# An Overview of Software Risk Management Principles

**Vasundhara Rathod, Monali Chim, Pramila Chawan**

***Abstract*— In this paper, risk-based methodologies for Software Risk Management (SRM) and its framework is discussed. Software Risk Management methodologies play important role the entire life cycle of software acquisition, development and maintenance. Good developed methodologies and tools for software engineering are very important for managing project more effectively. One should review and evaluate progress of the project in order to complete the project successfully. In this paper there are software risk management methodologies based on the seven management principles: product vision, teamwork, global perspective, future view, communication skills, integrated management, and continuity.**

***Index Terms*— Risk management principles, risk management framework, risk methodologies, Software risk management.**

## I. Introduction

There are two classes of functions for hierarchy of Software Risk Management (SRM) methodologies: software acquisition and software development. The basic methodological framework with which functions are managed is composed of the Software Acquisition-Capability Maturity Model (SACMM) and the Software Capability Maturity Model (SW-CMM) and their supporting practices and constructs. This framework for software risk management is supported by three groups of practices:
1. Software Risk Evaluation (SRE)
2. Continuous Risk Management (CRM)
3. Team Risk Management (TRM)
These are based on three basic concepts for software risk management: Risk Management Paradigm, Risk Taxonomy, and Risk Clinic. These are discussed in further sections. Above methodologies alone do not help to understand the complexity of the software risk management. The effect of this complexity is understood with the help of holographic modeling, in which there are two additional visions or dimensions: the temporal and human dimensions.

## II. Risk Management Principles

The developed software risk methodologies have three fundamentally different, though complementary, objectives:
1. Risk prevention
2. Risk mitigation and correction
3. Ensuring safe system failure
The following seven risk management principles are important to achieve these three objectives.

a. *Product vision* - product vision is based on common purpose, ownership, and collective commitment; focusing on results.
b. *Teamwork* – make team members to work in group to achieve a common goal. For this communication skill is needed.
c. *Global perspective* – In this potential impact of adverse effects, such as cost overrun, time delay, or failure to meet product specifications is included.
d. *Future view* – think about the risks that may arise in the future and try to minimize them as early as possible.
e. *Communication skills* – improve formal and informal communication. Communicate with all stakeholders through meetings.
f. *Integrated management* – Make habit of using risk management methods and tools in project development process.
g. *Continuity* – In all phases of project’s life cycle constantly identify and manage risks.

## III. Software Risk Management Methodologies

Risk Paradigm considered under the methodological dimension of risk management process. The Risk Paradigm explains all risk analysis activities. Similar reasoning applies to the Risk Taxonomy and to the Risk Clinic. The taxonomy provides a framework for organizing and studying the breadth of software development issues and hence provides a structure for surfacing and organizing software development risks. Since several of the methodologies discussed here make use of the Risk Taxonomy, it is presented along with the risk management paradigm as “Basic Constructs to Risk Management.” The Risk Clinic is a workshop that constitutes an important part of CRM and TRM.

### *Basic Constructs to Risk Management*

***Manuscript received Oct 15, 2011***.

***Vasundhara Rathod***, *Department of Computer Technology, Veermata Jijabai Tecnological Institute, Mumbai, India, +918055225407.*

***Monali Chim***, *Department of Computer Technology, Veermata Jijabai Tecnological Institute, Mumbai, India, +919762282884.*

***Pramila Chawan***, *Department of Computer Technology, Veermata Jijabai Tecnological Institute, Mumbai, India.*

Three basic constructs will be discussed here. All three constructs build on the seven risk management principles discussed earlier.

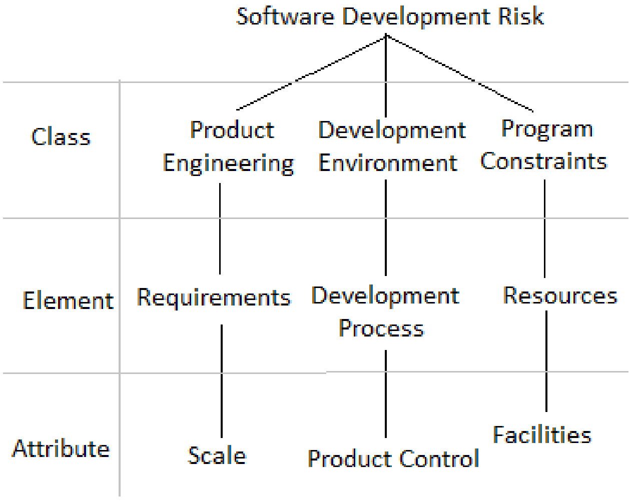

Fig. 1: Complete Taxonomy

### A. *Risk Management Paradigm*

The risk management paradigm states the different activities involved in the management of risk associated with software development. Essentially, the paradigm is a framework for software risk management. From this framework, a project may structure a risk management practice best fitting into its project management structure. A brief summary of each risk management paradigm activity is described below.

*Identify*- Before risks can be managed, they must be identified. Identification surfaces risks before they become problems. Techniques have been developed for surfacing risks by the application of a systematic process that encourages project personnel to raise concerns and issues. One such method, the SRE, is described in a subsequent section.

*Analyze*- Analysis is the conversion of risk data into risk decision-making information. Analysis provides the basis for the project manager to work on the "right" and most critical risks.

*Plan* -Planning turns risk information into decisions and actions. Planning involves developing actions to address individual risks, prioritizing risk actions, and creating an integrated risk management plan. The plan for a specific risk can take many forms. For example: Mitigate the impact of the risk by developing a contingency plan (along with an identified triggering event) should the risk occur; Avoid a risk by changing the product design or the development process; Accept the risk and take no further action, thus accepting the consequences if the risk occurs.

*Track*- Tracking consists of monitoring the status of risks and the actions taken to ameliorate them. Appropriate risk metrics are identified and monitored to enable the evaluation of the status of as well as of risk mitigation plans. Tracking serves as the "watchdog" function of management.

*Control* -Risk control corrects deviations from planned risk actions. Once risk metrics and triggering events have been chosen, there is nothing unique about risk control. Risk control melds into project management and relies on project management processes to control risk action plans, corrects for variations from plans, responds to triggering events, and improves risk management processes.

*Communicate* - Risk communication plays a vital role in risk management process. Without effective communication, no risk management approach can be viable. While communication facilitates interaction among the elements of the model, there are higher level communications to consider as well. In order to be analyzed and managed correctly, risks must be communicated to and between the appropriate organizational levels. This includes levels within the development project and organization, within the customer organization, and most especially, across that threshold between the developer, the customer, and, where different, the user. Because communication is pervasive, our approach is to address it as integral to every risk management activity and not as something performed outside of, or as a supplement to, other activities.

### B. *Risk Taxonomy*

The Risk Taxonomy follows the life cycle of software development and provides a framework for organizing data and information. The taxonomy-based identification method provides the organization developing software with a systematic interview process with which to identify sources of risk.

The taxonomy construct consists of a Taxonomy-Based Questionnaire and a process for its application. The taxonomy organizes software development risks into three levels: class, element, and attribute. The questionnaire consists of questions under each taxonomic attribute that are designed to elicit the range of risks and concerns potentially affecting the software product. The application process is designed such that the questionnaire can be used in a practical and efficient manner consistent with the objective of surfacing project risks. Both the questionnaire and the application process have been developed using extensive expertise and multiple field tests.

The taxonomy methodology is an instrument with which one can obtain a broad, system level of risks. These risks are commonly identified by program members, and are classified by categories within the hierarchical structure of the taxonomy. Moreover, the taxonomy identifies risk areas for more detailed investigation and is applied by interviewing peer groups of managers, engineers, and support personnel.

The taxonomy's risk identification method identifies and clarifies the uncertainties and concerns of a project's technical and managerial staff. The software taxonomy is organized into three major classes:

1. Product engineering: the technical aspects of the work to be accomplished
2. Development environment: the methods, procedures, and tools used to produce the product
3. Program constraints: the contractual, organizational, and operational factors within which the software is developed, but which are generally outside of the direct control of the local management.

These taxonomic classes are further divided into elements and each element is characterized by its attributes.

### C. *Risk Clinic*

A Risk Clinic is a workshop that takes the Continuous Risk Management (CRM) and Team Risk Management (TRM) and adapts and integrates it with a client's communication channels, infrastructure, existing practices, project management, risk management, and technical problem management.

## IV. Software Risk Management Framework

### A. *Software Risk Evaluation Practice*

The SRE practice is a formal method for identifying, analyzing, communicating, and mitigating software technical risk. It is used by decision makers for evaluating and mitigating the technical risks associated with a software-intensive program or project. The SRE is conducted at major milestones early and periodically in the acquisition life cycle. This practice consists of primary and support functions. Primary SRE functions are Detection, Specification, Assessment, and Consolidation. The support functions are Planning and Coordination, Verification, and Training and Communication.

Four primary functions are identified in the SRE practice—detection, specification, assessment, and consolidation.

*Detection* is the function of finding software technical risks of a target project. This function ensures systematic and complete coverage of all potential technical risk areas. It also ensures efficiency and effectiveness through the use of appropriate tools and techniques. Risk detection in the SRE practice is performed by using the following: Selection of appropriate individuals and guidelines for the make-up of the interview groups ensures coverage of all viewpoints including software development and support functions, technicians, and managers.

*Risk specification* is the function of recording all aspects of the identified software technical risk including its condition, consequences, and source. One representation of a software risk statement has several advantages. For instance, it serves as a simple, guiding structure for risk detection activities and for communicating risks coherently and with sufficient detail. It captures components of the risk and simplifies the task of prioritizing, isolating the condition within which the risk applies, and focusing the risk mitigation efforts to the source(s) of the risk. Additionally, risk specification records the source of the particular risk.

*Assessment* is a function that determines the magnitude of each software technical risk. By definition, magnitude is the product of severity of impact and the probability of an occurrence of the risk. Risk statements are assessed at one of three levels of magnitude— high, medium, or low. The level at which a particular risk is assessed depends on the separate assessments of its severity of impact and its probability of occurrence.

*Consolidation* is the function of merging, combining, and abstracting risk data into concise chunks of decision-making information. This is necessary due to multiple risk detection activities which identify elated risks from different sources. One example is similar risks that are identified in different interview sessions.

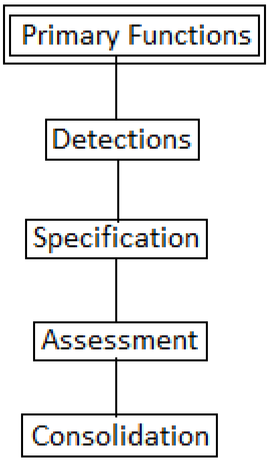

Fig. 2: SRE functional components

Only that set of risk statements which meets the defined criterion are considered as candidates for consolidation. Candidate risk statements must meet one of the following criteria for consolidation: manifestation of the same risk statement identical in every way except in the wording of the statements, fragmentation due to minor variations or different aspects of the same risk statement, differences in granularity; for example, a minor risk statement which is covered in the context of another risk statement of larger magnitude.

### B. *Continuous Risk Management (CRM)*

CRM is a principle-based practice for managing project risks and opportunities throughout the lifetime of the project. When followed, these principles provide an effective approach to managing risk regardless of the specific methods and tools used. These principles are composed of five groups: core, sustaining, integrated management, teamwork, continuous process, defining and future view.

*Core Principle* **-** Effective risk management requires constant attention to fostering the core principle of open communication. Clearly, the professionals associated with a project are the most qualified to identify the risks in their work on a daily basis. One should always ask, "Does project management provide a conductive environment for staffers to share their concerns regarding potential risks?" or, "Does management follow the 'killing the messenger' pattern instead of 'rewarding the messenger'?" Open communication requires encouraging free-flowing information at and between all project levels, enabling formal, informal communication, using consensus-based processes that value the individual voice, which can bring unique knowledge and insight to identifying and managing risk

*Sustaining Principles* **-** The sustaining principles focus on how project risk management is conducted on a daily basis. These are inward-directed, fundamental principles. If established early in the program and constantly nurtured, they should ensure that risk management becomes "the way we do business around here."

*Integrated management* **-** This principle helps to assure that risk management processes, paperwork, and discipline are consistent with established project culture and practice. Risk management is simply an area of emphasis in good project management; therefore, wherever possible, risk management tasks should be integrated into well-established project routine. Integrated management requires making risk management an integral and vital part of project management, adapting risk management methods and tools to a project's infrastructure and culture.

*Teamwork* - No single person can anticipate all the risks that face a project. Risk management requires that project members find, analyze, and work on potential risks together. Group synergy and interdependence in dealing with risk need to be rewarded. Teamwork requires working cooperatively to achieve a common goal, pooling talent, skills and knowledge

*Continuity* - The processes must be part of daily, weekly, monthly, and quarterly project management. The premise that risk management takes place only during "risk management seasons" is obviously foreign to true management. Continuous process requires sustaining constant vigilance, identifying and managing risks routinely throughout all phases of the project's life cycle

*Defining Principles* -The defining principles focus on how project staff members identify risks, and the extent to which staff and management are ready to address uncertainty. These principles are outward directed and concerned with focus; they foster the development of shared mental models that clarify the when, why, and what of risk management. *Future view* - This principle develops the ability to look ahead, beyond today's crisis and into the likely consequences and impacts of current decisions on future options. In this staff is also concerned with defining how far into the future to look, so that all risk mitigation efforts of project's staff are complementary.

### C. Team Risk Management (TRM)

TRM extends risk management with team-oriented activities involving the customer and supplier (e.g., government and contractor), where both customer and supplier apply the methodologies together. TRM establishes an environment built on a set of processes, methods, and tools that enables the customer and supplier to work cooperatively, continuously managing risks throughout the life cycle of a software-dependent development program. It is built on a foundation of the seven principles of risk management discussed in the preface of this paper, and on the philosophy of cooperative teams. Guided by the seven principles, TRM further extends the Risk Management paradigm by adding two functions—initiate and team. Each risk goes through these functions sequentially, but the activity occurs continuously, concurrently, and iteratively throughout the project life cycle (e.g., planning for one risk may identify another).

*Initiate*- Recognize the need and commit to create the team culture. Either customer or supplier may initiate team activity, but both must commit to sustain the teams.

*Team*- Formalize the customer and supplier team and merge the viewpoints to form a shared product vision. Systematic methods periodically and jointly applied establish a shared understanding of the project risks and their relative importance. Establish joint information base of risks, priorities, metrics, and action plans.

## V. Conclusion

This paper presents a brief summary of the methodologies for the management of risk associated with the acquisition, development, and use of software. Although software continues to grow in importance, as a critical system component and, more importantly as an overall system integrator, major sources of risk the user, the customer and the contractor communities. The methodologies presented in this paper shed some light on the professional community's effort to assess and ultimately control these inherent risks. Clearly, as systems become increasingly more complex, individual knowledge, judgment, and expertise will not be sufficient and systemic methodologies for risk management such as those presented in this paper become essential.

## References

[1] Brooks, Frederick P., "No Silver Bullet," Computer 20, 4 (April 1987).

[2] J. Kontio, V. Basili, "Emperical Evaluation of Risk Management Method", SEI Conference 1997, NJ

[3] Clyde Chittister, Yacov Y. Haimes, "Assessment and Management of Software Technical Risk", IEEE Transactions On Systems, Man, And Cybernetics, Vol. 24, No. 2, February 1994.

[4] Van Scoy, Roger L. "Software Development Risk: Opportunity, Not Problem". Pittsburgh, Pa.:Software Engineering Institute, Carnegie Mellon University, 1992.

[5] Sisti, Francis J. & Joseph, Sujoe. "Software Risk Evaluation Method", Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, 1994.

[6] Rasmita Dash, Rajashree Dash, "Risk Assessment Techniques for Software Development", European Journal of Scientific Research, ISSN 1450-216X Vol.42 No.4 (2010).

**Vasundhara Rathod** completed her B.Tech in Computer Science & Engineering in 2011 from GCOE, Amravati and currently doing M.Tech in Computer Technology at VJTI.

**Monali Chim** completed her B.Tech in Computer Technology in 2010 from COE, Pune and currently doing M.Tech in Computer Technology at VJTI.

**Pramila Chawan** is working as an associate professor in VJTI. She has 25 years of experience in teaching. Her fields of interest include software project management and computer architecture.