
# Risk Management Part I

Lecture 8 by Professor Vladimir Geroimenko

Module “Software Project Management“

13 November 2016 - Teaching Week 8

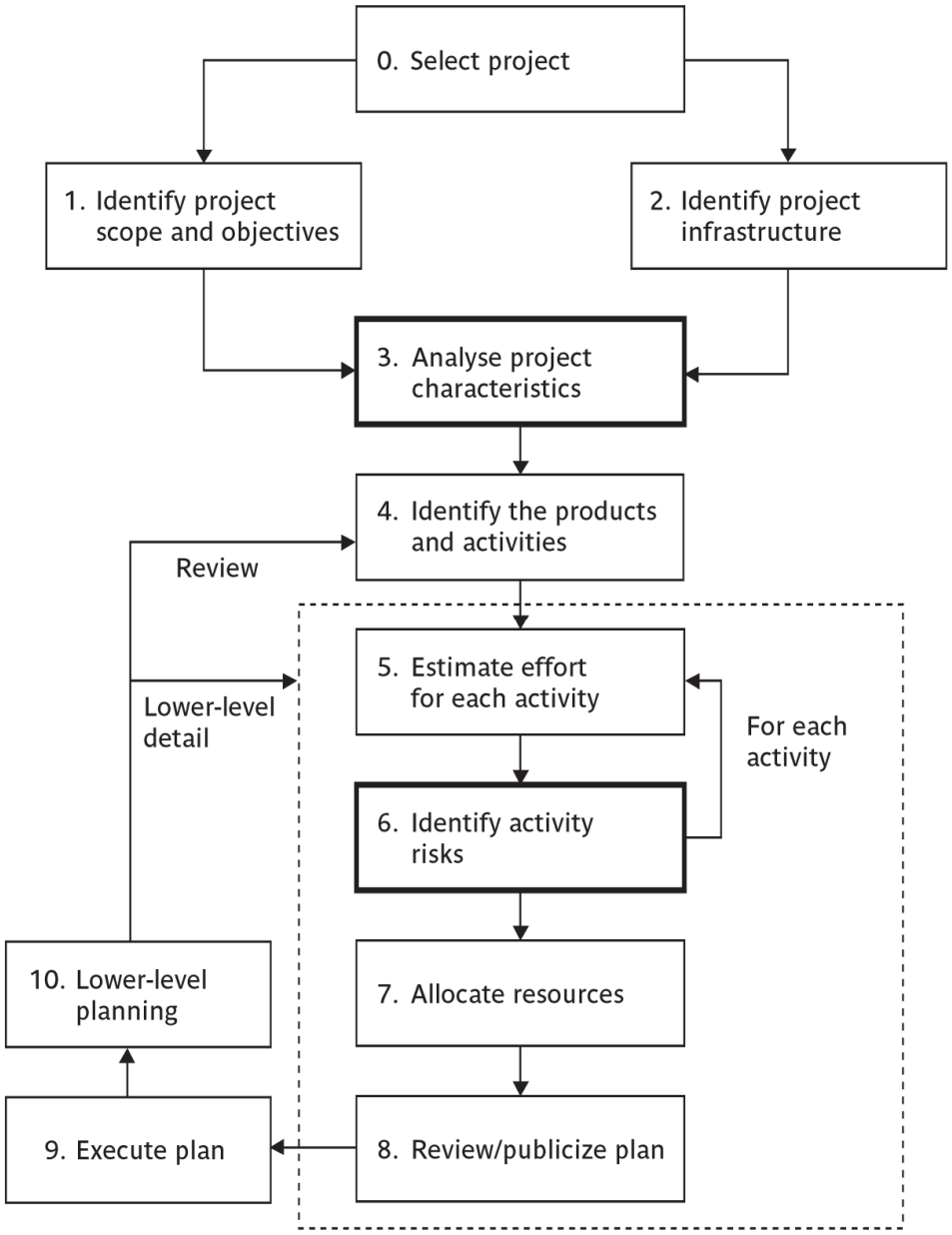
Textbook reference: Chapter 7

Copyright notice: lecture is based on the module textbook “Software Project Management, 5th edition” by Bob Hughes and Mike Cotterell and may use some royalty-free images from the Internet (unless it is stated otherwise on a particular slide)

# Lecture Outline

- Project risks
  - What causes project risks
- Risk Management Framework
  - Risk identification
  - Risk assessment
  - Risk reduction strategies
  - Risk monitoring
- Estimation techniques
  - Risk Exposure
  - Qualitative measures

# Managing risks in the Step Wise framework



# Project Risks

- *Factors that cause a project to be **delayed** or **over-budget***
- Project plans have to be based on ***assumptions***. ***Risk*** is the possibility that an assumption is wrong.
- Key elements of risk:
  - It relates to the ***future***
  - It evolves ***causes*** and ***effects***: inexperienced staff → low productivity.

# What Causes Project Risks?

- Planning assumptions
  - We all make assumptions due to uncertainties in the early stage of the project. What happen if the assumptions turn out to be invalid?
- Estimation errors
- Eventualities

# Estimation Errors

- Estimation can be improved by analyzing historic data for similar tasks and similar projects
  - Keep historic data of your estimation and the actual performance
  - Compare your estimation and the actual value
  - Classify the tasks that are easy or difficult to give accurate estimation

# Eventualities

- Unexpected and unimaginable events
- Examples of common unexpected events
  - Hardware cannot be delivered on time
  - Requirements specification needs to be rewritten
  - Staffing problem

# Some definitions of risk

*'an uncertain event or condition that, if it occurs, has a negative effect on a project's objectives'* PM-BOK

*'the chance of exposure to the adverse consequences of future events'* PRINCE2

*Please note:*

- Risks relate to **possible future problems**, not current ones
- They involve a possible **cause** and its **effect** e.g. developer leaves -> task delayed

# Exercise: Match causes and possible effects

**Causes**

(a) Staff inexperience

(b) Lack of top management commitment

(c) New technology

(d) Users uncertain of their requirements

**Possible affects**

(i) Testing takes longer than planned

(ii) Planned effort and time for activities exceeded

(iii) Project scope increases

(iv) Time delay in getting changes to plan agreed

# Cause a and effects i and ii

**Causes**

(a) Staff inexperience

(b) Lack of top management commitment

(c) New technology

(d) Users uncertain of their requirements

**Possible affects**

(i) Testing takes longer than planned

(ii) Planned effort and time for activities exceeded

(iii) Project scope increases

(iv) Time delay in getting changes to plan agreed

# Cause b and effect iv

**Causes**

(a) Staff inexperience

(b) Lack of top management commitment

(c) New technology

(d) Users uncertain of their requirements

**Possible affects**

(i) Testing takes longer than planned

(ii) Planned effort and time for activities exceeded

(iii) Project scope increases

(iv) Time delay in getting changes to plan agreed

# Cause c and effects i and ii

**Causes**

(a) Staff inexperience

(b) Lack of top management commitment

(c) New technology

(d) Users uncertain of their requirements

**Possible affects**

(i) Testing takes longer than planned

(ii) Planned effort and time for activities exceeded

(iii) Project scope increases

(iv) Time delay in getting changes to plan agreed

# Cause d and effect iii

**Causes**

(a) Staff inexperience

(b) Lack of top management commitment

(c) New technology

(d) Users uncertain of their requirements

**Possible affects**

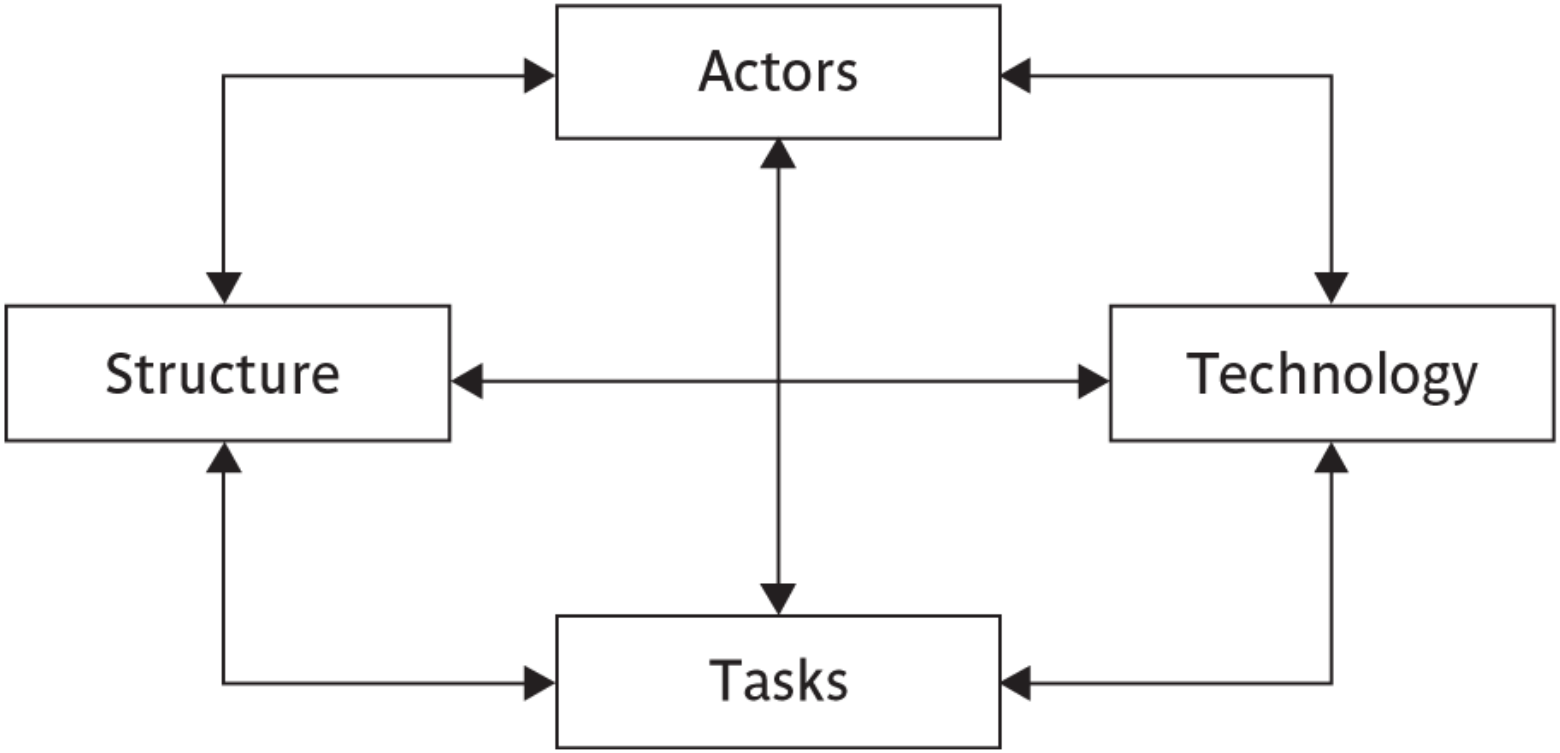
(i) Testing takes longer than planned

(ii) Planned effort and time for activities exceeded

(iii) Project scope increases

(iv) Time delay in getting changes to plan agreed

# Categories of risk

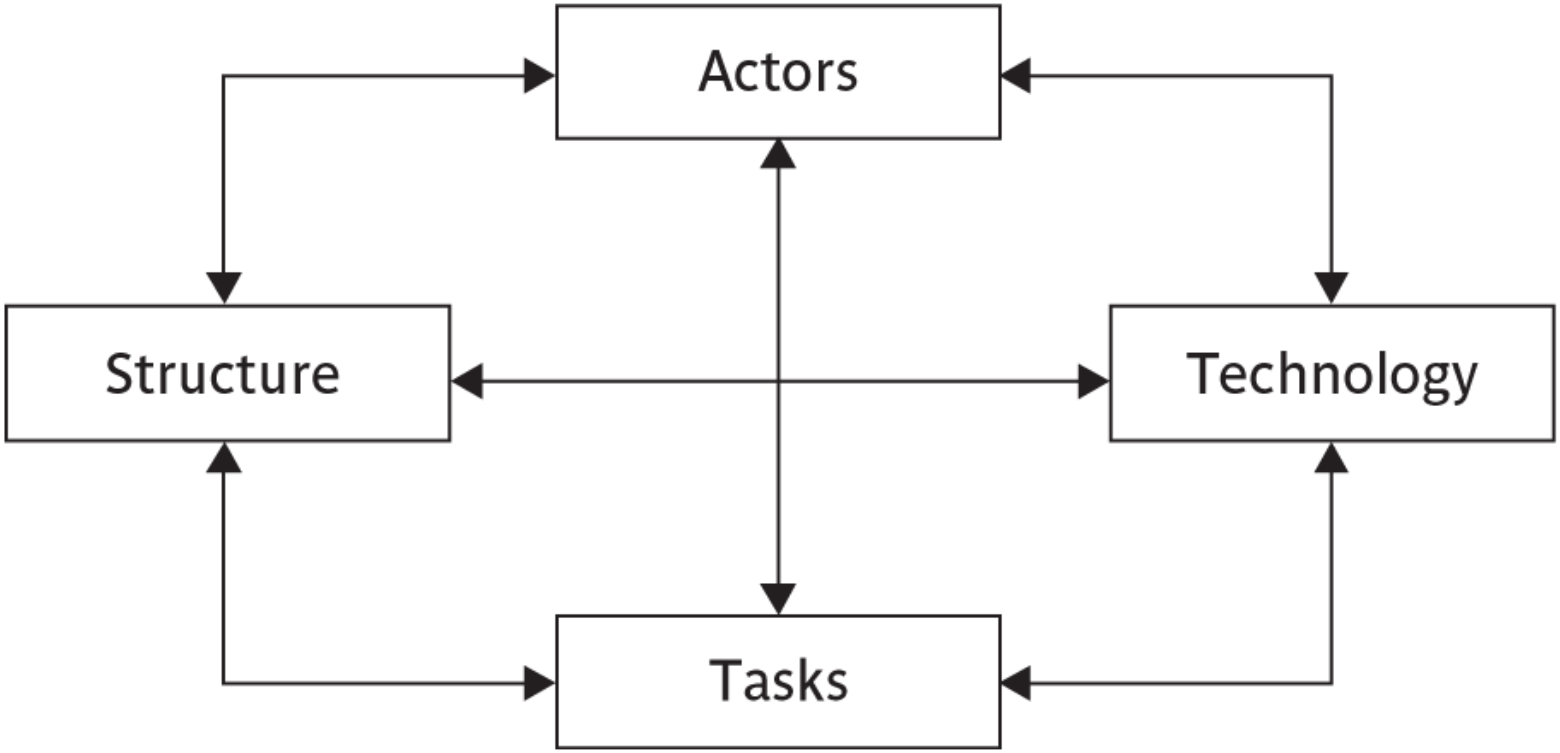


# Categories of risk explained

- **Actors** relate to all those involved in the project including both developers, users and managers.
- **Technology** – both that used to implement the project and that embedded in the project deliverables.
- **Structure** – this includes management procedures, risk here is that a group who need to carry out a particular project task are not informed of this need.
- **Tasks** – the work to be carried out. A typical risk is that the amount of effort needed to carry out the task is underestimated.

*(Based on Lyytinen's sociotechnical model of risk)*

# Categories of risk – all boxed are interlinked!



# A framework for dealing with risk

The planning for risk includes these steps:

- **Risk identification** – what risks might there be?
- **Risk analysis and prioritization** – which are the most serious risks?
- **Risk planning** – what are we going to do about them?
- **Risk monitoring** – what is the current state of the risk?

# Risk Identification Approaches

- **Checklists** – usually based on the experience of past projects
- **Brainstorming** – getting knowledgeable stakeholders together to pool concerns

# Type of risks

- *Generic risk* (common to all projects)
  - Standard checklist can be modified based on the risk analysis of previous projects
- *Specific risk* (only applies to individual projects)
  - Use brainstorming
  - More difficult to find
  - Need to involve project team members

# Common Risk Factors

- **Application** factors
- **Staff** factors
- **Project** factors
- **Hardware & software** factors
- **Changeover** factors
- **Supplier** factors
- **Environment** factors
- **Health and safety** factors

# Application Factors

- Nature of the application
  - A data processing application or a life-critical system (e.g. X-ray emission system)

- Expected size of the application
  - The larger is the size, the higher is the chance of errors, communication problems and management problems

# Staff Factors

- Experience and skills
- Appropriateness of experience
- Staff satisfaction
- Staff turn-over rates

# Project Factors

- Project objectives:
  - Ill defined
  - Unclear to every team member and user
- Project methods:
  - Ill specified methods
  - Unstructured methods

# Hardware and Software Factors

- New hardware
  - Stability of the new hardware system
- Cross platform development
  - Development platform is not the operation platform
  - Does the language used support cross platform development?

# Changeover Factors

- ‘All-in-one’ changeover
  - The new system is put into operation
- Incremental or gradual changeover
  - Adding new components to the system by phases
- Parallel changeover
  - Both the existing system and the new system are used in parallel

# Supplier Factors

- Late delivery of hardware
- Instability of hardware
- Late completion of building sites

# Environment Factors

- Changes in environment such as hardware platforms
- Changes in government policies
- Changes in business rules
- Restructuring of organizations

# Health and Safety Factors

- Health and safety of staff and environment
  - Staff sickness, death, pregnancy etc.
  - Any tragic accident to staff

# Boehm's top 10 development risks

| **Risk** | **Risk reduction techniques** |
|---|---|
| 1. Personnel shortfalls | Staffing with top talent; job matching; teambuilding; training and career development; early scheduling of key personnel |
| 2. Unrealistic time and cost estimates | Multiple estimation techniques; design to cost; incremental development; recording and analysis of past projects; standardization of methods |
| 3. Developing the wrong software functions | Improved software evaluation; formal specification methods; user surveys; prototyping; early user manuals |
| 4. Developing the wrong user interface | Prototyping; task analysis; user involvement |

# Boehm's top 10 development risks

| | |
|---|---|
| 5. Gold plating | Requirements scrubbing, prototyping, design to cost |
| 6. Late changes to requirements | Change control, incremental development |
| 7. Shortfalls in externally supplied components | Benchmarking, inspections, formal specifications, contractual agreements, quality controls |
| 8. Shortfalls in externally performed tasks | Quality assurance procedures, competitive design etc |
| 9. Real time performance problems | Simulation, prototyping, tuning |
| 10. Development technically too difficult | Technical analysis, cost-benefit analysis, prototyping , training |

# Risk Analysis: Risk Exposure Measure

- Risk estimation is to assess the **impact** and **likelihood** of each hazard
- Risk exposure (risk value)
  - It is the importance of the risk

*Risk exposure = risk impact × risk likelihood*

- Risk impact - The effect of the problem caused by the hazard
  - Delays to scheduled activities
  - Using additional expensive resources
  - Any compromise to the quality or functionality
- Risk likelihood - The probability that a hazard is going to occur

# Risk Exposure

Risk exposure (RE) = (potential damage) x (probability of occurrence)

Crudely analogous to the amount needed for an insurance premium

*For example:*

**Potential damage**: a money value e.g. a flood would cause £0.5 millions of damage

**Probability** 0.00 (absolutely no chance) to 1.00 (absolutely certain) e.g. 0.01 (one in hundred chance)

RE = £0.5m x 0.01 = £5,000

*Please note: In practice, with project risks, these quantitative approaches are usually impractical and more qualitative approaches are used instead.*

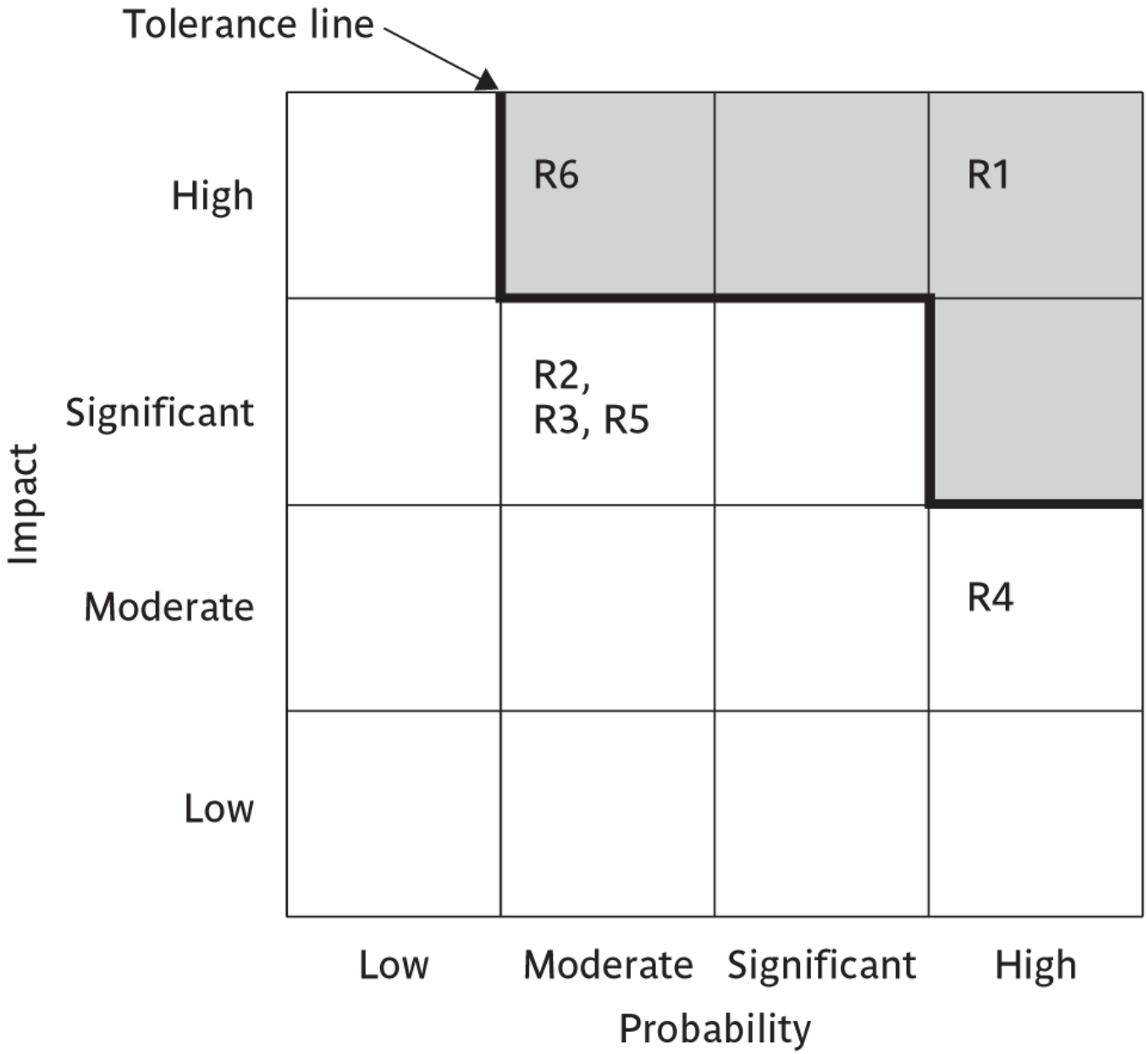
# Risk probability: qualitative descriptors

| *Probability level* | *Range* |
|---|---|
| High | Greater than 50% chance of happening |
| Significant | 30-50% chance of happening |
| Moderate | 10-29% chance of happening |
| Low | Less than 10% chance of happening |

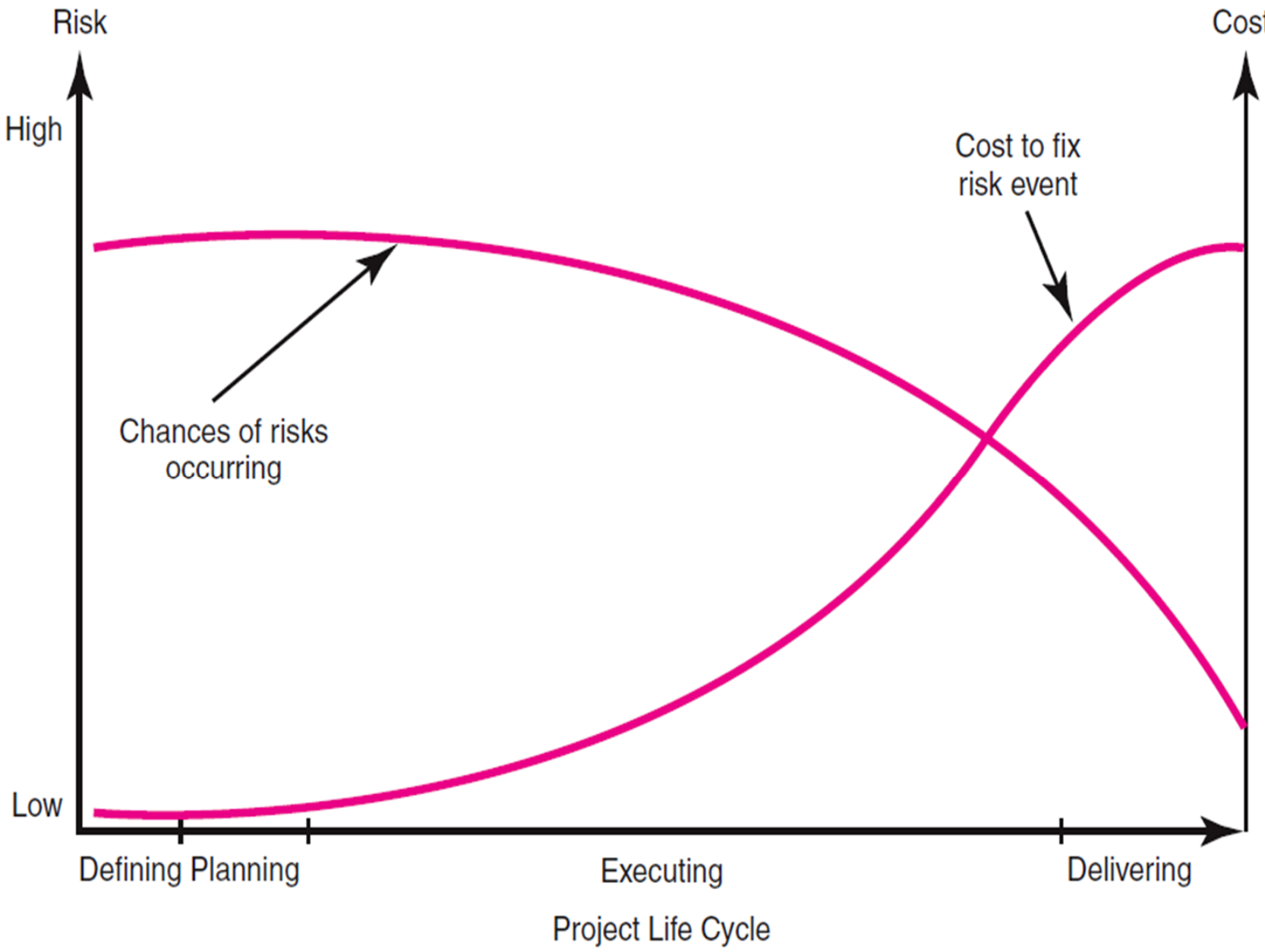
# Qualitative descriptors of impact on cost

| *Impact level* | *Range* |
|---|---|
| High | Greater than 30% above budgeted expenditure |
| Significant | 20 to 29% above budgeted expenditure |
| Moderate | 10 to 19% above budgeted expenditure |
| Low | Within 10% of budgeted expenditure. |

# Probability impact matrix



**Risk in Project Life Cycle**



# Risk planning

Five different ways of handling risk:

1. Risk acceptance
2. Risk avoidance
3. Risk reduction
4. Risk transfer
5. Risk mitigation/contingency measures

See next slide

# 1. Risk acceptance

- The cost of avoiding the risk may be greater than the actual cost of the damage that might be inflicted.

# 2. Risk avoidance

- Avoid the environment in which the risk occurs.
- For example, buying an off-the-shelf application would avoid a lot of the risks associated with software development.

# 3. Risk reduction

- The risk is accepted but actions are taken to reduce its likelihood.
- For example, prototypes ought to reduce the risk of incorrect requirements

# 4. Risk transfer

- The risk is transferred to another person or organization.
- For example, the risk of incorrect development estimates can be transferred by negotiating a fixed price contract with an outside software supplier.
- The impact of the risk can be transferred away from the project by contracting out or taking out insurance

# 5. Risk mitigation

- Tries to reduce the impact if the risk does occur
- For example, taking backups to allow rapid recovery in the case of data corruption

# Contingency planning

- Contingency Plan (a.k.a. “Plan B”)
  - An alternative plan that will be used if a possible foreseen risk event actually occurs.
  - A plan of actions that will reduce or mitigate the negative impact (consequences) of a risk event.
- Risks of Not Having a Contingency Plan
  - Having no plan may slow managerial response.
  - Decisions made under pressure can be potentially dangerous and costly.

# Risk Reduction Leverage (RRL)

- RRL is used to determine whether it is worthwhile to carry out the risk reduction plan.
- The **higher is the RRL** value, the **more worthwhile** is to carry out the risk reduction plan.

$$\text{RRL} = \frac{\text{RE}_{\text{before}} - \text{RE}_{\text{after}}}{\text{risk reduction cost}}$$

**If RRL > 1, it is worth doing**

# Risk Reduction Leverage: An Example

Risk reduction leverage =

$$(RE_{before} - RE_{after}) / (\text{cost of risk reduction})$$

$RE_{before}$ is risk exposure before risk reduction, e.g. 1% chance of a fire causing £200k damage

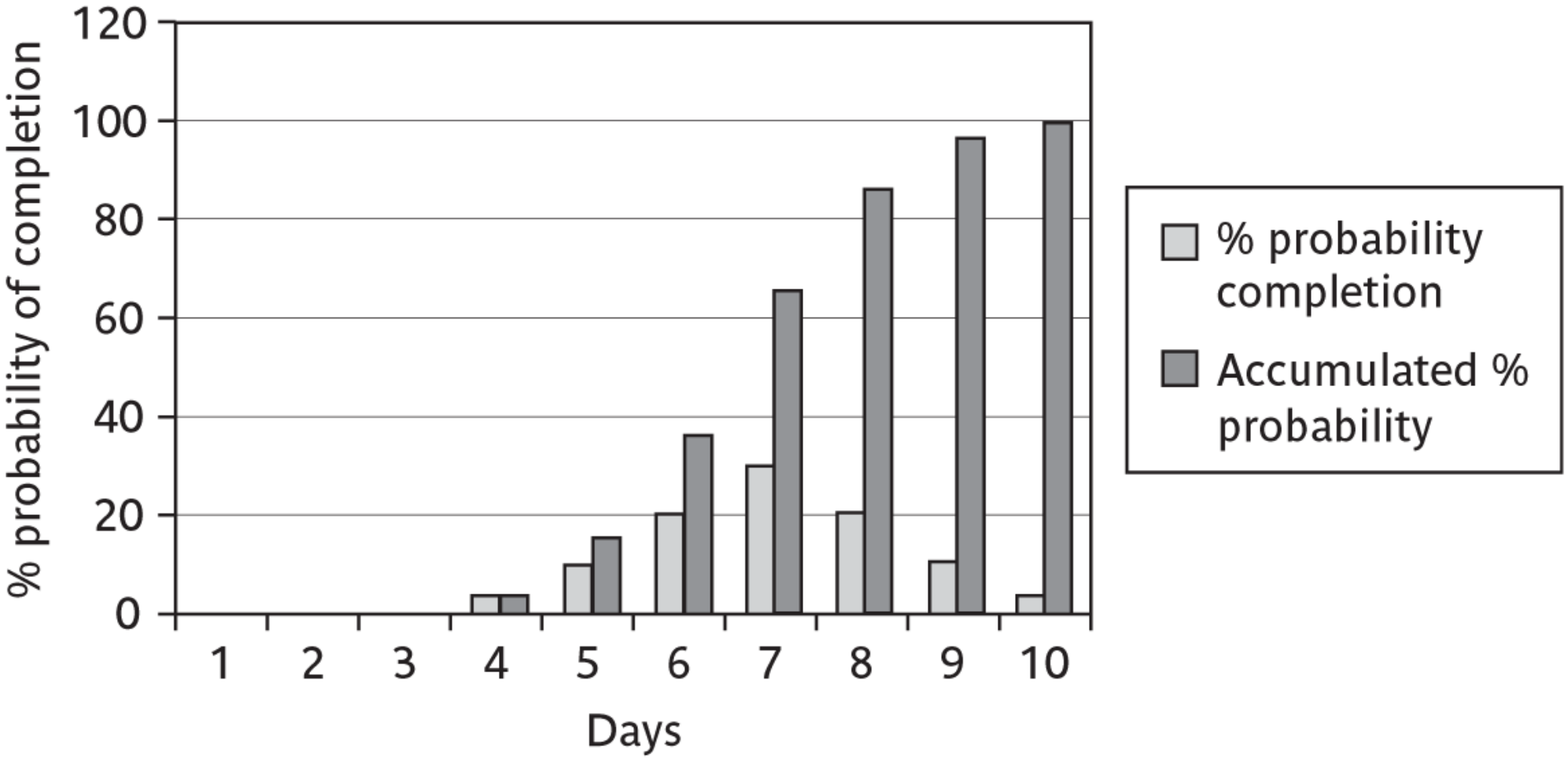
$RE_{after}$ is risk exposure after risk reduction, e.g. fire alarm costing £500 reduces probability of fire damage to 0.5%

RRL = (1% of £200k)-(0.5% of £200k)/£500 = 2

RRL > 1.00 therefore worth doing

**Please note:** You could think in terms of the analogy to insurance. An insurance company might reduce the fire insurance premium from £2k to £1k on condition that a fire alarm is installed. The insured would save £1k a year by investing £500, so it would be worth doing.

# Probability chart



| RISK RECORD | | | | | |
|---|---|---|---|---|---|
| Risk id | | Risk title | | | |
| Owner | | Date raised | | Status | |
| Risk description | | | | | |
| Impact description | | | | | |
| Recommended risk mitigation | | | | | |

| Probability/impact values | | | | |
|---|---|---|---|---|
| | Probability | Impact | | |
| | | Cost | Duration | Quality |
| Pre-mitigation | | | | |
| Post-mitigation | | | | |

| Incident/action history | | | |
|---|---|---|---|
| Date | Incident/action | Actor | Outcome/comment |
| | | | |

# Thank you for your attention

Any questions, please?