International Journal of Business, Human and Social Sciences
ISSN: 2517-9411
Vol:4, No:4, 2010

A New Dimension in Software Risk Managment

Masood Uzzafer

***Abstract*—**A dynamic risk management framework for software projects is presented. Currently available software risk management frameworks and risk assessment models are static in nature and lacks feedback capability. Such risk management frameworks are not capable of providing the risk assessment of futuristic changes in risk events. A dynamic risk management framework for software project is needed that provides futuristic assessment of risk events.

***Keywords*—**Software Risk Management, Dynamic Models, Software Project Managment.

I. Introduction

As the complexity of the software grows exponentially so as the need for better software management processes grew and areas which were not explored before are being taped to search for new management models. Risk management is certainly one such area in software engineering. Risk management has been used in different fields for many years, since last decade its application in software projects has gained momentum. Risk management in software engineering ensures that risk events of a software project are known and their severity of impact on different software project parameters is understood to make sure that the software project is completed within the projected schedule, budget and quality.

Risk managements roots are based on probability and uncertainty theory when Bernoulli presented expected utility theory [1], and discussed how people choose from different alternatives based on their expected utility. Different researchers adopted different expressions of risk that suites their fields of interest but in essence researchers refer risk as the possibility of loss which is the literal meaning of risk. In practice the risk management is about managing the uncertainties associated around certain events, and undertaking decisions to reduce the impact of undesirable events on a process.

The term risk management framework refers to the entire risk management process which consists of many sub-processes for example risk identification, risk analysis, risk assessment, risk control etc. Risk assessment is one of the sub-processes of risk management framework that represents the quantitative risk analysis process. It involves quantitative analysis of the impact of a risk event that is known as risk exposure or risk factor.

Software engineering researchers have contributed different software risk management frameworks and risk assessment models, few of these frameworks and models gained widespread attention and are briefly discussed.

Masood Uzzafer is a PhD candidate with University of Nottingham, Nottingham, NG8 1BB, UK. e-mail: keyx8muz@nottingham.edu.my.

Dynamic risk management has been used in the fields of finance where dynamic risk management models are called simulation models [7], in these models different financial strategies are simulated for a specific financial scenario. A financial scenario defines a setup, environment or parameters that may be affected by a financial strategy. A financial strategy is a set of decisions to achieve desirable results for a given financial scenario by combating risk events. Financial firms use dynamic risk management models to understand the impact of various management strategies under variety of futuristic scenarios, where each proposed strategy is associated with a set of desired outcomes. Hence the most optimum strategy is selected for a specific scenario, a strategy may work well in one scenario may not work in another scenario. We will redefine the terms scenario and strategy from software project perspective and define a dynamic risk management framework for software projects.

This paper is organized in this pattern, section II analyzes the currently practiced software risk management frameworks; section III dynamic risk management and finally some conclusions are drawn in section IV.

II. Software Risk management Frameworks

Barry Boehm proposed a model for software risk management framework [2] that was the first complete software risk management framework developed. The framework splits software risk management into primary and secondary risk management steps. It consists two primary steps with each further divided into three subsidiary steps. The first primary step is called risk assessment that involves subsidiary steps of risk identification, risk analysis, risk prioritization. The second primary step is called risk control that involves subsidiary steps of risk management planning, risk resolution and risk monitoring. The risk management process flows from first primary step to its subsidiary steps then to the second primary step to its subsidiary steps as illustrated in Figure -1.

This model uses decision tree for the risk assessment risk events. Each branch of the decision tree is assigned a probability of loss, where loss is represented as the risk exposure that is the product of the probability of loss and the magnitude of loss.

Software Engineering Institute (SEI) presented a six phase software risk management framework [3]. These phases are identify, analyze, plan, track, control and communication, these phases occur sequentially. The SEI risk management framework concurrently allows new risk events to be identified and analyzed, this way the risk management activity occurs continuously and iteratively throughout the project lifetime as shown in Figure -2.

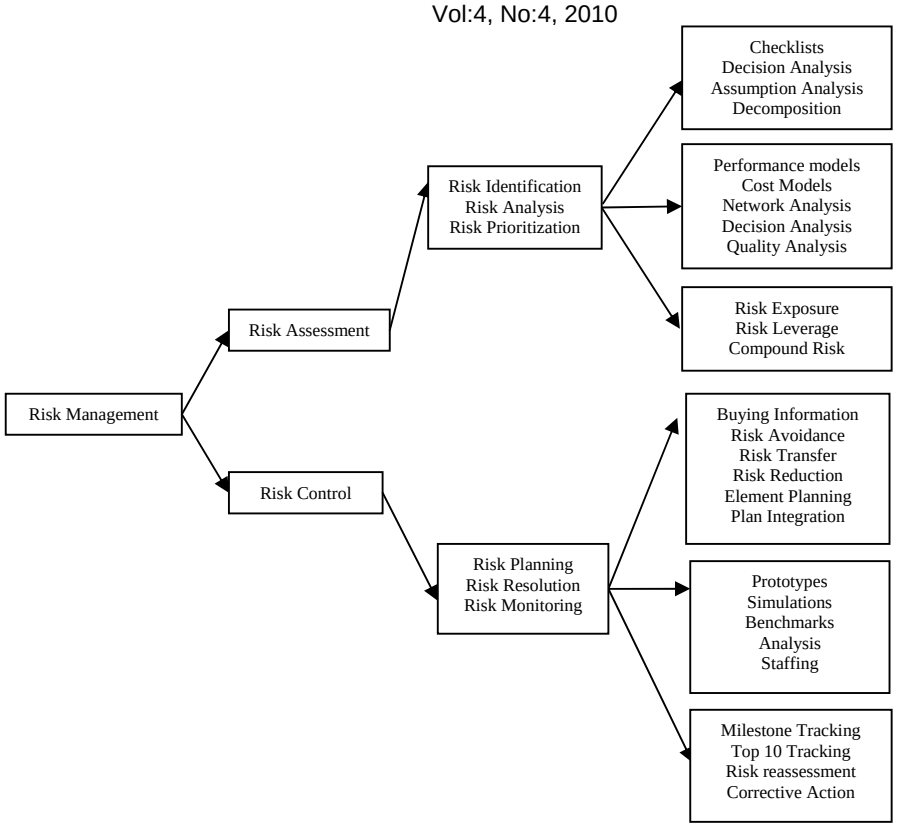

Fig. 1: Barry Boehm's Risk Management Steps

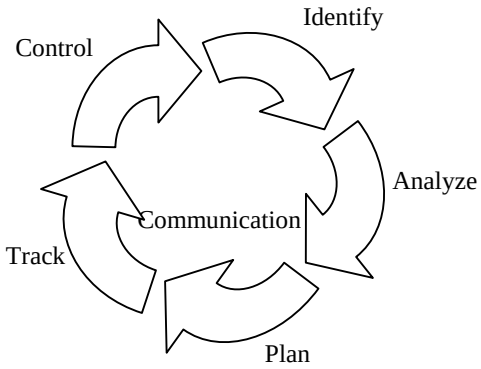

Fig. 2: SEI Six Phase Risk Management Framework

Another celebrated software risk management framework is Capability Maturity Model Integration (CMMI) Risk Management Guidelines [4]. CMMI divides risk management framework into three categories, defining a risk management strategy, identifying and analyzing risk and implementing risk mitigation plan for the identified risks, as shown in Table -1. Each one of these categories has multiple guidelines to help practitioners during the risk management planning. Risk assessment is performed in CMMI category two where each risk is evaluated and assigned values based on the impact and probability of risk event. CMMI requires that CMM level 4 companies must have risk management process implemented.

TABLE I CMMI RISK MANAGEMENT FRAMEWORK

CMMI Risk Management Plan – Maturity Level 3		
1: Prepare For Risk Management	2: Identify and Analyze Risks	3: Mitigate Risks
A: Determine Risk Sources and Categories	A: Identify Risks	A: Develop Risk Mitigation Plans
B: Define Risk Parameters	B: Evaluate, Categorize, and prioritize Risks	B: Implement Risk Mitigation Plans
C: Establish Risk Management Strategy		

Project Management Institute (PMI) describes risk management as "increase the probability of impact of positive events and decrease the probability and impact adverse to the project" [5]. It further describes that project risk is an uncertain event or condition that if occurs has a positive or negative effect on at least one project objective such as time, cost, scope or quality. It defines a sequential framework for project risk management which is equally applicable to software projects. This sequential framework consists off risk management planning, risk identification, qualitative and

quantitative risk analysis, risk response planning and risk monitoring and control, as shown in Figure -3.

PMI risk management framework further explains the flow of risk management sequences where all the activities occur in the sequence as described. Risk assessment is done in the sequence of quantitative risk analysis, which involves determining the impact of a risk event together with its probability distribution, which helps to determine the risk factor impact with a certain probability. PMI risk management framework further suggests the use of decision tree analysis to find the optimum decision based on the calculated risk assessment.

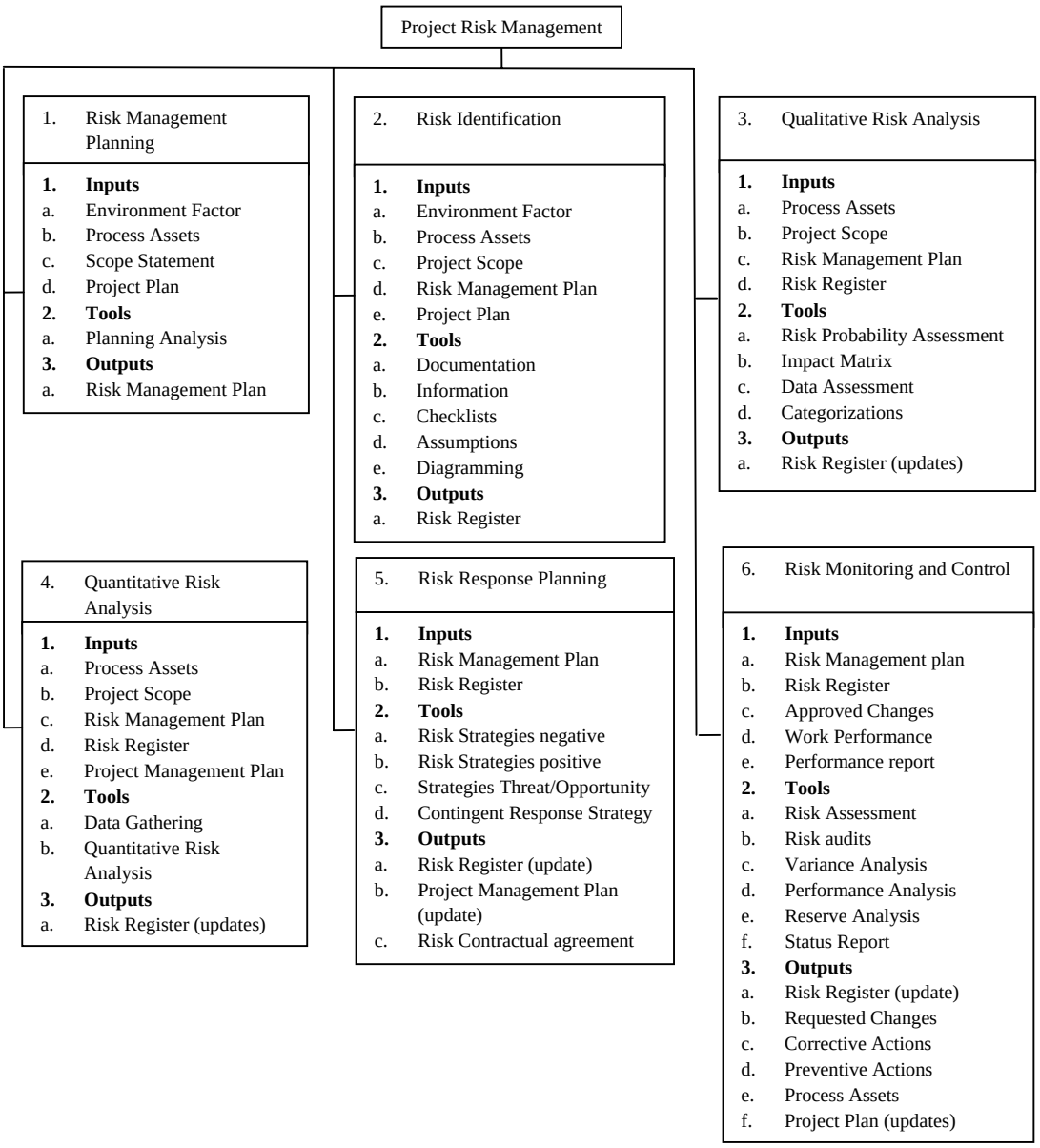

Fig. 3: PMI Risk Management Framework

All of the discussed risk management frameworks provide a foundation for software practitioners to manage software project risk using any of these models. These models describe in detail the risk management and assessment models and how to combat risk events to save the project from producing undesirable results. Hence any of these models could be useful if implemented according to the guidelines defined in these models. As new tools and techniques are tapped this opens an opportunity to improve upon to improve upon the available risk management framework and assessment models, one such tool is dynamic structure with feedback capability.

III. Dynamic Risk Management

Risk management frameworks discussed possess a static behavior hence are not capable of capturing the futuristic variations in the project parameters due to risk events. Further static risk management framework operates without any

feedback capability and lacks the dynamics of decision adjustments based on a futuristic risk assessment of risk events. Static risk management frameworks assume that the decisions are made one time only and are irreversible, treating decisions as one time only and irreversible limits the capability of risk assessment and hence recognizes the value of a framework which is dynamic in nature and capable of capturing the variations due to the changing environment [9].

Figure -4 illustrates a risk assessment example based on a static risk management framework. Risk assessment is performed at the beginning of the project and decisions are based on that assessment. Such framework is unable to accommodate the fact that risk impact and its probabilities could change and hence require risk re-assessment and eventually require decisions to be adjusted accordingly.

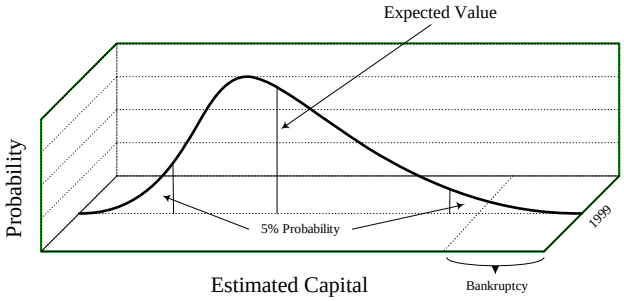

Fig. 4: Static Risk Exposure View

Risk impact factor varies as project moves through different phases of completion and the probability of impact of risk events change causing all the perceived assumptions and estimations to be updated accordingly. A static risk management framework describes how an enterprise should manage and reduce risk by deploying strategies based on the risk assessment taken at the beginning of the project. Although these models provide a rich insight as to how enterprises should manage risk lacks prediction capabilities of the nature and behavior of risk events in the future. It is vital for decision makers to understand the futuristic changes and to dynamically adjust the decisions.

A dynamic risk management framework with a dynamic risk assessment model is the way of moving forward for risk management in today's changing enterprise and market environment. Due to the fast paced environment of the software industry, where many parameters change rapidly, the use of dynamic risk management is a natural choice.

To understand the dynamic risk management framework let us define some definitions from software project perspective. A project is defined to be a set of parameters of interest for example a software project could have parameters such as software estimated cost, software quality, software deliverables and software project schedule etc. A collection of such parameters is defines a scenario. Risk events are defined as events which put negative impact on the project parameters or in other words put negative impact on a scenario. Strategy is defined as a set of decisions taken to combat risk events. The feedback loop feedback the difference between the actual and desired project parameters hence it measures the impact of risk events on the project parameters and also measures the effectiveness of the strategy applied to that scenario. Based on the feedback and the re-assessment, the decision set of a strategy may need to be updated. This allows practitioners to validate different strategies over a scenario and hence this process enables practitioners to choose the most optimum strategy. A project phase consists of project activities that occupy a specific time segment in the project during the project lifetime for example a project may be segmented based on its percentage of completion.

A dynamic risk management framework and assessment model gives more control over the risk management process and gets more insight into impact of a risk event on different parameters of a project as it moves through different phases of completion. Based on the future assessment of risk events the enterprise attempts to abate the risk by taking preventative actions at different times based on the feedback. These assessments are repeated over time and through feedback it is checked that the preventative actions are really effective. This results in a dynamic risk management framework in the form of feedback loops of assessments and preventative actions followed by new assessment and preventive actions.

Figure -4 illustrates a dynamic risk management framework for software projects. It consists of four building blocks sequentially connected; these blocks are labeled as Risk Process, Strategy, Scenario and Feedback block. The risk process block consists of risk identification, risk analysis and risk assessment that will generate risk impact factor of each identified risk together with its probability distribution. Risk process block acts as an input to the strategy block where set of decisions are taken based on the output of the risk process block. The strategy is applied on the scenario block which is a collection of project parameters of interest that may be affected by risk events. If risk events occur it will have a negative impact on the project parameters. The severity of impact of risk events will validate the effectiveness of the decisions defined in the strategy block. The activity ends at the end of the project phase.

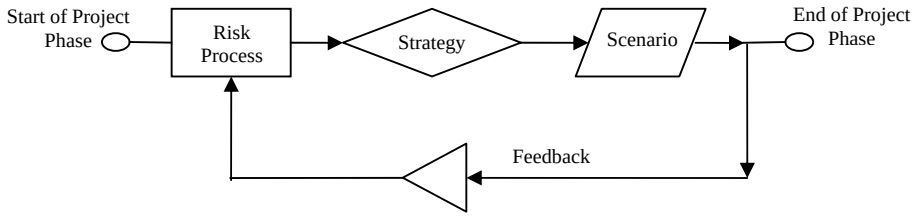

Fig. 5: Dynamic Risk Management Framework

Figure -5 illustrates a dynamic risk management framework for software projects. It consists of four building blocks sequentially connected; these blocks are labeled as Risk Process, Strategy, Scenario and Feedback block. The risk process block consists of risk identification, risk analysis and risk assessment that will generate risk impact factor of each identified risk together with its probability distribution. Risk process block acts as an input to the strategy block where set of decisions are taken based on the output of the risk process block. The strategy is applied on the scenario block which is a collection of project parameters of interest that may be affected by risk events. If risk events occur it will have a negative impact on the project parameters. The severity of impact of risk events will validate the effectiveness of the decisions defined in the strategy block. The activity ends at the end of the project phase.

The feedback block estimates the difference between the actual project parameters and the desired project parameters and this information is feedback to the risk process block. For the next project phase the risk process has the input from the previous project phase feedback. The feedback information is taken into account and the strategy is evaluated and adjusted for the subsequent project phase. The end of this project phase is connected with the start is another project phase the whole process is repeated until the end of the project.

As an example a software project scenario is presented. Software practitioners should consider similar scenarios in their own project setup. A software project faces a risk of a senior consultant leaving the project. At the start of the project risk process identifies that his departure at any stage of the project will cause project to delay. This event is flagged as a risk event that can cause project schedule to delay, so the software project schedule is the project parameter that is negatively affected by this risk event. As a strategy the project management decides that at every project phase the consultant will train one company employee and estimated that each newly trained employee will reduce the probability of schedule delay by 10%. This strategy is applied to all the project phases segmented with the percentage of project completion and its effectiveness is measured through the feedback.

Figure -6 illustrates how this strategy changes the shape of the perceived probability distribution of project delay. In the first phase of the project when project is 10% completed the expectation of project delay is around 80% due to this risk event. As project moves through different phases of completion and as software personal are trained the project delay expectation starts to shrink and towards the last project phase the expectation of the project delay is less than 10%.

In the beginning of the project the probability of long delay is high as the project moves through different phases of completion the probability of long delay decreases and moves towards the low delays. It is interesting to not that in the middle of the project it forms a normal distribution where long and short delays are equally probable, a shown in Figure -5. For initial project phases the probability distribution of the project delay is negatively skewed with a left tail showing more certainty in the long delay. This situation gradually improves and the probability distribution adopts a positive skew with a right tail this show certainty in the short project delays.

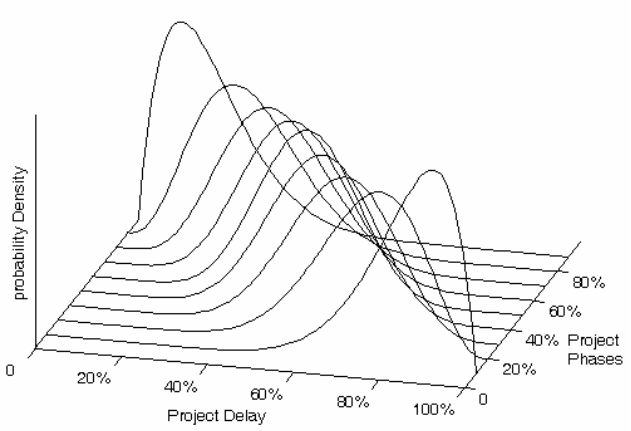

Fig. 6: Dynamic Risk Assessment of Project Delays due to Risk Event

IV. CONCLUSIONS

Software risk management frameworks and risk assessment models are discussed. These frameworks and models lack the capability of decision making and modeling assessment due the time variations in the perceived quantities over time. A dynamic framework and assessment model could provide a solution to deal with the short comings of the frameworks and assessment model being practiced. A dynamic framework analyzes different strategies for a specific scenario and through feedback loop it allows decision makers to re-access and re-adjust the decisions to help decision makers define strategies to combat different risk scenarios and how those strategies should evolve over time. This helps to choose the most optimum set of decisions for a specific scenario, hence giving decision makers more control over the process which increases the possibility of right decisions at the right time.

REFERENCES

[1] Daniel Bernoulli, Exposition of a new Theory on the Measurement of Risk (English Translation), The papers of Imperial Academy of Science in Petersburg, vol. V, 1738, pages 175-192.
[2] Barry W. Boehm, Software Risk Management: Principles and Practices, IEEE Software, pages 33-40, January 1991.
[3] Software Engineering Institute, Software Risk Evaluation Method Description, version 2.0, CMU/SEI-99-TR-029, Dec. 1999.
[4] Software Engineering Institute, CMMI for Development version 1.2, August 2006.
[5] Project Management Institute, A Guide to the Project Management Body of knowledge (PMBOK Guide), Third Edition, 2004.
[6] Mauro Gasparini et al, Dynamic Risk Control for Project Management, Statistical Methods and Applications, vol 13, pages 78 – 88, 2004.
[7] Casualty Actuarial Society, Dynamic Risk Modeling handbook, http://www.casact.org/research/drm/