605.401
Foundations of Software Engineering

Software Project Risk Management

Introduction

It is a fact of life that every software engineering project will be faced with one or more risks. Recall from our discussion of project planning that an important component of a project plan is a risk management plan. So, what's a risk management plan anyway?

Before answering that, a brief discussion of risk is in order. Many people talk about project risks but have different understandings of what a risk actually is. The dictionary definition of risk is ***the possibility of loss***. From a software engineering project management viewpoint we can think of a risk as an event that might occur that has an adverse impact on the project. A very pragmatic definition that I like to use is ***a risk is a potential problem***. When thinking about risks and talking about risks it is important to understand that a risk has two attributes:

- The ***likelihood*** of occurrence
- The consequence of occurrence…i.e., the ***loss***

A risks is not a sure thing…there's a probability that it might occur. If a risk does occur, then some loss would be incurred. Inherent in this definition of risk is also the fact that a risk is something that might occur in the ***future***. If it has already occurred it is no longer a risk…it is a reality.

Risk management is the process of ***dealing with risks*** over the course of the project. It consists of five activities:

- Risk identification
- Risk analysis
- Risk planning
- Risk tracking
- Risk resolution

Risk identification consists of creating and maintaining a list of project risks. **Risk analysis** consists of analyzing the impact of each risk, should it occur…that is, the associated loss. Risk analysis may also involve prioritizing project risks. **Risk planning** involves creating a plan that will reduce the probability of the risk occurring and/or reduce the impact if the risk should materialize. **Risk tracking** consists of monitoring the status of known project risks. **Risk resolution** consists of the actions that will be taken should a risk occur.

It's important to note that the risks associated with a project are dynamic. They may change over time. Some risks may not occur, some risks may, and new risks might pop up during the course of a project…so risk management must be an ongoing activity.

Software Project Risk Management

Risk Identification

Risk identification involves creating and maintaining a list of project risks. So…how exactly is that done? Basically, we start with identifying the possible sources of risk. There can be many sources, and sources can vary from organization to organization and from project to project. Four common sources of risk are uncertainty, knowledge, project concerns, and project issues. These aren't the only sources, but they can be useful starting points for the risk identification process.

Uncertainty is what we don't know or what we are unsure of. For example, having a poor understanding of requirements is a project risk. So is lack of understanding about necessary project deliverables. Estimates can also be a source of uncertainty, particularly if they are not based upon valid historical information or were arrived with no documented basis or methodology.

Knowledge is what we do know. We can use knowledge gained from previous software engineering to identify risks associated with a specific project. As an example, the project may be planning to use a particular external vendor for a portion of the work. We may know from prior experience that this vendor tends to deliver its work products late…which can cause project milestones to be missed.

Concerns are what we feel uneasy about. Concerns can arise based upon project resource constraints, unrealistic deadlines, client relationships, and so forth.

Issues usually involve things that need resolution and which may increase the likelihood that a risk will materialize. As an example, requirements that have not been approved, or a requirement that has been identified as ambiguous but hasn't been clarified, might be sources of project risk. Unresolved issues often involve other people or organizations collaborate with each other.

In addition to these sources it is often useful to think about risks that emanate from the software development process, the project management process, the project staffing, and the software product itself. Examples of process-related risk would be lack of a software test plan and lack of a review process for project work products. Examples of product-related risks include high code complexity and incomplete requirements.

Checklists for the potential sources of risk and possible specific risks can also be used as part of the risk identification activity. One example of such a checklist is the SEI Risk Taxonomy, developed by the Software Engineering Institute [Carr, 1993]. One of the things that history has revealed about project risks is that risks tend to repeat from project to project, so using data about historical risks can and should be incorporated into a risk checklist.

Software Project Risk Management

Risk Analysis

Risk analysis involves analyzing the impact of each risk if it occurs, and estimating the expected loss. It also involves producing a list of prioritized risks. There are many different techniques used to analyze and prioritize risks. Sometimes the techniques are informal and subjective, and sometimes they involve using a formal process. Common categories of loss for software engineering projects include schedule overruns, increased cost, and diminished work product and end product quality.

One technique for evaluating risk involves calculation of what is called risk exposure and risk severity. **Risk exposure** is the product of the likelihood that a risk will occur and the consequence (measured in dollars) if it does occur. For example, suppose the risk is that testing will take an additional 3 months. If this risk occurs the cost will be $30,000. If the likelihood that the risk occurs is .70 (70 percent), then the risk exposure is $21,000. **Risk severity** maps categories of risk exposure against time. An example of this is illustrated in the table below.

Risk Severity Table

		Risk Exposure		
		Low	Medium	High
Time	Short	5	2	1
	Medium	7	4	3
	Long	9	8	6

Adapted from [Hall, 1998]

In the above table, risk exposure is mapped to one of three categories: high, medium, and low. There would be dollar ranges associated with each category. For example, a risk exposure up to $10,000 might be considered low, $10,000+ to $20,000 may be considered medium, and more than $20,000 may be considered high. Each row in the table is associated with a time frame associated with how soon action is required to prevent a risk from occurring. Based on the table, the highest priority risks will have a risk severity of 1 and the lowest priority risks have a risk severity of 9. As an example, a risk with a high risk exposure value that needs to be acted on in a short time frame will have a severity score of 1, while a risk with low risk exposure and has a long time frame before it needs to be acted on will have a score of 9. Using this approach, the risk severity score for a given risk will change over time and, for a constant risk exposure value, the severity will increase as the action time frame nears.

Some examples of associating risk evaluation criteria are shown below. Likelihood can be specified using a probability of occurrence or on a scale of likely occurrence, as illustrated in the table below.

Software Project Risk Management

Probability Evaluation Table

Probability	Uncertainty Category	Likelihood Score
>80%	Highly Likely	5
61-80%	Likely	4
41-60%	Improbable	3
21-40%	Unlikely	2
1-20%	Highly Unlikely	1

Adapted from [Hall, 1998]

In the above table, if it is felt that a risk has up to a 20 percent chance of occurring, that risk would be assigned a category of *highly unlikely*. If it is felt that a risk has a 61-80 percent chance of occurring, that risk would be assigned a category of *likely*. Numeric likelihood scores may also be assigned for purposes of prioritizing risks, as illustrated in the third column in the table. The higher the number, the more likely the risk is to occur. An alternative to using a numeric score is to map the categories into high, medium, and low occurrence categories.

The table below illustrates sample consequence criteria for cost and project schedule. A risk that has a less than 5 percent cost increase and schedule slippage of up to 2 weeks would be categorized as a modest risk.

Consequence Evaluation Criteria

Criterion	Cost	Schedule
Low	Less than 1%	Slip 1 week
Moderate	Less than 5%	Slip 2 weeks
High	Less than 10%	Slip 1 month
Critical	More than 10%	More than 1 month

Adapted from [Hall, 1998]

It should be noted that the above values are not recommended values, but just examples. Actual values must be determined based on the specifics of each project.

The table below illustrates some sample criteria for time frames in which action must be taken. Again, it provides example values. Actual values must be determined based on the specifics of each project.

Time Frame Evaluation Criteria

Category	Time Frame
Short	1 month
Medium	2 months

Software Project Risk Management

Long	3 months

Adapted from [Hall, 1998]

Once risks have been analyzed, they can be reported and managed using a Top-N risk list. There are many formats for such a list. One sample is shown below.

Sample Top-N Risk List

Risk	Current Priority	Previous Priority	Action Plan Status	Risk Severity
Test plan completeness	1	3	Quality reviews being performed.	2
Hardware delivery	2	4	Not yet assigned.	5
Technical documentation	3	5	Scheduled training for documentation specialists.	6
…	…	…	…	…

Risk Planning & Risk Resolution

Risk planning involves creating a plan of action that will reduce the probability of the risk occurring and/or reduce the impact if the risk should materialize.

One technique for doing risk planning is to develop risk scenarios for the high-severity risks. A **risk scenario** is a projection of events and conditions that can lead to the occurrence of a risk. In developing a risk scenario, it is common to assume that the risk has occurred, and list the events and conditions that would precede the risk occurrence. As an example, suppose the risk of interest is that *software tests cannot be traced to the software requirements*. For this risk to materialize, several events may have to occur:

- Requirements-to-Tests traceability matrices have not been developed
- Requirements-to-Design traceability matrices have not been developed
- Design reviews have not been performed
- Test plan reviews have not been performed

These events can then be used to develop early-warning indicators and resulting risk resolution actions. For example, if design reviews have not been performed then the likelihood of discovering that requirements-to-design traceability matrices have not been developed is high…so that could be a trigger that causes actions to be taken to resolve the risk.

Risks may be resolved in several ways. One way is to avoid the risk altogether, by taking actions that make the risk not happen. In the above example, traceability matrices can be explicitly required, reviews

can be explicitly required which ensure that the matrices have, in fact, been produced. Management would then monitor these activities to ensure that it has been accomplished.

Another way to resolve risk is through risk reduction. Risk reduction decreases the probability that the risk will occur or decreases the impact if it does occur. As an example, the risk may be a *high level of defects in deliverable work products*. The risk action plan may then call for using work product walkthroughs or inspections, proven techniques for reducing defects in work products. There is a cost involved, but the cost will be recaptured by a reduction in project rework.

Yet another way to resolve risk is to accept it and live with the consequences…if that is acceptable. For example, some staff turnover may be expected over the course of the project. The expense of replacing staff might be the same as the expense of increasing compensation or other benefits to retain existing, trained staff. So, the risk resolution strategy might be to accept the turnover. The cost of hiring and training replacements is the consequence.

Risk Tracking

Risk tracking involves monitoring the known risks and being on the alert for new risks. The likelihood and consequences of risks will typically change during the course of the project. Some risks will not occur and can be taken off the risk list. Using early-warning indicators and metrics can help the tracking effort. Tracking risks is not enough, however. Project progress must also be tracked since project status, project events, and activity performance can cause risk parameters to change and new risks appear on the horizon. It is important that data associated with “actual” performance, such as actual time charged to project tasks for a particular time period, and metrics such as earned value (covered in this course module) be collected, calculated, and monitored on a routine basis. Risk tracking should be included as part of the normal, ongoing project tracking and control activities.

Software Project Risk Management

References

[Carr, 1993] M. Carr, S. Konda, I Monarch, F. Ulrich, C. Walker, **Taxonomy-Based Risk Identification**, Technical Report CMU/SEI-93-TR-6, Software Engineering Institute, Carnegie-Mellon University, 1993.

[Demasco, 2012] J. Demasco, **Effective Software Project Management**, seminar notes, 2012.

[Hall, 1998] E. Hall, **Managing Risk: Methods for Software System Development**, Addison-Wesley, 1998.