Risk Management in Software Development and Software Engineering Projects

Understanding Risk Management in Software Development

Software development is activity that uses a variety of technological advancements and requires high levels of knowledge. Because of these and other factors, every software development project contains elements of uncertainty. This is known as project risk. The success of a software development project depends quite heavily on the amount of risk that corresponds to each project activity. As a project manager, it's not enough to merely be aware of the risks. To achieve a successful outcome, project leadership must identify, assess, prioritize, and manage all of the major risks.

What Is Risk Management in Software Engineering?

Risk management means risk containment and mitigation. First, you've got to identify and plan. Then be ready to act when a risk arises, drawing upon the experience and knowledge of the entire team to minimize the impact to the project.

Risk management includes the following tasks:

- *Identify* risks and their triggers
- *Classify* and prioritize all risks
- Craft a *plan* that links each risk to a mitigation
- *Monitor* for risk triggers during the project
- Implement the *mitigating action* if any risk materializes
- *Communicate* risk status throughout project

Identify and Classify Risks

Most software engineering projects are inherently risky because of the variety potential problems that might arise. Experience from other software engineering

The goal of most software development and software engineering projects is to be distinctive—often through new features, more efficiency, or exploiting advancements in software engineering. Any software project executive will agree that the pursuit of such opportunities cannot move forward without risk.

Because risks are painfully real and quite prevalent on all software projects, it's critically necessary that stakeholders work hard to identify, understand, and mitigate any risks that might threaten the success of a project. For projects that have time and cost constraints, our experience shows most clearly that successful software development efforts are those in which risk mitigation is a central management activity.

What Is Risk In Software Engineering?

Very simply, a risk is a *potential* problem. It's an activity or event that may compromise the success of a software development project. Risk is the possibility of suffering loss, and total risk exposure to a specific project will account for both the *probability* and the *size* of the potential loss.

Guesswork and crisis-management are never effective. Identifying and aggregating risks is the only predictive method for capturing the probability that a software development project will experience unplanned or inadmissible events. These include terminations, discontinuities, schedule delays, cost underestimation, and overrun of project resources

projects can help managers classify risk. The importance here is not the elegance or range of classification, but rather to precisely identify and describe all the *threats to project success*. A simple but effective classification scheme is to arrange risks according to the areas of impact.

Five Types of Risk In Software Project Management

For most software development projects, we can define five main risk impact areas:

- New, unproven technologies
- User and functional requirements
- Application and system architecture
- Performance
- Organizational

New, unproven technologies. The majority of software projects entail the use of new technologies. Ever-changing tools, techniques, protocols, standards, and development systems increase the probability that technology risks will arise in virtually any substantial software engineering effort. Training and knowledge are of critical importance, and the improper use of new technology most often leads directly to project failure.

User and functional requirements. Software requirements capture all user needs with respect to the software system features, functions, and quality of service. Too often, the process of requirements definition is lengthy, tedious, and complex. Moreover, requirements usually change with discovery, prototyping, and integration activities. Change in elemental requirements will likely propagate throughout the entire project, and modifications to user requirements might not translate to functional requirements. These disruptions often lead to one or more critical failures of a poorly planned software development project.

Application and system architecture. Taking the wrong direction with a platform, component, or architecture can have disastrous consequences. As with the technological risks, it is vital that the team includes experts who understand the architecture and have the capability to make sound design choices.

Performance. It's important to ensure that software engineering projects don't disappoint user and partner expectations on performance. Consideration must be given to benchmarks and threshold testing throughout the project to ensure that the work products are moving in the right direction.

Organizational. Organizational problems may have adverse effects on project outcomes. Project management must plan for efficient execution of the project, and find a balance between the needs of the development team and the expectations of the customers. Of course, adequate staffing includes choosing team members with skill sets that are a good match with the project.

Risk Management Plan

After cataloging all of the risks according to type, the software development project manager should craft a risk management plan. As part of a larger, comprehensive project plan, the risk management plan outlines the response that will be taken for each risk—if it materializes.

Monitor and Mitigate

To be effective, software risk monitoring has to be integral with most project activities. Essentially, this means frequent checking during project meetings and critical events.

Monitoring includes:

- Publish project status reports and include risk management issues
- Revise risk plans according to any major changes in project schedule
- Review and reprioritize risks, eliminating those with lowest probability
- Brainstorm on potentially new risks after changes to project schedule or scope

When a risk occurs, the corresponding mitigation response should be taken from the risk management plan.

Mitigating options include:

- **Accept:** Acknowledge that a risk is impacting the project. Make an explicit decision to accept the risk without any changes to the project. Project management approval is mandatory here.
- **Avoid:** Adjust project scope, schedule, or constraints to minimize the effects of the risk.
- **Control:** Take action to minimize the impact or reduce the intensification of the risk.
- **Transfer:** Implement an organizational shift in accountability, responsibility, or authority to other stakeholders that will accept the risk.

- **Continue Monitoring:** Often suitable for low-impact risks, monitor the project environment for potentially increasing impact of the risk.

Communicate

Throughout the project, it's vital to ensure effective communication among all stakeholders, managers, developers, QA—especially marketing and customer representatives. Sharing information and getting feedback about risks will greatly increase the probability of project success.

Summary

Risk management is an extensive discipline, and we've only given an overview here. We leave you with a checklist of best practices for managing risk on your software development and software engineering projects.

- Always be forward-thinking about risk management. Otherwise, the project team will be driven from one crisis to the next.
- Use checklists, and compare with similar previous projects.
- Prioritize risks, ranking each according to the severity of exposure.
- Develop a top-10 or top-20 risk list for your project. Like most project managers, you can probably reuse this list on the next project!
- Vigorously watch for surfacing risks by meeting with key stakeholders—especially with the marketing team and the customer.
- As practicable, split larger risks into smaller, easily recognizable and readily-manageable risks.
- Strongly encourage stakeholders to think proactively and communicate about risks throughout the entire project.

Interested in closer look at Software Intelligence?

Schedule your Personal Tour (https://content.castsoftware.com/get-a-free-demo-of-cast-int

PRODUCTS

CAST Highlight (/products/highlight)

CAST Imaging (/products/imaging)

Add-on Modules (/products/imaging#add-on-modules)

LEARN MORE

Portfolio Rationalization for Cloud (/rapid-portfolio-analysis-for-cloud-migration)

Automated System Weakening (/automated-system-refactoring)

Greater Engineering Speed & Efficiency (/greater-engineering-speed-and-efficiency)

Software Composition Analysis (/software-supply-chain-control)

Automated M&A Technical Due Diligence (/automating-technical-due-diligence-mergers-acquisitions)

Software Supply Chain Transparency (/software-supply-chain-transparency)

ISO 5055 standard (/blog/iso-5055-standard-explained-in-your-software-projects&efficiency-and-safety)

RESOURCES

Research Library (/resources/research-library)

Education Library (/resources/education-library)

Video Interviews (/resources/education-library/video-interviews)

Upcoming Events (/resources/upcoming-events)

Recorded Webinars (/resources/events-library)

Support (https://help.castsoftware.com/)

ABOUT US

Why CAST (/discover-cast/why-cast)

Management Team (/discover-cast/management-team)

Careers (/discover-cast/join)

Investor Relations (/discover-cast/Investors)

Press Releases (/discover-cast/press-releases)

Contact Us (/discover-cast/contact-us)

International Presence (/discover-cast/worldwide)

(http://www.facebook.com/castonquality)

(http://twitter.com/onquality)

(http://www.linkedin.com/company/cast)

(https://www.youtube.com/channel/UCx_BN1Mr5gyYLh71yH3xsw)

(http://www.facebook.com/share.php?u=https://www.castsoftware.com/research-labs/risk-management-in-software-development-and-software-engineering-projects)

(https://twitter.com/intent/tweet?url=https://www.castsoftware.com/research-labs/risk-management-in-software-development-and-software-engineering-projects&original_referer=https://www.castsoftware.com/research-labs/risk-management-in-software-development-and-software-engineering-projects)

(http://www.linkedin.com/shareArticle?mini=true&url=https://www.castsoftware.com/research-labs/risk-management-in-software-development-and-software-engineering-projects)

(mailto:?subject=Check out https://www.castsoftware.com/research-labs/risk-management-in-software-development-and-software-engineering-projects&body=Check out https://www.castsoftware.com/research-labs/risk-management-in-software-development-and-software-engineering-projects)