RISK STUDY OF VARIOUS SOFTWARE DEVELOPMENT METHODOLOGIES

Vishal Sharma[1], Priyanka Yadav[2], Priti Yadav[3]

[1,2,3] *Computer Science Department, Dronacharya College of Engineering/*

Maharishi Dayanand University, (India)

ABSTRACT

There are different software methodologies that exist in the real world. And there are several factors to choose the methodology that will best fit the software project. One of the important factors is that how risky the project is. And the other factor is the degree to which each methodology will support risk management. The major contributor to project success is to control risk in software projects. In this paper, we will study the risk and risk management in various software development process models which are waterfall model, v-model, incremental development model, spiral model and agile development model so that it can help the project manager to adopt the best methodology for their projects and can improve the Software development process.

Keywords: ***Risk, Risk management, Software development methodology, Software development process model, Waterfall model, Spiral model, Incremental model, V-model, Agile model.***

I. INTRODUCTION

Risk is described as "the possibility of suffering loss that describes the impact on the project which could be in the form of poor quality of software solution, increased costs, failure, or delayed
Completion". We must identify, analyze and control the risks for successful management of a software project. According to Gilb's risk principle,"If you don't actively attack the risks, they will actively attack you". There is a basic component of risk management that are inherited in good project management. There are lot of differences between risk management and project management. Project management are designed to address general risks wherever risk management are designed to focus on risks that are unique to each project. Project management plan for details and risk management plans for
Contingencies. Project management plans for success and risk management plans to manage and mitigate potential cause of failures.

There are following four reasons defined by Boehm for implementing software risk management:

1. Avoiding software project disasters, including run away budgets and schedules, defect-ridden software products, and operational failures.
2. Avoiding rework that is caused by erroneous, missing, or ambiguous requirements, design or code, which typically consumes 40-50% of the total cost of software development.
3. Avoiding overkill with detection and prevention techniques in areas of minimal or no risk.

4. Stimulating a win-win software solution where the customer receives the product they need and the vendor makes the profits they expect.

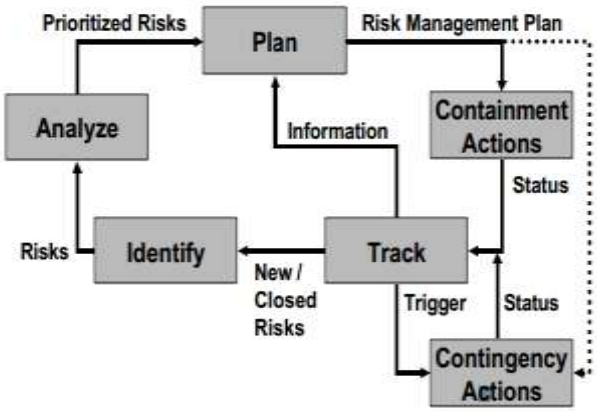

Fig. 1 Risk Management Process

II. ANALYSIS

In this section we will study the various software development methodologies and find the risk management in each of these models.

2.1 Waterfall Model

It was first defined by Winston W. Royce in 1970 and majorly used in software development processes. It is sequential design process in which progress is seen as steadlily flowing downward like waterfall which passes through different phases (Requirement, Design, Implementation, Verification, Maintenance).

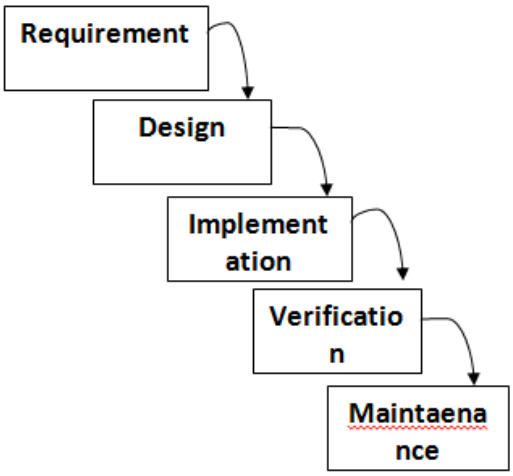

Fig 2 Stages of Waterfall Model

By the development of the waterfall model lot of time is saved during the implementation phase as it capture design errors before software is written, progress is easily measured, total cost of project is accurately estimated, testing becomes easier, user requirement are lucidly defined in advance.

Major Sources of Risk in the Waterfall Model

Risk evolved in the waterfall model are unavoidable.Some of the risk are:

- **Continuous requirements change**
  The major risk factor threatens the waterfall projects is the continuous requirements change during the development process. The waterfall model cannot accommodate with these changes due to its strict structure. The waterfall model requires that all requirements be clearly defined in advance in the requirements stage in order to guarantee that no change could appear later on during the development process.
- **No Overlapping between Stages**
  Another risk that is major in waterfall model is that each stage must be executed completely before proceeding to the other various stages. This led to time inefficient and unreliability.
- **Poor quality assurance**
  Since waterfall model deals with a single phase at a particular time. The quality is not checked at each stage and the single level testing is performed at the later stages hence, the quality of the project developed using waterfall is poor.
- **Relatively long stages**
  Another source of risk in this model resides in the relatively long stages, which makes it difficult to estimate, time, cost, and other resources required to complete each stage successfully. Additionally, in the waterfall model, there is no working product until late in the development process when the product is almost complete and any change is impossible. To make things worse; imagine if the product failed to meet users' expectations.

2.2 Incremental Development

Incremental development is a variant of the waterfall model which consists of a series of waterfall lifecycles wherein the software development project is broken down into smaller segments called increments.
The proposal of the incremental development was to accommodate with risks inherent from implementing the overall software project over a single lifecycle in the pure waterfall model.

Major Sources of Risk in Incremental Development

Following are major risk involved in Incremental Model:-

- **Propagation of bugs through increments**
  The major risk during increment development model is the propagation of undiscovered bugs through the increments to the later stages and it become even more complex to identify and handle the bugs at the later stages.
- **Underestimation of time and other resources required for each increment**
  Since it is required to estimate the time, cost and resources for the each increments which delays the project development. Also the errored estimations may led to the project fall.
- **Time and cost overrun**

Time and cost overrun is a critical factor too. This deadly interrupts the development process. Despite the fact that any interrupt at any point in the incremental development process results in a working system, mostly this system would be an uncompleted system wherein some functionalities are not implemented yet.

2.3 The V-Model

The major risk factors threaten the waterfall model is the poor verification and validation methods, which are restricted to a single testing phase conducted lately in the development process.

Another variant of the waterfall model that came out to deal with this risk is the V-model. The V-model is a testing-focused software development process. It gives equal importance to both development and testing. Its symmetrical shape allows the testing process to start early at the development process, and to be aligned with its different phases.

Moreover, test planning conducted at each stage helps at early identification of project's specific risks and reducing them through an improved process management. process model due to the inflexibility it exhibits against the current evolutionary nature of software projects.

- **User requirements should be precisely defined**

  In V model if user requirements are needed to be altered at later stages then it is not that much possible to make the changes. Hence this model is not flexible.There is direct layer to layer communication but only errors are detected not corrected.

2.4 Spiral Development

The spiral model was posit by Boehm in 1988 as a risk-driven software development process model, where the whole development process is guided by the involved risks. It aims at identifying and evaluating software project risks, and helps in reducing these risks and controlling project cost.

Spiral development supports risk management in software projects in several ways summarized in the following:

- The initial risk analysis aims at:
  - Identifying most risks threaten the project.
  - Classification of these risks under the category of their effect to the project.
  - Evaluate these risks to decide upon the risks to handle through each cycle.
- The risk analysis stage at each cycle that precedes each phase of the waterfall phases in purpose of:
  - Resolving program development start from the initial phase of project.
  - Evaluating the new risks that may arise and alter the project quality.
- The iterative feature of the spiral which allows the development process to go back to the first quadrant at any point in progress which allows:
  - Objectives, alternatives and constraints to change as more attractive alternatives exist.
  - New technology to be incorporated easily during the development process.

- The maximum optimization of project resources usage.
- To deal with poorly done activities in the earlier phases.

- The review conducted at the end of each cycle with main stakeholders as a decision point to avoid the lack of commitment risks during the next cycle.
- Time and cost overrun risks are best managed using spiral development due to the risk analysis stage conducted at each cycle. In this stage, the cost and time required for each cycle are analyzed in advance to give a clear picture about the critical state of the project. This helps the project manager and the developers get more control over these risks.
- Risks related to the increased complexity of the project are also managed using spiral. This is achieved by the partitioning activity conducted at the planning phase.
- Decomposing the project into portions to be developed in parallel spirals obviously reduces time contention related risks, since more work could be achieved during the same interval.

Major Sources of Risk in the Spiral Model

Despite its risk driven nature, spiral has its own sources of risks which are summarized in the following:

- **High reliance on the human factor**
  Since the risk is evaluated at each step of spirat quadrant hence there are experts of each phase to concentrate on the spiral processing. This led to the more emphasize on the human factor. Also the cost and time reliability is low.
- **Detailed risk management process**
  Cost and schedule risks might increase using spiral due to its iterative feature, especially for low risk projects wherein risk assessment is not required to be at this level of granularity.

2.5 Agile Development

In February 2001, 17 software developers (see below) met at the Snowbird, Utah resort, to discuss lightweight development methods. They published the *Manifesto for Agile Software Development* to define the approach now known as agile software development. Some of the manifesto's authors formed the Agile Alliance, a non-profit organization that promotes software development according to the manifesto's values and principles.

Agile software development is a group of software development methods in which requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. It promotes adaptive planning, evolutionary development, early delivery, continuous improvement and encourages rapid and flexible response to change. It is a conceptual framework that focuses on frequently delivering small increments of working software.

The major principal in the Agile development is the informal communication between the stakeholders and the developers. This communication includes the planning, resource allocating and other phases of the project.

Building upon the literature, we can say that there are two contrasting views regarding risk management in the agile context. The first claims that agile is an inherent risk driven approach and implicitly supports risk management by nature. The proponents believe that there is no need to enhance risk management in these

projects. In contrast, the second believes that the risk management state in agile does not differ significantly from other traditional models and that risk management should be enhanced in agile to compensate for the lack of risk management in the agile projects. The advocates to the second view believe in that in some situations the inherent risk management driven nature of the agile is insufficient .

Major Sources of Risk in the Agile Development

In spite of the assertions it makes regarding managing risks, the agile development lacks for any detailed suggestions for managing these risks. Thus, many sources of risks will be left unhandled. The following are the major sources of risk in the agile development:

- **Very large software system**
  The risk management in the Agile development is not suitable for the large and complex projects as this would increase the timespan.
- **Large development team**
  This agile development is not suitable for the large team cosisity large no of members as it is not reliable to make communication effectively between large no of people.
- **High reliance on human factor**
  It relies entirely on the experience of the development team and their abilities to communicate successfully with customers. If the project misses these conditions, then the failure is an inevitable issue.
- **Distributed development environment**
  Since Agile Development requires face to face daily communication hence it is not suitable for distributed environment.
- **Scope creep**

Under this Agile development the developer mostly get disatracted from the main objective of the project.

III. CONCLUSION

This paper focuses on the leading software development process models and explored the state of risk management in each of these models. We conclude that the risk management is deep rooted involved with some software development methodologies. Risk Management must be enhanced in all the software development methodologies as risk is unavoidable in most software development methodologies. The fascinatingaspect for future research is to discovera approach that aims at further improving quality of risk management in the divergent software development methodologies.

REFERENCES

[1] Standish Group, “CHAOS report,” 2009, Boston.

[2] L. Guimares and P. Vilela, “Comparing Software Development Models Using CDM,” Proceedings of The 6th Conference on Information Technology Education, New Jersey, 20-22 October 2005, pp. 339-347.

[3] N. Ruparelia, "Software Development Lifecycle Models," ACM SIGSOFT Software Engineering Notes, Vol. 35, No. 3, 2010, pp. 8-13.

[4] I. Sommerville, "Software process models," ACM Computing Surveys, Vol. 28, No. 1, 1996, pp. 269-271.

[5] B. Shahzad and S. Safvi, "Effective Risk Mitigation: A User Prospective," International Journal of Mathematics and Computers in Simulation, Vol. 2, No. 1, 2008, pp. 70-80.

[6] L. Rodrguez, M. Mora, and F. Alvarez, "A descriptive Comparative Study of the Evolution of Process Models of Software Development Lifecycles (PM-SDLCs)," Proceedings of the Mexican International Conference on Computer Science, 2009, pp. 298–303.

[7] J. Nyfjord and M. Kajko-Mattsson, "Outlining A Model Integrating Risk Management and Agile Software Development," Proceedings of the 34th Euromicro Conference Software Engineering and Advanced Applications, 2008, pp. 476-483.

[8] R. Dash and R. Dash, "Risk assessment techniques for software development," European Journal of Scientific Research, Vol. 42, No. 4, 2010, pp. 629–636.

[9] N. Munassar and A. Govardhan, "A Comparison between Five Models of Software Engineering," International Journal of Computer Science Issues (IJCSI), Vol. 7, No. 5, 2010, pp. 94–101.

[10] W. Royce, "Managing the development of large software systems," IEEE WESCON, 1970, pp. 1-9.

[11] GSAM, "Condensed GSAM Handbook, chapter 2: Software Life Cycle," 2003.

[12] G. Tate and J. Verner, "Case Study of Risk Management, Incremental Development and Evolutionary Prototyping," Information and Software Technology, Vol. 32, No. 3, 1990, pp. 207-214.

[13] B. Boehm, "A Spiral Model of Software Development and Enhancement," Computer, 1988, pp. 61-72.

[14] B. Gotterbarn, "Enhancing risk analysis using software development impact statements," Proceedings of the 26th Annual NASA Goddard Software Engineering Workshop, 2001, pp. 43-51.

[15] V. Szalvay, "An Introduction to Agile Software Development," technical report, Danube Technology, 2004.

[16] J. Miller and J. Grski, "A Method of Software Project Risk Identification and Analysis," Ph.D. Thesis, Faculty of Electronics, Telecommunications and Informatics, Gdansk University Of Technology, 2005.

[17] A. Schmietendorf, E. Dimitrov, and R. Dumke, "Process Models for the Software Development and Performance Engineering Tasks," Proceedings of the 3rd International Workshop on Software and Performance, 2002, pp. 211-218.

[18] F. Nasution and R. Weistroffer, "Documentation in Systems Development a Significant Criterion for Project Success," Proceedings of the 42nd Hawaii International Conference on System Sciences, 2000, pp. 1-9.

[19] S. Murthi, "Preventive Risk Management for Software Projects," IT Professional, Vol. 4, No. 5, 2002, pp. 9-15.