and risk level helps to prioritise the requirements. The activities and tasks also ease the integration as similar techniques can be used for both requirements engineering and GSRM activities, such as workshop and brainstorming sessions which are effective for both types of activities.

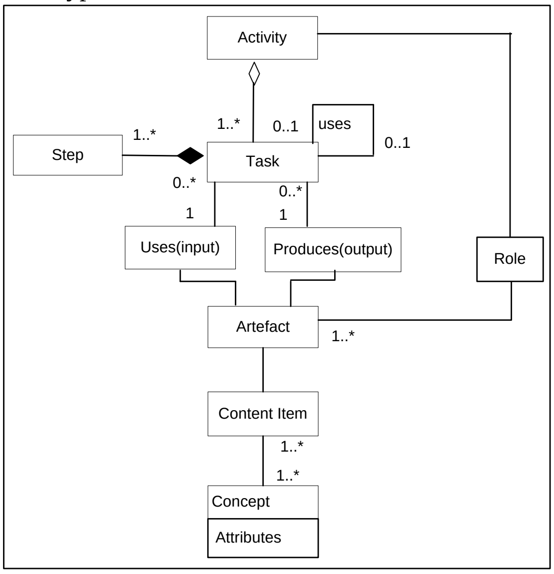

Figure 2. Process model for GSRM

C. GSRM Activities

It is commonly agreed among risk management standards and relevant research studies that to be successful risk management must be run as a continuous process involving repeated risk assessment and project specific risk mitigation activities throughout the system life cycle [1,2,10,16]. The standard ISO/IEC 16085:2006 recommends agreeing on and approving a risk management plan before initiating any risk management activities. GSRM follows these guidelines in its process model, i.e., develop and agree on a risk management plan as well as define the scope of the analysis before executing any risk management activities. The activities of the GSRM process are performed sequentially starting with the initialise risk management activities such as defining the risk management scope and plan and ending with the treat and monitor activities, specifically for the initial iteration of GSRM. Figure 3 gives an overview of the activities and the information flow of the goal-based risk management process. The process contains five activities that each defines a major area of concern for GSRM. A short overview of the activities is given in the following.

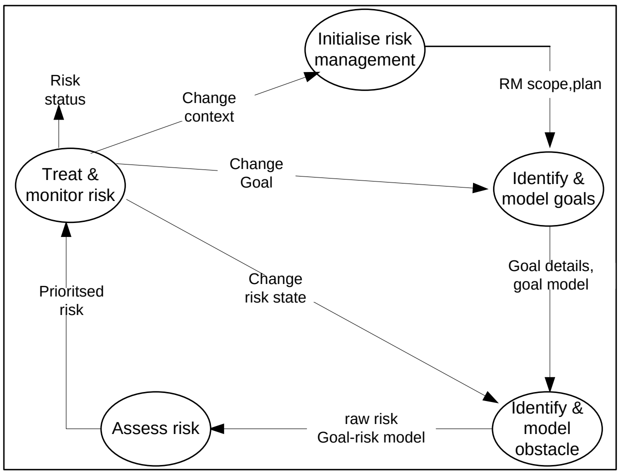

Figure 3. GSRM activities

Initialise goal-driven risk management. The first activity of the GSRM process initiates the implementation of the model into the development and aligns it with the requirements engineering activities. Plan risk management is the only task that focuses on the initial project artefacts such as business goals, project authorisation document including project plan, budget and schedule related information which are used to specify the risk management scope and to assign responsibility, authority and schedule for the risk management activities. The scope shall also define the boundary and limitations of risks to the project. Furthermore in cases where some of the information needs to be updated, such as change to the business process, goals, main project stakeholder (sponsor, management, and development team members), project scope, risk status and organisation structure, the activity should be repeated. A complete risk management plan is the main artefact produce from this activity. This plan needs to be agreed upon and approved by the main project stakeholders and properly communicated to the development team members.

Identify and model goals. Once the risk management plan is approved the GSRM process moves to the next activity which is to understand the main goals of the project. This is done by examining the development components and by mapping these to the project success indicators. It consists of two different tasks: identify and categorise the goals and construct the goal model. The identified goals are if necessary refined or revised so that they reflect the stakeholder expectation, project success factors, project scope, business goals and environmental expectations. The goals need to be well understood, modelled and documented through the artefacts. The goal model connects the higher level goals to the lower level goals by means of refinement links. The more the goals are elaborated the easier it is to identify obstacles. The component-element-factor hierarchy eases the identification and categorisation of the goals. We follow several types of goals relevant in software development domain such as information, satisfaction, maintain, improve, reduce and product quality factors goal [1]. The activity produces the goal details and constructs the goal model.

Identify & model obstacles. This activity identifies the initial raw list of risk factors capable to obstruct the development goals. This activity includes two tasks: identify and categorise the obstacles and construct the goal-risk model. Component-element-factor hierarchy is used to identify and categorise the risk factors. The initial focus is to identify as many obstacles as possible so that their consequence to the goal negation can be properly and completely analysed. Obstacles are various events or circumstances which can result in goal negation for the software development project. Several techniques can be employed for identifying obstacles and the main aim is on how a goal can be prevented from being fulfilled. Initially we follow a questionnaire consisting of 93 close questions to identify the state of the development components based on the project context. The questionnaire is arranged by following the development component hierarchy and is constructed based on the result of our investigation of literature discussing risk factor and factors related to project success in offshore outsourced development. Brainstorming sessions among the development team and project stakeholders is also effective for both obstacle and goal identification. This activity constructs the goal-risk model and documents all the identified risk factors.

Assess risks. This activity aims to prioritise the risk by quantifying the risk level of each identified obstacle. Risk assessment in software engineering is always challenging because factors are inherently fuzzy, subjective and hard to fit into a single agreed quantitative value. Furthermore historical data are rarely available in adequate volume to make statistically reliable assumptions on risk events. Therefore we rely on subjective probability estimation based on a prior experience and other relevant observations and subjective expert judgments (belief). GSRM uses a causal relationship model to estimate the risk level which are derived by analysing the identified risk factors and how these relates to the risk event and their consequences on the goals by following the Bayesian Belief Network (BBN) methodology [8]. This activity consists of two different tasks: estimate the risk level and prioritise the risks. The risk level estimation further includes two steps: construct the causal relationship model and assess likelihood and impact of the risk events. Risk factors are considered as target nodes and risk event as observable variables. We use the same qualitative scale (i.e., three different levels) for estimating likelihood, impact and priority. This to simplify the risk estimation and prioritising process

Once the risk factor value is identified by observing the state of the development components the risk event likelihood is determined by following the causal relationship from the factors to the risk events. This means that the factors are considered as causes or circumstances affecting the occurrence of risk events. However the risk event likelihood value also depends on individual observation in particular when factors are not adequately estimated. The next task is to determine the consequence that the risk events have on goal negation. Project context and goals are important when evaluating consequence. Factors that directly obstruct the goals but that are beyond the control of the project manager such as numerous change requests issued by the user may pose severe impact on the development. Risk event likelihood and consequence determine the risk priority. Normally one will focus on the high and medium ranked risks at the early development stages. At the end of this activity detailed risk artefacts are described and the goal-risk model is fully constructed including the refinement links from the risk factors to the risk events.

Treat & monitor risk. The final activity focuses on controlling the risks as early as possible and monitors the effectiveness of the control action throughout the development. Risk monitoring is concerns with the status of risks once the selected control actions are implemented and identifies any new risk throughout the development. This activity should be performed continually and throughout the development and is comprised of four main tasks: identify the possible countermeasure, select the most potential ones, assign the agent responsibility and monitor risk throughout the development. Initially possible countermeasures are identified by focusing on risk control action strategies such as prevention, reduction and avoidance. Once the countermeasures are identified the most promising ones are selected for implementation. We use risk factor review techniques to evaluate the alternative countermeasures and to evaluate their ability to control the factors causally related to the risk event. The goal is to control as many factors as possible. Furthermore project context such as project scope, budget, availability of resource and risk management scope are important parameters to consider when selecting countermeasures. Risk treatment actions generate single or multiple task lists and define the agents responsible to perform the task and by that prevent goal negation. Final task of the activity is to monitor the risk status and to review the effectiveness of the control actions to ensure that the selected controls are indeed effective and appropriate for future risk mitigation. Goals and risks evolve over time and new risks may be discovered during the course of the development making the risk monitoring an important and continuous activity of software development projects.

III. Case Study

The GSRM framework has been tested in a survey [19] and a case study [20]. The survey was mainly aimed at identifying the feasibility of introducing a goal-driven approach for risk management. The case study partially implemented the GSRM into a medium size software development project to understand the aspects involved in integrating risk management into requirements engineering.

To evaluate the efficiency and applicability of the whole GSRM process in practice we carried out an empirical investigation; a case study. The case used for the study was a running offshore software development project by a vendor company in Bangladesh for its offshore client located in Australia. We implement the full set of GSRM activities in the project and observed how the results of GSRM activities help to manage the project. In our previous case study GSRM was not employed in the whole project scope due to the framework still being in a prototype version, strict time constraints and lack of involvement from practitioners to carry out the study. These factors in general make it hard to carry out a full-scale case study in an industrial setting. Another aspect of investigating research results in practice is that it requires several case studies to conclude on its scalability and efficiency. Several coincidences may affect the case study results such as exceptionally well or ill functioning development teams. In this case study we combined the case study method with action research [14]. This allowed us on the one hand to guide the development team for managing risks and to attain goals during the development and on the other hand to identify ways to improve the GSRM process as the framework is still under development. This combination also enables us to take a more active and guiding part in the risk management activities such that the case study results would not suffer from lack of knowledge and experience with risk management among the practitioners. A brief overview of the case study testing the GSRM process and main outcomes of the study are presented in the following. Note that we only include the non-confidential part of the study.

Study goal

The main aim of the case study was to investigate the efficiency and applicability of the GSRM process in practice. To elaborate on these issues we focused on the following two objectives:

- Implement formal risk management practice into offshore software project during requirements engineering (**O1**).
- Characterise the usefulness of the GSRM process in terms of advantage and limitations of performing risk management activities using GSRM (**O2**).

Study Context

The project concerns the development of a business information system to support the client’s core sales business processes. The project focused on the two modules: account and reporting, which both are comprised of a number of features such as bar code readable sales system, inventory and purchase. The existing software on the client side that the project aimed at extending had modules supporting item management and sales. The tasks for the development team were therefore to extend this existing software with accounting and reporting features. The challenging part of the project was transformation of old data and data format into the new modules, which was built on a new platform, and to integrate this new platform with existing hardware. The project size was estimated to be approximately nine man months with a total duration of eleven months.

Study Design

The project manager initially decided not to follow any formal risk management practice in the project and did as many others rather prefer an informal and more ad-hoc approach. The development team members also had lack of motivation regarding performing a formal risk management as part of the project. This situation is not rare in software project. Ropponen et al [12] found that 75% of project managers did not follow any detailed risk management approach and do not even clearly understand the concept of software risk. Therefore initially it was challenging to include a formal risk management practice in the project and to convince the project manager to include GSRM activities into the project. This challenge was solved after a number of discussions with the project manager by one of the co-authors (former part time employee of the company). However before the project manager finally gave his approval, we managed to convince the companymanagement to include risk management activities by arguing forthe inherent challenge involved in the project. In practice we ended up combining a case study approach with action research to ensure the quality of the risk management part of the project and to help integrate GSRM activities into the requirements engineering phase under agreement with the company management and the project manager. This was done by introducing two students of the co-author as part of the risk management team in the project. Initially a kick-off workshop was carried out by the student members giving the team an overview of GSRM and a plan on how to employ it in the project. To ensure effective data gathering both for the project and for our study context we used a series of interviews and brainstorming sessions. We also examined the project documents to get as complete data set as possible for the risk management sessions. We used close question interviews (interview guided by a questionnaire) to identify the risk factors and a feedback session with 25 open questions to characterise the usefulness in practice of the GSRM process.

A. Implement GSRM activities into the running project

Initialise goal-driven risk management. Initialise goal-driven risk management. The project manager emphasised on controlling risks related to the customer end as the main scope of GSRM. Risk management team was comprised of the project manager, the student members and two development team members. The project manager agreed to include GSRM into the late phase of requirements engineering. Initially there were two brainstorming sessions scheduled and interviews were planned for the development team members and five members from the client. The student members were mainly assigned the task to perform the interviews and to analyse the project documents. The project manager took the role as the risk manager and a feedback session was at this point already scheduled to review GSRM.

Identify and model goals. Practitioners had a general idea about goal oriented requirements engineering prior to start up the GSRM activities. This eased to identify and agree goals among the risk management team related to project execution, requirements, customer/user factors, customer business process and the existing software. Project documents were also reviewed for this purpose and the initially identified goals were mapped to the project success indicators. However detailed refinements of the goals were not done as the project manager were more interested in prioritising the goals as this would ensure that high prioritised goals could get immediate attention. The goal Success project was agreed as the top goal. In addition goals related to a clear business process and effective collaboration with the customer were also considered as important for the project context. The project manager strongly believed that an effective collaboration was the key to identify details about the existing software at the client.

Identify and model obstacles. In practise interviews were conducted with the practitioners of the development, risk management team and two from the clients rather than the five according to the plan. The close questions were arranged by following the earlier identified component-element-factor hierarchy and were used to identify states of the development components based on the project context. The interview with the two client users were carried out by conference call. The responses from all the interviews were documented and signed by the participants except the client representatives. The risk management team then identified the risk factors from the interview response and by reviewing the project artefacts: project scope, requirements and customer related factors. A brainstorming session was used for this purpose. A large number of risk factors were initially identified from the project context but the project manager was not interested to consider all of these, rather only the risk factors related to the important goals. Risk factors were documented and the goal-risk model was constructed.

Assess risk. The same session of the previous activity was used for assessing the risk and to produce a list off pre-prioritised risks. Note that that the causal relationship models were considered only for the three main goals. Work was then undertaken to link the risk factors with the goals via the risk events and to assess the extent to which the factors may contribute to goal negation. Note that some of the factors were linked to more than one risk event. These factors were important for determining how to control risks as treatments were employed on the factors rather than the events and goals, because they are the root causes. The relationship between the risk events, i.e., how the events affect the goals were mainly considered based on assumptions and experience of the project manager and the rest of the risk management team. Assessment was done according to the qualitative scale: {low, medium, high}.

Treat and monitor risks. The project manager decided to concentrate on preventing or reducing the high and medium level risks as early as possible, which is a reasonable approach. However due to the budget and schedule constraints control actions were restricted by the resources already assigned to the project. At this point there was a high chance that important risk factors could be overlooked if the project did not decide to start following a formal risk management process. The uncertainties already uncovered and the situation in the development project then led the project manager to reconsidering his earlier decision on not following a formal risk management process. Therefore the risk information, the goal-risk model and the identified control information were at this point communicated to the company management. The goal-risk model made it easy to communicate with the management as well as with the customers, which further increased the trust of the project manager in GSRM and in risk management. Some risks and control actions were also agreed with the customer during the discussion of the goal-risk model and this led to the core stakeholders approving the suggested control actions and enabled an immediate start of the risk mitigation activities. The project manager also scheduled three risk monitoring meetings once the control actions were implemented. However, at this stage in the project the student members left the risk management team and the project manager took the overall responsibility of the risk monitor activity. Before leaving the project the student members conducted an individual interview with the risk management team members as well as with three other practitioners to obtain feedback about GSRM. The open interviews were structured around 25 questions of which five of this are listed below.

Feedback on GSRM based on your experience

1) Is goal-oriented approach for risk management useful for software development project?
2) Is it feasible to integrate risk management activities in requirements engineering?
3) Do you think the process used for GSRM is appropriate and adequate for your project?
4) Which techniques of individual activities are useful from your opinion and why?
5) Are the GSRM artefacts practical for software risk management and why?

B. Result

Table 1 shows a brief summary of findings about the goals, risk factors, events, and treatment actions. Note that we follow the same temporal notation as used in KAOS to represent goals and risk factors.

TABLE I.
PARTIAL STUDY RESULT

Goals	**Risk factors**	**Events**	**Treat.**
• SuccessProject • EffectiveCustomerCollaboration • Reduce[ErrorFromRequirements] • Maintain[EstimatedBudget ThroughoutDevelopment] • Maintain[EstimatedSchedule ThroughoutDevelopment] • ClearBusinessProcess • ClearMilestones	• UnclearBusiness Process • InadequateInformationabout ExistingApplication • LackofCooperation • RequirementsFaults • IncompleteRequirement Specification • IneffectiveCommunication • PractitionerLackof DomainKnowledge	• BudgetOverruns • ScheduleOverruns • ErrorneousRequirements • PassiveCustomer Involvement • ProjectComplexity • UnclearGoals • FailtoConvertOldData • CustomerDissatisfaction	• ObtainDetailsBusiness Process • Customer/UserActive Involvement • AdequateInformationabout ExistingSoftware • InvovleOnePractitionerin Development

The project operated under a very tight schedule, therefore maintain estimated budget and schedule throughout development are prioritised goals directly relevant for the top goal Success Project. Effective customer collaboration and reduce requirements error are also important goals as the project was an enhancement to an existing application. The team identified several risk factors as shown in table I which causally link multiple risk events and obstruct top goals. Hence these are the important risk factors. The project manager was concerned about the risk factors and events that negatively influenced the three important goals. Based on the interview response and brainstorming session the high prioritised risk events were identified as being: budget and schedule overruns erroneous requirements and passive customer involvement. Customer/user factors like passive involvement and poor cooperation in the development led to incomplete information about the existing application, business process and requirements faults. In offshore development projects these factors are common and important to control as early as possible otherwise estimated schedule and budget cannot be maintained and requirements errors cannot be reduced. The project manager also believed that the development team have adequate skills to manage the project but did not have sufficient knowledge about the client business processes and the existing application. Concerning risk mitigation the risk management team observed that controlling risk factors and events which obstruct multiple goals are effective. Mitigation of these obstacles directly or indirectly helps to control other obstacles and by that better attain the main project goals. E.g., the actions controlling customer/user lack of cooperation reduce the likelihood of risk events such as inadequate information, requirements errors and unclear goals which further control schedule overruns. The project manager emphasised on control actions which was believed to increase customer/user involvement in the project. The goal-risk model showed to be an effective approach for this analysis as it resulted in the client realising and agreeing with the problem and to dedicate three of the future product end-users making them available to the development team. Besides active customer involvement the company management (in the Bangladesh development company) approved committed an additional resource available to the development team to manage the schedule pressure. The development team also put more emphasise on obtaining information about the customer business process and existing application. Figure 4 shows the partial resulting GSRM goal-risk-treatment model.

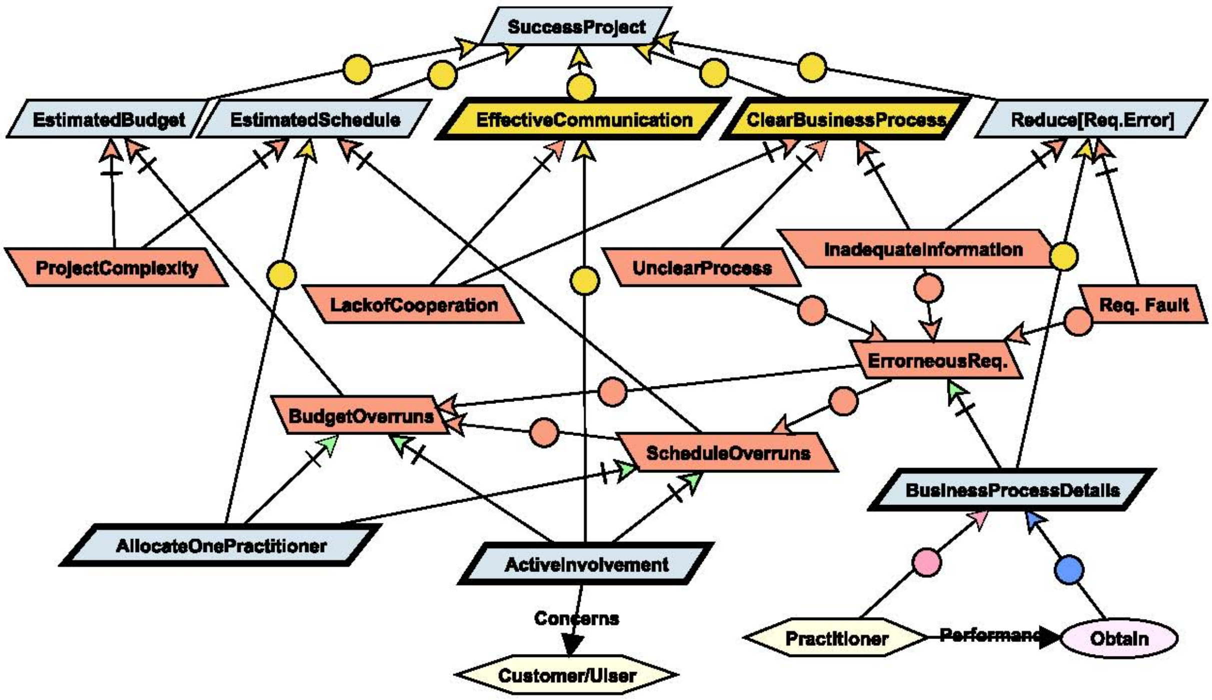

Figure 4. Goal-driven risk management model for success project

B. Study Validity

Case studies are prone to bias [14]. General threats to case studies are related to the difficulties of collecting reliable results and on generalising the findings. To counter such threats we examined possible validity threats from the beginning of the study. As an example data was collected not only from the interview responses but also from the project documents. Therefore the reliability of the data collection was improved by using multiple data sources and by transcripting the interview responses. This for traceability purposes and to enable backtracking in cases where the observation made gave conflicting results. In addition the case study was systematically planned and implemented. However this does not completely reduce biases, it merely attempts to reduce them and to provide a more versified information basis.

Internal validity. Internal validity concerns the set-up of the study and the independence of the findings. We mitigated risks to internal validity by not involving any of the principles investigators of GSRM and related research directly in the case study. The realism in the case study was also considered being valid as the project manager and development team initially were reluctant to execute a formal risk management process. This attitude was changed later on though as a consequence of a management decision and the outcome of the early GSRM activities. This means that the GSRM approach demonstrated the importance of risk management and quickly produced visual and critical insight. Furthermore apart from the student members the risk management team members did not have any prior knowledge about GSRM. Therefore the integration of GSRM into the project was executed in a realistic environment and allows to generalise over the easiness of employing and integrating the GSRM process into requirements engineering. Finally the interview responses and result of the project document inspection was further analysed and discussed during a brainstorming session intended to uncover additional risk factors and project goals or other constraints of relevance.

External validity. External validity concerns the ability in which the result from the case study can be generalised beyond the actual study context. We evaluated this by comparing the study results with results from similar studies. Namaste et al. [17] concluded that inadequate user involvement, lack of communication, poor change controls and lack of business know-how appear to be unique in offshore development context. Tsuji et al. [9] observed similar issues related to the project management and communication affect in offshore development context. The ACM task force result [21] recommends to consider both technical and non-technical aspects of offshore software development when examining project risks in such development contexts. Our findings conform with the above mentioned observations and in addition provided additional insight into the importance of understanding how the customer context may affect the project execution beyond the lack of user involvement. We could also generalize our previous study result as risk manangement showed to integrate well into requirements engineering [9]. Some studies investigated the main barriers of effective risk management practice during the development. E.g., study results indicate among other things: practitioners' lack of motivation, an over-focus on the tangible development cost and difficulty to quantify risk management benefits as the main barriers for the implementation of effective risk management [7, 12]. We also observed similar issues with respect to the project manager and the development team members. Software project undertakes multidimensional challenges during course of the development. Therefore validating a single method into a active onging software development project is a difficult task. Our case study is related to a single geographical region and therefore cultural biases may exist and the findings may have limited generality as a consequence of this.

IV. Lessons Learned

The case study was instrumental in moving the work on GSRM forward and provided us with additional insight that supported the objectives of the study in terms of how to integrate formal risk management practice into offshore development projects and on the usefulness of GSRM as such.

- Risk management is essential activity for any software project. Particularly, factors related to the customer/user perspective in an offshore development context should be analysed as early as possible. A project should explicitly set aside time and resource for risk management.
- The activities in GSRM are perceived to be systematic for managing risk in software development. Goal-driven risk management eases to consider generic and project specific goals and risk factors. GSRM artefacts such as goals and goal-risk model ease to communicate risk information to the project stakeholders and contribute to make informed decision about the control actions.
- Our study result suggests that non-technical risk such as customer/user involvement, communication and coordination, and domain knowledge about customer business process are essential in an offshore development context and have great influence on important risk events like erroneous requirements and budget and schedule overruns.
- In reality, it is difficult to obtain a precise value for the risk event likelihood and consequence that these may have to the goals. This makes subjective estimates important and BNN an effective tool, as it can reason under uncertainty and are based on the Bayesian interpretation of probability [3].
- Participants were also not always giving spontaneous answers to the close questions due to the difficulties of quantifying development component status. There are no industrial standard for such and no easy to use and widely adopted industrial best practices. Furthermore, modelling goal and risk and determine the causal relationship between these is a time consuming task. It is not always feasible to construct the model for every risk event especially for projects with continuous time pressure. Also creating and managing detailed textual representation about the risk artefacts requires extra effort. However, once the artefacts are developed they can be reused as a repository for other projects and act as generic risk specification information.

V. Related Works

There exist several relevant and well documented risk management approaches targeting the software engineering domain. Some of these approaches focus on risk factors derived from survey result rather than practical experience and case studies. The O-OSD trend brings additional challenges and considerations and has demonstrated a particular need for risk control for many reasons.

Prikaldnicki et al. in [16] proposed an integrated risk management process across three different organizational levels (strategic, tactical and operational) for distributed IT projects and emphasized the need of effective risk management practice from the early stage. Tafti [13] provides a framework for the major risk factors that must be taken into consideration when stakeholders decide to outsource IT activities. The framework categorise risks in several dimensions such as contract, privacy and security, decision process, outsourcing scope, technical returns, hidden cost and loss of IT expertise. Risks related to these dimensions must be assessed before an organization decides to outsource to an offshore location. Aron et al. [15] emphasised on what is sensible and business beneficial outsourcing specifically on how to determine which business activities to outsource. However due to its inherent nature it is difficult to understand and manage risks of complex business process. The authors propose a redesign of the process to address risk related to strategic, operational, intrinsic risks of atrophy and intrinsic risks of location. Kontio [10] describes the Riskit methodology which focuses on identifying stakeholder goals and risks that threaten these goals. Risks are analyzed and prioritized by deriving scenarios, which is a non-trivial task as a scenario involves several probabilistic elements. Procaccino et al., [11] identified seven early development factors and discussed how these contribute to the success or failure of a software project. Ropponen et al. [12] conducted a survey to investigate six software development risk components and showed how to provide assistance in addressing these components. In our previous works [19, 20] we proposed a goal-driven risk management model (GSRM) for managing software development risk by extending the KAOS goal modelling language [1]. KAOS aims to model not only what and how aspect of requirements but also why, who, and when and contribute to a comprehensive goal oriented requirement engineering (GORE). The model also includes obstacle as unintended risks that associates with undesirable behaviour and anti goal as intended risk that associates with intended risk.

Iacovou et al. [5] summarised top ten risk factors in offshore-outsourced development projects . The risk factors are ranked by focusing on three main areas: communication, client’s internal management and vendor capabilities. Namaste et al. [17] compared the risk factors between offshore and domestic outsourcing. The result showed that risk factors related to project management such as lack of top management commitment, inadequate user involvement and failure to manage end user expectation commonly appeared on the top of both domestic and offshore risk lists. Some risks are unique in offshore context such as lack of communication, poor change controls, lack of business know-how and failure to consider all costs. An ACM task force report by Aspray et al. [21] emphasised to consider risks from both technical and nontechnical issues. Tsuji et al. [9] proposed questionnaires assessment scheme considering software, vendor and project properties to quantify risk of offshore software outsourcing. The survey result concluded that vendor properties and capabilities such as communication and project management are the main factors affecting the result of development.

GSRM follows the basic concepts of KAOS and enhances the existing risk management practice by explicitly include it into the requirements engineering phase using a goal-driven approach for risk management. The model considers a holistic view on software development risk management taking both technical and nontechnical perspectives into consideration. KAOS also includes risk management activities within requirements evaluation but its main focus is on the completeness of the requirement specification.

VI. CONCLUSION

Effective risk management practice increases the likelihood of project success. Making timely and well informed decision is important for controlling uncertainty during the development. We believe that GSRM contributes in this direction and enables software risk management activities to be explicitly integrated into requirements engineering, i.e., from early development. The paper discusses the results of a case study implementing the GSRM process into a running offshore software development project. The result from the study showed that GSRM is systematic and easy to employ in practice. Software development projects contain a number of goals which need to be attained to achieve a successful project execution. Our goal oriented approach for risk management is applicable in any software development project not only in an offshore context. Further work includes a revision of GSRM based on the lessons learned and to conduct follow-up studies to further enhance the framework. We hope this will produce an implementation guideline pushing risk management for software development into an industrial best practice.

REFERENCES

[1] A. van Lamsweerde, Requirements Engineering: From System Goals to UML Models to Software Specifications, Wiley, January 2009.

[2] B. Boehm, Software Risk Management: Principles and Practices, IEEE Software, Vol. 8, pp. 32-41, January 1991.

[3] B. D. Finetti. Theory of Probability Volume 1. John Wiley & Sons, 1973.

[4] B. Schätz. Model-Based Development of Software Systems: From Models to Tools. Habilitation, Technische Universität München, 2008.

[5] C. Iacovou and R. Namaste, Risk Profile of Offshore-outsourced development project, Communication of the ACM, Vole 51, No. 6, pp. 89-94, June 2008.

[6] D. Bradstreet, Dun & Bradstreet's Barometer of Global Outsourcing, vol. 2007: Dun & Bradstreet, 2000.

[7] E.E. Odzaly and P.S. Des Greer, Software risk management barriers: An empirical study. In Proceedings of the 3rd international Symposium on Empirical Software Engineering and Measurement, 2009.

[8] F.V. Jensen, Introduction to Bayesian Networks. Springer-Verlag New York, Inc., Secaucus, NJ, USA 1996.

[9] H. Tsuji, A. Sakurai, K. Yoshida, A. Tiwana and A. Bush, Questionnaire-Based Risk Assessment Scheme for Japanese Offshore Software Outsourcing, B. Meyer and M. Joseph (Eds.): SEAFOOD 2007, LNCS 4716, pp. 114–127, 2007.

[10] J. Kontio, 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] J. D. Procaccino and J.M. Verner, Case Study: Factors for Early prediction of software development success; Information and Software Technology; Vol. 44, 2002.

[12] J. Ropponen and K. Lyytinen, Component of Software Development Risk: How to address them? IEEE Transactions on Software Engineering, Vol.26, Issue: 2, Feb 2000.

[13] M. H. A. Tafti, Risks factors associated with offshore IT outsourcing", Industrial Management & Data Systems, Vol. 105 Iss: 5, pp.549 – 560, 2005.

[14] P. Runeson and M. Höst, 2009. Guidelines for conducting and reporting case study research in software engineering, Empirical Software Engineering, 14(2), 2009.

[15] R. Aron, E. K. Clemons and S. Reddi, Just Right Outsourcing: Understanding and Managing Risk , Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05) - Track 8, 2005.

[16] R. Prikladnicki, R. Evaristo, J. L. N. Audy, and M. H. Yamaguti, Risk Management in Distributed IT Projects: Integrating Strategic, Tactical, and Operational Levels in a Headquarter-Subsidiary Relationship, International Journal of e-Collaboration, v. 2, p. 1-18, Oct-Dec, 2006.

[17] R. T. Namaste and C. L. Iacovou, A comparative study of important risk factors involved in offshore and domestic outsourcing of software development projects: A two-panel Delphi study, Information and management journal, Elsevier Science Publishers. Volume 46 , Issue 1, January 2009.

[18] McConnell S. Rapid Development. Microsoft Press, 1996.

[19] S. Islam, Md. M. A. Joarder and S. H. Houmb, Goal and Risk Factors in Offshore Outsourced Software Development from Vendor's Viewpoint, Proceedings of the Fourth IEEE International Conference on Global Software Engineering, Limerick, Ireland , 2009.

[20] S. Islam and S. H. Houmb, Integrating Risk Management Activities into Requirements Engineering, In Proc. of the 4th IEEE International Conference on Research Challenges in Information Science (RCIS2010), Nice, France.

[21] W. Astray, F. Maydays, and M.Y. Verdi, Globalization and of shoring of Software: A Report of the ACM Job Migration Task Force. ACM, NY, 2006.