Volume : 2 | Issue : 6 | June 2013 • ISSN No 2277 - 8179

Risk Analysis and Mitigation Steps in Different Phases of Software Development

Engineering

KEYWORDS : Risk Analysis, Software Development Life Cycle

Sonam Mittal B.K. Birla Institute of Engineering & Technology, Pilani, Rajasthan

ABSTRACT

Software Engineering is the discipline of computer science concerned with developing small and large applications. There are different phases of software development life cycle (SDLC) model. The different phases of SDLC model include requirement analysis, design, coding, testing, implementation and maintenance. Each phase in the model has some risks involved in it. Mainly there are two things involved in the risk i.e. uncertainty and loss. Due to these risks software can face various problems that can lead to some negative outcomes and in some extreme cases can lead to a great loss. This paper explores different risks involved in various phases of software development life cycle along with their mitigation steps.

I. INTRODUCTION

Software Engineering is defined as the application of tools and methods for producing an efficient solution to a real world problem [1]. Software engineering covers not only the technical aspects of building the software but also management issues such as directing programming team, scheduling & budgeting [2]. Managing software projects is a very complex task. Proper software management requires a balanced approach for different processes that include planning, organizing, staffing, controlling and coordination. Improper management can lead to failure of the project.

Risks are involved in every type of software development [3]. In some cases the risks are negligible and in some cases the risks are very severe that they can't be ignored. When the risks are negligible they have a low affect on cost, schedule and other phases of the project.

Software risks are involved while developing software. They can occur in any phase of the software development life cycle. Risk can be defined as suffering from harm or loss due to some unfavorable conditions. Risk is something that adversely affects the project. Risks are associated with projects and proper precautions must be taken care of to handle the risks.

II. SOFTWARE RISK MANAGEMENT

Since there could be various risks associated with the software development projects, the key to identify and manage those risks is to know about the concepts of software risk management. Many concepts about software risk management could be identified but the most important are risk index, risk analysis, and risk assessment [7].

2.1 Risk Index: Risk index is the multiplication of impact and probability of occurrence. Risk index can be characterized as high, medium, or low depending upon the product of impact and occurrence. Risk index is very important and necessary for prioritization of risk.

2.2 Risk Analysis: Basically, risk analysis is used to identify the high risk elements of a project in software engineering. Also, it provides ways of detailing the impact of risk mitigation strategies. Risk analysis has also been found to be most important in the software design phase to evaluate criticality of the system, where risks are analyzed and necessary counter measures are introduced. The main purpose of risk analysis is to understand risks in better ways and to verify and correct attributes. A successful risk analysis includes important elements like problem definition, problem formulation, data collection.

2.3 Risk Assessment: Risk assessment is another important case that integrates risk management and risk analysis. Risk assessment requires correct explanations of the target system and all security features. It is important that a risk referent levels like performance, cost, support and schedule must be defined properly for risk assessment to be useful.

III. RISK CLASSIFICATION

The key purpose of classifying risk is to get a collective viewpoint on a group of factors. These are the types of factors which will help project managers to identify the group that contributes the maximum risk. Risk classification is considered as an economical way of analyzing risks and their causes by grouping similar risks together into classes. Software risks could be classified as internal or external. Those risks that come from risk factors within the organization are called internal risks whereas the external risks come from out of the organization and are difficult to control. Internal risks are project risks, process risks, and product risks. External risks are generally business with the vendor, technical risks, customers' satisfaction, political stability and so on. In general, there are many risks in the software engineering which is very difficult or impossible to identify all of them. Some of most important risks in software engineering project are categorized as software requirement risks, software cost risks, software scheduling risk, software quality risks, and software business risks. These risks are explained detail below [6, 7].

3.1 SOFTWARE REQUIREMENT RISKS

1. Lack of analysis for change of requirements
2. Change extension of requirements
3. Lack of report for requirements
4. Poor definition of requirements
5. Ambiguity of requirements
6. Invalid requirements

3.2 SOFTWARE COST RISKS

1. Lack of good estimation in projects
2. Unrealistic schedule
3. The hardware does not work well
4. Lack of testing
5. Lack of monitoring
6. Complexity of architecture
7. Extension of requirements change
8. The tools does not work well
9. Personnel change, Management change, technology change, and environment change
10. Lack of reassessment of management cycle

3.3 SOFTWARE QUALITY RISKS

1. Inadequate documentation
2. Lack of project standard
3. Lack of design documentation
4. Inadequate budget
5. Unrealistic schedule
6. Lack of enough skill
7. Lack of testing and good estimation in projects
8. Inadequate knowledge about techniques, programming language, tools

3.4 SOFTWARE SCHEDULING RISKS

1. Inadequate budget
2. Change of requirements and extension of requirements
3. Inadequate knowledge about tools and techniques

4. Long-term training
5. Lack of employment of manager experience
6. Lack of enough skills
7. Lack of good estimation in projects

IV. METHODOLOGY

Each of phases of software engineering has some risks involved in it. The probability that software will suffer an event of a negative impact is determined from various factors, the ease of executing the risk, lack of interest, a system's existing vulnerabilities and cost or impact in a particular business context. During the software development process various strategies for risk management could be identified and defined according to the amount of risk influence. The identified risks were analyzed for their impact on cost, schedule, some other phases and some other factors. Further the Probability of occurrence and sensitivity of these risks were also identified and analyzed. The effects of risks were noted on the following Parameters:

a) Cost (Directly)
b) Schedule
c) Reputation of Organization
d) Other phases of Development

Most of the risks affect on the schedule of the project, which indirectly affects the cost. After the schedule the second affected parameter is cost of the project. The risks affecting other phases and other parameters are very few. There are some risks that affect the repute of the organization. The Complexity Risk is more sensitive and has at least minimal affect, so it can be categorized one of the Major risks. Availability of resource person has mostly negligible effects. But it doesn't mean that it has no effect. Sometime it has minimal and sometimes considerable effects. Change in Requirements is a major risk, which can have sometimes negligible, sometimes minimal and sometimes considerable risks. The major risks involved in SDLC & mitigation steps are [3, 5]:

A. Requirement Phase:

i. Risk due to change in requirements (Minor or Major)?

Mitigation:

a) Using Proper Performa or pattern to get requirements from the user.
b) Conduct interview from time to time with the person in-charge.
c) Feasibility of the changed requirement should be measure.

ii. Risk due to lack of client/ user/ Customer knowledge?

Mitigation:

a) If Client has not much knowledge about system to be built, contact any other person from that organization.
b) Client should be informed properly about the technology and the system.

B. Analysis Phase:

i. Risk due to ignorance of non-functional requirements?

Mitigation:

a) Performa for requirement gathering should be perfect enough to capture the non functional requirements as maximum as possible and all the documents regarding previous system should be collected.

C. Design Phase:

i. Risk due to use of reusable components?

Mitigation:

a) Reusable components should be designed in such a way that they can be easily modified.
b) Iteration and prototyping should be used.

ii. Risks due to less expertise on reusable components?

Mitigation:

a) While selecting the reusable component, it should be noted carefully that it is compatible with the other modules used in project.

D. System Development or Coding

i. Risk due to less command on programming language used?

Mitigation:

a) Choose the language on which the expertise is more.
b) Take proper help from some senior persons.

ii. Risk due to availability of resource person having command on the technology?

Mitigation:

a) This Risk can be covered by searching the right person having the command on the technology.

iii. Risk due to skills / competency of the resource person on the technology?

Mitigation:

a) Skills and competency of the person should be high.

iv. Risk due to team harmony and working relationship.

Mitigation:

a) Different HRM techniques should be practiced to develop harmony among the team members.

v. Risks due to missing comments?

Mitigation:

a) Proper comments should be added during function, classes and other major modules.

E. Integration and Testing

i. Risk due to testing?

Mitigation:

a) Software should be tested according to the complexity and nature of project.
b) This testing level should be defined at the Requirement specification time.

F. Implementation

i. Risk due to restructuring of the site where software is to be installed?

Mitigation:

a) New Structure of the site should be planned carefully keeping view of need of the restructure or according to the place.
b) If the site is restructured developers should be given prior information about the restructuring so that they can develop the product accordingly.

V. CONCLUSION

Software risks are involved while developing the software products. Risk can be defined as the probability of suffering harm or loss. The risk requires two things i.e. Uncertainty and Loss of the risks are Dynamic and some are Static. In Dynamic there might be some profit or loss associated with risk. And in Static only Loss is associated with the risk. This research focused on the importance of risk analysis and stated different types of risks involved at each level of software development life cycle. The risks stated above are of different nature and the severity of these risks can be varying from nature of project to other project nature. Sometimes it is very severe and sometimes it is negligible. These risks comprise of the different risks in the different development phases. The most of the risks lie in the most basic development phase i.e. Requirement phase. And the impact of the risks in this phase is the highest. Sometimes these risks directly affect the cost of the project, sometimes it affects the schedule and sometimes it affects some other phase of the development. The process of risk analysis is continuous and applies to many different levels, at once identifying system-level vulnerabilities, assigning probability and impact, and determining reasonable mitigation strategies. By considering the resulting ranked risks, business stakeholders can determine how to manage particular risks and what the most cost-effective controls might be.

REFERENCE

[1] K.L.James (2011), Software Engineering 2nd ed., PHI Learning. | [2] Roger S. Pressman, Software Engineering. | [3] Tahir Abdullah, Ahmed Mateen, Ahsan Raza Sattar, and Tasleem Mustafa, (2010) "Risk Analysis of Various Phases of Software Development Models., vol.420, pp. 369-376. | [4] Denis Verdon, Gary McgrawRisk Analysis in software Design. | [5] J. Viega and G. McGraw (2001), Building Secure Software: How to Avoid Security Problems the Right Way, Addison-Wesley. | [6] Karl G. (2002) "Software Development Risk Management", CSCI 510 Reports, 2002. | [7] Hoodat, H, & Rashidi, H. (2009). "Classification and Analysis of Risks in Software Engineering". World Academy of Science, Engineering & Technology. | [1] K.L.James (2011), Software Engineering 2nd ed., PHI Learning. | [2] Roger S. Pressman, Software Engineering. | [3] Tahir Abdullah, Ahmed Mateen, Ahsan Raza Sattar, and Tasleem Mustafa, (2010) "Risk Analysis of Various Phases of Software Development Models., vol.420, pp. 369-376. | [4] Denis Verdon, Gary McgrawRisk Analysis in software Design. | [5] J. Viega and G. McGraw (2001), Building Secure Software: How to Avoid Security Problems the Right Way, Addison-Wesley. | [6] Karl G. (2002) "Software Development Risk Management", CSCI 510 Reports, 2002. | [7] Hoodat, H, & Rashidi, H. (2009). "Classification and Analysis of Risks in Software Engineering". World Academy of Science, Engineering & Technology. |