

Chapter 25

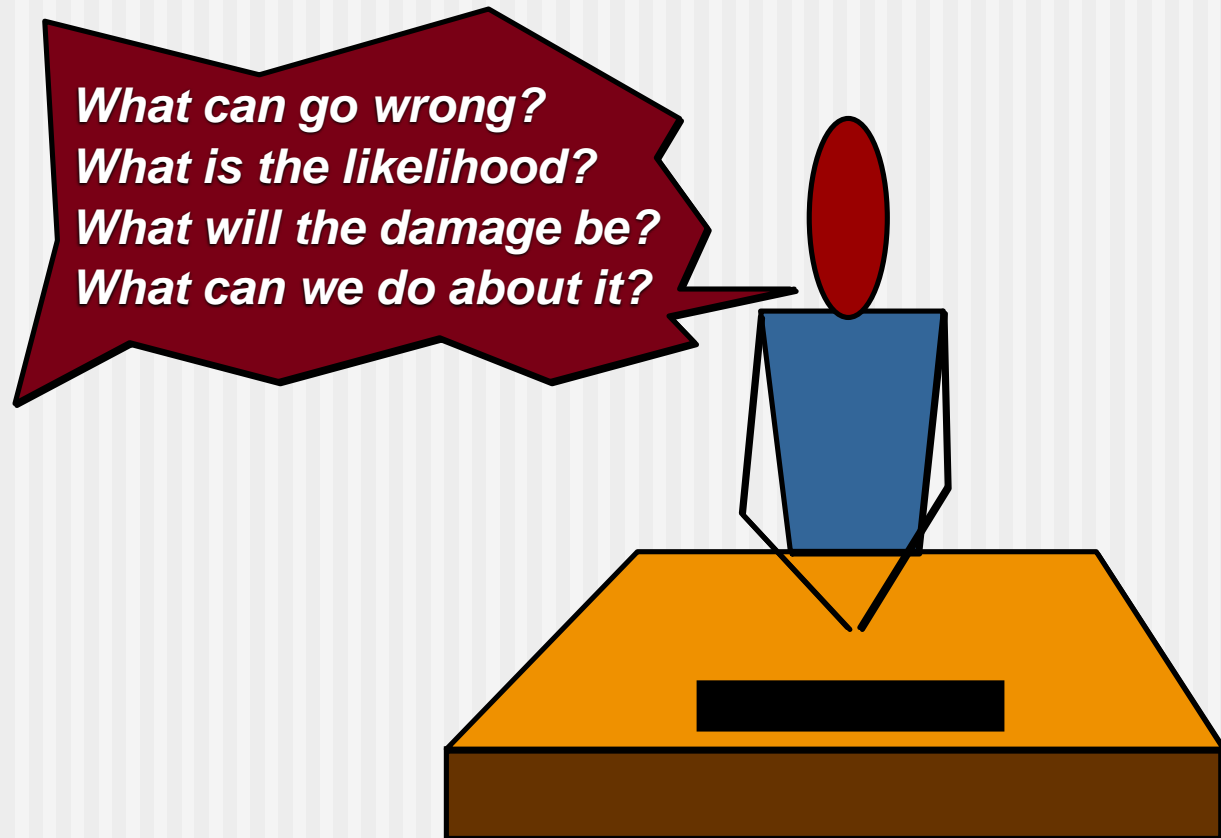
■ Risk Management

Software Engineering: A Practitioner's Approach, 6/e
by Roger S. Pressman

Slides copyright © 1996, 2001, 2005 by Roger S. Pressman

Introduction

Project Risks



Definition of Risk

- A risk is a potential problem – it might happen and it might not
- Conceptual definition of risk
 - Risk concerns future happenings
 - Risk involves change in mind, opinion, actions, places, etc.
 - Risk involves choice and the uncertainty that choice entails
- Two characteristics of risk
 - **Uncertainty** – the risk may or may not happen, that is, there are no 100% risks (those, instead, are called constraints)
 - **Loss** – the risk becomes a reality and unwanted consequences or losses occur

Reactive vs. Proactive Risk Strategies

- **Reactive** risk strategies
 - "Don't worry, I'll think of something"
 - The majority of software teams and managers rely on this approach
 - **Nothing is done about risks until something goes wrong**
 - The team then flies into action in an attempt to correct the problem rapidly (fire fighting)
 - Crisis management is the choice of management techniques
- **Proactive** risk strategies
 - **Steps for risk management are followed** (see next slide)
 - Primary objective is to avoid risk and to have a contingency plan in place to handle unavoidable risks in a controlled and effective manner

Risk Categorization – Approach #1

- **Project risks**

- They threaten the project plan
- If they become real, it is likely that the project schedule will slip and that costs will increase

- **Technical risks**

- They threaten the quality and timeliness of the software to be produced
- If they become real, implementation may become difficult or impossible

- **Business risks**

- They threaten the viability of the software to be built
- If they become real, they jeopardize the project or the product

Risk Categorization – Approach #1 (continued)

- Sub-categories of Business risks
 - **Market risk** – building an excellent product or system that no one really wants
 - **Strategic risk** – building a product that no longer fits into the overall business strategy for the company
 - **Sales risk** – building a product that the sales force doesn't understand how to sell
 - **Management risk** – losing the support of senior management due to a change in focus or a change in people
 - **Budget risk** – losing budgetary or personnel commitment

Risk Categorization – Approach #2

- **Known risks**

- Those risks that can be uncovered after careful evaluation of the project plan, the business and technical environment in which the project is being developed, and other reliable information sources (e.g., unrealistic delivery date)

- **Predictable risks**

- Those risks that are extrapolated from past project experience (e.g., past turnover)

- **Unpredictable risks**

- Those risks that can and do occur, but are extremely difficult to identify in advance

Steps for Risk Management

- 1) **Identify** possible risks; recognize what can go wrong
- 2) **Analyze** each risk to estimate the probability that it will occur and the impact (i.e., damage) that it will do if it does occur
- 3) **Rank** the risks by probability and impact
 - Impact may be **negligible**, **marginal**, **critical**, and **catastrophic**
- 4) **Develop** a contingency plan to manage those risks having high probability and high impact

Risk Management Paradigm



Risk Identification

Background

- Risk identification is a systematic attempt to specify threats to the project plan
- By identifying known and predictable risks, the project manager takes a first step toward avoiding them when possible and controlling them when necessary
- **Generic risks**
 - Risks that are a potential threat to every software project
- **Product-specific risks**
 - Risks that can be identified only by those a with a clear understanding of the technology, the people, and the environment that is specific to the software that is to be built
 - This requires examination of the project plan and the statement of scope
 - "What special characteristics of this product may threaten our project plan?"

Known and Predictable Risk Categories

- *Product size* — risks associated with the overall size of the software to be built or modified.
- *Business impact* — risks associated with constraints imposed by management or the marketplace.
- *Customer characteristics* — risks associated with the sophistication of the customer and the developer's ability to communicate with the customer in a timely manner.
- *Process definition* — risks associated with the degree to which the software process has been defined and is followed by the development organization.
- *Development environment* — risks associated with the availability and quality of the tools to be used to build the product.
- *Technology to be built* — risks associated with the complexity of the system to be built and the "newness" of the technology that is packaged by the system.
- *Staff size and experience* — risks associated with the overall technical and project experience of the software engineers who will do the work.

Risk Projection (Estimation)

Risk Projection

- ***Risk projection***, also called ***risk estimation***, attempts to rate each risk in two ways
 - the likelihood or probability that the risk is real
 - the consequences of the problems associated with the risk, should it occur.
- **The are four risk projection steps:**
 - establish a scale that reflects the perceived likelihood of a risk
 - delineate the consequences of the risk
 - estimate the impact of the risk on the project and the product,
 - note the overall accuracy of the risk projection so that there will be no misunderstandings.

Building a Risk Table

Risk	Probability	Impact	RMMM
			Risk Mitigation Monitoring & Management

Building the Risk Table

- Estimate the **probability** of occurrence
- Estimate the **impact** on the project on a scale of 1 to 4, where
 - 1 = catastrophic impact on project success
 - 4 = low impact on project success
- sort the table by probability and impact

Assessing Risk Impact

- The overall risk exposure formula is

$$RE = P \times C$$

- P = the probability of occurrence for a risk
- C = the cost to the project should the risk actually occur
- Example
 - P = 80% probability that 18 of 60 software components will have to be developed
 - C = Total cost of developing 18 components is \$25,000
 - $RE = .80 \times \$25,000 = \$20,000$

Risk Mitigation, Monitoring, and Management

Background

- An effective strategy for dealing with risk must consider three issues
(Note: these are not mutually exclusive)
 - Risk mitigation (i.e., avoidance)
 - Risk monitoring
 - Risk management and contingency planning
- Risk mitigation (avoidance) is the primary strategy and is achieved through a plan
 - Example: Risk of high staff turnover (see next slide)

Background (continued)

Strategy for Reducing Staff Turnover

- ❑ Meet with current staff to determine causes for turnover (e.g., poor working conditions, low pay, competitive job market)
- ❑ Mitigate those causes that are under our control before the project starts
- ❑ Once the project commences, assume turnover will occur and develop techniques to ensure continuity when people leave
- ❑ Organize project teams so that information about each development activity is widely dispersed
- ❑ Define documentation standards and establish mechanisms to ensure that documents are developed in a timely manner
- ❑ Conduct peer reviews of all work (so that more than one person is "up to speed")
- ❑ Assign a backup staff member for every critical technologist

Background (continued)

- During risk monitoring, the project manager monitors factors that may provide an indication of whether a risk is becoming more or less likely
- Risk management and contingency planning assume that mitigation efforts have failed and that the risk has become a reality
- RMMM steps incur additional project cost
 - Large projects may have identified 30 – 40 risks
- Risk is not limited to the software project itself
 - Risks can occur after the software has been delivered to the user

Background (continued)

- **Software safety and hazard analysis**

- These are software quality assurance activities that focus on the identification and assessment of potential hazards that may affect software negatively and cause an entire system to fail
- If hazards can be identified early in the software process, software design features can be specified that will either eliminate or control potential hazards

Summary

- Whenever much is riding on a software project, common sense dictates risk analysis
 - Yet, most project managers do it informally and superficially, if at all
- However, the time spent in risk management results in
 - Less upheaval during the project
 - A greater ability to track and control a project
 - The confidence that comes with planning for problems before they occur
- Risk management can absorb a significant amount of the project planning effort...but the effort is worth it