SWINBURNE UNIVERSITY OF TECHNOLOGY

SWE30010
Development Project 2: Design, Planning and Management

Lecture 9

Traditional Project Risk Management

Disclaimer

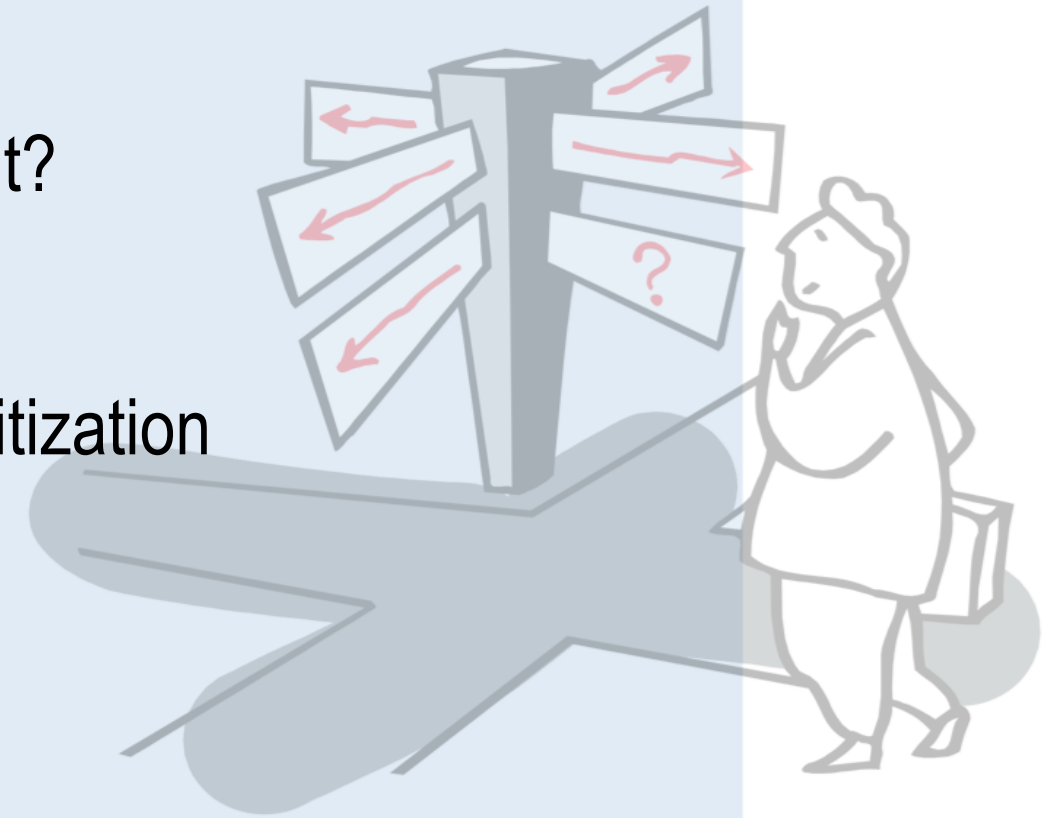
- http://creativecommons.org/licenses/by-sa/3.0/

Principal References

- Roger S. Pressman, *Software Engineering - A Practitioners Approach* (7th Edition), McGraw Hill, 2010, Chapter 28.
- Bob Hughes and Mike Cotterell, *Software Project Management* (4th Edition), Wiley, 2006, Chapter 7.
- Pankaj Jalote, *Software Project Management in Practice*, Addison-Wesley, 2002, Chapter 6.

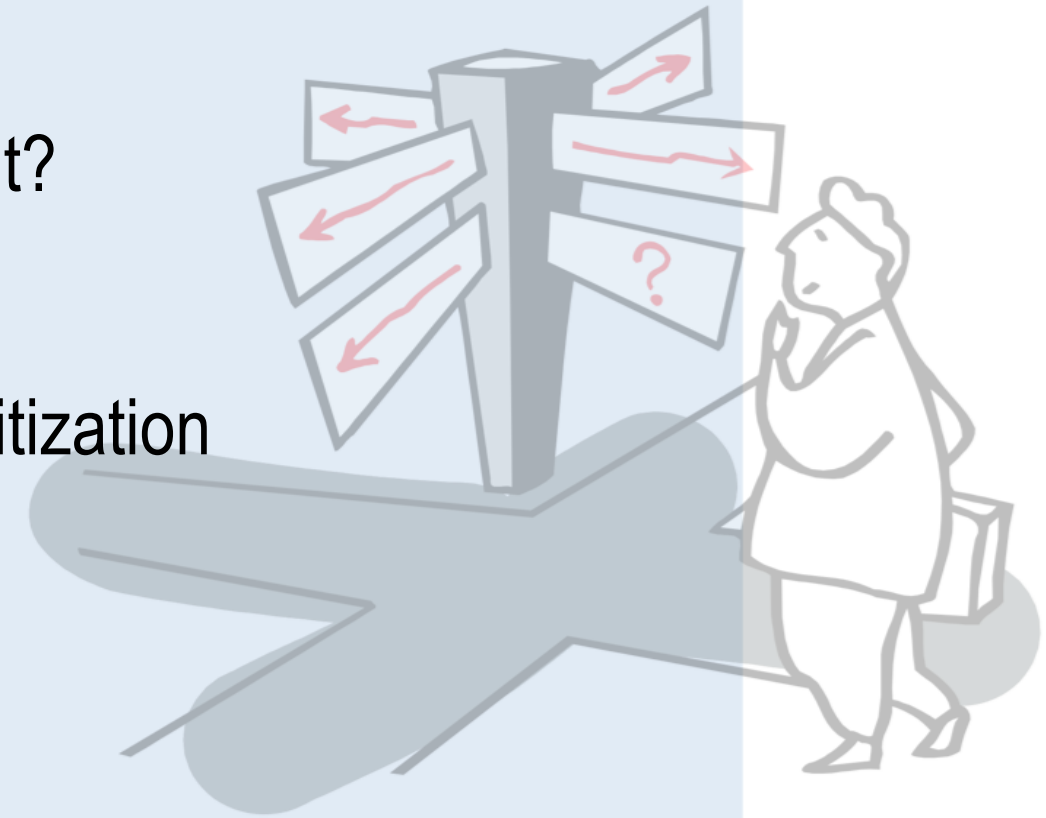
Roadmap

- What are Risks?
- What is Risk Management?
- Risk Identification
- Risk Estimation and Prioritization
- Risk Mitigation Strategies



Roadmap

- **What are Risks?**
- What is Risk Management?
- Risk Identification
- Risk Estimation and Prioritization
- Risk Mitigation Strategies



What are Risks?

"First, risk concerns future happenings. Today and yesterday are beyond active concern, as we are already reaping what was previously sowed by our past actions. The question is, therefore, by changing our actions today, create an opportunity for a differently and hopefully better situation for ourselves tomorrow. This means, second that risk involves change, such as in changes of mind, opinion, actions, or places. Third, risk involves choice, and the uncertainty that choice itself entails."

– Robert Charette, 1989

What are Risks (cont.)?

- PMBOK: *"an uncertain event or condition that, if it occurs, has a positive or negative effect on a project's objectives."*
- PRINCE2: *"the chance of exposure to the adverse consequences of future events."*

☞ Key elements:

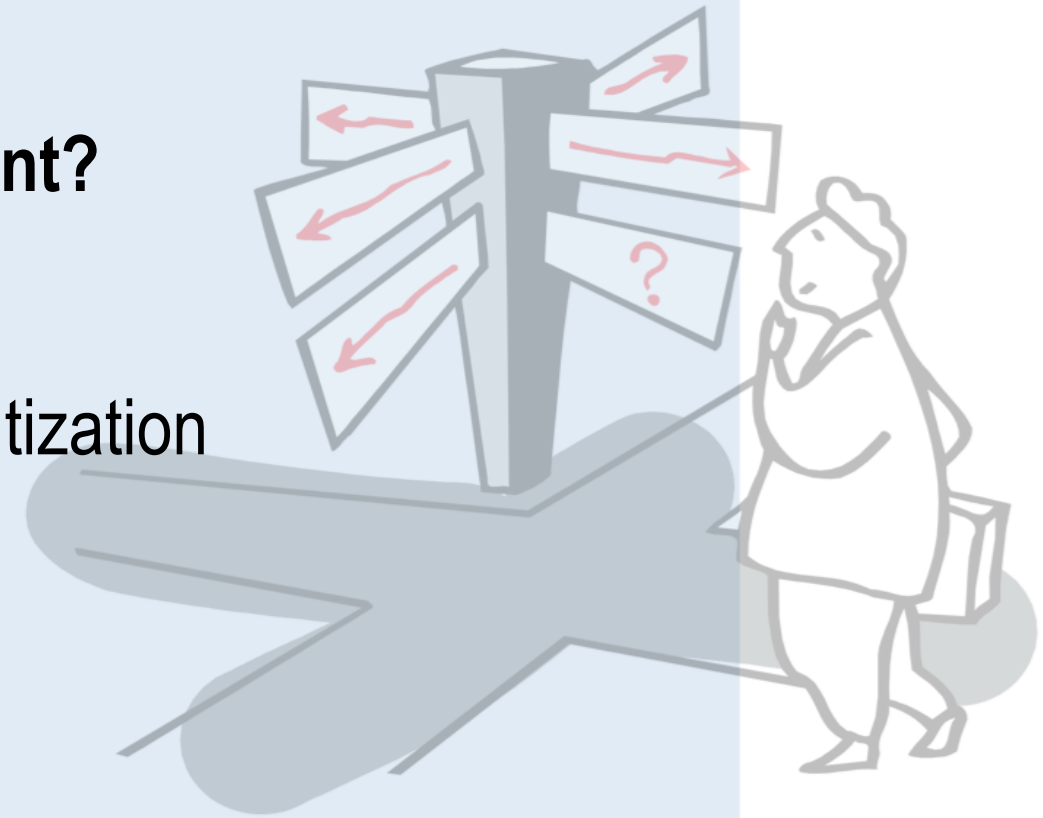
- ☐ Risks relate to the future (*"speculating about future events"*)
- ☐ Risks involve cause (*"why"*) and effect (*"measurable consequence"*).

“I never look back, dahling. It detracts from the now.”

– Edna Mode, The Incredibles

Roadmap

- What are Risks?
- **What is Risk Management?**
- Risk Identification
- Risk Estimation and Prioritization
- Risk Mitigation Strategies



Risk Management

Software project risk management defines the way to
identify, *analyse*, and *respond*
to software risks during the execution of a project.

Why Risk Management?

Be prepared!

Why Risk Management (cont.)?

Risk management

- has a positive impact on
  - Selecting projects
  - Determining the scope of projects
  - Developing realistic schedules and cost estimates
- helps project stakeholders to understand the nature of the project
  - and may give additional justifications why a project should *not* be undertaken!

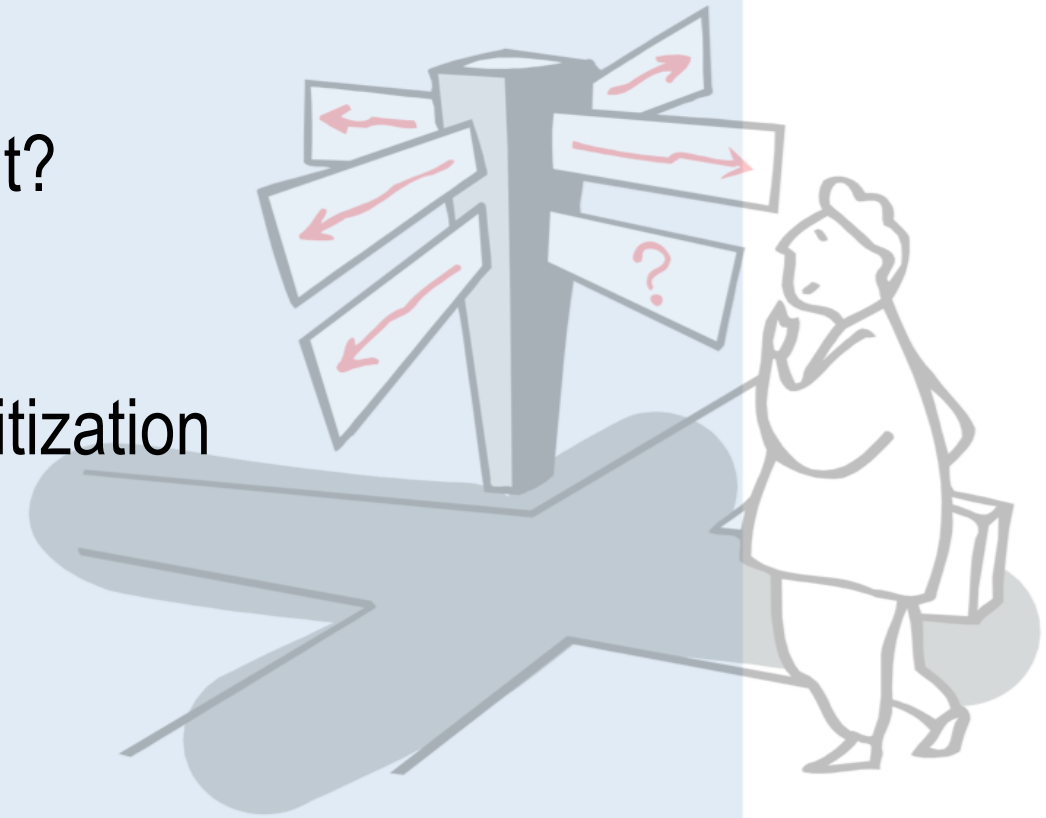
Risk Management Framework

- Risk assessment:
  - Risk identification
  - Risk analysis and estimation
  - Risk prioritization
- Risk mitigation/control:
  - Risk avoidance
  - Risk planning
  - Risk monitoring

(Source: P. Jalote, *Project Management in Practice*)

Roadmap

- What are Risks?
- What is Risk Management?
- **Risk Identification**
- Risk Estimation and Prioritization
- Risk Mitigation Strategies



Risk Identification

- Identification of the *hazards* that might affect the cost, duration or resources of a project:

  Hazard → Problem → Risk

- A hazard is an event that *might* occur and will create a problem for the successful completion of the project, if it *does* occur.
- Examples of hazards:
  - New, unproven technology
  - Unclear requirements
  - Lack of experience in problem domain
  - Overall size/complexity of problem

Risk Identification (cont.)

- Guidelines:
  - Use checklist that lists the potential hazards and their corresponding factors
  - Identify both, *cause* **and** *effect* of risks!
  - Maintain an updated checklist for future projects
  - Think of other things that may go wrong…

Knowledge

Information = Data + Meaning

Knowledge = Information + Processing (Domain Context)

Domain Knowledge: scopes context we are dealing with!

Risk Drivers

Principal risk drivers (KoST):

- Knowledge Gap (*don't know*)
- Skill Gap (inexperience)
- Technology Gap (unknown/young or unavailable)

Other important risk drivers:

- Team Dynamics + Management
- Research & Development Component

Risk Causes

The two most common causes for project failures are:

- Problem framing (solving “wrong” problem)
- Project approach (methodology, resources, processes etc.)

Other, more “traditional” risks causes include:

- Project planning:
  - budget, schedule, resources, size, personnel, morale, ...
- Business:
  - market, sales, management, commitment, ...

Project vs. Business Risks

- Typically, a *project risk*, if it occurs, will threaten a project's cost and schedule.
- A *business risk* will threaten the viability of the software to be built.
  - Examples:
    - Building a product that no one really wants.
    - Building a product that does not fit into the overall business strategy of the organization (any more).
    - Losing support from senior management due to a change in focus or a change in people.
    - Losing budgetary or personnel commitment.

Barry Boehm's Top Ten Risks

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

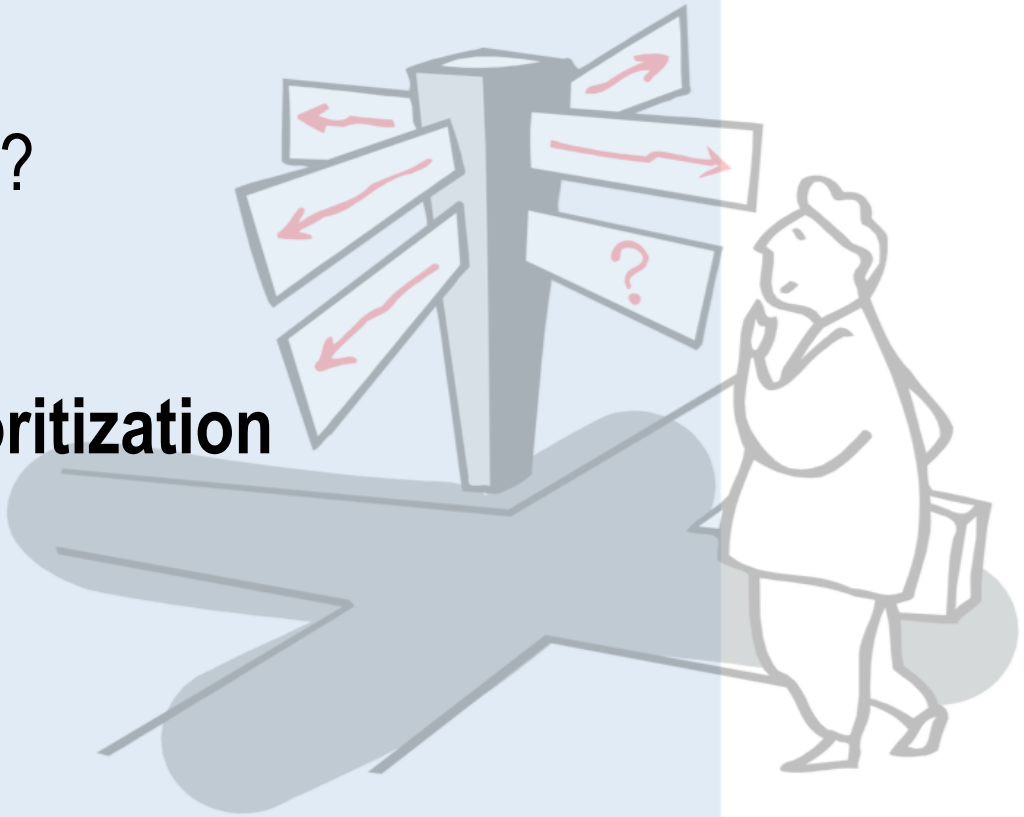
Other Common Risks

- *Lack of communication*
  - ☐ within team
  - ☐ between team and client/customer
- Lack of resources/time for *testing*
  - ☞ or Quality Assurance in general
- Lack of trust
- Development platform vs. deployment platform
- “Uninformed” decision making (about technology etc.)
- “Heroism”

“The Truck Factor”

Roadmap

- What are Risks?
- What is Risk Management?
- Risk Identification
- **Risk Estimation and Prioritization**
- Risk Mitigation Strategies



Risk Exposure

- Risk *probability*
  - also known as *rate of occurrence*
- Risk *impact*
  - severity of the consequences
- **Risk Exposure = Risk Impact * Risk Probability**

☞ *No "real" risk management can be performed unless both, probability and impact of a risk is known!*

Risk Exposure (cont.)

- Advantages
  - Risk exposure provides a way to compare or *rank* risks
  - Having a good *quantitative* estimate of the risk exposure, the extra effort can provide a better understanding of the problem
- Disadvantages
  - Difficult to have good estimation
  - Estimation is *subjective* and often time-consuming
  - Risk exposure alone neglects *dependencies* between risks!
    - but this goes beyond the scope of an introduction into risk management

Risk Estimation

- Risk probability
  - Use appropriate ranking criteria
    - E.g., Low, Moderate, High, Very High
    - E.g., scale from 1 (least likely) to 10 (most likely)
    - ☞ Do not use a too fine-grained scale (uncertainty factor!)
- Risk impact
  - Define suitable units of impact
    - E.g., Insignificant, Tolerable, Serious, Catastrophic
    - E.g., scale from 1 (insignificant) to 10 (catastrophic)
    - ☞ Again, do not use a too fine-grained scale…
- Use experience data (can be difficult!)
- Define “suitable” values for probability * impact
  - What does “Moderate” * “Tolerable” mean??

Risk Estimation – Example 1

Risk Item	Likelihood	Impact	Exposure
It is impossible to recruit staff with suitable skills for the project	0.70	200,000	140,000
Organizational financial problems force to cut down the project budget	0.20	100,000	20,000
The time required to develop th		40,000	26,000
Software components that shou which make the system unrelia		25,000	10,000
The DBMS component cannot process as many requests as it is expected	0.45	20,000	9,000
Customers fail to understand the impact of requirements changes	0.20	15,000	3,000
User training is postponed	0.10	1,000	100

Where do the "magic" numbers come from?

Adapted from Fig. 5.12 (Sommerville, 2007)

Risk Estimation – Example 2

Risk Item	Likelihood	Impact
It is impossible to recruit staff with suitable skills for the project	High	Catastrophic
Organizational financial problems force to cut down the project budget	Low	Catastrophic
The time required to develop th		Serious
Software components that shou which make the system unrelia		Serious
The DBMS component cannot process as many requests as it is expected	Moderate	Serious
Customers fail to understand the impact of requirements changes	Low	Tolerable
User training is postponed	Low	Insignificant

Is it impossible to quantify the risk items?

Risk Estimation – Example 2 (cont.)

	Low (1)	Moderate (2)	High (3)	Very High (5)
Insignificant (1)	1	2	3	5
Tolerable (2)	2	4	6	10
Serious (3)	3	6	9	15
Catastrophic (5)	5	10	15	25

Risk Prioritisation

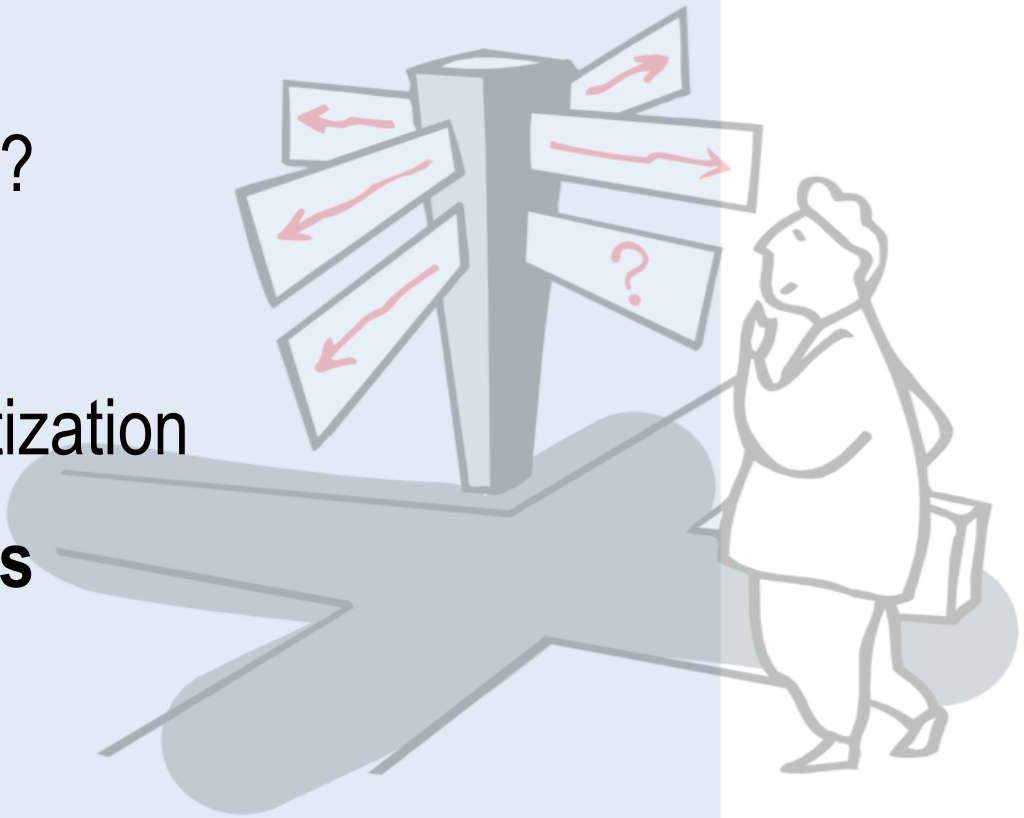
- Ranking risks
  - Rank the risks based on their *risk exposure*
  - Note: ranking only shows the *order of importance*
  - Need to consider factors like
    - Confidence of the risk assessment
    - Total number of risks
    - Cost of action (cf. next slide)
- Select top "few" risks (e.g., Top 10)
  - ☞ Risks that have a *direct impact on project goals and objectives*!
  - ☞ Risks that require *risk control/mitigation strategies*
- ☞ *Risks are not static, they will change over time!*
  - ☞ *This includes both, probability and impact*

Cost of Action

- Risk management is *not for free*, there are costs associated with mitigation strategies
  - E.g., to prevent a (short-term) power failure, need to acquire a UPS (Universal Power Supply)
- Cost-Benefit analysis
  - Does the cost associated with the mitigation strategy merit its implementation?
  - E.g., cost of risk exposure of power failure smaller than cost of UPS, no need for mitigation strategy.
- If a mitigation strategy costs too much
  - Look for an alternative
  - If not possible, carefully monitor the risk!

Roadmap

- What are Risks?
- What is Risk Management?
- Risk Identification
- Risk Estimation and Prioritization
- **Risk Mitigation Strategies**



Risk Mitigation Strategies

- 4 different types of mitigation strategies in a generic sense
  - Risk prevention/avoidance
  - Likelihood reduction
  - Impact reduction
  - Risk transfer
- If above fails to address risk
  - Contingency planning

☞ *Sometimes, the distinction between them are fuzzy*

Risk Prevention

- Prevent a hazard from occurring or reduce its likelihood to an insignificant level
- Examples
  - Use of unknown technology can be prevented by choosing existing, proven technology
  - Unclear/ambiguous requirements can be prevented by using formal requirement specification techniques
  - (Too) frequent changes of requirements can be managed by using a time-boxed SDLC methodology

Likelihood Reduction

- Reduce the likelihood of an unavoidable risk by prior planning
- Examples
  - Choosing a wrong technology can be mitigated by spikes/prototypes
  - Decline of team morale (resulting in lower productivity) can be reduced by providing free coffee, staff BBQs, etc.

Impact Reduction

- Reduce the impact of an unavoidable risk by adding "buffers"
- Examples
  - Reduce the impact of the "Truck Factor" by *distributing knowledge amongst team members*
  - Reduce impact of "develop the system wrong" by having more than one team developing a system ("NASA Principle")
    - Traditional jargon – N-version programming

Risk Transfer

- The impact of the risk can be transferred away from the project by contracting out or taking out insurance
- Example
  - The risk of shortfalls in external supplied (software or hardware) components can be transferred away by quality assurance procedures and certification, and contractual agreements.

Contingency Planning

- It assumes that the previous attempts (strategies for hazard prevention, likelihood reduction, impact reduction and risk transfer) are not successful
- Contingency plans (i.e. "Plan B") are needed to reduce the exposure of those risks that cannot be avoided
  - If new, unproven technology is part of the project specification, risks cannot be avoided
- Example
  - The impact of any unplanned absence of programming staff can be minimized by using agency programmers.

☞ *Risks that require contingency plans need careful monitoring!*

Risk Reduction Strategies – Example

Risk Item	Strategy	Category
Poor database performance	Invest on a higher-performance DBMS	Hazard prevention
Staff lack of skills	Outsourcing, staff training, buying components	Likelihood reduction
Defective components	Replace potential defective components with bought-in components of high reliability	Risk avoidance
Underestimate development time	Outsourcing some components to contractors or agency	Risk transfer
Organizational financial problems	Prepare a briefing report for senior management showing how the project is making a significant contribution to the goals of the business	Contingency planning

Adapted from Fig. 5.13 (Sommerville, 2007)

SEI Principles of Risk Management

- Maintain a global perspective
- Take a forward-looking view
- Encourage open communication
- Integrated management
- Emphasize a continuous process
- Develop a shared product vision
- Encourage teamwork

(Source: http://www.sei.cmu.edu/risk/principles.html)

Balance of Power

Business decisions should be made by business people

(and not technical people)

Technology decisions should be made by technical people

(and not business people)

☞ *A project is at immediate risk if this is reversed!*

Summary

- Risk: an uncertain event that may effect a project
  - has cause and effect
- Risk exposure = Risk probability * Risk impact
- Risk drivers: Knowledge, Skill, Technology (KoST)
- Risk management: identify, analyze, respond to risks
- Risks are not fixed, they will change over time!
- Risk management is not for free, costs are associated
  - Cost of risk not always “obvious” to identify

Closing Remark

“Projects with no real risks are losers.
They are almost always devoid of benefit;
that is why they were not done years ago.”

– Tom DeMarco, Tim Lister

What you should know!

- What are the key characteristics of risks?
- What is risk management?
  Why do we need to manage risks in software projects?
- What are the main risk drivers?
  How do they assist in identifying risks in a project?
- What is the relationship between risk probability, risk impact, and risk exposure?
- Why is it recommended to regularly watch the “Top 10” risks?

Can you answer these questions?

- Should a risk mitigation strategy be implemented at all cost?
- What may happen if risk probability and risk impact are measured at a too fine-grained scale?
- What kind of risks does N-Version Programming attempts to mitigate?
- What kind of risks does Pair Programming attempt to mitigate? What other strategies could be put in place to achieve the same outcome?

Recommended Reading Lecture 8

- Bob Hughes and Mike Cotterell, *Software Project Management* (4th Edition), Wiley, 2006, Chapter 9.
- Kent Beck, Martin Fowler, *Planning Extreme Programming*, Addison-Wesley, 2001, Chapters 19.
- Ian Sommerville, *Software Engineering* (8th Edition), Addison-Wesley, 2007, Chapter 28.