Risk Management (3C05/D22)

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"

Probability/ uncertainty

Something bad happening

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"

More examples?

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

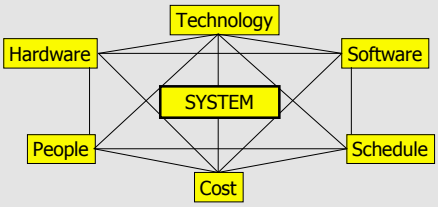
Why is the software world interested in risk?

- 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!

Browse the forum on "Risks To The Public In Computers & Related Systems" http://catless.ncl.ac.uk/Risks

Successful project managers are good risk managers!

Sources of software risk (systems context)



Reproduced from [Higuera 1996] "Software Risk Management", Technical Report CMU/SEI-96-TR-012, ESC-TR-96-012, June 1996

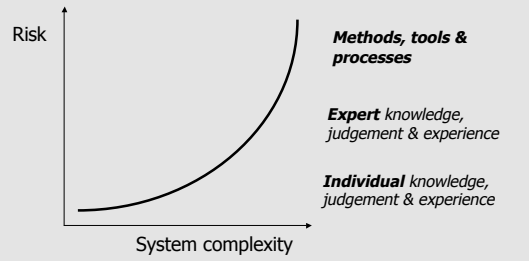
Why is it often forgotten?

- Optimistic enthusiasm at the start of projects
- Software process can lead to over-commitment & binding requirements much 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



Reproduced from [Higuera 1996]

The questions

What can go wrong?

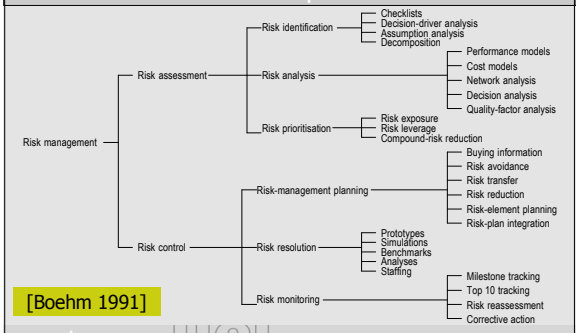
What is the likelihood it will go wrong?

What are the consequences?

What can be done?

What options are available?

Software risk management steps & techniques



[Boehm 1991]

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

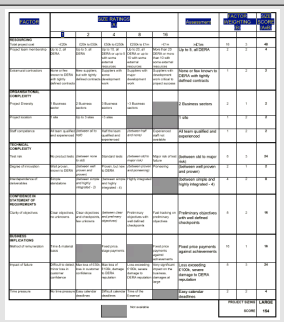
Top 10 risks in software project mgmt

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

[Boehm 1991]

Determine a risk-management technique to deal with each of these!

E.g. project sizing matrix



[Used@QuinetQ]

Always a question of balance - full risk analysis may not improve risk probability estimation significantly!

E.g. prioritisation scheme

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

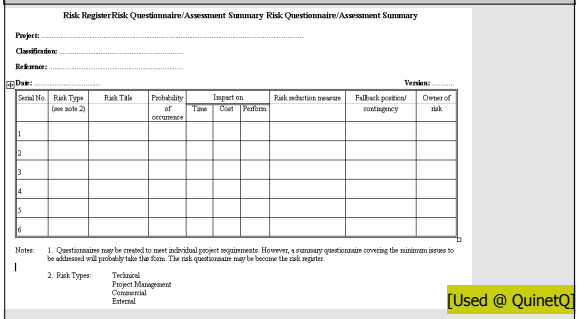
Unsatisfactory outcome	Probability of unsatisfactory outcome	Loss caused by unsatisfactory outcome	Risk exposure
Software error loses key data	3-5	8	24-40
Processor memory insufficient	1	7	7

- 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 & tracking risks; developing risk-handling methods; & planning for adequate resources to handle each risk, should they occur
- The RMP also assigns specific risk management responsibilities & describes the documenting, monitoring & reporting processes to be followed

PMP summarised as a risk register



Risk Register/Risk Questionnaire/Assessment Summary Risk Questionnaire/Assessment Summary

Project: .....

Classification: .....

Reference: .....

Date: ..... **Version:** .....

Serial No.	Risk Type (see note 2)	Risk Title	Probability of occurrence	Impact on			Risk reduction measure	Fallback position/ contingency	Owner of risk
				Time	Cost	Perform			
1									
2									
3									
4									
5									
6									

Notes: 1. Questionnaires may be created to meet individual project requirements. However, a summary questionnaire covering the minimum issues to be addressed will probably take this form. The risk questionnaire may be become the risk register.

2. Risk Types: Technical
Project Management
Commercial
External

[Used @ QuinetQ]

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

Implement & ……. track

- An on-going 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)

UCL ComputerScience

Putting risk management into practice

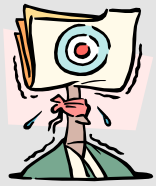
- Insert risk management principles & practices into your software development process, so they are risk-oriented & risk-driven - do this gradually & incrementally
- Start with a top 10 risk-item tracking process - lightweight, cheap & good returns!
- Develop a WWWWWHHM RMP template to populate
- Not a prescription - relies on *good* human judgement!

A focus on CSFs can help you win work!

UCL ComputerScience

The BIGGEST risk?

Not knowing what the risks are!



UCL ComputerScience

Key points

- The enemy of the software manger is risk
- Software projects must manage risks to minimise their consequences
- Time spent identifying, analysing & 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

- B. W. Boehm, "Software Risk Management: Principle and Practices," IEEE Software, Vol. 8, No. 1, January 1991, pp. 32-41
- Roger Pressman, "Software Engineering: A Practitioner's Approach", McGraw-Hill, 5th edition, ISBN: 0-07-709677-0 (Chapter 6)
  - *Contains pointers to lots more refs*

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

- Ian Sommerville, "Software Engineering", Addison-Wesley, 6th Edition, ISBN: 0-201-39815-X (Chapter 4.4)

Supplementary references

- P. G. Neumann, "Computer Related Risks", ACM Press, 1995
- J. Adams, "Risk", UCL Press, 1995

LOTS of general risk info on the web!

- B. W. Boehm, "Software Risk Management", CS Press, 1989
- Tom Gilb, "Principles of Software Engineering Management", Addison-Wesley, 1998, ISBN: 0-201-19246-2 (Chapter 6)
- IEEE Software - Special issues on Risk - May 1994 & May/June 1997