ISSN (Online) : 2348 - 2001

International Refereed Journal of Reviews and Research

Volume 3 Issue 2 March 2015

International Manuscript ID : 23482001V3I2032015-08

(Approved and Registered with Govt. of India)

EFFECTIVE MODEL AND ALGORITHMIC APPROACH TOWARDS SOFTWARE RISK MANAGEMENT AGAINST ASSORTED ATTACKS

Ankur Tiwari

Research Scholar

Computer Science and Engineering

Chandigarh University, Gharuan

Punjab, India

Isha Sharma

Assistant Professor

Computer Science and Engineering

Chandigarh University, Gharuan

Punjab, India

ABSTRACT

Software Risk alludes to the potential future mischief that may emerge because of some present activities on programming applications. Hazard administration in software development is identified with the different future risk that could be conceivable on the product because of some minor or non-detectable oversights in programming improvement

ISSN (Online) : 2348 - 2001
International Refereed Journal of Reviews and Research
Volume 3 Issue 2 March 2015
International Manuscript ID : 23482001V3I2032015-08
(Approved and Registered with Govt. of India)

venture or procedure. "Programming activities have a high likelihood of disappointment so viable programming improvement means managing risks enough." Risk administration is the most essential issue included in the product venture advancement. This issue is for the most part overseen by Software Project Management (SPM). Amid the life cycle of programming undertakings, different risks are connected with them. These risks in the product venture is recognized and oversaw by programming risk administration which is a piece of SPM. A portion of the vital parts of risk administration in programming building are programming risk administration, hazard grouping and methodologies for risk administration.

Keywords - Software Risk Management, Software Vulnerability, Software Attacks

FOREWORD

A risk can be defined as a consideration that has some degree of probability of compromising the success of a software development project (Karolak, 1995). Formally, risk can be described as project variables that designate project success (Capers, 1994). Risk defines the probability that the software development project will experience unwanted and inadmissible events, such as termination, delays in project schedule and overrun of project resources. Risk is the possibility of suffering loss that describes the impact on the project which could be in the form of poor quality of the software solution, increased costs, failure, or delayed completion. Software development is specific because risks in software development can be classified in many categories. It is difficult to clearly define risk categories that could apply to a wide area of software development projects. The essence of risk classification is not in precisely defining risk categories, but in identifying and describing as many risks as possible on the project. Due to that, it is suggested that risks on software development projects should be classified according to the main impact areas (Ould, 1996).

ISSN (Online) : 2348 - 2001
International Refereed Journal of Reviews and Research
Volume 3 Issue 2 March 2015
International Manuscript ID : 23482001V3I2032015-08
(Approved and Registered with Govt. of India)

The most important risk impact area is connected to the problems related to new technologies. The majority of software development projects is connected with the application of new technologies. The improper use of new technologies usually leads to project failure. Knowledge is the main problem connected with the use of new technologies (Sertić, 2002). In order to implement new technology, a development team should have sufficient knowledge about it. The importance of knowledge about technologies involved in a software development project is often disregarded. Software solution is built upon technological features, which are often new to a project development team. These technological features are often predicted at the beginning of a software development project, and only a software prototype can prove these predictions (Larman, 2002). The knowledge about these technological features can refine predictions about features and minimize the risk connected with the use of these technologies. A software development team should have sufficient knowledge in order to efficiently exploit technological features. The knowledge about the involved technologies is a precondition for a successful software development project (Ould, 1998).

The second important risk impact area is connected with software requirements. Software requirements are a common term for all users' needs regarding software system functionality and quality of service. It is often very hard to develop the right software solution that absolutely meets users' expectations. In order to develop a software solution according to user expectations, a software development team should discover a whole set of user requirements (Larman, 2002). These requirements, that must guide the entire development process, can be divided into functional and non-functional (Larman, 2002). Functional requirements define behavior of the software solution, while non-functional requirements define qualities and constraints to which the software solution must conform.

ISSN (Online) : 2348 - 2001

International Refereed Journal of Reviews and Research

Volume 3 Issue 2 March 2015

International Manuscript ID : 23482001V3I2032015-08

(Approved and Registered with Govt. of India)

The process of the identification and definition of requirements is long and complicated, so it is common that requirements change during the realization of a software development project (Booch, Rambaugh & Jacobson, 2001). The change of requirements is a major problem in software development because the change of one or few requirements could impact the complete software solution and lead to the failure of a software development project. As a result, it is absolutely necessary to find the right requirements and to manage the change of requirements during a software development project.

The third important risk impact area is the architecture of a software solution. Software architecture could be described as a set of significant decisions about the organization and components of a software solution (Booch, Rambaugh & Jacobson, 2001). Software architecture must be defined in the early development phases in order to build a quality software solution. It is possible that software architecture defined in the early development phases does not satisfy all of the requirements set on a software solution. The software architecture can be verified only with a software prototype, which can be realized only in later development phases. Due to the importance of software architecture, many development processes focus directly on software architecture in order to build a quality software solution according to the defined requirements (Royce, 1998).

The fourth risk impact area is connected with software system performance. In order to satisfy non-functional requirements, a software solution should have satisfactory performance. The performance of a software solution could be tested only on a real and realized software solution. Thus, it is necessary to make predictions about software system performance in the early development phases. These predictions are very important because it is possible to develop a software solution that satisfies functional requirements, but it is too slow to fulfil performance requirements (Karolak, 1995). As a result, development team

ISSN (Online) : 2348 - 2001
International Refereed Journal of Reviews and Research
Volume 3 Issue 2 March 2015
International Manuscript ID : 23482001V3I2032015-08
(Approved and Registered with Govt. of India)

members, with experience in given technologies, should do performance predictions and assumptions.

The fifth risk impact area could be considered as the organizational and non-functional area. The risks connected with this area could be described as organizational problems and problems related to the project resources and schedule. Organizational problems may affect the realization of a software solution since only the efficient organization of software development leads to a successful software development project (Sertić, 2002). A defined project schedule could become a risk because there are many unwanted events that could cause a delay in software realization. It is a management problem to define a project schedule in order to satisfy both customers and developers (Ould, 1996). In order to fulfil defined project deadlines, resources given to the project should be sufficient. This means that every software development project should have enough project members with the right competencies and all the required resources for the planned completion.

Besides risk identification and specification, risks in software development should be addressed and managed. In software development, there are many choices of dealing with risks (Ould, 1998). Based on the risk impact area and type, risks can be avoided, restricted, mitigated and monitored (Jones, 1994). The best possible way of risk addressing is to completely avoid risks. There are a number of risks that can completely be avoided by the use of different technologies, changing requirements or project plans. The next best way to address risks on the project is to confine them. A risk can be confined in order to restrict the risk impact area to affect only a small part of a software development project. In order to reduce the impact area, risks must be identified and some changes on the project should be made regarding the technology and resources used. This is a proper way of addressing risks that cannot be avoided.

ISSN (Online) : 2348 - 2001

International Refereed Journal of Reviews and Research

Volume 3 Issue 2 March 2015

International Manuscript ID : 23482001V3I2032015-08

(Approved and Registered with Govt. of India)

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 (Hoodat, H. & Rashidi, H.).

1. **Risk Index:** Generally risks are categorized into two factors namely impact of risk events and probability of occurrence. 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 (Hoodat, H. & Rashidi, H.).
2. **Risk Analysis:** There are quite different types of risk analysis that can be used. 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 (Hoodat, H. & Rashidi, H.). 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 (Hoodat, H. & Rashidi, H.).
3. **Risk Assessment:** Risk assessment is another important case that integrates risk management and risk analysis. There are many risk assessment methodologies that focus on different types of risks. Risk assessment requires correct explanations of the target system and all security features (Hoodat, H. & Rashidi, H.). It is important that a risk referent levels like performance, cost, support and schedule must be defined properly for risk assessment to be useful.

RISK CLASSIFICATION

ISSN (Online) : 2348 - 2001

International Refereed Journal of Reviews and Research

Volume 3 Issue 2 March 2015

International Manuscript ID : 23482001V3I2032015-08

(Approved and Registered with Govt. of India)

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. A best and most scientific way of approaching risks is to classify them based on risk attributes. Risk classification is considered as an economical way of analyzing risks and their causes by grouping similar risks together into classes (Hoodat, H. & Rashidi, H.). 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 (Hoodat, H. & Rashidi, H.).

RISK VERSUS ISSUE MANAGEMENT

Risk management is the overarching process that encompasses identification, analysis, mitigation planning, mitigation plan implementation, and tracking. Risk management should begin at the earliest stages of program planning and continue throughout the total life-cycle of the program. Additionally, risk management is most effective if it is fully integrated with the program's systems engineering and program management processes—as a driver and a dependency on those processes for root cause and consequence management. A common misconception, and program office practice, concerning risk management is to identify and track issues (vice risks), and then manage the consequences (vice the root causes). This

ISSN (Online) : 2348 - 2001

International Refereed Journal of Reviews and Research

Volume 3 Issue 2 March 2015

International Manuscript ID : 23482001V3I2032015-08

(Approved and Registered with Govt. of India)

practice tends to mask true risks, and it serves to track rather than resolve or mitigate risks. This guide focuses on risk mitigation planning and implementation rather on risk avoidance, transfer or assumption.

THE RISK MANAGEMENT PROCESS MODEL

The risk management process model includes the following key activities, performed on a continuous basis:

- Risk Identification,
- Risk Analysis,
- Risk Mitigation Planning,
- Risk Mitigation Plan Implementation, and
- Risk Tracking.

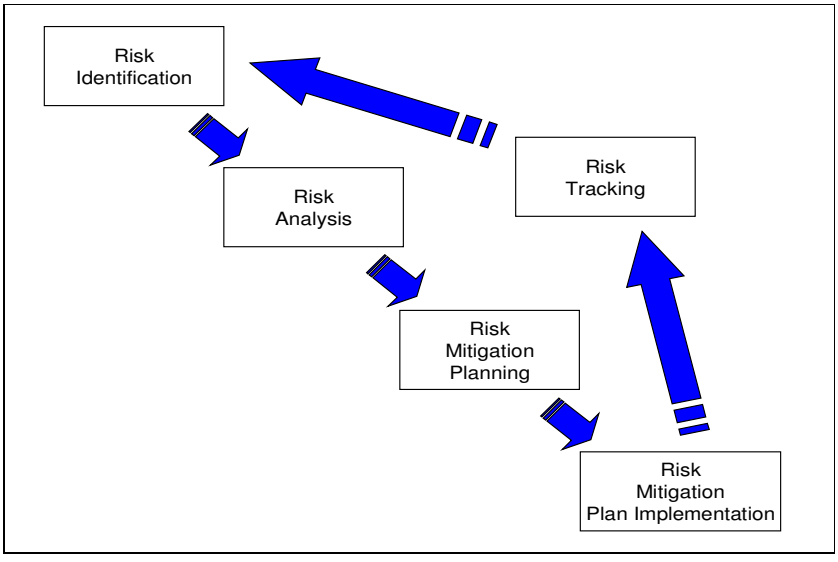

Figure 1 - Risk Management Process

ISSN (Online) : 2348 - 2001

International Refereed Journal of Reviews and Research

Volume 3 Issue 2 March 2015

International Manuscript ID : 23482001V3I2032015-08

(Approved and Registered with Govt. of India)

Acquisition programs run the gamut from simple to complex procurements and support of mature technologies that are relatively inexpensive to state-of-the-art and beyond programs valued in the multibillions of dollars. Effective risk management approaches generally have consistent characteristics and follow common guidelines regardless of program size. Some characteristics of effective risk management approach are discussed below.

STRATEGIES FOR RISK MANAGEMENT

During the software development process various strategies for risk management could be identified and defined according to the amount of risk influence. Based upon the amount of risk influence in software development project, risk strategies could be divided into three classes namely careful, typical, and flexible (Boban, M. et.). Generally, careful risk management strategy is projected for new and inexperienced organizations whose software development projects are connected with new and unproven technology; typical risk management strategy is well-defined as a support for mature organizations with experience in software development projects and used technologies, but whose projects carry a decent number of risks; and flexible risk management strategy is involved in experienced software development organizations whose software development projects are officially defined and based on proven technologies (Boban, M. et.).

The key to risk management is the identification and mitigation of all true risks or the development of a contingency plan in case the potential risk becomes a reality (Charette, 1989). The process of risk management and risk mitigation is connected with preventing huge losses in software development. Risk management should focus on risk reduction and prevention. Software risk management is defined as practice for managing risks that occur in a software development project (Hall, 1998). Risk management should continuously assess

ISSN (Online) : 2348 - 2001
International Refereed Journal of Reviews and Research
Volume 3 Issue 2 March 2015
International Manuscript ID : 23482001V3I2032015-08
(Approved and Registered with Govt. of India)

possible problems on a software development project and define potential risks, determine what risks are important to deal with and implement strategies to deal with those risks. This means that the global project picture is required for successful risk identification, but every project member should assess and identify risks in project areas defined by his role in the project. There are four basic steps in risk definition: identification, assessment, mitigation and conclusion (Capers, 1994).

The first step in risk definition is risk identification, which is responsible for the recognition of potential losses and their causes. In order to implement successful risk management, project team members should have a global perspective on the software development project. Risk assessment should determine the level of exposure to potential loss caused by risk materialization (Jones, 1994). The mitigation step is responsible for the creation of a risk avoidance plan, while the conclusion step describes the execution of risk avoidance and mitigation plans. These steps will lead to a complete description of all risks, which should be captured in a formal document called the Risk List. This document should contain all risks with the description of definition, consequence, likelihood, risk ranking, indicators, risk mitigation strategy and contingency plan for every possible risk on a software development project.

The process of risk management should start with risk identification. The purpose of risk identification is to discover all factors that could lead to project failure (Hall, 1998). These factors are connected with the technology used on the project, software development process and organizational factors. These areas should be observed and assessed in order to capture all of the potential risks. It is necessary to capture details connected with the discovered risk, like risk description, probability of risk occurrence, costs connected with materialized risk and possible risk solutions and avoidance strategies.

ISSN (Online) : 2348 - 2001

International Refereed Journal of Reviews and Research

Volume 3 Issue 2 March 2015

International Manuscript ID : 23482001V3I2032015-08

(Approved and Registered with Govt. of India)

The second step of the risk management process is to assess the level of exposure for each risk. In this step, discovered risks should be ranked in levels according to risk impact (Capers, 1994). Risks should be classified according to the degree of impact in order to choose important risks to be solved first. Risks with a devastating impact should be assessed before risks with a low impact. This is important because risks with a huge impact should be considered in the early development phases, when the costs connected with risk materialization and project failure is smaller than in later development phases (Booch, Rambaugh & Jacobson, 2001).

After risk assessment, risk mitigation is the next step in the risk management process. Risk mitigation is an attempt to avoid or prevent the consequences of risk materialization (Ould, 1998). There are three main strategies of risk mitigation: risk avoidance, risk reduction and risk transfer (Hall, 1998). Risk avoidance is the best possible answer to risk materialization, but there are times when it is very difficult to avoid risks on a software development project. The best way to avoid risks is to completely reorganize the software development project, which is sometimes impossible. If risks cannot be avoided, they can be reduced or transferred. Risk reduction is connected with re-planning the software development project in order to reduce the probability of risk occurrence. Risk transfer could be described as a project reorganization strategy in order to forward the risk to areas where it would cause less damage. Risk mitigation plans should be defined as soon as possible for every identified risk.

The final step in the risk management process is risk conclusion. This step is taken after the definition of a risk mitigation plan and includes all the actions needed for risk avoidance or actions, which are required after the risk materializes. The actions taken in the conclusion step should be defined in the contingency plan. The contingency plan should describe actions, which are taken once a risk becomes a reality.

ISSN (Online) : 2348 - 2001

International Refereed Journal of Reviews and Research

Volume 3 Issue 2 March 2015

International Manuscript ID : 23482001V3I2032015-08

(Approved and Registered with Govt. of India)

The selection of a suitable risk management strategy should be primarily based on organizational experience and the quality of project organization, but software development project complexity, use of new technologies, stability of requirements and competencies of project members should also be considered. We believe that the careful risk management strategy should be used in the initial project phase and that the risk management strategy should be reconsidered and possibly changed to typical or flexible on each project milestone. Proposed strategies are the result of a theoretical study based on real experiences from software development projects at Ericsson Nikola Tesla. Activities defined in the careful risk management strategy are a result of experiences from the IP Telex development project, whose project goal was to build a telecom exchange solution based on a commonly used computer platform. The two other strategies are a theoretical proposition for more mature development projects than the current ones at Ericsson Nikola Tesla Company and should be tested in the future.

CONCLUSION

This paper portrays risks in the software development. Risks are available in every product advancement venture on the grounds that product improvement is taking into account information and new innovations, and the chances for achievement of a product improvement undertaking are firmly associated with effective risk tending to. As an aftereffect of that, we have firmly researched risks and risk sway ranges in software development ventures. With this paper, we propose a key component of advanced programming improvement practices to be programming risk administration. With a specific end goal to accomplish proficient risk administration, we have proposed three risk administration techniques suitable for distinctive programming advancement undertakings as indicated by the measure of risk effect. We have also underlined a risk-based approach to development planning and risk management as an

ISSN (Online) : 2348 - 2001

International Refereed Journal of Reviews and Research

Volume 3 Issue 2 March 2015

International Manuscript ID : 23482001V3I2032015-08

(Approved and Registered with Govt. of India)

attempt to address and retire the highest impact risks as early as possible in the development process. The risk-based approach to software development should enable early risk addressing and conclusion when the expenses connected with risk materialization and project failure are small and insignificant.

REFERENCES

[1] Hall, E. (1998): Managing Risk: Methods for Software System Development, Addison-Wesley, New York

[2] Jones, C. (1994): Assessment and Control of Software Risk, Prentice-Hall, New York

[3] Sertic, H. (2002): Applying Unified process on complex software system development, Master Thesis, Economic Faculty, University of Zagreb, Zagreb

[4] Booch, G.; Rambaugh, J.; Jacobson, I. (2001): The Unified Software Development Process, Addison-Wesley, New York

[5] Karolak, W. (1995): Software Engineering Risk Management, Wiley-IEEE Press, San Francisco

[6] Royce, W. (1998): Software Project Management: A unified framework, Addison-Wesley, New York

[7] Larman, C. (2002): Applying UML and Patterns, Prentice Hall, Upper Saddle River

[8] Robertson, S.; Robertson J. (2001): Mastering the Requirements Process, Addison-Wesley, New York

[9] Capers, J. (1994): Assessment and Control of Software Risks, Prentice Hall PTR, Upper Saddle River NJ

[10] Ould, M. (1996): Strategies for Software Engineering: The Management of Risk and Quality, John Wiley & Sons, San Francisco

ISSN (Online) : 2348 - 2001

International Refereed Journal of Reviews and Research

Volume 3 Issue 2 March 2015

International Manuscript ID : 23482001V3I2032015-08

(Approved and Registered with Govt. of India)

[11] Charette, R. (1989): Software Engineering Risk Analysis and Management, McGraw Hill, New York

[12] Ould, M. (1998): Managing Software Quality and Business Risk, John Wiley & Sons, San Francisco

[13] Hoodat, H., & Rashidi, H. (2009). " Classification and Analysis of Risks in Software Engineering". World Academy of Science, Engineering & Technology, 56446-452. Retrieved from EBSCOhost.

[14] Boban, M., Pozgaj, Z., Sertic, H. " Strategies for successful software development risk management", <www.efst.hr/management/Vol8No2-2003/4-boban-pozgaj-sertic.doc>

[15] "Risk Management" <http://en.wikipedia.org/wiki/Risk_management>

[16] "Software Engineering Risk: Understanding & Management (SERUM)" <http://www.thedacs.com/databases/url/key/270/277/3535>.

[17] Strategies For Successful Software Development Risk Management, Marija Boban, Željka Požgaj, Hrvoje Sertic