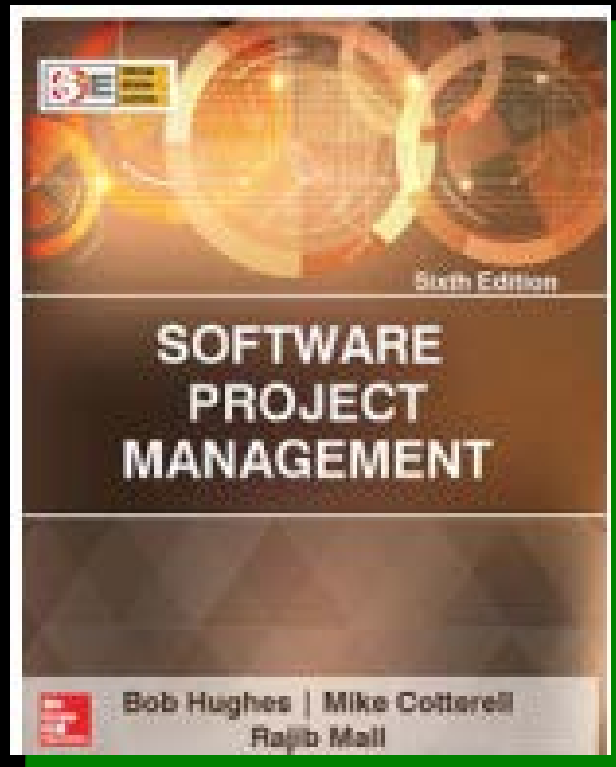


# Software Project Management



## Chapter One

## An Introduction

# Outline of talk

In this introduction the main questions to be addressed will be:

- ◆ What is software project management? Is it really different from 'ordinary' project management?
- ◆ How do you know when a project has been successful? For example, do the expectations of the customer/client match those of the developers?

# Why is project management important?

- Large amounts of money are spent on ICT e.g. UK government in 2003-4 spent £2.3 billions on contracts for ICT and only £1.4 billions on road building
- Project often fail – Standish Group claim only a third of ICT projects are successful. 82% were late and 43% exceeded their budget.
- Poor project management a major factor in these failures

# What is a project?

Some dictionary definitions:

*“A specific plan or design”*

*“A planned undertaking”*

*“A large undertaking e.g. a public works scheme”*

Longmans dictionary

Key points above are *planning* and *size of task*

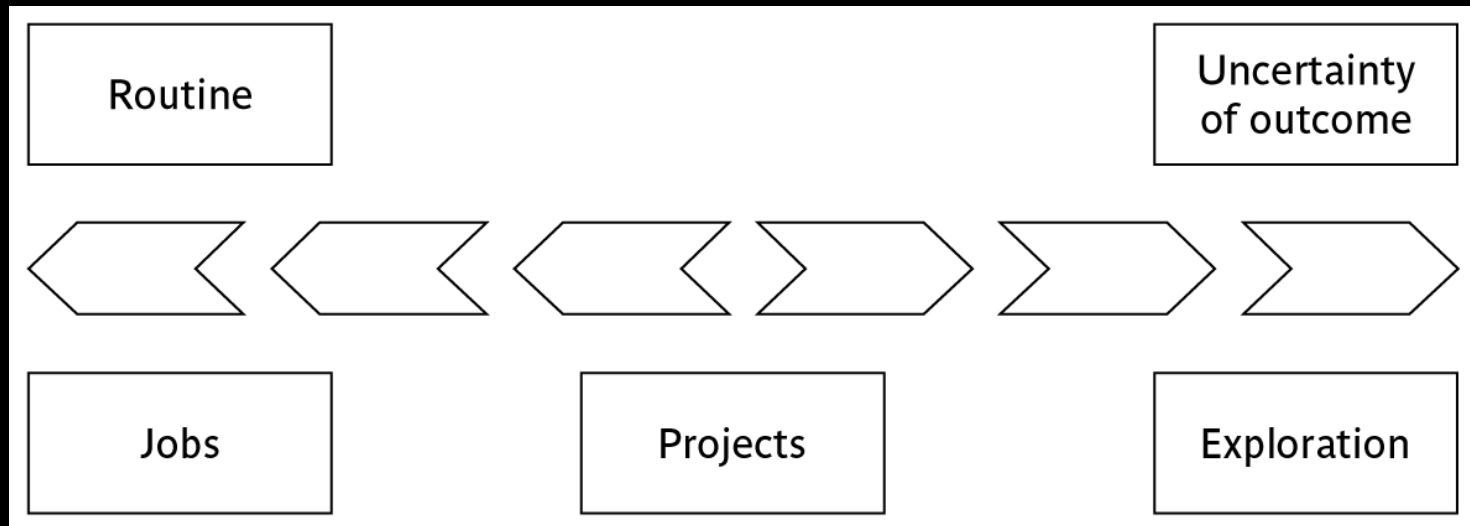
# What is a Project?

- An endeavor with specific objectives:
  - ◆ Usually consists of multiple tasks
  - ◆ With defined precedence relationships
  - ◆ With a specific time period for completion
- **Non-Software Examples:**
  - ◆ A wedding
  - ◆ An MBA degree
  - ◆ A house construction project
  - ◆ A political election campaign

# What is a Task?

- A small piece of work:
  - ◆ Meant to accomplish a straightforward goal
  - ◆ Effort of no longer than a few person-hours
  - ◆ Involves only a few people
  - ◆ May or may not be a part of some project
  - ◆ Usually repetition of a previously accomplished task
  - ◆ Process management may be relevant!
- Non-software Examples:
  - ◆ Attend a lecture class
  - ◆ Buy a chocolate from the market
  - ◆ Book a railway ticket

# Jobs versus projects



‘Jobs’ – repetition of very well-defined and well understood tasks with very little uncertainty

‘Exploration’ – e.g. finding a cure for cancer: the outcome is very uncertain

Projects – in the middle!

# Characteristics of projects

A task is more 'project-like' if it is:

- Non-routine
- Planned
- Aiming at a specific target
- Carried out for a customer
- Carried out by a temporary work group
- Involving several specialisms
- Made up of several different phases
- Constrained by time and resources
- Large and/or complex



# Are *software* projects really different from other projects?

Not really ...but... The factors

- Invisibility
- Complexity
- Conformity
- Flexibility

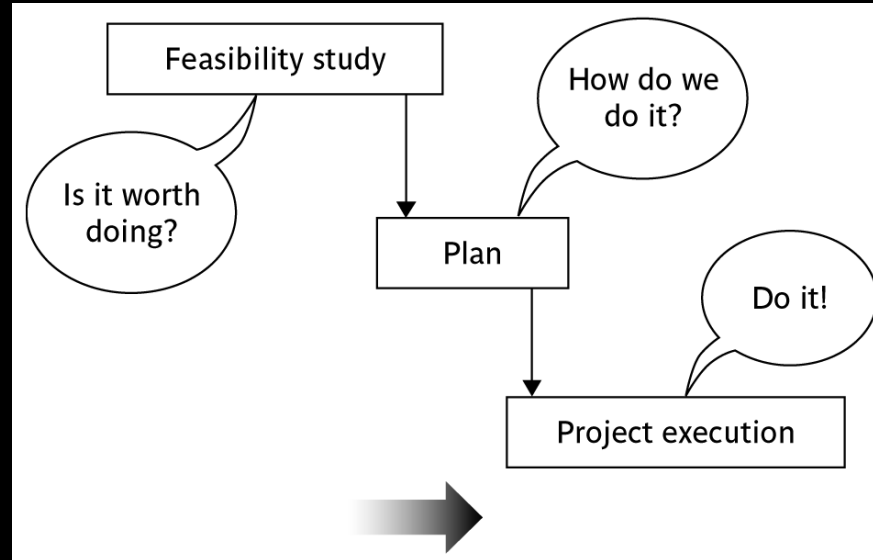
make software more problematic to build than other engineered artefacts.

# Contract management versus technical project management

Projects can be:

- **In-house:** clients and developers are employed by the same organization
- **Out-sourced:** clients and developers employed by different organizations
- 'Project manager' could be:
  - ◆ a 'contract manager' in the client organization
  - ◆ a technical project manager in the supplier/services organization

# Activities covered by project management



## Feasibility study

Is project technically feasible and worthwhile from a business point of view?

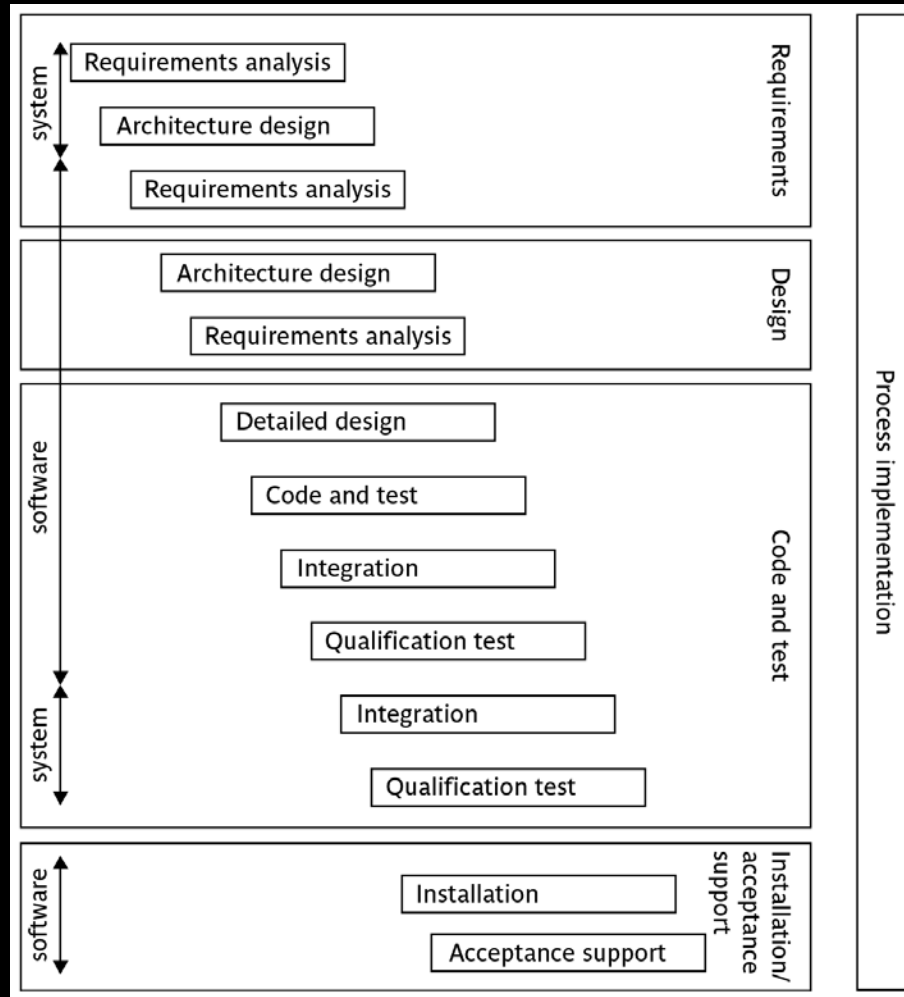
## Planning

Only done if project is feasible

## Execution

Implement plan, but plan may be changed as we go along

# The software development life-cycle (ISO 12207)



# ISO 12207 life-cycle

- Requirements analysis
  - ◆ Requirements elicitation: what does the client need?
  - ◆ Analysis: converting 'customer-facing' requirements into equivalents that developers can understand
  - ◆ Requirements will cover
    - Functions
    - Quality
    - Resource constraints i.e. costs

# ISO 12207 life-cycle

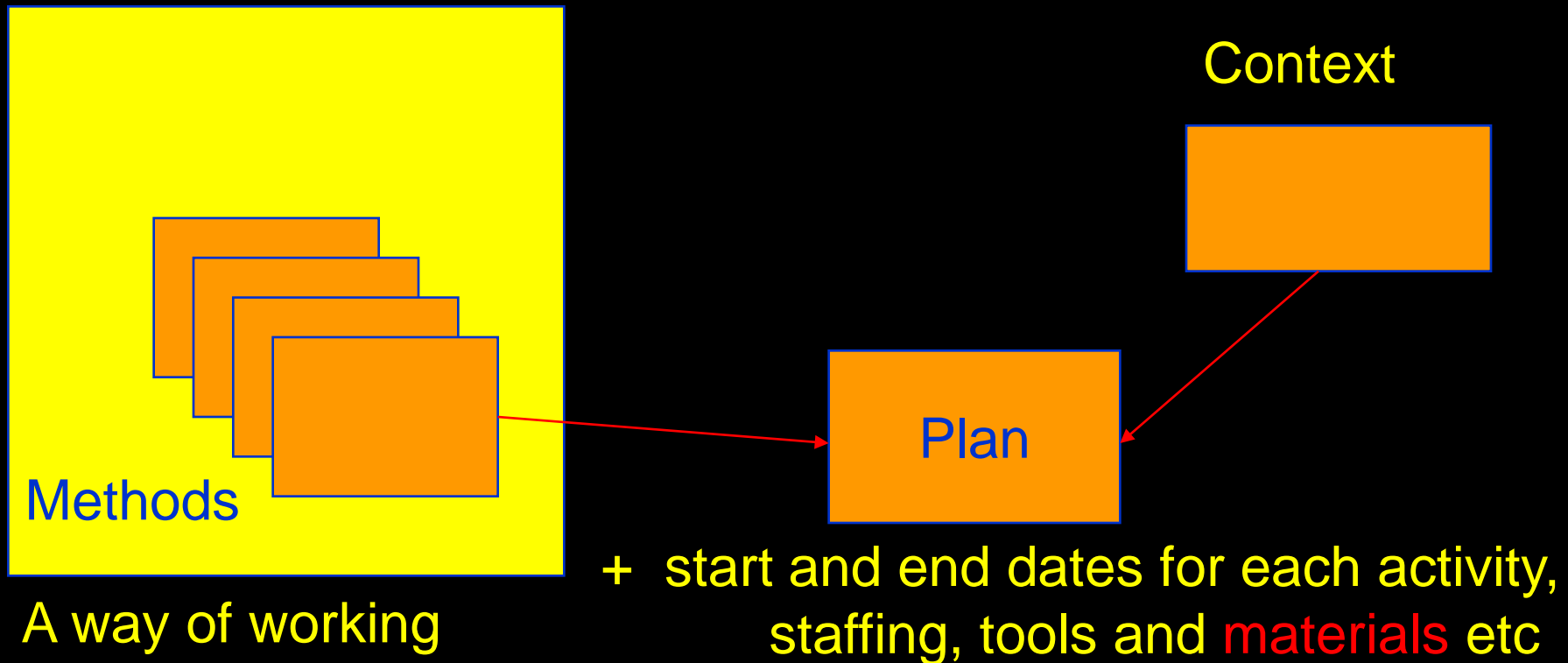
- Architecture design
  - ◆ Based on *system requirements*
  - ◆ Defines components of system: hardware, software, organizational
  - ◆ *Software requirements* will come out of this
- Code and test
  - ◆ Of individual components
- Integration
  - ◆ Putting the components together

# ISO12207 continued

- Qualification testing
  - ◆ Testing the *system* (not just the *software*)
- Installation
  - ◆ The process of making the system operational
  - ◆ Includes setting up standing data, setting system parameters, installing on operational hardware platforms, user training etc
- Acceptance support
  - ◆ Including maintenance and enhancement

# Plans, methods and methodologies

Methodology = a set of methods



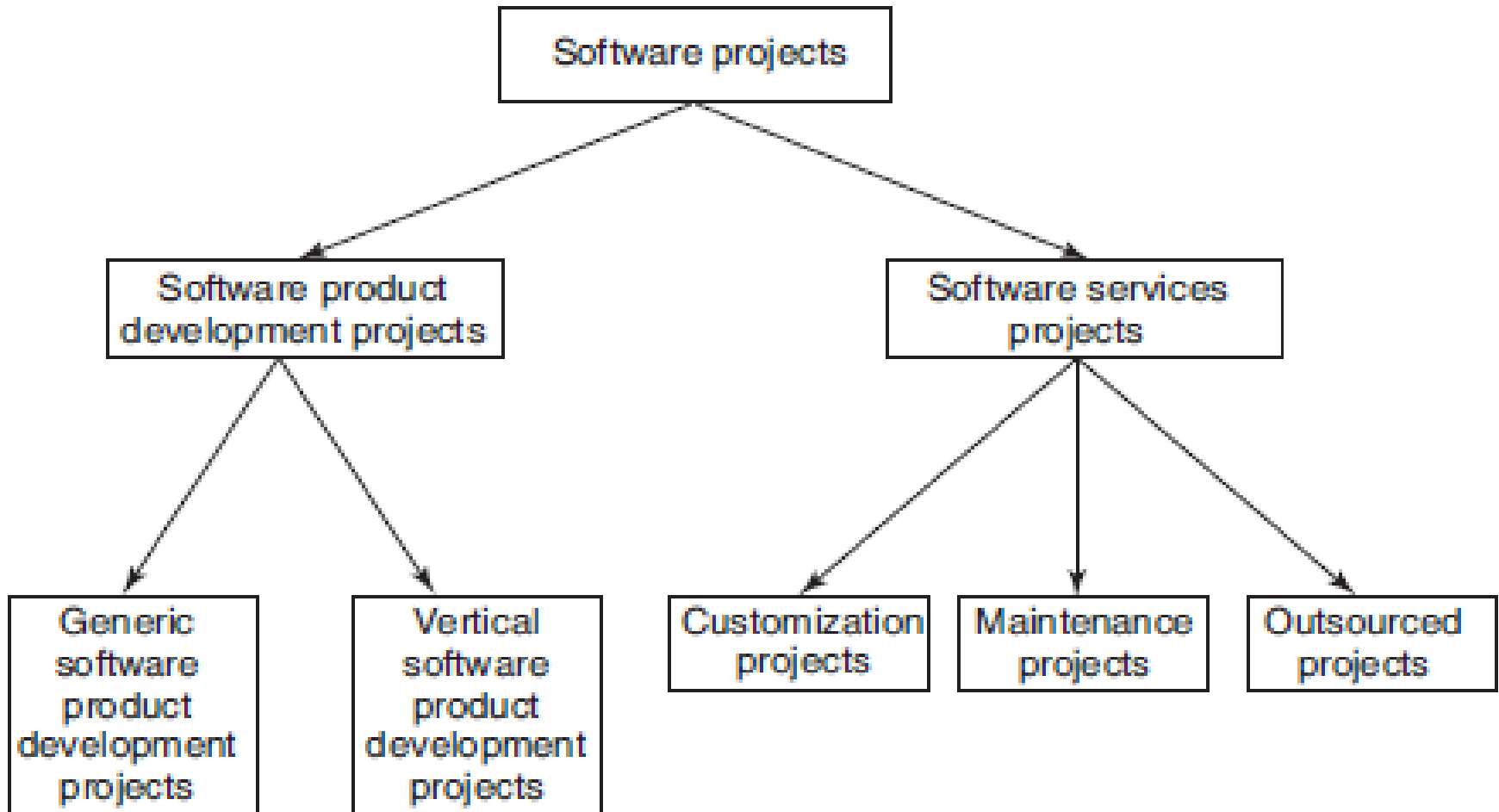


# Some ways of categorizing projects

Distinguishing different types of project is important as different types of task need different project approaches e.g.

- Voluntary systems (such as computer games) versus compulsory systems e.g. the order processing system in an organization
- Information systems versus embedded systems
- Objective-based versus product-based
- Product-development versus outsourced

# A Categorization of Software Projects



# Two Types of Software Projects

- Software product development projects
- Software services projects

# Software Services

- Software service is an umbrella term, includes:
  - ◆ Software customization
  - ◆ Software maintenance
  - ◆ Software testing
  - ◆ Also contract programmers who carry out coding or any other assigned activities.

# Stakeholders

These are people who have a stake or interest in the project

In general, they could be *users/clients* or *developers/implementers*

They could be:

- Within the project team
- Outside the project team, but within the same organization
- Outside both the project team and the organization

Different stakeholders may have different objectives – need to define common project objectives

# Setting objectives

- Answering the question '*What do we have to do to have a success?*'
- Need for a *project authority*
  - ◆ Sets the project scope
  - ◆ Allocates/approves costs
- Could be one person - or a group
  - ◆ Project Board
  - ◆ Project Management Board
  - ◆ Steering committee

# Objectives

*Informally*, the objective of a project can be defined by completing the statement:

*The project will be regarded as a success if.....*

.....

Rather like *post-conditions* for the project

Focus on *what* will be put in place, rather than *how* activities will be carried out

# Objectives should be SMART

S – specific, that is, concrete and well-defined

M – measurable, that is, satisfaction of the objective can be objectively judged

A – achievable, that is, it is within the power of the individual or group concerned to meet the target

R – relevant, the objective must be relevant to the true purpose of the project

T – time constrained: there is a defined point in time by which the objective should be achieved



# Goals/sub-objectives

These are steps along the way to achieving the objective

Informally, these can be defined by completing the sentence

To reach objective X, the following must be in place

A.....

B.....

C..... etc

# Goals/sub-objectives continued

Often a goal can be allocated to an individual

Individual might have the capability of achieving goal on their own, but not the overall objective  
e.g.

*Overall objective* – user satisfaction with software product

*Analyst goal* – accurate requirements

*Developer goal* – reliable software

# Measures of effectiveness

How do we know that the goal or objective has been achieved?

By a practical test, that can be objectively assessed.

e.g. for user satisfaction with software product:

- Repeat business – they buy further products from us
- Number of complaints – if low etc etc

# The business case

Benefits of delivered project must outweigh costs

Costs include:

Development

Operation

Benefits

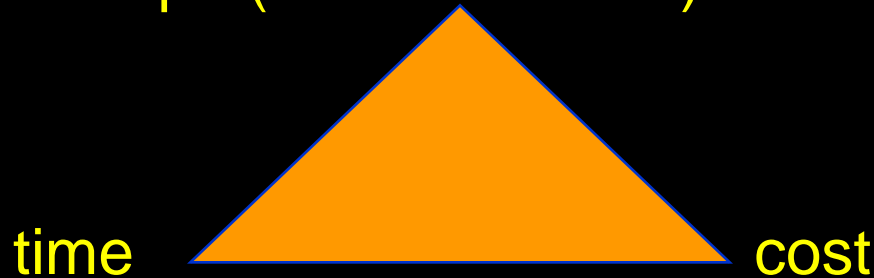
Quantifiable

Non-quantifiable



# Project success/failure

- Degree to which objectives are met  
scope (of deliverables)



In general if, for example, project is running out of time, this can be recovered for by reducing scope or increasing costs. Similarly costs and scope can be protected by adjusting other corners of the 'project triangle'.

# Other success criteria

These can relate to longer term, less directly tangible assets

- Improved skill and knowledge
- Creation of assets that can be used on future projects e.g. software libraries
- Improved customer relationships that lead to repeat business

# What is management?

This involves the following activities:

- Planning – deciding what is to be done
- Organizing – making arrangements
- Staffing – selecting the right people for the job
- Directing – giving instructions

continued...

# What is management? (continued)

- Monitoring – checking on progress
- Controlling – taking action to remedy hold-ups
- Innovating – coming up with solutions when problems emerge
- Representing – liaising with clients, users, developers and other stakeholders



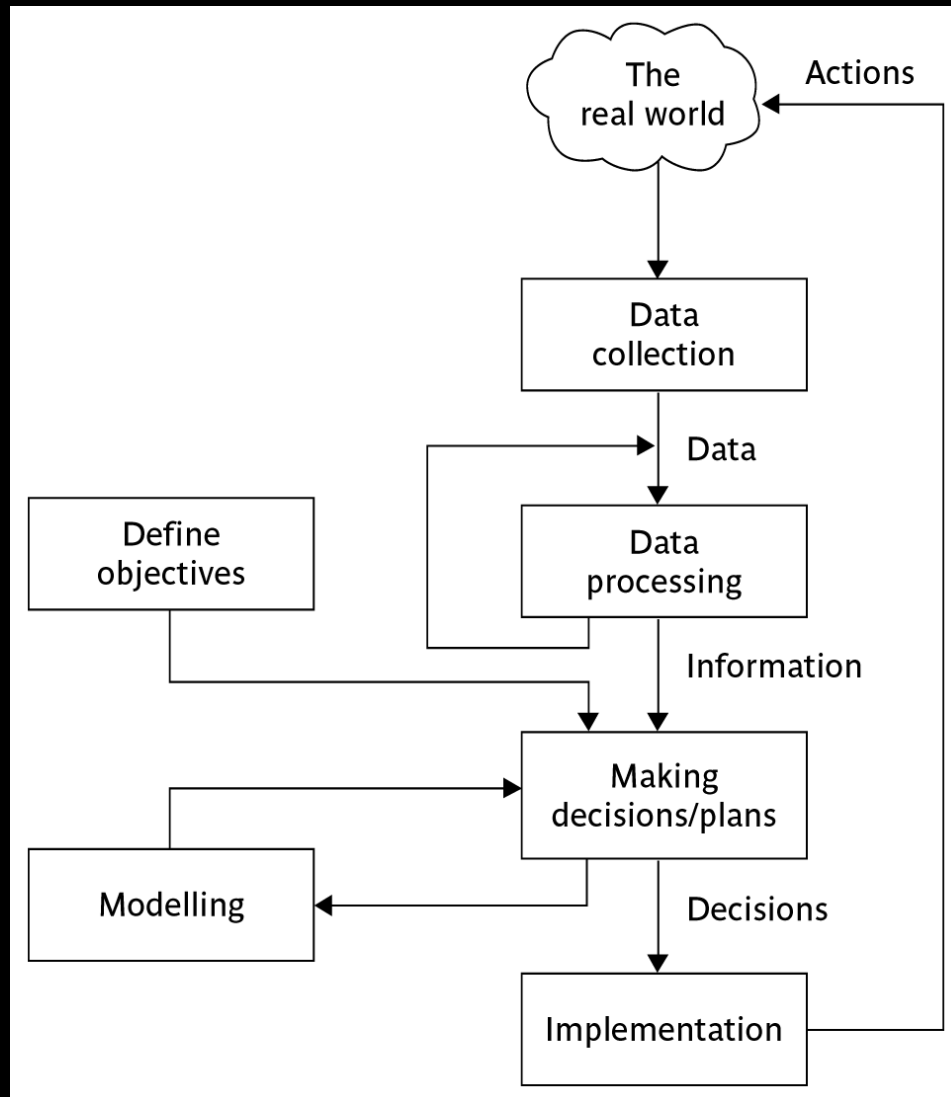
# Project Planning

- **Carried out before development starts.**
- **Important activities:**
  - ◆ **Estimation**
  - ◆ **Scheduling**
  - ◆ **Staffing**
  - ◆ **Risk management**
  - ◆ **Miscellaneous plans**

# Traditional versus Modern Project Management

- Projects are increasingly being based on either tailoring some existing product or reusing certain pre-built libraries.
- Facilitating and accommodating client feedbacks
- Facilitating customer participation in project development work
- Incremental delivery of the product with evolving functionalities.

# Management control



# Management control

Data – the raw details

*e.g. '6,000 documents processed at location X'*

Information – the data is processed to produce something that is meaningful and useful

*e.g. 'productivity is 100 documents a day'*

Comparison with objectives/goals

*e.g. we will not meet target of processing all documents by 31<sup>st</sup> March*

continued.....

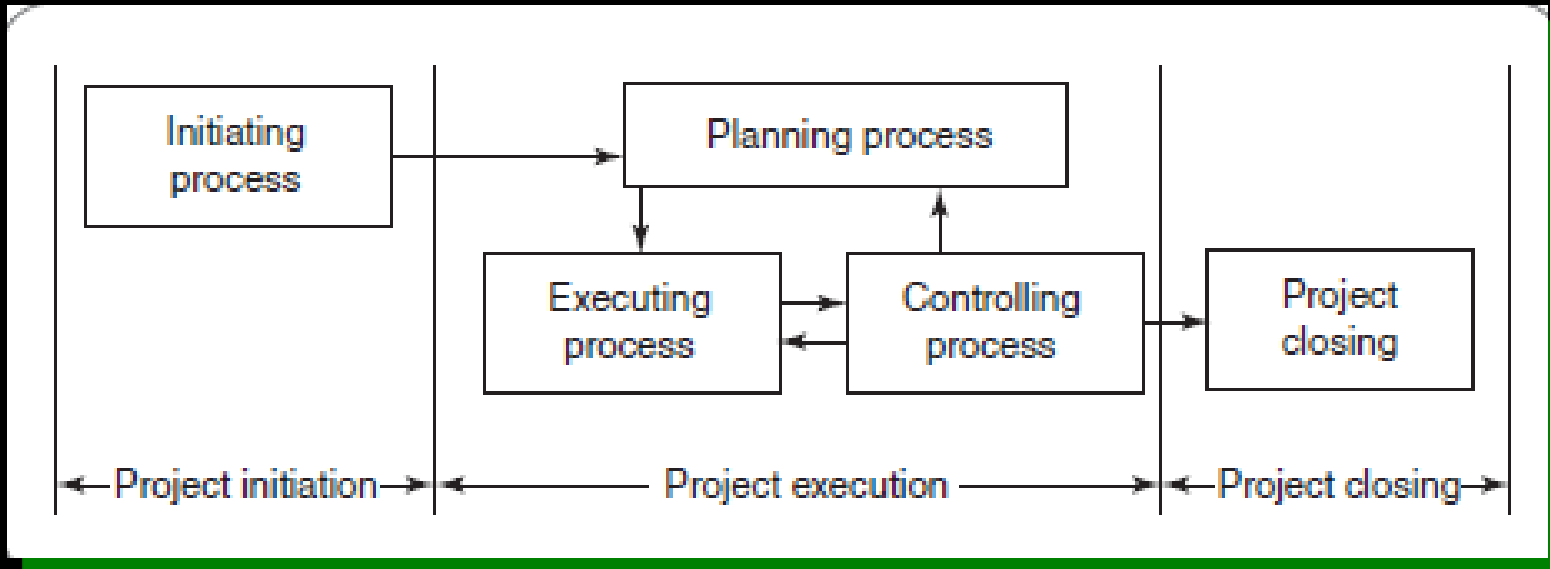
# Management control - continued

Modelling – working out the probable outcomes of various decisions

e.g. if we employ two more staff at location X how quickly can we get the documents processed?

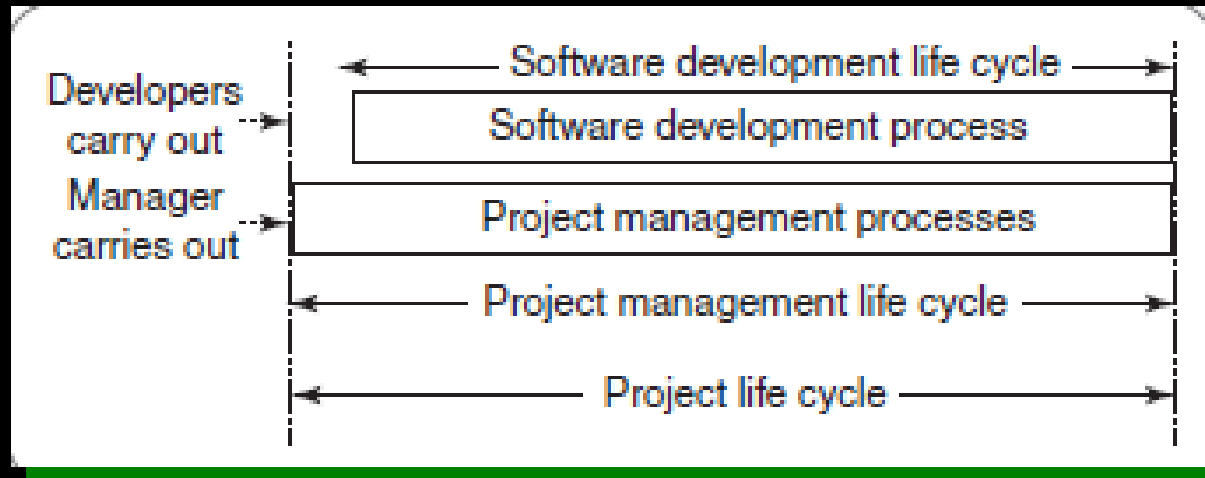
Implementation – carrying out the remedial actions that have been decided upon

# Project Management Processes



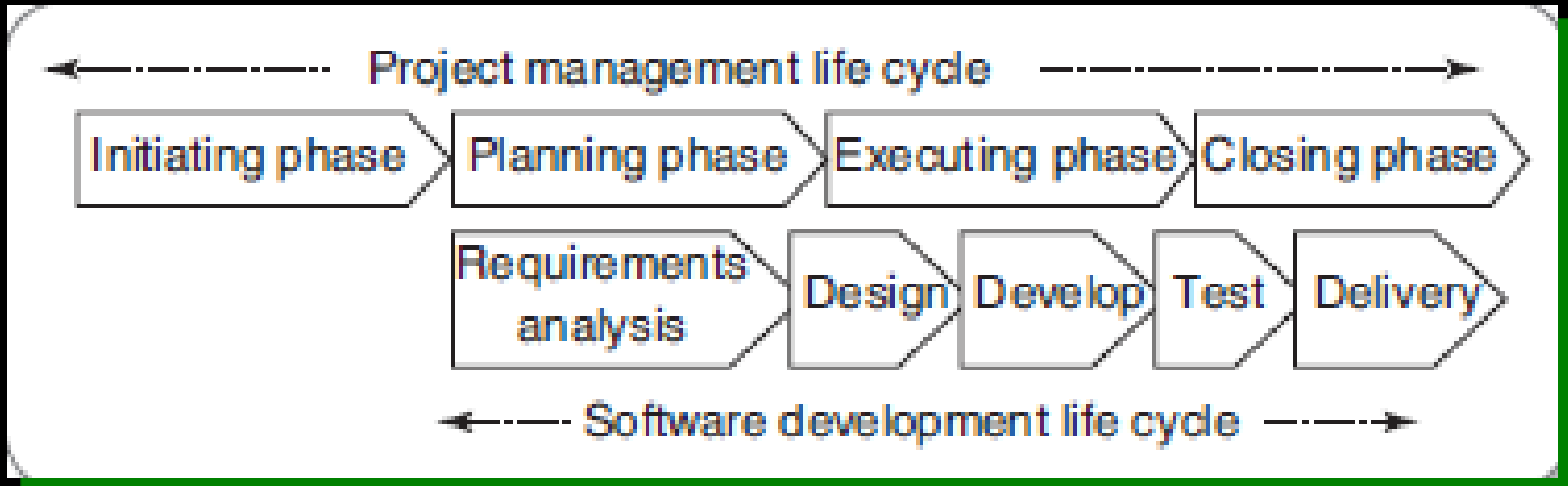
In the project initiation stage, an initial plan is made. As the project starts, the project is executed and controlled to proceed as planned. Finally, the project is closed.

# Project Management Life Cycle Versus Product Development Life Cycle



During the software development life cycle, the software developers carry out several types of development processes. On the other hand, during the software project management life cycle, the software project manager carries out several project management processes

# Phases of Project Management Life Cycle





# Project Initiation

- During the project initiation phase it is crucial for the champions of the project to develop a thorough understanding of the important characteristics of the project.
- In his W5HH principle, Barry Boehm summarized the questions that need to be asked and answered in order to have an understanding of these project characteristics.

# W5HH Principle

- A series of questions that lead to a definition of key project characteristics:
  - ◆ Why is the software being built?
  - ◆ What will be done?
  - ◆ When will it be done?
  - ◆ Who is responsible for a function?
  - ◆ Where are they organizationally located?
  - ◆ How will the job be done technically and managerially?
  - ◆ How much of each resource is needed?

# Project Planning

- Various plans are made:
  - ◆ *Project plan: Assign project resources and time frames to the tasks.*
  - ◆ *Resource plan: List the resources, manpower and equipment that required to execute the project.*
  - ◆ *Financial plan: plan for manpower, equipment and other costs.*
  - ◆ *Quality plan: Plan of quality targets and control.*
  - ◆ *Risk plan: Identification of the potential risks, their prioritization and a plan for the actions that would be taken to contain the different risks.*

# Project Execution

- Tasks are executed as per the project plan
- Monitoring and control processes are executed to ensure that the tasks are executed as per plan
- Corrective actions are initiated whenever any deviations from the plan are noticed.

# Project Closure

- Involves completing the release of all the required deliverables to the customer along with the
- necessary documentation.
- Subsequently, all the project resources are released and supply agreements with the vendors are terminated and all the pending payments are completed.
- Finally, a post-implementation review is undertaken to analyze the project performance and to list the lessons learnt for use in future projects.

# Key points in lecture

- Projects are non-routine - thus uncertain
- The particular problems of projects e.g. lack of visibility
- Clear objectives which can be objectively assessed are essential
- Stuff happens. Not usually possible to keep precisely plan – need for control
- Communicate, communicate, communicate!