Strategy of Risk Management for a Distributed Software Engineering Environment

Lafaiete Henrique Rosa Leme, Tania Fatima Calvi Tait and Elisa Hatsue M. Huzita

Computer Science Department, State University of Maringá
Colombo Av., 5790, Maringá, Paraná, Brazil

Abstract. The risk management is a subset of the entire software development project management and includes the process concerned about identify, analyze and control any threat to the project success. The objective of this paper is to present a strategy for an effective risk management, for supporting the project management model integrated to DiSEN (Distributed Software Engineering Environment). This strategy proposes supplying the lack of a well-defined process for the risk management not only for DiSEN, but for distributed environments in general.

1 Introduction

A project of software development differs from other sorts of projects due to the innate characteristics of software. The software development by geographically distributed teams, in addition to the inherent difficulties to the software project, has increased the own characteristics of distributed environments (Enami et al 2006). Those characteristics can be related to the software (parallelism, concurrence, distribution and synchronization) as well as the project by itself (culture, language, mediated communication, knowledge dispersion, etc).

The software project manager has to keep in mind that the project should intend the quality, productivity and cost reduction through the planning and execution of the product development (Project Management Institute, 2004). Thus, a Risk Management must be done rigorously and permanently. Knob (2005) defines risk management in Software Engineering as an evolution of the risk concept, that evolved from an analysis in the development model into a control that must permeate the entire software life cycle. But, some approaches of Risk Management, still used, does not regard totally the specific characteristics of a software project. Even more when thinking about development in a distributed environment.

The strategy of risk management, here presented, is worried about the typical characteristics of distributed environments and was based on PMBOK (Project Management Body of Knowledge), CMMI (Capability Maturity Model Integration), no MSF (Microsoft Solutions Framework) e no No-Risk. This paper is organized as follows: item 2 presents DiSEN and the Project Management Model developed, in item 3 are presented some approaches of risk management for software projects, item 4 presents some models of risk management, in item 5 is described the risk management strategy here proposed, and finally in chapter six are found the conclusions of this research.

Henrique Rosa Leme L., Fatima Calvi Tait T. and Hatsue M. Huzita E. (2007).
Strategy of Risk Management for a Distributed Software Engineering Environment.
In *Proceedings of the 4th International Workshop on Computer Supported Activity Coordination*, pages 66-75
DOI: 10.5220/0002435100660075

2 DISEN

The DiSEN (Distributed Software Engineering Environment) is an agent-based environment, and its goal is to give support to the distributed development of software products. This work contributes to that environment with a well-defined strategy of risk management. It has concerns not only about the classic risks of software development, but also to typical risks of distributed environments.

2.1 Project Management Model

This PMM (Project Management Model) proposed to DiSEN considers aspects related to the project management area, and factors associated with the physical distribution of members of the team. The organization is hierarchically organized in three levels: strategic, tactical and operational (Enami et al, 2006), each one with its specifics controls and rules. All the members of the environment must receive information that was recorded in the environment repository, and the DIMANAGER (Pedras 2003) is the responsible tool to the information propagation.

The project information control in a distributed environment is necessary to record and control more information about the project, as: human and material resources of each unit of development registered in DiSEN; a responsibility/authority of the project; knowledge, skill and training of the team members; and, the cultural aspects of everyone in the organization.

3 Approaches of Risk Management in Software Projects

3.1 Approach of SEI: The CMMI

The objective of CMMI is supply orientations to make better the process of an organization and its capability to manage the development, acquisition and maintenance of products and services (SEI 2002). It offers two kinds of representation: continuous or staged. The continuous allows to select the improvement sequence that better attends the objectives of the organization, and reduces its risk areas. The staged representation offers a supported sequence of improvements, starting with basic practices of management, and evolving for successive levels, each one basing the next (SEI 2002).

Due to the interest, in this work, to the analysis of the risk management process as a whole, it is used the staged representation, once the continuous one uses qualification levels, which dissolves the risk management on the levels (SEI 2002). The staged representation of the CMMI presents five maturity levels: Initial, Managed, Defined, Quantitatively Managed and Optimized. Each maturity level consists of a pre-defined set of process areas, each one with its objectives to reach and various practices that helps in search of those objectives (SEI 2002). The Risk Management is an area presented only in the level 3, Defined, where the processes are already marked and understood and are described in patterns, procedures, tools and methods.

3.2 Approach of PMI: The PMBOK

The PMBOK (Project Management Body of Knowledge) is composed by nine areas of knowledge of project management: project integration management, project scope management, project time management, project cost management, project quality management, project human resource management, project communications management, project risk management, and project procurement management.

Its risk management proposal is composed of the process worried about the risk management planning, identification, analysis, response, and monitoring and control of risks in a project. Its objectives are increase the probability and impact of positive events, and decrease the probability and impact that adverse events could cause to the project objectives (PMI 2004).

Its processes are: risk management planning (decides how to tackle, plan and execute the risk management activities for a project); risk identification (determines which risks might affect a project and documents their characteristics); qualitative risk analysis(prioritizes risks for further analysis through the valuation and combination of their probabilities of occurrence and impact); quantitative risk analysis (analyses numerically the effects of identified risks to the project global objectives); risk response planning (develops options and actions to increase opportunities and decrease threats to the project objectives); and risk monitoring and control (tracks the identified risks, monitors residual ones, identifies new ones, run response plans to risks and evaluate the effectiveness in the entire project life cycle).

3.3 Approach of MSF (Microsoft Solutions Framework)

The MSF Risk Management Discipline (MSF 2002) defines six logic steps to plan and execute strategies for the management and extract knowledge for the organization: (1) *Risk identification* should be done as soon as possible, and repeated during the project life cycle. It allows the involved members to identify the risks so that the team stays warned about this potential problem; (2) *Risk analysis* converts the data obtained in risk identification in a form so that the team can use it to make decisions about the risk prioritizing. This prioritization gives parameters to the team in manage the most important risk, in accordance with the project resources; (3) *Risk planning* and schedule use the results of analysis and prioritization to compose strategies, plains and actions. They should ensure that plans are merged to the routine of the team. The scheduling links directly the risk planning to the project planning; (4) *Risk tracking* watches the status of specific risks and the progress of the respective action plans. It includes monitoring the probability, impact, exposure and other measures whose changes may affect the project characteristics, modify its resources or schedule. Report gives to the team and other stakeholders the status about the project risks and the plans to manage them; (5) *Risk control* is the process of executing the action plans for risks and the associated reports; (6) *Learn* formalizes the learned experiences and the relevant project artefacts. It is made in a form, for reuse of the team and the whole organization.

These are logic steps that does not must be followed strictly in this chronological order. For instance, a team can pass through steps 2, 3 and 4 more frequently for a determined risk class, and pass just periodically through the step of learning, to report the knowledge for the organization.

3.4 Approach of No-Risk

The No-Risk is a process for the application of risk management in software projects for information systems. It was conceived by a study with project managers, with the objective of identify the relevant risk factors on software projects, about cost, time and quality (Oliveira 2006).

It is defined a process risk management in software projects, associated to a software development process, which proposes the risk analysis based in factors previously mapped. No-Risk is organized in stages, as follows: risk management planning, risk identification, risk control planning, risk monitoring, risk communicating, and risk learning.

4 Risk Management Models

4.1 Management and Analysis Model of Peters

Peters (2001) presents the risk management clearly together with its analysis. According to his proposal, the risk management is divided in five main activities: planning, control, monitoring, direction, and recruitment. Risks might be assessed about their seriousness, in qualitative and quantitative units. For instance, the qualitative evaluations of severity should be made by and expert risk manager, using words like "zero", "very low", "low", "average", "high" or "very high". The quantitative evaluations are made in the interval from 0 to 1.

The main objective or risk analysis is to develop a set of strategies of risk prevention. This consists of execute a detailed project that embody the resources of fail tolerance in the software architecture, and also of improving the project so that it presents a system behavior that accomplishes system security, testability and maintenance. The risk engineering is present in all the software life cycle, and there is a permanent interchange between risk analysis and the risk management processes.

4.2 Evaluation and Control Model of Rook

Rook (1993, apud Pfleeger 1998) distinguish risks from other project events focusing in three characteristics: (1) risk impact: a loss associated with the event. The event must create a situation where something negative happens to the project: a loss of time, quality, money, control, understanding, and so on; (2) risk probability: the likelihood that the event will occur. We must have some iadea of the probability that the event will occur. The risk probability is assessed from 0 (impossible) from 1 (certainty), and in this last case, once the risk occurrence is sure, it is called problem; (3) risk control: the degree to which we can change the outcome. For each risk, we must determine what we can do to minimize or avoid the impact of the event. It involves a set of actions taken to reduce or eliminate a risk. The risk probability changes over time, as can the impact. It is task of the project manager to track these values over time, and plan for the events accordingly.

Pfleeger (1998) defines two main types of risk: (1) Generic risks (common to all software projects as: misunderstanding the requirements or staff turnover); (2) Project-specific risks (threats that result from the particular vulnerabilities of the given project as a network software needed for the project process was not available).

The risk management evolves some important steps. Firstly, we should assess the risks of the project to be developed, to understand what might happen during the development or maintenance process. This evaluation is divided in: identification (that can be made with the use of several techniques), analyse and prioritize risks. The notion of risk control allows the manager to know that he might be not able to eliminate all the risks. Instead, they might be minimized or mitigate by taking action to handle the unwanted outcome in an acceptable way. Risk control involves risk planning, reduction and resolution.

4.3 Model of Sommerville

Sommerville (2000) says that risks can threat the project (its schedule and resources), the software being developed (its quality and performance) or the organization (the one which develops or needs the software). The sorts of risks that might affect a project depends on the project and the organizational environment where it is being developed. However, it is possible to abstract some of them in general lines, as represented in Table 1.

Sommerville (2000) proposes a process of risk management with the following steps: (1) identification (knowing the project, product or business risks), (2) analysis (probability and consequences or risks, if become real), (3) planning (make plans for tracking risk, to avoid or minimize its effects on the project) and (4) monitoring (the risk is constantly assessed and plans for mitigation are revises in proportion to more information of risk become a available).

Table 1. Potential software risks (Sommerville 2000).

Risk	Risk sort	Description
Staff turnover	Project	Experienced staff will leave the project before it is finished.
Management change	Project	There will be a change of organisational management with different priorities.
Hardware unavailability	Project	Hardware which is essential for the project will not be delivered on schedule.
Requirements change	Project and product	There will be a larger number of changes to the requirements than anticipated.
Specification delays	Project and product	Specifications of essential interfaces are not available on schedule
Size underestimated	Project and product	The size of the system has been underestimated.
CASE tool under-performance	Product	CASE tools which support the project do not perform as anticipated
Technology change	Business	The underlying technology on which the system is built is superseded by new technology.
Product competition	Business	A competitive product is marketed before the system is completed.

5 Risk Management Strategy

Karolak (1998) says that risks in distributed software development projects are inclined to be more centred in not so visible aspects, when compared to the classical software development. There must be activities of risk identification and planning of mitigation strategies in distrib-

uted projects: organizational, technical and communication. Moreover, maybe there are risks in more than one category, and these must be on the top of priorities list.

The strategy presented here is an extension of the project management model proposed by Enami et al (2006), which classifies the users in distinct levels in the organization, such as operational, tactical and strategical. And also it is used the categorization of Sommerville, stratified by the scope where the risks may impact: project, product and business.

In this strategy, information is always disseminated in two directions. Firstly, bottom-up direction, in which the organization lower levels generate information related to the risk management process, and each manager, when receive the data sent from the immediately lower level, upgrades it and send the information again, until reaching the tactical level. In this level, the information is upgraded again, and then is started the propagation to the lower ones in a top-down direction. This is the documentation which will guide all the risk management process.

The stages of this strategy are: Risk Discovering, Risk Analysis, Risk Mitigation Strategies, and Risk Monitoring and Learning.

5.1 Risk Discovering

As described in by MSF (2002), risks must be clearly identified and classified so that the team can enter into an arrangement before evaluation them. During the risk identification, the focus of the team must be intentionally expansible for any new risk that might be spotted, besides the already known ones from previous projects. It is important to give special attention for looking gaps in what is already know about the project and its environment that may affect, in an unfavorable way, the project, or limit its success.

When treating an outsourcing project, policies and organizational procedures will, certainly, be disagreeing. But, in early project management is necessary a standardization of absolutely all the elements project related, for instance, documentation, procedures, representation and implementation characteristics.

In 1989, Boehm presented a risk classification structure, also called risk taxonomy. This classification is critical to establish the workflow and bases the organizational risk knowledge because provides a base for indexing new contributions and searches, and recover the already existing work.

In distributed projects, communication and organizational aspects deserves special attention, once the geographic distribution makes these factors more complex. If we are leading with a project in a distributed environment, the complexity is much bigger, with the insertion of other factors, as: language, time zone, local cultures, and others.

Fortified with all the presented framework, the software engineers and project managers, in each distributed unit, proceed the risk discovering process. Must be used methods, as suggested by McManus (2004), as follows: brainstorming, swot analisys, checklists and questionnaires, and interviews.

The result of risk discovering must be recorded by the project manager and reported to the local manager, in case the last one has not participated of the process yet. It is task of the local manager reports to the general managers, informing the process current situation, and what was produced in this stage.

5.2 Risk Assessment

As said by McManus (2004), the objective of this stage must be investigate the probability of each risk and the impact that the identified risks will have on the project.

It can be used a relevant scale representing probabilistic values, since "unlikely" until "certainly", or numeral values (for instance, 10%, 30%, 50%, …).

Each risk must have a probability assigned (SEI 2002). Each one is assessed and then values are assigned, that might include probability, consequence (impact) and limits. The values assigned to the risk parameters can be integrated to produce additional measures, as: risk exposition, which can be use to prioritize the risks to be treated.

MSF (2002) says that the probability and impact, if combined, turn possible to evaluate the risk exposition, also treated in PMI (2004).

The project manager is the main responsible for the entire process or risk assessment, and he can include software engineers and other stakeholders on the evaluation. Once the result is reported, the local manager can respect the result of the works, include or consider with the project manager some relevant points.

It is task of the local manager keep informed the general manager, about the results of that evaluation. SEI (2002) says that is very important that this communication carries: summary of the most critical risks, and key-parameters of risks (as the probability and consequence of these risks).

The local manager makes the integration of all the evaluations received and communicates, again, the general managers any appropriated verification.

5.3 Risk Mitigation Strategies

Some organizations facilitates the risk mitigation implementation more than others, depends on the risk maturity level conscience. The general manager must be attentive to the characteristics of each development unit.

It is role of the project manager, in each unit, to manage the risk mitigation on its elaboration and implementation. MSF (2002) informs that each organization is apt to adopt one of the following strategies developes plans for risk mitigations: research, accept, avoid, transfer, mitigation and contingency. The resulting status of these mitigation efforts must be communicate between the managers.

Research:

Many times many things are not known related to a specific risk. This strategy defines what must be learned about a risk for acquiring more information and be able to determine better the risk characteristics before make any decision.

Many risks presented in projects come from incomplete information. Risks related to the uncertainties because of the knowledge missing are commonly solved or managed when you learn more about their domain.

If the team chooses the research, the risk plan must include an adequate research plan, which contains the hypothesis to be tested, team, and any necessary hardware.

Accept:

Might be possible to admit the consequences of a risk, being not necessary take other initiatives over the already planed ones.

Accept a risk is not, necessarily, doing nothing. The reason for this chosen must be, instead of being mitigated or moderated. These risks must be monitored anyway, since changes may happen on its probability or impact.

Avoid:

As shown by PMI (2004), it evolves changes on the project management plan to eliminate the threat, isolate project objectives of the risk impact, or "relax" the objective in danger, as schedule extending or scope reduction. Many risks that can arises on the beginning of the project can be avoided cleaning requirements, obtaining information, improving communication or acquiring skills. Project changes sugestions can be made, as adopting less complex process, doing more testes, or chose another

McManus (2004) says that, when developing risk scape strategies, the organization should take into account the potential value of the activity and the potencial risk, once those strategies are expensive e could severely restrict the company skill of its own business management. Even though, some risks that arises on the beginning of the projects can be avoided by a better requiriments eliciting, information obtaining of communication improvement, the last one very important in a distributed software engineering environment.

Transfer:

Sometimes it is possible to transfer a risk so that it is can be better managed by another entity outside the project (MSF 2002). For instance: insurance, external consultants, component purchasing, and outsourcing services. It consists in transfer the negative impact of a threat, add with the response responsibility, for a third part. Risk transfer simply outsources a risk responsibility; it does not eliminate the risk. Risk responsibility transferring is more effective when treating budget areas. Sometimes it evolves payment to the part that received the risk.

The main reason to take this strategy is having all the protections about that risk, once it cannot be avoided and it need more resources than those available on the organization.

Mitigation:

It implies in a reduction of the probability and/or impact of a risk event for an acceptable level. Actions for reduce the probability/impact of a risk to occur are always more efficient trying to repair the damage after the risk occurrence. The strategy that the teams always adopt that can make anything to reduce the probability or impact of a risk, so that it become more acceptable for the project and the organization.

Not all risk projects has a reasonable and good cost-profit mitigation strategy. In the cases where the mitigation is not possible, it is essential to consider a contingency planning.

Contingency:

Evolves the elaboration of one or more actions and activities to reduce the impact of a risk through a planed reaction. Such activities have the objective of prevent any adverse fail to project objectives. Where was not possible to reduce the probability, it is possible to address the risk impact focusing coupling that reduce the impact. For instance, developing redundance in a subsystem can reduza de impact of a fail of the original component.

Risks in this category are those ones that can be critical or catastrophic to the project. It is important that the team and the managers to enter into an agreement as early as possible, about the contingency starters and their values, thus no delays happen on the budget draw up or necessary resources to put in practice the contingency plan.

5.4 Risk Monitoring and Learning

In this step the risk documentation produces on the prevous steps should be recovered and compared to the facts the really happened, upgrade and then, the new documentation is added to the institutional knowledge base. PMI (2004) says that planned risk responses, that are included in the project management plan, are executed during the project life cycle, but the process must be constantly monitored for new risks or to any change in the ones already identified.

The monitoring should: identify, analyze and plane recent raised; to keep the conduct for the identified risks; to reanalyze already existent risk; monitoring starting conditions for the contingency plan; to monitor residual risks; to review the risk response execution while evaluating its effectiveness.

Other objectives are to determine if: project assumptions are still valid; risk, as assessed, have changed from the previous status, with course analysis; peculiar risk management policies and procedures are being followed; cost or schedule contingency reserves should be modified according to the project risks.

SEI (2002) presents the following examples of actions taken in monitoring and control: identified, managed, tracked and controlled risk quantity; risk subjection and for each assessed risk and as a summary percentage of management funds; activity of changes for the risk mitigation plans (for instance: processes, schedule, resources); occurrence of unforeseen risks; volatility of the risk classification; effort comparison of estimated risk mitigation impact versus the real one.

During the project development, the software engineer should report the happened problems observed during the development, that stays recorded in a base of the system. That informação will be used by the project manager in monitoring and control step.

In accordance to MSF (2002), organizations that use risk management techniques frequently judge necessary to create a structured approach for the project risk management. Conditions to succeed that are: (1) an individual should be given ownership of a specific risk classification (might be the new ones or those that had their mitigation strategies succeeded) and also responsibility for approving changes; (2) risk classifications should balance the need for a comprehensive coverage or risks against complexity and usability; may be created different risk classifications for different projects; (3) a risk knowledge base should be configured to store classifications, definitions, diagnosis criteria and risk assessment systems, and capture the feedback from the experience of the team; (4) the process of risk revision should be managed to ensure that all the knowledge was captured.

Thus, the suggested changes and corrective and preventive actions, that were taken and the result of previous activities should be documented.

6 Conclusions

Risks in software development projects should be managed since the early phases of project conception, from its identification and passing through all the stages described in this paper.

The risk management in a distributed software engineering environment should be even more detailed, because of the inclusion of factors inherent to these environments, for instance: the time zone, cultural differences, etc.

In an environment which organization is organized in strategic, tactical and operational levels, the information should be always available to the ones who take interest in them. In each cycle of the risk management process, the information should bottom-up flow, for the well aware of the manages about the activities running in each one of the organizational layers. Then, the way should be drop-down in which the general manager has already received all the information, made updates and distribute again to the under layers.

Future works are the treatment not only of the negative risks, but also the positive ones, so that it became possible to take advantage of the opportunities that a project holds, and the implementation of a tool to be connected to the DiSEN, and its objective is to support the risk management in that distributed environment. There is, still, the necessity of elaboration of a knowledge repository for this environment, so that not only the risk knowledge is recorded, but the all the knowledge or the environment.

Acknowledgements

We are thankful to CNPq for the financial support to the DiSEN project, process no. 50651/2004-9.

References

1. Boehm, B. W.: Software Risk Management. New York: IEEE Press (1989)
2. Enami, L.N.M.; Tait, T. F. C.; Huzita, E. H. M.: A project management model to a distributed software engieering environment. International Conference on Enterprise Information Systems (2006).
3. Karolak, D. W.: Global Software Development – Managing Virtual Teams and Environments. Los Alamitos, EUA: IEEE Computer Society (1998)
4. Knob, F. et al.: RiskFree – Uma Ferramenta de Apoio à Gerência de Riscos em Projetos de Software. In: Anais do Simpósio Brasileiro De Sistemas De Informação, 2 (2005)
5. McManus, J.: Risk Management in Software Development Projects. Burlington, Elsevier Butterworth-Heinemann (2004)
6. Microsoft Solutions Framework: MSF Risk Management Discipline. version 1.1 (2002) Avaiable from: http://www.microsoft.com/msf/. [cited 20 February 2004]
7. Oliveira, G. C.:No-Risk – Um Processo para Aplicação de Gerência de Riscos de Software Focados em Sistemas de Informação. Master. Pontifícia Universidade Católica do Rio Grande do Sul (2006)
8. Pedras, M. E. V.: Uma Ferramenta de Apoio ao Gerenciamento de Desenvolvimento de Software Distribuído. Master. Universidade Estadual de Maringá (2003)
9. Peters, J. F.: Engenharia de software. Trad. Ana Patrícia Garcia. Rio de Janeiro: Elsevier (2001)
10. Pfleeger, S. L. Software engineering: theory and practice. [s.l.]: Prentice Hall (1998)
11. Prikladinicki, R. et al.: Risk Management in Distributed IT Projects: Integrating, Strategic, Tactical and Operational Levels. International Journal of e-Collaboration, 2(4) (2005) 1-18
12. Project Management Institute: A guide to the project management body of knowledge: PMBOK guide. 3rd ed. Pennsylvania: PMI (2004)
13. Software Engineering Institute: Capability Maturity Model Integration (CMMI) Version 1.1. Carnegie Mellon University (2002)
14. Sommerville, I.: Engenharia de Software. São Paulo: Addison Wesley (2000)