Risk Monitoring

Abstract

Software development is a highly complex and unpredictable activity associated with high risks. With more and more organizations investing substantial resources in software development, risk management becomes crucial.This paper presents the various kinds of risks involved while Software Project development. The different risks need different kinds of strategies to mitigate the risk, in order to have a successful and error-free project.

1. Introduction

1.1. Risk : A dictionary definition of risk is "the possibility of loss or injury". Project risk involves understanding potential problems that might occur on the project and how they might impede project success. Risk management is like a form of insurance; it is an investment.

Software maintenance projects have certain features that make them different from other engineering ones. These include increased complexity and higher project failure rates. To increase the chances of software projects to be successful, it is necessary to identify its risks and monitoring them. Different risks are present in the whole project, even before and after it. In fact, effective risk management is a critical issue in software projects. This will help practitioners to control risks factors in software maintenance projects. Software development is a highly complex and unpredictable activity associated with high risks. With more and more organizations investing substantial resources in software development, risk management becomes crucial.

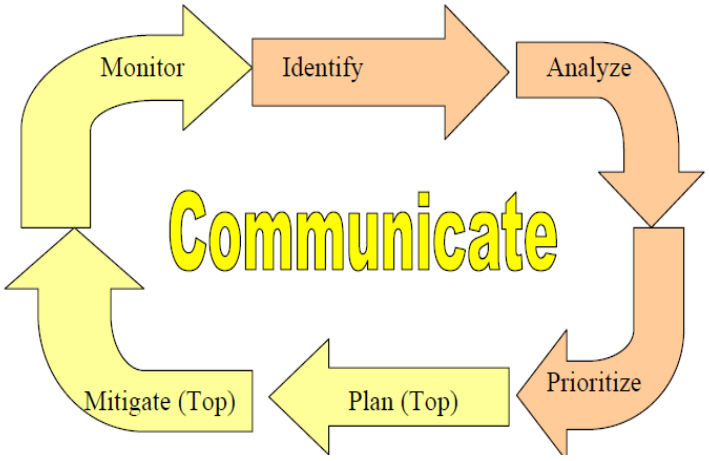

Figure: Risk Management Cycle

Three questions that should be dealt with:

1. What are the most frequently used risk management approaches?
2. What is the process of risk management?

How to apply risk management in practice?

1.2. Software Development Process

A software development process is concerned primarily with the production aspect of software development, as

opposed to the technical aspect, such as software tools. These processes exist primarily for supporting the management of software development, and are generally skewed toward addressing business concerns. Many software development processes can be run in a similar way to general project management processes. Examples are:

- Risk management is the process of measuring or assessing risk and then developing strategies to manage the risk. In general, the strategies employed include transferring the risk to another party, avoiding the risk, reducing the negative effect of the risk, and accepting some or all of the consequences of a particular risk.
- Requirements management is the process of identifying, eliciting, documenting, analyzing, tracing, prioritizing and agreeing on requirements and then controlling change and communicating to relevant stakeholders. New or altered computer system . Requirements management, which includes Requirements analysis, is an important part of the software engineering process; whereby business analysts or software developers identify the needs or requirements of a client; having identified these requirements they are then in a position to design a solution.
- Change management is the process of identifying, documenting, analyzing, prioritizing and agreeing on changes to scope (project management) and then controlling changes and communicating to relevant stakeholders. Change impact analysis of new or altered scope, which includes Requirements analysis at the change level, is an important part of the software engineering process; whereby business analysts or software developers identify the altered needs or requirements of a client; having identified these requirements they are then in a position to re-design or modify a solution. Theoretically, each change can impact the timeline and budget of a software project, and therefore by definition must include risk-benefit analysis before approval.
- Software configuration management is the process of identifying, and documenting the scope itself, which is the software product underway, including all sub-products and changes and enabling communication of these to relevant stakeholders. In general, the processes employed include version control, naming convention (programming), and software archival agreements.
- Release management is the process of identifying, documenting, prioritizing and agreeing on releases of software and then controlling the release schedule and communicating to relevant stakeholders. Most software projects have access to three software environments to which software can be released; Development, Test, and Production. In very large projects, where distributed teams need to integrate their work before releasing to users, there will often be more environments for testing, called unit

testing, system testing, or integration testing, before release to User acceptance testing (UAT).

- A subset of release management that is gaining more and more attention is Data Management, as obviously the users can only test based on data that they know, and "real" data is only in the software environment called "production". In order to test their work, programmers must therefore also often create "dummy data" or "data stubs". Traditionally, older versions of a production system were once used for this purpose, but as companies rely more and more on outside contributors for software development, company data may not be released to development teams. In complex environments, datasets may be created that are then migrated across test environments according to a test release schedule, much like the overall software release schedule.

1.3. Project Planning, Monitoring and Control

The purpose of project planning is to identify the scope of the project, estimate the work involved, and create a project schedule. Project planning begins with requirements that define the software to be developed. The project plan is then developed to describe the tasks that will lead to completion.

The purpose of project monitoring and control is to keep the team and management up to date on the project's progress. If the project deviates from the plan, then the project manager can take action to correct the problem. Project monitoring and control involves status meetings to gather status from the team. When changes need to be made, change control is used to keep the products up to date.

1.3.1. Issues

In computing, the term **issue** is a unit of work to accomplish an improvement in a system. An issue could be a bug, a requested feature, task, missing documentation, and so forth. The word "issue" should not be misunderstood as just a synonym for "problem," as in other English usage.

Problems occur from time to time and fixing them in a timely fashion is essential to achieve correctness of a system and avoid delayed deliveries of products.

1.3.2. Severity levels

Issues are often categorized in terms of **severity levels**. Different companies have different definitions of severities, but some of the most common ones are:

High

The bug or issue affects a crucial part of a system, and must be fixed in order for it to resume normal operation.

Medium

The bug or issue affects a minor part of a system, but has some impact on its operation. This severity level is assigned when a non-central requirement of a system is affected.

Low

The bug or issue affects a minor part of a system, and has very little impact on its operation. This severity level is assigned when a non-central requirement of a system (and with lower importance) is affected.

Cosmetic

The system works correctly, but the appearance does not match the expected

one. For example: wrong colors, too much or too little spacing between contents, incorrect font sizes, typos, etc. This is the lowest severity issue.

1.4.Reasons of Project Failure:

Software project management failures has shown that the following are the most common causes:

1. Unrealistic or unarticulated project goals
2. Inaccurate estimates of needed resources
3. Badly defined system requirements
4. Poor reporting of the project's status
5. Unmanaged risks
6. Poor communication among customers, developers, and users
7. Use of immature technology
8. Inability to handle the project's complexity
9. Sloppy development practices
10. Poor project management
11. Stakeholder politics
12. Commercial pressures

1.4.1.Issues

The main issues involved in a project include:

- Decreasing the complexity
- Decreasing the risks involved

1.5. Types of Risk

1.5.1. Schedule Risk

Project schedule get slip when project tasks and schedule release risks are not addressed properly. Schedule risks mainly affect on project and finally on company economy and may lead to project failure.

Schedules often slip due to following reasons:

- Wrong time estimation
- Resources are not tracked properly. All resources like staff, systems, skills of individuals etc.
- Failure to identify complex functionalities and time required to develop those functionalities.
- Unexpected project scope expansions.

1.5.2. Budget Risk:

- Wrong budget estimation.
- Cost overruns
- Project scope expansion

1.5.3. Operational Risk:

Risks of loss due to improper process implementation, failed system or some external events risks.

Causes of Operational risks:

- Failure to address priority conflicts
- Failure to resolve the responsibilities
- Insufficient resources
- No proper subject training
- No resource planning
- No communication in team.

1.5.4. Technical Risk:

Technical risks generally leads to failure of functionality and performance.

Causes of technical risks are:

- Continuous changing requirements
- No advanced technology available or the existing technology is in initial stages.
- Product is complex to implement.
- Difficult project modules integration.

1.5.5. Programmatic Risk:

These are the external risks beyond the operational limits. These are all uncertain risks are outside the control of the

program.
These external events can be:

- Running out of fund.
- Market development
- Changing customer product strategy and priority
- Government rule changes. **[5]**

1.6. Process of Risk Monitoring

1.Risk Identification: Risk identification is the process of understanding what potential unsatisfactory outcomes are associated with a particular project
Several risk identification tools and techniques include

- Brainstorming
- The Delphi technique
- Interviewing
- SWOT analysis (strengths, weaknesses, opportunities, and threats)

2.Risk Analysis: Assess the likelihood and impact of identified risks to determine their magnitude and priority. Risk quantification tools and techniques include :

- Probability/Impact matrixes
- The Top 10 Risk Item Tracking technique
- Expert judgment

3. Risk Response Control: Risk response control involves executing the risk management processes and the risk management plan to respond to risk events. Risks must be monitored based on defined milestones and decisions made regarding risks and mitigation strategies. Sometimes workarounds or unplanned responses to risk events are needed when there are no contingency plans. After identifying and quantifying risks, you must decide how to respond to them. Four main response strategies for negative risks:

- Risk avoidance
- Risk acceptance
- Risk transference
- Risk mitigation

4. Risk Monitoring and Control

- Monitoring risks involves knowing the status of risk.
- Controlling risks involves carrying out the risk management plans as risks occur.
- Workarounds are unplanned responses to risk events that must be done when there are no contingency plans.
- The main outputs of risk monitoring and control are corrective action, project change requests, and updates to other plans.

1.7. Project Risk Management.

“A little risk management saves a lot of fan cleaning”. A disciplined, though simple, approach to risk management can reduce crisis management and the clean ups that result. It will also increase the chance of project success and probably reduce the project manager's stress level into the bargain. The process is summarised under these five headings: Measure, Minimise, Mention, Monitor and Modify, which you will notice make the easily remembered acronym 'mmmmm'.

1. Measure

The project manager needs to measure, assess, understand:

- the risk that the project will exceed the budget he is thinking of committing to
- the risk that the project will miss any dates he has in mind
- the risk that the project will fail to meet any other commitments he is about to make.

Many organisations have checklists - things that have gone wrong in previous similar projects - which are therefore things that might cause problems in future projects.
A better approach might be to invite the team and other stakeholders to a Risk Identification Workshop. Brainstorm: "team, what do we think could cause us problems, or even cause us to fail?" When that has been exhausted, zip through any checklists you've managed to lay your hands on to

- Prioritise the risks, listing first those which would cause major problems and are most likely to happen. These high impact, high probability risks will clearly need most attention. You may well decide to ignore low probability low impact risks to avoid cluttering up your risk management process.

2. Minimise
Do something about it! There's no point in assessing risks if you don't then take action to reduce them.
For example: define that user's role in writing, get a written commitment from his boss that he will be available and empowered to make decisions on behalf of his department. Line up backfills for people you think might leave the team during the project. Increase the budget to include contingency tasks for risks that you can't eliminate at the start, i.e. tasks that describe what you'll do if the risk bites you during the project. Now, if you are an experienced PM you'll know how to deal with most risks. If you're a novice you may have no idea how to reduce the risk that, for example, too much change to the Requirements during the project might cause you to miss the proposed end date.
Some companies record how projects deal successfully with risks and make this information available to future project managers. Getting a list of the dozen or so ways PMs have reduced that change risk in the past could give you some very useful pointers as to how you might reduce the risk in your project.

3. Mention
Who owns the project risk, who ultimately is taking the risk? The Project Sponsor. But many sponsors have no idea what the risks might be, particularly if they are sponsoring technical, e.g. IT, projects - and why should they? Project managers have a duty to explain the risks to the sponsor before the sponsor gives the go ahead. (Image you don't mention the risks and when it's all going wrong you tell the sponsor that you knew all along it was high risk - you're going to get shot and quite rightly too.)
But there are good ways and bad ways of presenting risks to sponsors. Do not go in with 50 overheads listing hundreds of risks. No. The way to do it is briefly to explain the process you have used to identify and analyse the risks and then to describe the major risks you initially foresaw. Then explain what you have already done to eliminate or reduce some or most of these major risks. This builds real credibility for the next bit where you tell the sponsor about the high risks that remain, the likelihood of their coming to fruition and the consequence if they do. If you were sponsor and you're told "this risk is almost certain to happen and when it does your £5M project will be a total write off", would you give the project the go ahead? Probably not, unless the benefits are measured in billions and it's worth the gamble. But the sponsor may accept the position that if a particular risk should happen it would delay the end date by a month - although he will of course exhort you to meet the end date in spite of the risk.

4. Monitor
But that isn't the end of it. Far from it. Clearly we must make sure we monitor and manage risks during the project to make it less likely that they will happen and to minimise the impact if they do.

Create a Risk Register which lists risks in priority order. For each risk, the register might include:

- Risk number
- Description
- Consequence if risk happens
- Probability (high, medium or low)
- Planned actions to mitigate the risk
- Contingency plan (what you'll do if the risk happens)
- Risk owner (a member of the project team)
- Status (e.g. closed: no longer a risk)

At weekly team meetings risk owners report, briefly, on the status of their risks. What planned (or unplanned) actions are being taken to reduce the risk. Is the risk receding or growing? This should ensure that the team is involved in risk reduction activities and in refining the risk management plan. Keep the register updated - relegate receding risks, promote growing ones.

New risks can always crawl out of the woodwork as you go along - if they are significant add them to the register.

Each month the project manager should report to the sponsor on the status of key risks. With any luck you'll be reporting your success in dealing with the risks, but if risks are growing you should obviously alert the sponsor to that.

5. Modify

At the end of each stage of your project hold a post mortem. Look at the risks you identified at the outset. Record what you did successfully to deal with each risk, what you tried that did not work, and what you would do next time with the benefit of hindsight. These ways of addressing risks will be useful not only to you and your next project, but also to other project managers.

Some organisations have a Project Support or Project Assurance or similarly named project management 'centre of competence'. If such exists in your organisation, send your list of addressing factors to them. When a future project manager is starting a project and has identified a risk he has no idea how to mitigate, he can call up Project Support and ask "what has worked well in the past?"

Ideally Project Support will be proactive in modifying the organisation's risk checklists. They will add new risks that are actually being experienced and delete risks that are no longer applicable within the organisation. Ideally they would also seek out successful strategies for addressing risks and proactively feed these forward to future projects. **[2]**

1.8. Risk Mitigation

Risk mitigation planning is the process of developing options and actions to enhance opportunities and reduce threats to project objectives. Risk mitigation implementation is the process of executing risk mitigation actions. Risk mitigation progress monitoring includes tracking identified risks, identifying new risks, and evaluating risk process effectiveness throughout the project.

Risk mitigation handling options include:

- Assume/Accept: Acknowledge the existence of a particular risk, and make a deliberate decision to accept it without engaging in special efforts to control it. Approval of project or program leaders is required.
- Avoid: Adjust program requirements or constraints to eliminate or reduce the risk. This adjustment could be accommodated by a change in funding, schedule, or technical requirements.
- Control: Implement actions to minimize the impact or likelihood of the risk.
- Transfer: Reassign organizational accountability, responsibility, and

authority to another stakeholder willing to accept the risk.

- Watch/Monitor: Monitor the environment for changes that affect the nature and/or the impact of the risk.

Each of these options requires developing a plan that is implemented and monitored for effectiveness. More information on handling options is discussed under best practices and lessons learned below.

From a systems engineering perspective, common methods of risk reduction or mitigation with identified program risks include the following, listed in order of increasing seriousness of the risk :

1. Intensified technical and management reviews of the engineering process
2. Special oversight of designated component engineering
3. Special analysis and testing of critical design items
4. Rapid prototyping and test feedback
5. Consideration of relieving critical design requirements
6. Initiation of fallback parallel development. **[3]**

1.9. A continuous process of risk management

Software development is a system. There are so many factors affect software development project. The process of software development is dynamic and involves many evolutionary and revolutionary changes.

1.9.1.Basic concept of continuous process of risk management

The paradigm of risk management proposed by Software Engineering Institute (SEI) is very similar to the six steps proposed by Boehm (1991). Figure 9 shows the paradigm from SEI. This paradigm illustrates a set of functions that are identified as continuous activities throughout the life cycle of a project.

Function	Description
Identify	Search for and locate risks before they become problems.
Analyze	Transform risk data into decision-making information. Evaluate impact, probability, and timeframe, classify risks, and prioritize risks.
Plan	Translate risk information into decisions and actions (both present and future) and implement those actions.
Track	Monitor risk indicators and mitigation actions.
Control	Correct for deviations from the risk mitigation plans.
Communicate	Provide information and feedback internal and external to the project on the risk activities, current risks, and emerging risks. **Note**: Communication happens throughout all the functions of risk management.

Figure : Adapted from SEI

Continuous risk management can be defined as a software engineering practice with **process**, **methods,** and **tools** for managing risks in a project. It provides an environment for decision makers to assess continuously what can go wrong, to determine which risks are most important, and to implement strategies to deal with these risks.

1.9.2.Benefits and costs of continuous risk management

Implementing continuous risk management will give organizations benefits. Simultaneously, some costs will occur. The benefits of continuous risk management include preventing problems before they occur, improving product quality, enabling better use of resources, and promoting teamwork. The costs that will occur during continuous risk management include infrastructure costs, risk management cost, mitigation costs.

Cost-benefit value is difficult to determine when some costs and benefits cannot be quantified. Thus, the project managers need to balance the costs against the expected benefits and the cost of not doing risk management.

1.10. A practical model for software risk management

The action research to develop risk management approach for software performance improvement is used.

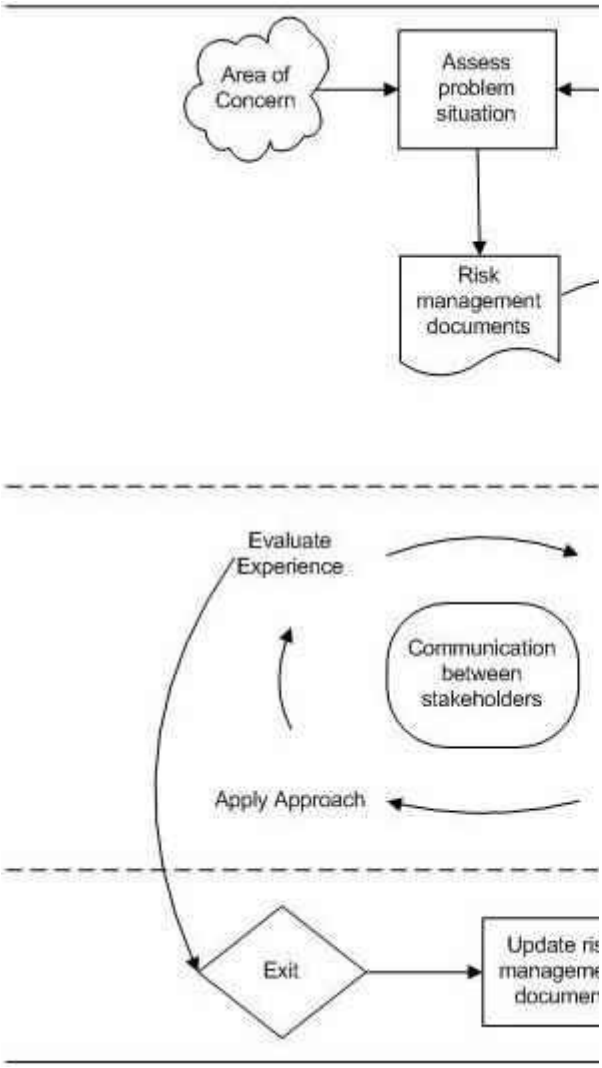

Figure : A practical model for software risk management

Identify three major stages for developing risk approaches for software performance improvement, including initial stage, iterating stage, and closing stage.

1.10.1. Initialing stage

In the initial stage, stakeholders assess the problem situation based on the area of concern such as users with negative attitudes, system requirements not adequately identified, project involved the use of new technology, lack of an effective project management methodology, team members lack specialized skills required by the project, employees turnover, organization undergoing restructuring during the project. Goals of risk management is another factor should be considered when assessing the problem situation. Goals of risk management usually refer to the success of software development . After evaluating the problem situation, project manager and other stakeholders need to select a risk approach based on existing risk documents. If organization use risk-action list more often and have extensive documents, they can consider to utilize the previous method. This doesn't mean that they have to follow the previous risk management approach. They can explore new approach based on the context and the project characteristics.

1.10.2. Iterating stage

In this stage, it is a continuous process including a repeating set of activities. The first iteration is to develop a risk framework. For instance, you select risk list approach. You need to identify the risk items in it, and also think about how to avoid, mitigate or transfer this risk items. The second iteration is to design the risk process. This iteration can include reformulating risk items and introducing a first step in which the software development team should interpret the risk model in their particular context. The third iteration is to apply the risk approach based on the first two iterations and the lessons learned from first two iterations. The iteration step in this stage is to evaluate experience and then the risk approach was used again without any changes. The core of iterating stage is

communication between stakeholders. Through the communication, software development team and project managers can get feedbacks from internal and external, in turn can improve and refine risk management approach.

1.10.3. Closing stage

Two situations can exit software risk management cycle. One is that software development project is finished. The other one is that software development project is stopped in the middle way. Both two situations can provide valuable experience to software risk management since organizational learning is a dual loop. In this stage, we need to update the risk management documents either from successful experience or failed experience which can help organization better manage software risk management in the future. **[4]**

2.Literature Survey

2.1.Hooman Hoodat, and Hassan Rashidi "Classification and Analysis of Risks in Software Engineering"

Despite various methods that exist in software risk management, software projects have a high rate of failure. When complexity and size of the projects are increased, managing software development becomes more difficult. In these projects the need for more analysis and risk assessment is vital. In this paper, a classification for software risks is specified. Then relations between these risks using risk tree structure are presented. Analysis and assessment of these risks are done using probabilistic calculations. This analysis helps qualitative and quantitative assessment of risk of failure. Moreover it can help software risk management process. This classification and risk tree structure can apply to some software tools.

2.2.Robert Stern, José Carlos Arias "*Review of Risk Management Methods"*

Project development, especially in the software related field, due to its complex nature, could often encounter many unanticipated problems, resulting in projects falling behind on deadlines, exceeding budgets and result in sub-standard products. Although these problems cannot be totally eliminated, they can however be controlled by applying Risk Management methods. This can help to deal with problems before they occur. Organisations who implement risk management procedures and techniques will have greater control over the overall management of the project. By analysing five of the most commonly used methods of risk management, conclusions will be drawn regarding the effectiveness of each method. The origin of each method will be established, along with the typical areas of application, the framework of the methods, techniques used by each and the advantages and disadvantages of each of the methods. Each method will be summarised, then an overall comparison will be drawn. Suitable references will be included to highlight features, along with diagrams and charts to illustrate differences in each approach.

2.3.Mike Harding Roberts "A Straightforward Approach to Project Risk Management "

Managing risk is not the same as being afraid to take risk. On the contrary, doing very high risk projects is sometimes the right thing to do. But if the project is very high risk everyone must be aware of that and the project must be wrapped in cotton wool to make it succeed in spite of all the risks. And with appropriate management attention, contingency and whatever, i.e. with proper risk management, very high risks projects can be perfectly successful.

2.4.Aihua Yan, "Risk Management in Software Development: A Continuous Process", 2008

Software development is a highly complex and unpredictable activity associated with high risks. With more and more organizations investing substantial resources in software development, risk management becomes crucial. The present paper tries to ask these three questions: what are the most frequently used risk management approaches? What is the process of risk management? How to apply risk management in practice? An analysis of previous literatures identifies four types of risk management approach: risk-list, risk-action list, risk-strategy model, and risk-strategy analysis. A continuous process is introduced based on the dynamic system theory. A framework for applying risk management approach in practice is also proposed from action research perspective. Finally, the relevant practical implications for software development project are discussed.

2.5. Sunil Sapoka "Risk Management in Software Engineering" , 2011

Software risk management, risks classification, and strategies for risk management are clearly described in this paper. If risk management process is in place for each and every software development process then future problems could be minimized or completely eradicated. Hence, understanding various factors under risk management process and focusing on risk management strategies explained above could help in building risk free products in future.

2.6. DR. E. WALLMÜLLER "Risk Management for IT and Software Projects"

Risk Management can be defined as a systematic process for identifying, analyzing and controlling risks in projects or organizations. Definitions and illustrations of risks are given especially by a list of ten risk factors, which occur most frequently in IT and Software projects. For complex, high-risk projects it is very useful to implement a formal risk-management process, supported by effective methods in the individual process steps. As variants, risk-management processes according to Barry Boehm, Ernest Wallmüller and Jyrki Kontio are presented. The importance of a sound operational preparation of each step of the risk-management process is emphasised and illustrated by examples.

2.7. Cristina López' , Jose L. Salmeron "Monitoring Software Maintenance Project Risks", 2012

Software maintenance projects have certain features that make them different from other engineering ones. These include increased complexity and higher project failure rates. To increase the chances of software projects to be successful, it is necessary to identify its risks and monitoring them. Different risks are present in the whole project, even before and after it. In fact, effective risk management is a critical issue in software projects. In this line, this research proposes the use of a framework based on IEEE 1074. This will help practitioners to control risks factors in software maintenance projects.

2.8. Ville Ylimannela "A MODEL FOR RISK MANAGEMENT IN AGILE SOFTWARE DEVELOPMENT"

This paper researches risk management in agile software development. In traditional

waterfall-model risks were usually managed by using project risk management frameworks. Nowadays agile methods have started replacing the traditional models. One of the reasons was old models inability to respond constantly changing business requirements. Agile development is based on short iteration cycles, which allow them to respond to changes in business environment. Using agile development is itself risk management at project level. Problems started arising, when people tried to merge the old school heavy project risk management models with agile models. One of the key principles in agile development is to avoid unnecessary bureaucracy and documentation. The traditional risk management is heavily centered around documentation. Two companies were interviewed during the making of this paper. The goal was to address the issues which arose during the interviews.

Conclusion

Project risk management is the art and science of identifying, analyzing, and responding to risk throughout the life of a project and in the best interests of meeting project objectives.

Main processes include:

- Risk identification
- Qualitative risk analysis
- Risk response planning
- Risk monitoring and control

References

[1] www.wikipedia.com

[2] Mike Harding Roberts "A Straightforward Approach to Project Risk Management "

[3] Kossiakoff, A. and W.N. Sweet, , "Systems Engineering Principles and Practice", 2003.

[4] Aihua Yan, "Risk Management in Software Development: A Continuous Process", 2008

[5] Hooman Hoodat, and Hassan Rashidi "Classification and Analysis of Risks in Software Engineering"