



## Topics covered

- ✧ Software process models
- ✧ Process activities
- ✧ Coping with change
- ✧ The Rational Unified Process
  - An example of a modern software process.

## Chapter 2 – Software Processes

### Lecture 1

Chapter 2 Software Processes

1

## The software process

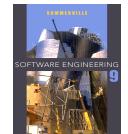


- ✧ A structured set of activities required to develop a software system.
- ✧ Many different software processes but all involve:
  - Specification – defining what the system should do;
  - Design and implementation – defining the organization of the system and implementing the system;
  - Validation – checking that it does what the customer wants;
  - Evolution – changing the system in response to changing customer needs.
- ✧ A software process model is an abstract representation of a process. It presents a description of a process from some particular perspective.

Chapter 2 Software Processes

2

## Software process descriptions



- ✧ When we describe and discuss processes, we usually talk about the activities in these processes such as specifying a data model, designing a user interface, etc. and the ordering of these activities.
- ✧ Process descriptions may also include:
  - Products, which are the outcomes of a process activity;
  - Roles, which reflect the responsibilities of the people involved in the process;
  - Pre- and post-conditions, which are statements that are true before and after a process activity has been enacted or a product produced.

Chapter 2 Software Processes

3

Chapter 2 Software Processes

4

## Plan-driven and agile processes

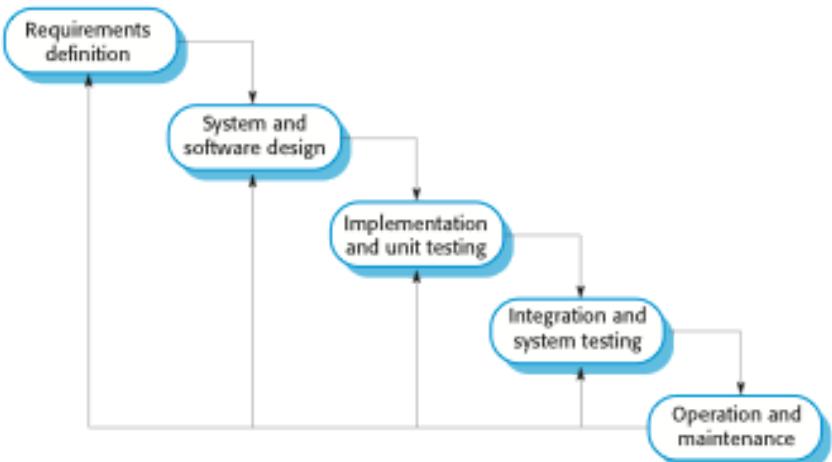


- ❖ Plan-driven processes are processes where all of the process activities are planned in advance and progress is measured against this plan.
- ❖ In agile processes, planning is incremental and it is easier to change the process to reflect changing customer requirements.
- ❖ In practice, most practical processes include elements of both plan-driven and agile approaches.
- ❖ There are no right or wrong software processes.

Chapter 2 Software Processes

5

## The waterfall model



Chapter 2 Software Processes

7

## Software process models



- ❖ The waterfall model
  - Plan-driven model. Separate and distinct phases of specification and development.
- ❖ Incremental development
  - Specification, development and validation are interleaved. May be plan-driven or agile.
- ❖ Reuse-oriented software engineering
  - The system is assembled from existing components. May be plan-driven or agile.
- ❖ In practice, most large systems are developed using a process that incorporates elements from all of these models.

Chapter 2 Software Processes

6

## Waterfall model phases



- ❖ There are separate identified phases in the waterfall model:
  - Requirements analysis and definition
  - System and software design
  - Implementation and unit testing
  - Integration and system testing
  - Operation and maintenance
- ❖ The main drawback of the waterfall model is the difficulty of accommodating change after the process is underway. In principle, a phase has to be complete before moving onto the next phase.

Chapter 2 Software Processes

8

## Waterfall model problems



- ❖ Inflexible partitioning of the project into distinct stages makes it difficult to respond to changing customer requirements.
  - Therefore, this model is only appropriate when the requirements are well-understood and changes will be fairly limited during the design process.
  - Few business systems have stable requirements.
- ❖ The waterfall model is mostly used for large systems engineering projects where a system is developed at several sites.
  - In those circumstances, the plan-driven nature of the waterfall model helps coordinate the work.

Chapter 2 Software Processes

9

## Incremental development benefits

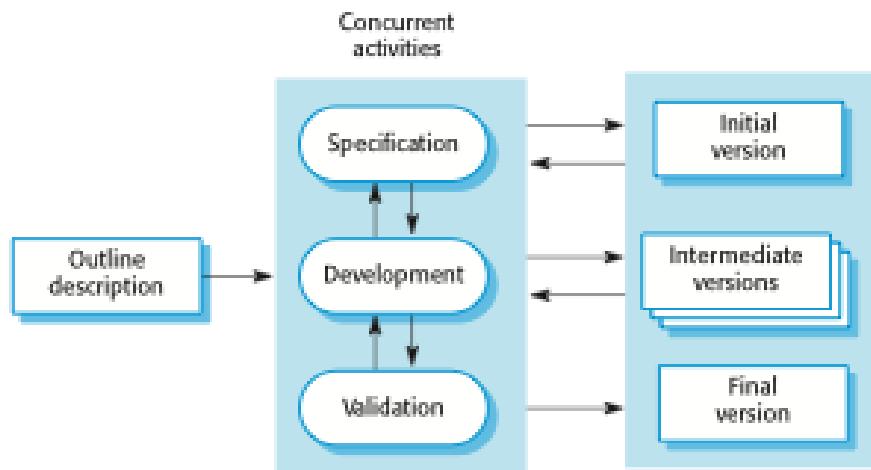


- ❖ The cost of accommodating changing customer requirements is reduced.
  - The amount of analysis and documentation that has to be redone is much less than is required with the waterfall model.
- ❖ It is easier to get customer feedback on the development work that has been done.
  - Customers can comment on demonstrations of the software and see how much has been implemented.
- ❖ More rapid delivery and deployment of useful software to the customer is possible.
  - Customers are able to use and gain value from the software earlier than is possible with a waterfall process.

Chapter 2 Software Processes

11

## Incremental development



Chapter 2 Software Processes

10

## Incremental development problems



- ❖ The process is not visible.
  - Managers need regular deliverables to measure progress. If systems are developed quickly, it is not cost-effective to produce documents that reflect every version of the system.
- ❖ System structure tends to degrade as new increments are added.
  - Unless time and money is spent on refactoring to improve the software, regular change tends to corrupt its structure. Incorporating further software changes becomes increasingly difficult and costly.

Chapter 2 Software Processes

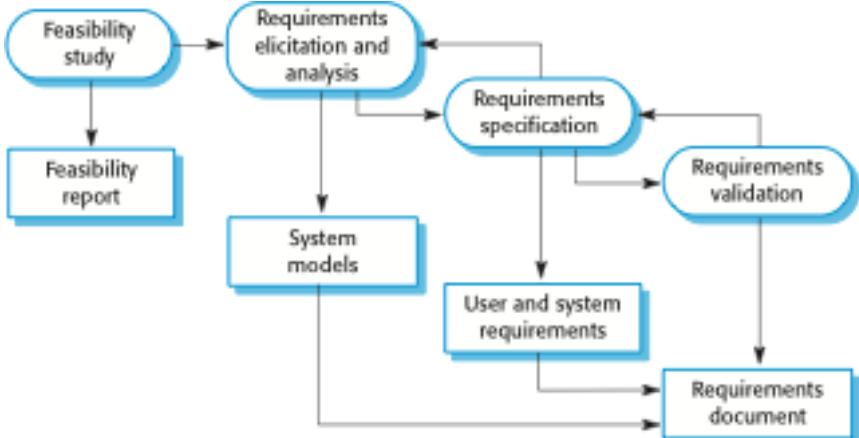
12

## Process activities



- ❖ Real software processes are inter-leaved sequences of technical, collaborative and managerial activities with the overall goal of specifying, designing, implementing and testing a software system.
- ❖ The four basic process activities of specification, development, validation and evolution are organized differently in different development processes. In the waterfall model, they are organized in sequence, whereas in incremental development they are interleaved.

## The requirements engineering process



## Software specification



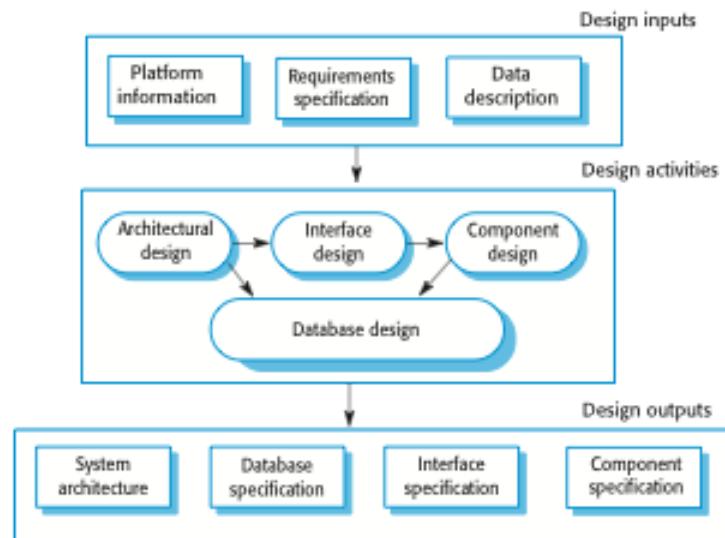
- ❖ The process of establishing what services are required and the constraints on the system's operation and development.
- ❖ Requirements engineering process
  - Feasibility study
    - Is it technically and financially feasible to build the system?
  - Requirements elicitation and analysis
    - What do the system stakeholders require or expect from the system?
  - Requirements specification
    - Defining the requirements in detail
  - Requirements validation
    - Checking the validity of the requirements

## Software design and implementation



- ❖ The process of converting the system specification into an executable system.
- ❖ Software design
  - Design a software structure that realises the specification;
- ❖ Implementation
  - Translate this structure into an executable program;
- ❖ The activities of design and implementation are closely related and may be inter-leaved.

## A general model of the design process



Chapter 2 Software Processes

17

## Software validation



- ✧ Verification and validation (V & V) is intended to show that a system conforms to its specification and meets the requirements of the system customer.
- ✧ Involves checking and review processes and system testing.
- ✧ System testing involves executing the system with test cases that are derived from the specification of the real data to be processed by the system.
- ✧ Testing is the most commonly used V & V activity.

Chapter 2 Software Processes

19

## Design activities



- ✧ *Architectural design*, where you identify the overall structure of the system, the principal components (sometimes called sub-systems or modules), their relationships and how they are distributed.
- ✧ *Interface design*, where you define the interfaces between system components.
- ✧ *Component design*, where you take each system component and design how it will operate.
- ✧ *Database design*, where you design the system data structures and how these are to be represented in a database.

Chapter 2 Software Processes

18

## Boehm's spiral model

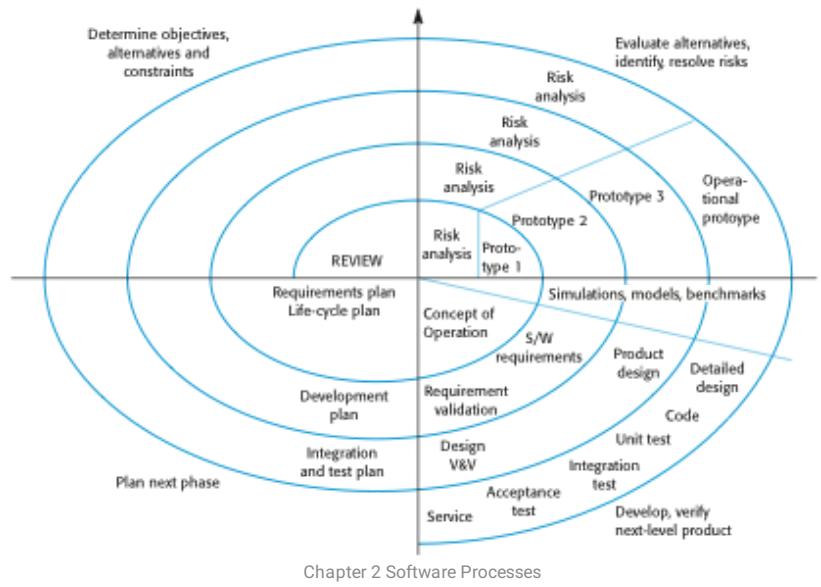


- ✧ Process is represented as a spiral rather than as a sequence of activities with backtracking.
- ✧ Each loop in the spiral represents a phase in the process.
- ✧ No fixed phases such as specification or design - loops in the spiral are chosen depending on what is required.
- ✧ Risks are explicitly assessed and resolved throughout the process.

Chapter 2 Software Processes

20

## Boehm's spiral model of the software process



21

## Spiral model usage



- ❖ Spiral model has been very influential in helping people think about iteration in software processes and introducing the risk-driven approach to development.
- ❖ In practice, however, the model is rarely used as published for practical software development.

## Spiral model sectors



- ❖ **Objective setting**
  - Specific objectives for the phase are identified.
- ❖ **Risk assessment and reduction**
  - Risks are assessed and activities put in place to reduce the key risks.
- ❖ **Development and validation**
  - A development model for the system is chosen which can be any of the generic models.
- ❖ **Planning**
  - The project is reviewed and the next phase of the spiral is planned.

Chapter 2 Software Processes

22

23