IJITKMSpecial Issue (ICFTEM-2014) May 2014 pp. 11-13 (ISSN 0973-4414)

Software Risk Management

Rajesh Garg[1], Nitika Bansal[2],Pallavi Sharma[3]
[1]Lect (ECE Deptt),Damla Polytechnic for Engineering,
[2, 3]Lect (CSE Deptt.)Ganpati Polytechnic for Engg.Bilaspur,
[1] rajesh_damla@yahoo.co.in.,[2] nitikabpr@gmail.com, [3]girlcutester@gmail.com

Abstract: Generally, in software development, there are large, medium, and small software Projects that each of them can be affected or influenced by a risk. Therefore, it requires a distinctive assessment process of the potential risks that may cause failure or loss of the project whenever they occur. Risk management is a series of steps whose objectives are to identify, address, and eliminate software risk items before they become either threats to successful software operation or a major source of expensive rework.

1. Introduction

Present software projects are often facing expanding and changing client demands and are put under schedule pressure. The systems are growing in size and become increasingly complex. To shorten the development time the systems are built out of reused (but often not reusable) components. The personnel turnover is high and the size and diversity of project groups are growing. Consequently, despite of the progress in technology, the software engineering project management still faces similar problems as twenty years ago .The requirements are not defined precisely which results in constant expansion of the system scope or even in rejection of the final system. The involvement of people is relentlessly adding the factor of human mind and personality to the technical difficulties of the project. The resulting software is constantly error-prone, the cooperation among the project members is often poor and the expectations of the clients are not satisfied. Altogether, it calls for some significant improvements to the software engineering process. One of such innovative approaches is risk management.

2. Risk

Risk is a combination of the probability of a negative event and its consequences. If an event is inevitable but inconsequential, it does not represent a risk, because it has no impact. Alternatively, an improbable event with significant consequences may not be a high risk. These two factors are combined in what we experience as the possibility of loss, failure, danger.

$$\text{Project Risk} = \sum (\text{Events} * \text{Probabilities} * \text{Consequences})$$

An easy way to reduce risk is to have less ambitious goals. After evaluating risks, one can choose a path of risk avoidance or risk mitigation and management. If we understand the risks on a project, we can decide which risks are acceptable and take actions to mitigate or forestall those risks. If our project risk assessment determines risks are excessive, we may want to consider restructuring the project to within acceptable levels of risk.
Risks that do not offer the potential for gain (profit?) should be avoided. Risks associated with achieving challenging and worthwhile goals should be managed. One way to reduce risk is to gather information about relevant issues to lower the level of uncertainty.
A software project may encounter various types of risks:

Technical risks include problems with languages, project size, project functionality, platforms, methods, standards, or processes. These risks may result from excessive constraints, lack of experience, poorly defined parameters, or dependencies on organizations outside the direct control of the project team.

Management risks include lack of planning, lack of management experience and training, communications problems, organizational issues, lack of authority, and control problems.
Financial risks include cash flow, capital and budgetary issues, and return on investment constraints.

Contractual and legal risks include changing requirements, market-driven schedules, health & safety issues, government regulation, and product warranty issues.

Personnel risks include staffing lags, experience and training problems, ethical and moral issues, staff conflicts, and productivity issues.

Other resource risks include unavailability or late delivery of equipment & supplies, inadequate tools, inadequate facilities, distributed locations, unavailability of comp uter resources, and slow response times.

2.1 Risk Component

Risk has following components:-

- Performance Risk-the degree of uncertainity that the product will meet its requirements.
- Cost Risk-the degree of uncertainity that the project budget will be maintained.
- Support Risk-The degree of uncertainity the project will be easy to correct, adapt.
- Schedule Risk-the degree of uncertainity that the project schedule will be maintained and that the product will be delivered.

3. Risk management process

The risk management process can be broken down into two interrelated phases, risk assessment and risk control, as outlined in Figure 1. These phases are further broken down. Risk assessment involves risk identification, risk analysis, and risk prioritization. Risk control involves risk planning, risk mitigation, and risk monitoring.

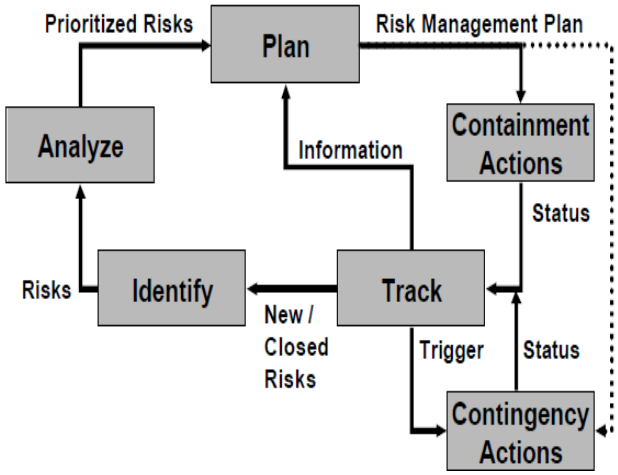

Fig. 1 - Risk Management Process

3.1 Risk Identification

During the first step in the software risk management process, risks are identified and added to the list of known risks. The output of this step is a list of project-specific risks that have the potential of compromising the project's success. There are many techniques for identifying risks, including interviewing, reporting, decomposition, assumption analysis, critical path analysis, and utilization of risk taxonomies.

3.2 Risk Analysis

During the risk analysis step, each risk is assessed to determine:

- Likelihood: the probability that the risk will result in a loss
- Impact: the size or cost of that loss if the risk turns into a problem
- Timeframe: when the risk needs to be addressed (i.e., risk associated with activities in the near future would have a higher priority then similar risks in later activities)

3.3Plan

Risk management plans should be developed for each of the “above the line” prioritized risks so that proactive action can take place. These actions are documented in the Action column of the Risk Table.

3.4 Taking Action

During the action step, we implement the risk reduction plan. Individuals execute their assigned tasks. Project effort estimations are adjusted to take into consideration additional effort needed to perform risk reduction activities or to account for projected additional effort if the risk turns into a problem. Some tasks may be moved forward in the schedule to ensure adequate time to deal with problems if they occur. Other tasks may be moved back in the schedule to allow time for additional information to be obtained.

3.5 Tracking

Results and impacts of the risk reduction implementation must be tracked. The tracking step involves gathering data, compiling that data into information, and then reporting and analyzing that information.. The results of the tracking can be:

- Identification of new risks that need to be added to the risk list.
- Validation of known risk resolutions so risks can be removed from the risk list because they are no longer a threat to project success.
- Information that dictates additional planning requirements
- Implementation of contingency plan

4. Conclusions

With ever-increasing complexity and increasing demand for bigger, better, and faster, the software industry is a high risk business. When teams don't manage risk, they leave projects vulnerable to factors that can cause major rework, major cost or schedule over-runs, or complete project failure. Adopting a Software Risk Management Program is a step every softwaremanager can take to more effectively manage software development initiatives. Risk management is an ongoing process that is implemented as part of the initial project planning activities and utilized throughout all of the phases of the software development lifecycle.

References

[1] Barry W. Boehm, Tutorial: Software Risk Management, Les Alamitos, CA, IEEE Computer Society,1989.
[2] Hall, E. M. (1998). Managing Risk: Methods for Software Systems Development, Addison Wesley.
[3] Pfleeger, S. L. (1998). Software Engineering: Theory and Practice. Upper Saddle River,NJ, Prentice Hall.
[4] Software Risk Management: Importance and Practices by Abdullah Al MuradChowdhuryandShamsulArefeen