A Risk Assessment Framework to Reduce Risk Level and Optimize Software Quality

Sanjeev Puri[1*]

ABSTRACT

Risk management for software projects is intended to minimize the chances of unexpected events, or more specifically to keep all possible outcomes under tight management control with making judgments about how risk events are to be treated, valued, compared and combined. It is necessary to have some well-founded infrastructure for the identification of software security risks as well as the application of appropriate controls to manage risks. To be truly beneficial, the risk analysis framework must be granular and practical enough to produce a customizable roadmap of which problems exist, and to rank them in order of severity. The paper a risk assessment framework for a precise, unambiguous and efficient risk analysis with qualitative risk analysis methodologies and tree based techniques by exploiting the synthesis of risk analysis methods with object-oriented modeling, semi-formal methods and tools, in order to improve the security risk analysis of software and security policy implementation of security-critical systems to reduce risk levels and optimize quality instructions.

Keywords : *Risk Assessment Framework, Software Security Risk, Qualitative Risk Analysis Methodologies, Tree Based Techniques.*

I. INTRODUCTION

RISK management is the human activity which integrates recognition of risk, risk assessment, developing strategies to manage it, and mitigation of risk using managerial resources. In ideal risk management, a prioritization process is followed whereby the risks with the greatest loss and the greatest probability of occurring are handled first, and risks with lower probability of occurrence and lower loss are handled in descending order. In practice the process can be very difficult, and balancing between risks with a high probability of occurrence but lower loss versus a risk with high loss but lower probability of occurrence can often be mishandled [5].

Intangible risk management identifies a new type of risk - a risk that has a 100% probability of occurring but is ignored by the organization due to a lack of identification ability. For example, when deficient knowledge is applied to a situation, a knowledge risk materializes. Relationship risk appears when ineffective collaboration occurs. Process-engagement risk may be an issue when ineffective operational procedures are applied. These risks directly reduce the productivity of knowledge workers, decrease cost effectiveness, profitability, service, quality, reputation, brand value, and earnings quality [7]. Intangible risk management allows risk management to create immediate value from the identification and reduction of risks that reduce productivity.

A Risk is defined as "The possibility of suffering harm or loss; danger." Software Risk Management may be defined as a well defined, continual set of activities that together with the necessary tools and metrics can be used to identify, analyze and mitigate the risks involved in the Software Development Life Cycle (SDLC) of Software Projects through the usage of well defined policies, procedures and practices [2].

The management of risks is generally known to be an important aspect of project management, and yet

1* Sanjeev Puri is Professor (IT), with Sri Ramswaroop Memorial College of Engineering and Management, Lucknow; e-mail: purispuri_2005@rediffmail.com.

it often presents one of the greater challenges for project managers. Risks can be classified into three general types which coincide with the three primary concerns in project management: risk of delay in schedule; risk of over-spending; and risk of under-performance. Obviously, these three concerns are very much related such that one affects the other. The amount of available resources affects the rate at which the project progresses, and also affects the overall performance

Software development risk has been defined as the exposure to one or more of four types of risk [1]:

- performance risk, or the failure to obtain all of the anticipated benefits of the systems and software under development
- cost risk, or significantly exceeding budgeted or estimated cost
- schedule risk, or the failure to deliver satisfactory software products by scheduled milestones and user need dates
- support risk, or the delivery of a product that has excessive life cycle maintenance costs due to deficiencies in maintainability, flexibility, compatibility or reliability

Risk management has been defined as the practice of controlling risks that have the potential for causing unwanted program effects. This control is an entire development life cycle activity, starting with planning for risk at the earliest stages of the project and continuing with monitoring and alleviating risk though the support stage. Several risk management methodologies have recently been offered in the literature. The following is an overview of several approaches, with at brief look at the tools that have been proposed to aid those processes

II. RECENT WORK

Organizations are complex systems and for effective risk management, a systemic view is vital. A systemic approach implies an interconnected complex of functionally related components. The effectiveness of each component relies on how it fits into the whole, and the effectiveness of the whole depends on the way each component functions. A systemic approach considers the larger environment that affects processes and other work. The environment includes inputs, but, more importantly, it includes pressures, expectations, constraints, and consequences. Moore proposes a cyclic systemic approach to risk management systems development. It is important to distinguish a systemic approach from a systematic or process model.

Many existing risk management models and methodologies are found to be systematic. Webster's dictionary defines a system that is characterized by order and planning as systematic [1, 2]. A systematic system is also formed with regular connection and relating to the design as a whole. According to Wiegers, a barrier to effective process improvement or 'adaptive' behavior is the checklist mentality as exhibited by systematic models. As described by Fastenersources.com a checklist is 'a tool used to ensure that all important steps or actions in an operation have been taken'.

The Rand Corporation has developed an excellent "Guide for the Management of Expert Systems Development" using Boehm's risk driven spiral model. In this guide, expert system development is evolutionary, taking place through six phases: initiation, concept, definition/design, development, deployment, and post-deployment. For each phase, the guide discusses the risk containment activities, but no tools are recommended.

Boehm discusses sample tools for use at each of his steps, ranging from checklists to cost models to cost-benefit analysis. No mention is made of any standard project management metrics as a potential tool for aiding risk managers.

Richard Fairley offers a seven step risk management process based on his work identifying and overcoming risk factors on software development projects. Outside of assorted plans and his mathematical model for determining risk probabilities and effects, Fairley discusses no tools or tool methodology.

Rockwell risk management process, based on the principles of Dr. Robert Charette, is that of Rockwell, which is made up of to identify, characterize, prioritize and avert risks and lastly track/control risks. This methodology is tool-based and lists many tools for possible use for each of these five steps, but none is related to common project management metrics [4].

Risk management process is that of the F/A-18E/F, in which under risk identification, this approach asks, "What causes a risk to be surfaced?" and then

suggests a set of tools including "negative trends or forecasts" along with a set of metrics [9].

III. STATEMENT OF THE PROBLEM

Risk Management has traditionally been associated with risk elimination, insurance and compliance. Most software vendors have predictably added some risk features onto their existing compliance packages because it is easier from them to sell [3, 7].

- Traditional process models, methodologies and tools that are used to manage risks. The weakness of traditional risk management is the focus on historical precedence rather than forward looking investigative approach. Traditional design (Top-down design) and waterfall model is used for risk assessment.
- Traditional risk management framework is to allow a standalone and not-repeatable expert-driven approach to risk management in SDLC.
- Simple or no metrics is related to common software project management.
- Qualitative approach is more preferred than quantitative approach, to measure and evaluate risk.
- No systematic assorted plans and their mathematical and statistical model for determining risk probabilities and effects. Prioritization to rank of the identified risk items according to their compound risk are not is traditional risk management frameworks.

IV. RISK ASSESSMENT

Risk management is a structured approach to managing uncertainty through, risk assessment, developing strategies to manage it, and mitigation of risk using managerial resources. The strategies 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 [4]. After establishing the context, the next step in the process of managing risk is to identify potential risks. Risks are about events that, when triggered, cause problems. Hence, risk identification can start with the source of problems, or with the problem itself.

A. Source analysis

Risk sources may be internal or external to the system that is the target of risk management. Examples of risk sources are: stakeholders of a project, employees of a company or the weather over an airport.

B. Problem analysis

Risks are related to identify threats. For example: the threat of losing money, the threat of abuse of privacy information or the threat of accidents and casualties.

Unlike traditional risk management, Enterprise Risk Management [4] avoids this silo mentality by using a root cause approach to take a comprehensive view of risk. The root cause method looks at risks, such as information security, from all angles including processes and relationships as well as people, systems and external sources. Enterprise Risk Management recognizes that the chain is only as strong as the weakest link. Over investment in one area without the others is understood as not a good use of resources:

i) Use root cause as part of self-assessments to understand the source of risk.
ii) Use best practice risk indicators that are forward looking in nature to uncover risks.
iii) Develop clear measures of the penetration of your Enterprise Risk Management program.
iv) Measure the progress of your Enterprise Risk Management program roll-out and don't allow the timetable to slip.

V. IDENTIFIED KEY CHARACTERISTICS

Root Cause: A framework that gets to the cause of issues makes follow-up straight forward and logical.

Motivation: Performance Management functionality that makes it easy to help line managers achieve process improvements to reduce costs, bottlenecks, and unnecessary risk translates into their embracing risk management.

Process Driven: Selecting the most relevant 30 to 50 key risk indicators for each core business process from thousands of possibilities.

Cross Functional Risk: Features to deliver a portfolio view with interactive dashboards to drill down or cut across silos to identify dependencies between risks.

Operational Controls: Go beyond financial controls to also quantify the effect of controls on business goal achievement while maintaining accountability throughout the process.

Risk Tolerance: Embedding risk management pro-

cesses within the existing corporate culture from enterprise-wide board room strategy to tactical planning and analysis.

Maturity Model: Enable the risk management department itself to accelerate adoption of best practices, to set program objectives and measures and to manage ERM program activities.

With these criteria you can evaluate new software coming to the market from true ERM vendors and use risk tolerance to achieve the strategy and performance targets for your organization In the process of quantifying risk, there are two categories that generally stand out; they are Risk Measurement and Risk Metrics. These two things are often interchangeably confused, which should not be the case. Risk Measurement and Risk Metrics are two complete different processes. Risk Measurement is the process by which risk is measured and Risk Metrics is the value attached to the measured Risk. These two items have to be fully understood in order to have a proper understanding of the risk report presented for a project. Most metric sets deal with a variation of these attributes and are chosen to help project managers gain insight into their product (size, software quality, and rework), process (rework, software quality) and project (effort, schedule) [8].

Risk management is the process of continually assessing and addressing risk throughout the life of the software. It encompasses four processes: (1) asset identification, (2) risk analysis, (3) risk mitigation, and (4) risk management and measurement. During each of these phases, business impact is the guiding factor for risk analysis. The architectural risk analysis process includes identification and evaluation of risks and risk impacts and recommendation of risk-reducing measures. The Risk Assessment Framework content area of this site contains more detail of the life cycle of risk management.

Assessing security risks in software is predominately a qualitative process. Traditionally, efforts to deal with security vulnerabilities focus on hardening networks and peripherals that have access to computer systems [6]. Efforts have been underway to deal with application vulnerabilities early in the software development life cycle (SDLC). These efforts have underscored the fact that risk management should drive the software development process, which assures that security is made an emergent feature of the development process.

A. State of Risk Assessment

Making risk management an integral part of the software development process allows it to drive the development process so that security issues are ameliorated early in the product's life. Developers are expected to identify, rank, mitigate and manage risk throughout the software product life cycle. Methodologies such as threat/vulnerability identification, software testing and assessment, software reliability and the traditional risk assessment approaches that are used to allow risk to drive the development process have in large part been qualitative in nature.

B. Risk Assessment and Risk Management

A formal risk framework can be a useful tool for decomposing the problem of risk management. In such a framework, risks are assessed by evaluating preferences, estimating consequences of undesirable events, predicting the likelihood of such events, and weighing the merits of different courses of action. In this context, risk is formally defined as a set of ordered pairs of outcomes (O) and their associated likelihoods (L) of occurrence.

Risk $\{(L_1, O_1), ..., (L_i, O_i), ..., (L_n, O_n)\}$

Risk assessment is the process of identifying, characterizing, and understanding risk; that is, studying, analyzing, and describing the set of outcomes and likelihoods for a given endeavor. Modern risk assessment traces its roots to the nuclear power industry, where carefully constructed risk assessment methodologies were developed to analyze the operations of the very new and potentially dangerous nuclear power facilities. These methodologies centered on fault/event trees that were used to illustrate and to capture all possible plant failure modes in a graphical representation. Risk management is a policy process wherein alternative strategies for dealing with risk are weighed and decisions about acceptable risks are made. The strategies consist of policy options that have varying effects on risk, including the reduction, removal, or reallocation of risk. In the end, an acceptable level of risk is determined and a strategy for achieving that level of risk is adopted. Cost-benefit calculations, assessments of risk tolerance, and quantification of preferences are often involved in this decision-making process.

The three key elements in risk analysis are; (1) A statement of impact or the cost of a specific difficulty if it happens, (2) A measure of the effectiveness countermeasures, and (3) A series of recommendations to correct or minimize identified problems [5, 9].

The report's technical details should include, as a minimum:

- Vulnerability levels
- Applicable threats and their frequency
- The use environment
- System connectivity
- Data sensitivity level(s)
- Residual risk, expressed on an individual vulnerability basis
- Detailed Annual Loss Expectancy calculations

So, which methodology for security risk analysis is best; qualitative, quantitative? or hybrid? Should the process be manual or automated? The most basic function of any security risk analysis process is to determine, as accurately as possible, the risk to assets. Of course, the procedure for determining the risk can be complex or simple, depending on the asset and on the analysis methodology used. The amount of risk can be expressed as good/bad; high/low (qualitative), as a calculated metric (quantitative), or a combination of the two (hybrid) [5, 8, 9].

Furthermore, there are two ways in which metrics can help with risk identification. Using feasibility analysis, measures can be used at the initial risk identification step to help managers create a risk list. (Explain in figure1) Since risk identification should be an ongoing task that happens throughout the life cycle, using performance analysis. To develop an Adaptive Risk Assessment System that can [5, 9]:

- Identify various situations by matching them with a library of Risk Management Frameworks.
- Anticipate danger.
- Detect a possible threat which does not match any of the known frameworks

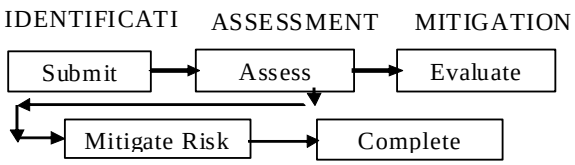

Fig. 1: Risk Management Process Steps

The basic idea behind Risk Management in SDLC is through a regular planning and assessment of risk that are measured based on the probability and impact on the Software Project Development Plan or schedule along with the proposed risk mitigation strategies to avoid risks and their impact on SDLC processes.

All risks can never be fully avoided or mitigated simply because of financial and practical limitations. Therefore all organizations have to accept some level of residual risks." Risk Management involves the following activities:

- **Risk Identification -** This is the step where a risk is identified before it becomes a problem, or, rather a hindrance to the success of any Software Project.
- **Risk Analyzing -** This is a step that determines which risks are the most important ones to address based on their priority and impact. Once the risks are prioritized based on their importance, the adverse effects that they can inject into the SDLC process and their probability of occurrence is analyzed.
- **Risk Planning -** Risk Planning involves a decision making process that prioritizes the risks and creation of Risk Mitigation Plans. Risk Prioritization involves the quantitative measurement of risks and estimating the probability of their re-occurrence and the relative loss that they could incur in the SDLC process.
- **Risk Response Actions -** This identifies and describes the action (such as acceptance, transfer, avoidance, or mitigation) and the necessary response strategies to address the risks based on the priority of the identified risks. This is the step that also identifies the target date for completion of the risk response action and the resource(s) that is/are responsible for the same.
- **Risk Monitoring -** This phase monitors the risks and their evaluation of their current status based on the defined metrics so as to ensure that the risks identified are addressed as per the stated timelines in the SDLC process of a Software Project.
- **Control -** This process controls the Risk Action or the Risk Mitigation Plans and improves the overall Risk Management Process. It involves the tracking of the progress of the SDLC

process towards resolving the risk items that have already been identified.

- Risk Reporting and Communication - This is a step that defines the methodologies that are used to report risk mitigation activities, review and present the Software Project risks and communicate the risks and their status effectively.

C. Qualitative Risk Analysis Methodologies

In this section, we will deal with the qualitative methods used in risk analysis namely preliminary risk analysis, hazard and operability study (HAZOP), and failure mode and effects analysis (FMEA/FMECA) [5].

Preliminary Risk Analysis: Preliminary Risk Analysis or hazard analysis is a qualitative technique which involves a disciplined analysis of the event sequences which could transform a potential hazard into an accident. In this technique, the possible undesirable events are identified first and then analyzed separately. For each undesirable events or hazards, possible improvements, or preventive measures are then formulated [2].

Hazard and Operability studies (HAZOP): HAZOP can be defined as the application of a formal systematic critical examination of the process and engineering intentions of new or existing facilities to assess the hazard potential that arise from deviation in design specifications and the consequential effects on the facilities as a whole. This technique is usually performed using a set of guidewords: NO/NOT, MORE OR/LESS OF, AS WELL AS, PART OF REVERSE, & OTHER THAN. From these guidewords, a scenario that may result in a hazard or an operational problem is identified. Consider the possible flow problems in a process line, the guide word MORE OF will correspond to high flow rate, while that for LESS THAN, low flow rate. The consequences of the hazard and measures to reduce the frequency with which the hazard will occur are then discussed.

Failure Mode and Effects Analysis (FMEA/ FMECA): Failure mode and effects analysis is a procedure by which each potential failure mode in a system is analyzed to determine its effect on the system and to classify it according to its severity. When the FMEA is extended by a criticality analysis, the technique is then called failure mode and effects criticality analysis (FMECA). Failure mode and effects analysis has gained wide acceptance by the aerospace and the military industries. In fact, the technique has adapted itself in other form such as misuse mode and effects analysis.

D. Tree Based Techniques

In this section, fault-tree analysis (FTA), event-tree analysis (ETA), cause- consequence analysis (CCA), and safety management organization review technique (SMORT) is to be discussed.

Fault tree analysis: A fault tree is a logical diagram which shows the relation between system failure, i.e. a specific undesirable event in the system, and failures of the components of the system. It is a technique based on deductive logic. An undesirable event is first defined and causal relationships of the failures leading to that event are then identified. Fault tree can be used in qualitative or quantitative risk analysis [9].

Event tree analysis: Event tree analysis is a method for illustrating the sequence of outcomes which may arise after the occurrence of a selected initial event. This technique, unlike fault tree uses inductive logic. It is mainly used in consequence analysis for pre-incident and post-incident application. The left side connects with the initiator, the right side with plant damage state; the top defines the systems; nodes (dots) call for branching probabilities obtained from the system analysis.

Cause-Consequence Analysis: Cause-consequence analysis (CCA) is a blend of fault tree and event tree analysis. This technique combines cause analysis (described by fault trees) and consequence analysis (described by event trees), and hence deductive and inductive analysis is used. The purpose of CCA is to identify chains of events that can result in undesirable consequences calculated in Figure 2.

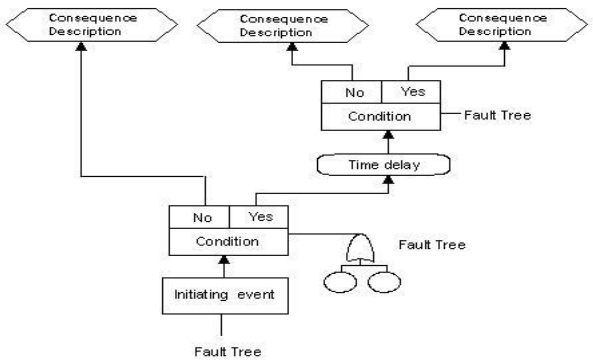

Fig. 2: A typical Cause-Consequence Analysis

E. Safety Management Organization Review Technique

Safety management organization review technique (SMORT) is a simplified modification of MORT. This technique is structured by means of analysis levels with associated checklists, while MORT is based on a comprehensive tree structure. Owing to its structured analytical process, SMORT is classified as one of the tree based methodologies. The SMORT analysis includes data collection based on the checklists and their associated questions, in addition to evaluation of results. The information can be collected from interviews, studies of documents and investigations.

VI. CONCLUSION

Security factors that take into account the innate characteristics of each vulnerability is incorporated into the calculation of the risk model; resulting in an empirical assessment of the potential threats to a development effort based on the risk assessment strategies and methods adopting in this adaptive framework. A Continuous risk management process is a necessary part of any approach to software security. A high-level approach to iterative risk analysis that is deeply integrated throughout the software development life cycle.

REFERENCES

[1] John Viega and Gary McGraw. Building Secure Software McGraw, Gary Software Security Building Security in Addison-Wesley Software Security Series, Boston, MA. 2006.

[2] Verdon, Denis, Gary McGraw "Risk Analysis in Software Design." IEEE Security and Privacy 2.4 2004.

[3] Mc-Graw "Software Security" IEEE Security & Privacy Vol.2 no.2 2004 pp80 83.

[4] Software Engineering Risk Analysis and Management - Page 113 by Robert N. Charette - Business & Economics - 1989 - 325 pages.

[5] Bennett, S.P., Kailay, M.P, "An application of qualitative risk analysis to computer security for the commercial sector", Eighth Annual IEEE Computer Security Applications Conference, Nov.-4 Dec. 1992, pp.64-73.

[6] Bishop, M., Computer Security, Art and Science, Addison Wesley, 2003.

[7] Alberts, C., Dorofee, A., Managing Information Security Risks, The Octave Approach, SEI Series, Addison Wesley, 2003.

[8] Clements, Paul C. and Ray C. Williams Final Report From the Workshop "Managing Software Risk at NASA", 21-22 October 2003CMU-Software Engineering Institute (November, 2003).

[9] Kaplan, Stan and B.John Garrick, "On the quantitative definition of risk," Risk Analysis Vol. 1, No. 1, (1981), pp. 11-27.