Unit 3: Risk Management

Objectives
- To explain the concept of risk & to develop its role within the software development process
- To introduce the use of risk management as a means of identifying & controlling risk in software development

What is risk?

It is not just a game!

Definitions of risk
- "The possibility of suffering harm or loss; danger"
- "The possibility of loss or injury"
- "Chance of danger, injury, loss"
- "A measure of the probability & severity of adverse effects"

Risks in the everyday world
- Financial risks - "your house is at risk if you fail to repay your mortgage or any loans secured on it"
- Health risks - "the chance that a person will encounter a specified adverse health outcome (like die or become disabled)"
- Environmental & ecological risks - "the likelihood of extinction due to exposure of terrestrial wildlife to contaminants"
- Security risks - "there is a significant risk that widespread insertion of government-access key recovery systems into the information infrastructure will exacerbate, not alleviate, the potential for crime and information terrorism"

How is risk dealt with?
- Basic process: identify the risk -> analyse its implications -> determine treatment methods -> monitor performance of treatment methods
- Techniques & heuristics for the identification, analysis, treatment & monitoring of risk
- Insurance companies depend on understanding risk

Risk management is a project management tool to assess & mitigate events that might adversely impact a project, thereby increasing the likelihood of success
Why is the software world interested in risk?

- Many post-mortems of software project disasters indicate that problems would have been avoided (or strongly reduced) if there had been an explicit early concern with identifying & resolving high-risk elements!
- An obvious cost factor!

Successful project managers are good risk managers!

Sources of software risk (systems context)

- Technology
- Hardware
- Software
- SYSTEM
- Schedule
- Cost
- People

Reproduced from [Higuera 1996]
CMU/SEI-96-TR-012, ESC-TR-96-012, June 1996

Why is it often forgotten?

- Optimistic enthusiasm at the start of projects
- Software process can lead to over-commitment & binding requirements too early on
- Premature coding
- The “add-on” syndrome
- Warning signals are missed
- Legal implications
- Poor software risk management by project managers

Software risk management

- Objectives
  - To identify, address & eliminate risk items before they become either threats to successful software operation or major sources of software rework
  - Necessary that some form of measurement is undertaken to determine & classify the range of risks a software development project faces, & to identify areas where a significant exposure exists
  - The discipline attempts to provide a set of principles & practices to achieve the above
  - A response to change & uncertainty

The need to manage risk

- Methods, tools & processes
  - Expert knowledge, judgement & experience
  - Individual knowledge, judgement & experience

The questions

- What can go wrong?
- What is the likelihood it will go wrong?
- What are the consequences?
- What can be done?
- What options are available?
Software risk management steps & techniques

Risk management
- Risk identification
- Risk assessment
- Risk control
- Risk resolution
- Risk monitoring
- Risk-plan integration
- Quality-factor analysis
- Buying information
- Risk management planning
- Risk resolution
- Risk monitoring
- Top 10 tracking
- Risk reassessment
- Corrective action

Risk assessment
- Risk identification - listing project-specific risk items that are likely to compromise a project's success
- Risk analysis - assessing the loss probability & loss magnitude for each identified risk item, & assessing compound risks
- Risk prioritisation - ordering & ranking the risk items identified & analysed

Risk control
- Risk-management planning - doing the ground work so as to be in a position to address each risk item
- Risk resolution - producing a situation in which risk items are eliminated or resolved
- Risk monitoring - tracking the project's progress towards resolving risk items & taking corrective action where required

E.g. top 10 risks in software project mgmt
1. Personnel shortfalls
2. Unrealistic schedules & budgets
3. Developing the wrong functions & properties
4. Developing the wrong user interface
5. Gold-plating
6. Continuing stream of requirements changes
7. Shortfalls in externally furnished components
8. Shortfalls in externally performed tasks
9. Real-time performance shortfalls
10. Straining computer-science capabilities

E.g. project sizing matrix

E.g. prioritisation scheme
- Risk-exposure quantity an effective technique for risk prioritisation
- Assess risk probabilities & losses on a scale 0-10
- Multiply probability by loss to determine exposure

Risk exposure quantity
- Unsatisfactory outcome: Probability of unsatisfactory outcome: Loss caused by unsatisfactory outcome: Risk exposure
- CPU error from key data: 30 - 3 - 90
- Process or memory crash: 5 - 7 - 35

- Relies on accurate estimates of the probability & loss associated with an unsatisfactory outcome
E.g. risk management plan

- The Risk Management Plan (RMP) presents the process for implementing proactive risk management as part of overall project management.
- The RMP describes techniques for identifying, analysing, prioritising, and tracking risks; developing risk-handling methods; and planning for adequate resources to handle each risk, should they occur.
- The RMP also assigns specific risk management responsibilities and describes the documenting, monitoring, and reporting processes to be followed.

Ways of dealing with risks

- Elimination: where exposure to risk is terminated.
- Retention: where the risk is made tolerable, perhaps after some modification.
- Avoidance: where the risk is negated in some way, possibly by redesign of work methods.
- Transfer: where the risk is passed to a third party, either contractually or via insurance.
- Need to balance acceptable risks.

Putting risk management into practice

- Insert risk management principles and practices into your software development process, so they are risk-oriented and risk-driven—do this gradually and incrementally.
- Start with a top 10 risk-item tracking process—lightweight, cheap & good returns!
- Develop a WWWWWW! RMP template to populate.
- Not a prescription—relies on good human judgement.
- A focus on CSFs can help you win work.

E.g. PMP summarised as a risk register

Implement & track

- An ongoing process of measuring the effect that implementation of a risk management programme has had & its ability to continue.
- Focus on the high-risk, high-leverage critical success factors:
  - Rank a project’s most significant risk items (prepare).
  - Establish a regular schedule for review of progress (meet).
  - Summarise progress on top risk items (discuss).
  - Focus on handling any problems in resolving the risk items (act).

The BIGGEST risk?

Not knowing what the risks are!
### Key points

- The enemy of the software manager is risk.
- Software projects must manage risks to minimize their consequences.
- Time spent identifying, analyzing & managing risk pays off!
- You can use the 6 stage conceptual framework with its associated techniques as a solid starting point.
- If nothing else, be risk aware...

### Core references


### Supplementary references


*You are strongly advised to read one of these!*

*LOTS of general risk info on the web!*