Software Risk Management: Importance and Practices

Abdullah Al Murad Chowdhury *and* Shamsul Arefeen

Abstract— Software risk management is a software engineering practice with processes, methods, and tools for managing risks in a project. It provides a disciplined environment for proactive decision-making to assess continuously what can go wrong; determine what risks are important to deal with; and implement actions to deal with those risks. Risk management planning addresses the strategy for risk management, the risk management process, and the techniques, methods, and tools to be used to support the risk management process. This paper recognizes the increasing role of risk management in present software projects and aims at providing more support in this area. First we take a look how software risk enters into the enterprise then we show a data model of risk identification with the overview of risk assessment and mitigation process. We then show how we can apply insurance to Software. And finally we present a solution to managing software risks.

Index Terms— Software project risk, Risk identification, Outsourcing, Risk mitigation, Risk insurance

————————————— ◆ —————————————

1 INTRODUCTION

THE software industry is one of the largest manufacturing industries in the world, with $350 billion in off-the-shelf software sold each year and over $100 billion in customized code on top of that. Risk management is an investment; that is, there are costs associated with identifying risks, analyzing those risks, and establishing plans to mitigate those risks. Software risk management is a software engineering practice with processes, methods, and tools for managing risks in a project. The main objective of Risk Management is to identify potential problems before they occur so that risk handling activities can be planned and invoked as needed across the life of the product or project to mitigate adverse impacts on achieving objectives. It should begin at the earliest stages of project planning and continue throughout the total life cycle of the project.

Software project risk management is an ethic in which the project team continually asseses what may negatively impact the project, determines the probability of such events occuring, and determines the impact should such events occur. It provides a disciplined environment for proactive decision-making to assess continuously what can go wrong; determine what risks are important to deal with; and implement actions to deal with those risks. Risk management planning addresses the strategy for risk management, the risk management process, and the techniques, methods, and tools to be used to support the risk management process.

————————————————

- **Abdullah Al Murad Chowdhury** is with the Department of Computer Science and Engineering, University of Asia Pacific (www.uap-bd.edu), Dhanmondi, Dhaka-1209, Bangladesh. E-mail: abamurad@gmail.com.
- **Shamsul Arefeen** is the General Manager of Softwarepeople, Dhaka, Bangladesh. He is attached with the Department of Computer Science and Engineering, University of Asia Pacific (www.uap-bd.edu), Dhanmondi, Dhaka-1209, Bangladesh. E-mail: shovon.arefeen@gmail.com.

2 RISK

Risk is a function of the likelihood of a given threat-source's exercising a particular potential vulnerability, and the resulting impact of that adverse event on the organization [1]. In IT systems, risk can be introduced from the internet, servers, networks, malicious insiders and even lapses in physical security. Risk is the possibility of loss. It is a function of both the probability of an adverse event occurring and its impact; the impact manifests itself in a combination of financial loss, time delay, and loss of performance. A risk is the precursor to a problem; the probability that, at any given point in the software life cycle, the predicted goals cannot be achieved within available resources. Risk cannot be eliminated from a software project, but it can be managed. Risk management is critical to the success of any software effort and is a strategic aspect of all software projects.

2.1 What Creates Risk?

Forces that contribute to loss or damage constitute elements of risk. Some influences are external to the enterprise and other influences are internal to the enterprise. These forces cannot be completely eliminated, and, hence, the enterprise has to take a calculated risk on its IT investment. IT risks are somewhat peculiar to each industry and/or firm.

Risk can be classified into systematic and unsystematic risk [2]. Systematic risk refers to that portion of risk caused by external factors; this is common and may affect all firms. Virus, hacking, fire, natural disasters and power loss are sources of systematic risk. Their effect is felt by many of the companies that are placed in the same position. For example, a loophole in the Internet browser that is vulnerable for hacking affects all of the firms that use the browser. Whereas, unsystematic risk is the portion of total risk that is unique to the firm. The factors such as misuse of data, loss of data, application error, human interaction, inside attack

and equipment malfunction can be cited for unsystematic risk. Unsystematic factors are largely independent of factors affecting the IT industry in general. Since these factors affect one firm, they must be examined for each firm.

The proportion of systematic and unsystematic risk denotes degree of vulnerability of the firm to the external or internal factors. Systematic risk is also known as generic risk, and unsystematic risk is also known as specific risk. Even though systematic risk is common for all firms of similar nature, its effect is not the same across all firms. This may be due to differences in the level of exposure and countermeasures taken by firms.

2.2 Software: Today's Biggest Security Risk

Today's application has become the enterprise's ''new perimeter''. With better network-level security technology hardening the network perimeter, malicious attackers are now focusing their efforts to strike at the least defended points - the application. While hackers were once satisfied with defacing Web sites, unleashing denial-of-service attacks and trading illicit files through targeted networks, modern attackers are profit-driven. Financial and customer data have become valuable commodities and applications must be secure enough to protect them.

Recent industry statistics confirm this trend. Data from Computer Emergency Response Team (CERT) reveals that the number of software vulnerabilities has risen dramatically and has eclipsed 7,000 new software vulnerability disclosures in the past year [3] – for example: personnel shortfalls, unrealistic schedules and budgets, developing the wrong software functions, developing the wrong user interfacing, gold plating, continuing stream of requirements changes, shortfalls in externally furnished components, shortfalls in externally performed tasks, real-time performance shor falls, straining computer science capabilities, etc.

Meanwhile, Gartner and NIST report that 95% of all reported vulnerabilities are in software [4], 78% of threats target business information, and 75% of attacks target the application level [5]. Yet, even with these findings, most enterprises allocate less than 10% of their security spending to application security.

2.3 How Software Risk Enters the Enterprise

As the myriad of applications deployed within organizations increases, some developed internally, some brought in from outside –the effort needed to manage risk becomes greater. Applications are inherently more complex – with many being based on a mixed code base from a wide range of sources, teams, and geographic locations. Gone are the days when companies developed their own source code. Now over two-thirds of the world's largest companies are engaged in offshore outsourcing [6].

Additionally enterprises have lacked an efficient matter to analyze the security of off-the-shelf commercial applications that are purchased further decreasing their security posture. Traditional tools cannot rise to this challenge as they typically require source code which usually isn't available in mixed code based applications. Manual penetration testing is time consuming, costly and simply doesn't scale. Today's applications are made up of multiple pre-compiled components, libraries and open source. The US Department of Homeland Security calls this "SOUP" or software of unknown pedigree. Simply put, the software supply chain is increasingly complex and whether software is purchased from an established vendor or developed in-house, the liability and risk that the application poses rests with the enterprise and organizations must take steps to test applications for security vulnerabilities prior to accepting and deploying software.

3 SOFTWARE RISK MANAGEMENT PROCESS

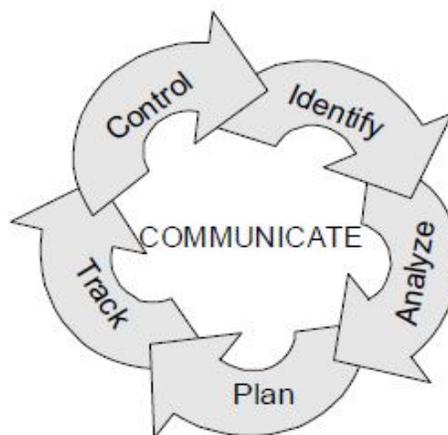There are several models available for risk management. The model recommended in this section was developed by

Fig. 1. Software Risk Management Paradigm

the Software Engineering Institute (SEI) [7] and is shown in Figure 1.

Identify: Before risks can be managed its must be identified before adversely affecting the project. Establishing an environment that encourages people to raise concerns and issues and conducting quality reviews throughout all phases of a project are common techniques for identifying risks.

Analyze: Analysis is the conversion of risk data into risk decision-making information. It includes reviewing, prioritizing, and selecting the most critical risks to address. The Software Risk Evaluation (SRE) Team analyzes each identified risk in terms of its consequence on cost, schedule, performance, and product quality.

Plan: Planning turns risk information into decisions and actions for both the present and future. Planning involves developing actions to address individual risks, prioritizing risk actions and creating a Risk Management Plan. The key to risk action planning is to consider the future consequences of a decision made today.

Track: Tracking consists of monitoring the status of risks and the actions taken against risks to mitigate them.

Control: Risk control relies on project management processes to control risk action plans, correct for variations from plans, respond to triggering events, and improve risk

management processes. Risk control activities are documented in the Risk Management Plan.

Communicate: Communication happens throughout all the functions of risk management. Without effective communication, no risk management approach can be viable. It is an integral part of all the other risk management activities.

3.1 Risk Assessment

Risk assessment is the first process in the risk management methodology. It is based on three concepts: reviews, snapshots and reports that underpin the three layers of processing the risk-related information: identification, analysis and reporting. Reviews establish the framework for risk identification, snapshots pass the identified risks for further analysis and reports communicate the results of risk assessment.

The risk identification layer uses reviews to gather risk-related information from a project. Reviews differ in terms of their scope, duration, participants and identification techniques. It is possible that two reviews overlap in time, however differing in their scope and/or participants. Risk-related information collected during a review is represented as risk indication and identifies a particular risk, the involved project stakeholder, timestamp, the identification technique and possible comments. After the identification and analysis, the risk assessment report is generated. The report is a sort of "risk summary" of the present view at risks. It can then be used as input for risk mitigation related activities. It may also be taken as an input to the next risk review action. The output of the risk assesment process helps to identify appropriate controls for reducing or eliminating risk during the risk mitigation process [1].

The risk assessment methodology encompasses nine primary steps such as System Characterization, Threat Identification, Vulnerability Identification, Control Analysis, Likelihood Determination, Impact Analysis, Risk Determination, Control Recommendations, and Results Documentation.

3.2 Review based Risk Assessment Process

We assume that there is a risk identification and analysis process performed by the project stakeholders and controlled by the risk manager (the role usually played by the project manager except large projects where it could be assigned separately). The process is structured as a sequence of reviews as is shown in Figure 2. It is assumed that at any time some review is open. The review remains open over its time window. Time windows of subsequent reviews are adjacent.

We distinguish between two types of reviews:

Active review: Its starting and ending times are set by the risk manager as well as its scope and participants (the stakeholders involved in the review). The review has a defined set of inputs (reports, checklists, questionnaires, etc.) and associated risk identification techniques. As a rule, the snapshot from the last continuous review is included as an input of the active review. The active review ends with the risk analysis session that aims at assessing and prioritizing

the identified risks and produces a relevant report.

Continuous review: It starts with the end of the previous review and ends with the start of the next review (being it active or continuous). It just keeps the communication channel open enabling the communicated risk information
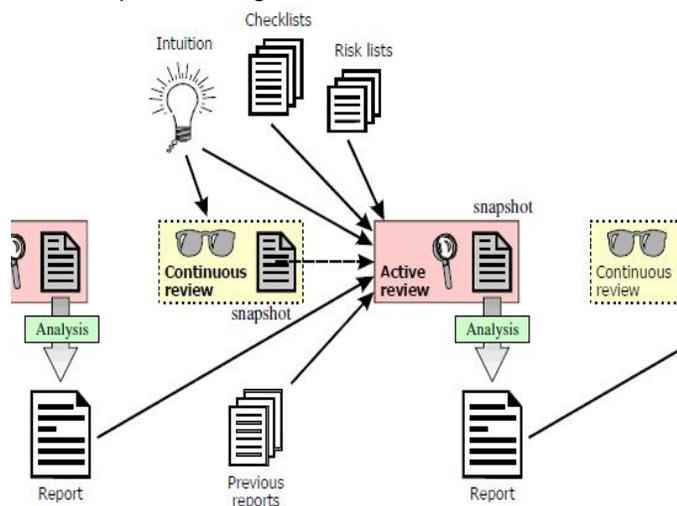

Fig. 2. Review Based Risk Assessment

being memorized. The set of its input documents is not controlled by the risk manager. Any project stakeholder can pass risk-related information disregarding the way of its generation.

Typically, a snapshot is taken at the end of the continuous review to provide an input to the subsequent active review. A snapshot is also taken at the end of an active review to summarize the effects of risk identification activities (as shown in Figure 2). The risk assessment report is generated at the end of an active review. We assume that the process has the active and continuous reviews interleaved, their extent (in time) and scope (in terms of inputs and participants) being controlled by the risk manager. This way we achieve the following benefits:

1. The communication channel is constantly open.
2. The identification actions are being planned (active and continuous reviews).
3. All communicated risk-related information is being memorized.
4. The identified risks are periodically reviewed and assessed and the frequency and scope of those assessments is under control of the risk manager.
5. The results of the analyses are kept in the form of reports and are available downstream of the process (can support further identification and analysis).

3.3 Data Model of Risk Management

The model comprises the following elements:

Project: General project description (process, methodology, organization, size, initiation date).

Mitigation area: Area of a project that is exposed to a common type of risks (e.g. requirement specification, personnel management etc.)

Review: This is the root object of the identification phase.

Opening a new review starts risk identification activities whereas closing the review ends the risk information acquisition.

Checklist: Checklists are used to collect information that helps to identify risks. A checklist includes its name, description, author's identification and its components.

Chapter: It is a component of a checklist. It can be hierarchically decomposed into more fine structuring elements as shown in Figure 3.

Question: This is the lowest structuring level of a checklist (nevertheless it may include some sub-questions).

Answer: It represents the answer to a checklist question (it may be of different type: yes/no, range etc).

Predefined risk: Risk that is stored in the risk knowledge base. It may be selected by one or more answers to the questions.

Predefined risk factor: Risk factor providing the context for a risk stored in the risk knowledge base.

Identified risk: Detailed risk description (from the risk knowledge base) in the context of a particular project. It is extracted from the knowledge base using the selection of
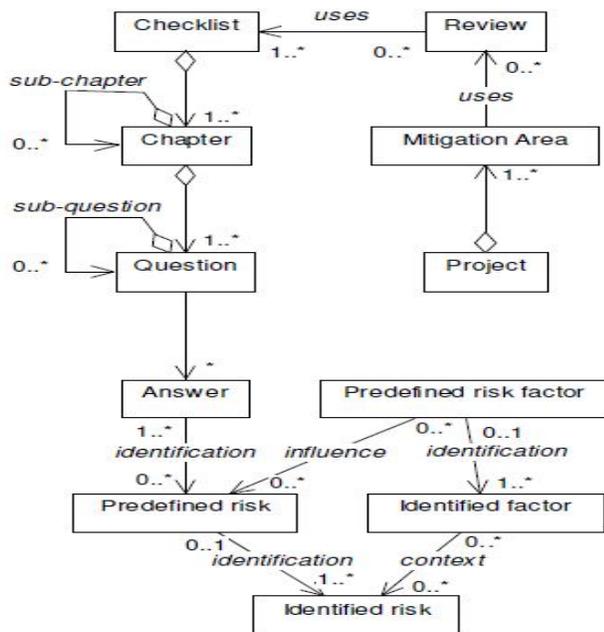

Fig. 3. Risk Identification Data Model

predefined risks resulting from the answers to the checklist questions.

Identified factor: Context of the identified risk extracted from the risk knowledge base.

3.4 Risk Mitigation

Risk mitigation, the second process of risk management, involves prioritizing, evaluating, and implementing the appropriate risk-reducing controls recommended from the risk assessment process. Because the elimination of all risk is usually impractical or close to impossible, it is the responsibility of senior management and functional and business managers to use the least-cost approach and implement the most appropriate controls to decrease mission risk to an acceptable level, with minimal adverse impact on the organization's resources and mission.

Senior management, the mission owners, knowing the potential risks and recommended controls, may ask, "When and under what circumstances should I take action? When shall I implement these controls to mitigate the risk and
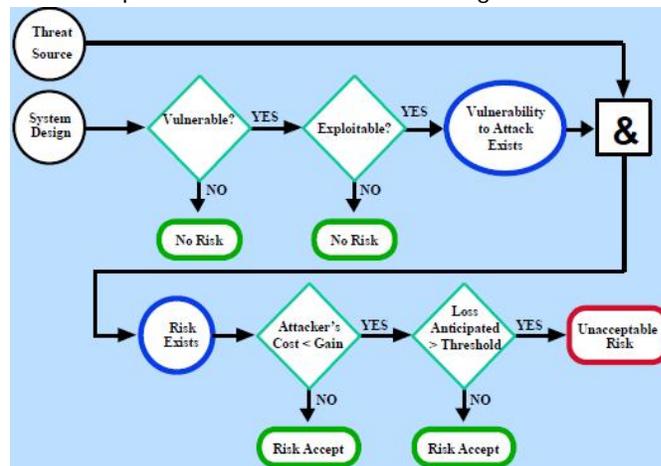

Fig. 4. Risk Mitigation Action Points

protect our organization?" The risk mitigation chart in Figure 4 addresses these questions. Appropriate points for implementation of control actions are indicated in this figure by the word YES.

This strategy is further articulated in the following rules of thumb, which provide guidance on actions to mitigate risks from intentional human threats:

When vulnerability (or flaw, weakness) exists -> implement assurance techniques to reduce the likelihood of a vulnerability's being exercised.

When a vulnerability can be exercised -> apply layered protections, architectural designs, and administrative controls to minimize the risk of or prevent this occurrence.

When the attacker's cost is less than the potential gain -> apply protections to decrease an attacker's motivation by increasing the attacker's cost (e.g., use of system controls such as limiting what a system user can access and do can significantly reduce an attacker's gain).

When loss is too great -> apply design principles, architectural designs, and technical and nontechnical protections to limit the extent of the attack, thereby reducing the potential for loss.

The strategy outlined above, with the exception of the third list item ("When the attacker's cost is less than the potential gain"), also applies to the mitigation of risks arising from environmental or unintentional human threats (e.g., system or user errors). (Because there is no "attacker," no motivation or gain is involved.)

4 APPLYING INSURANCE TO SOFTWARE

Despite all precautions, we cannot completely avoid risks. Insurance provides a means of reducing financial loss due to the consequences of risks, by spreading or pooling the

risks over a group of people. The group needs to be as homogeneous as possible regarding risk characteristics, yet large enough to share the risk. This way people who did not experience a loss help to repay the losses of the few who did. The process of finding these risk groups is called risk classification. It enables determination of coverage and its price for each group.

It seems attractive to apply insurance to the software economic risk remediation problem.Our interest is in software reuse risks. These are risks that arise when the online code, data or services you are using misbehave and as a result cause you an insurable loss. We are interested in using insurance to alleviate the consequences of such failures, thus enabling broader reuse. The risks we are interested in arise due to the characteristics of software.There are existing forms of insurance for risks related to the business setting, mainly risks related to a traditional business going online. These include insurance against potential lawsuits involving fraud, libel or invasion of privacy and against risks related to external attacks such as theft, tampering and destruction of information resources.

We need both to recognize loss events and to determine loss. We concentrate here on recognizing loss events. Consequences of loss event are beyond the scope of our work here.Loss events in our domain are software failures. We classify such failures into communication failures (cannot get data), syntax and format failures (cannot parse the data) and semantic failures (data doesn't make sense). Failure detection techniques exist mainly for the first kind and for some parts of the second. We therefore concentrate on detection of semantic failures. These are failures to supply the intended semantic due to software failures. Examples include failure to provide timely updates of near real-time data, such as current weather conditions, changing data that should be static, and neglecting to provide complete information, such as missing an important news item about a requested topic.

In software development, risks are everywhere. External systems are a risk. Unmaintainable code is a risk. Having a high number of defects is a risk. Even the choice of technology is a risk. We can apply insurance in several IT areas, some of them we propose are following:

1. Data Processing/Warehousing Services
2. Education & Training
3. Facilities Management or Outsourcing
4. Hardware Installation
5. Maintenance and Repair
6. Hardware Sales
7. ISP/Web/Internet Service
8. IT Consultants
9. IT Recruitment & Placement Services
10. Project Management
11. Software Development
12. Software Installation
13. Software Maintenance
14. Software Sales – Customised/Own Developed
15. Software Sales – Packaged/Third Party Software
16. Systems Analysis
17. Systems Audit
18. Systems Integration
19. Telecommunication Services
20. Troubleshooting

The software industry has unique liability exposures due to the crossover between the provision of professional services and supply of goods with many services providers in this industry having a mix of both. Nowadays the insurance industry has developed a range of insurance products that are commonly referred to as IT liability policies. These policies represent a combination of professional indemnity and public and products liability insurances combined within the one product with a view to minimising the prospect of an uninsured claim due to it "falling between the gaps" between the two traditional insurance products. The IT Liability insurance market is relatively young with many of the current insurers being relative newcomers to the market and their products varying significantly.

5 VERACODE SECURITYREVIEW

Veracode SecurityReview [8] is the industry's first solution to use patented binary code analysis and dynamic web analysis to uniquely assess any application security threats, including vulnerabilities such as cross-site scripting (XSS), SQL injection, buffer overflows and malicious code. SecurityReview performs the only complete and independent security audit across any internally developed applica-

Fig. 5. Veracode: a simple solution

tions, third-party commercial off-the-shelf software and offshore code without exposing a company's source code. Delivered as an on-demand service, Veracode delivers the simplest and most-cost effective way to implement security best practices, reduce operational cost and achieve regulatory requirements such as PCI compliance without requiring any hardware, software or training.

As an expert in application security, Veracode is uniquely suited to provide independent verification and validation (IV&V) of software applications without the need for costly on-site consultants. Veracode's Ratings System produces a software security rating based on respected industry standards including MITRE's Common Weakness Enumeration (CWE) for classification of software weaknesses and FIRST's Common Vulnerability Scoring System (CVSS) for severity and ease of exploitability and NIST's application assurance levels. These universally accepted vulnerability

scoring methods provide a clear audit trail enabling enterprises to automate the security acceptance testing of outsourced applications and meet both internal and external security and compliance requirements and reduce their exposure to risk.

6 CONCLUSION

The most important thing for a software project to do is to get focused on its critical success factors. For various reasons, including the influence of previous document-driven software management guidelines, projects get focused on activities which are not critical for their success. We can take some steps such as, ranking the project's most significant risk items, establishing a regular schedule for higher management reviews of the project's progress and so on to keep tracking on major risk factors. In this paper, we discuss about the risk management process with risk assessment and risk mitigation techniques, and also present software insurance concepts with a real life solution. We observed, still now software risk management reside in back seat but we should keep more focus on it. However, risk management is not a cookbook approach. To handle all of the complex people-oriented and technology-driven success factors involved in software projects, a great measure of human judgment is required.

ACKNOWLEDGMENT

REFERENCES

[1] NIST Risk Management Guide for Information Systems Special Publication 800-30. July, 2002

[2] Reilly, F.K.; K. Brown; Investment Analysis and Portfolio Management, Harcourt College Publishers, 2002

[3] Microsoft Security Intelligence Report 2008 –Based on data from the DHS NVD & CERT)

[4] Mark Curphey, "SoftwareSecurity Testing: Let's Get Back to Basics" October, 2004, SoftwareMAG.com

[5] Theresa Lanowitz, "Now Is the Time for Security at the Application Level" 2005, Gartner

[6] Mary Hayes Weier, "The Second Decade Of Offshore Outsourcing: Where We're Headed", Nov.2007, InformationWeek

[7] Software Engineering Institute Web site: http://www.sei.cmu.edu/risk

[8] Based in Burlington, Mass., Veracode is backed by .406 Ventures, Atlas Venture and Polaris Venture partners. Web site: http://www.veracode.com

Abdullah Al Murad Chowdhury has been serving as a Programmer at ICDDR,B from 2010. He obtained his B.Sc. degree in Computer Science and Engineering (CSE) from State University of Bangladesh, Dhaka in the year of October 2009. He also completed a Professional Diploma in Information Technology from NIIT, Dhaka in the year of June 2007.

Shamsul Arefeen is serving as the General Manager of Softwarepeople, a multinational software development organization at Dhaka, Bangladesh. Soon after his return from USA in 2007, Arefeen worked as the Program Manager for CMMI implementation in five software industries in Bangladesh for the first time. Under his leadership with joint collaboration with QAI India all five software organizations achieved the CMMI level 3 maturity. Arefeen's experience in Software project management, quality assurance and International Softwarel Business Development is worth mentioning.
Arefeen achieved his first Master's degree in Computer Science and Engineering from University of Dhaka. He completed his 2nd MS in Software Engineering from Texas State University, USA. His working experience in USA in different software organizations has given him various exposures to the working methodology of various software development organizations. He takes special interest in knowledge sharing and has been engaged in teaching M.Sc level courses on Software Engineering in different universities in Bangladesh.