THE WATERFALL MODEL

Sommerville, P.65

Activities such as:

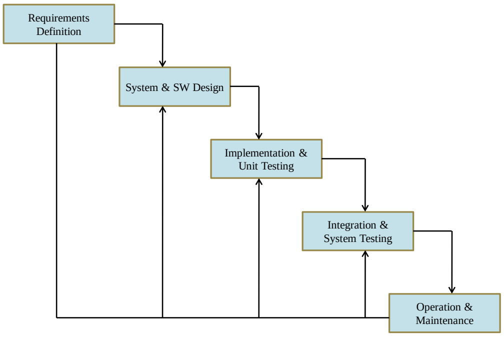
- Specification
- Development
- Validation
- Evolution

are represented as separate *phases eg:*

- *Requirements specification phase*
- *Software design phase*
- *Implementation phase*
- *Testing phase*
- *etc*

- The first published model of software development process
- Evolved from general system engineering processes
- This process cascades from one phase to the next
- Hence the name waterfall model or *system life cycle*
- Process phases map onto development activities

The software life cycle

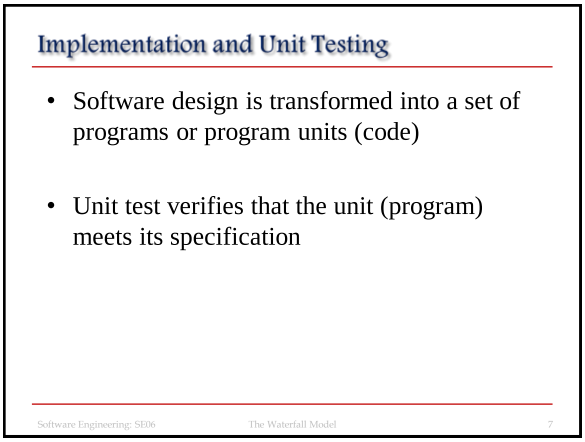


Requirements Analysis & Definition

- This phase establishes the system’s services, constraints and goals
- Consultation with the system users/stakeholders
- When all information is gathered, it is defined in detail
- Document serves as a system specification

System and Software Design

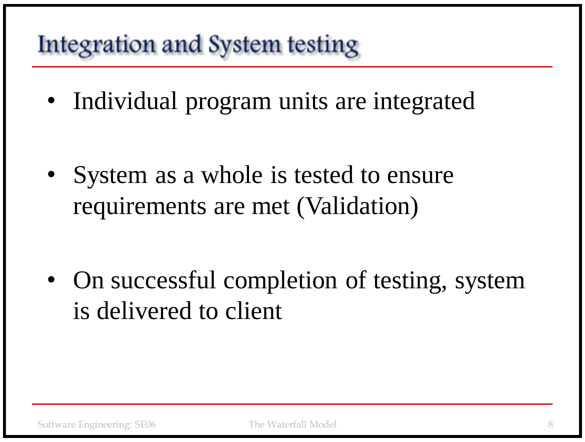
- Partitions the requirements into hardware or software requirements
- Establishes overall system architecture
- Identifies and describes the basic system abstractions and their relationships



Implementation and Unit Testing

- Software design is transformed into a set of programs or program units (code)
- Unit test verifies that the unit (program) meets its specification

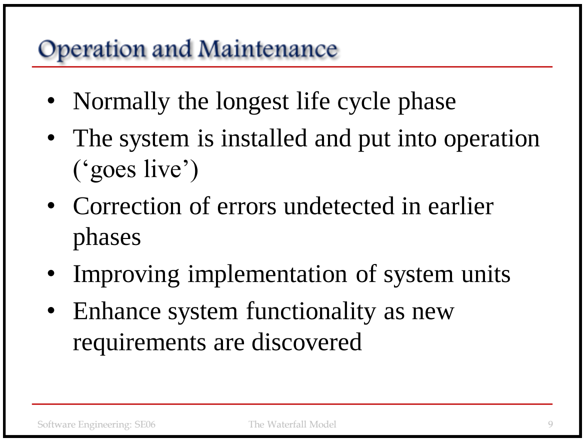
Software Engineering: SE06 The Waterfall Model 7



Integration and System testing

- Individual program units are integrated
- System as a whole is tested to ensure requirements are met (Validation)
- On successful completion of testing, system is delivered to client

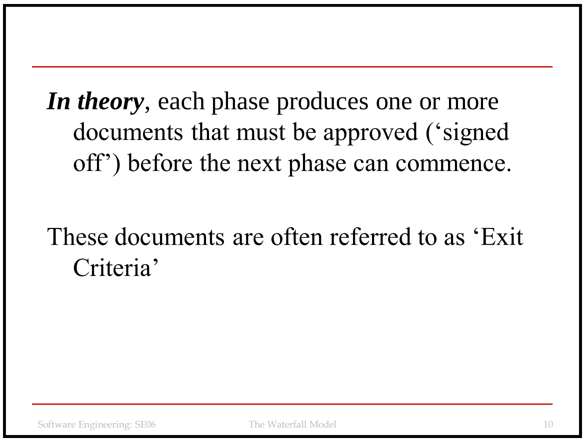
Software Engineering: SE06 The Waterfall Model 8



Operation and Maintenance

- Normally the longest life cycle phase
- The system is installed and put into operation (‘goes live’)
- Correction of errors undetected in earlier phases
- Improving implementation of system units
- Enhance system functionality as new requirements are discovered

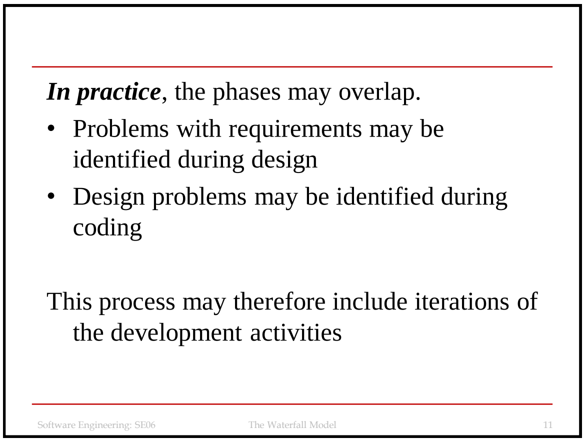
Software Engineering: SE06 The Waterfall Model 9



In theory, each phase produces one or more documents that must be approved (‘signed off’) before the next phase can commence.

These documents are often referred to as ‘Exit Criteria’

Software Engineering: SE06 The Waterfall Model 10

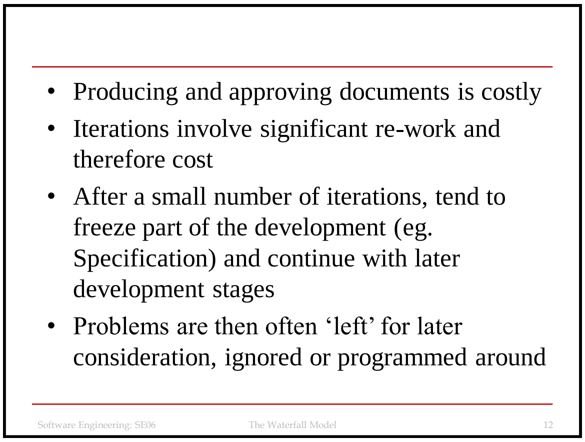


In practice, the phases may overlap.

- Problems with requirements may be identified during design
- Design problems may be identified during coding

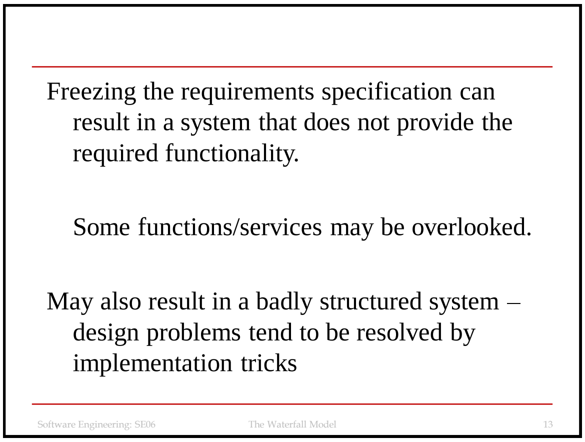
This process may therefore include iterations of the development activities

Software Engineering: SE06 The Waterfall Model 11



- Producing and approving documents is costly
- Iterations involve significant re-work and therefore cost
- After a small number of iterations, tend to freeze part of the development (eg. Specification) and continue with later development stages
- Problems are then often ‘left’ for later consideration, ignored or programmed around

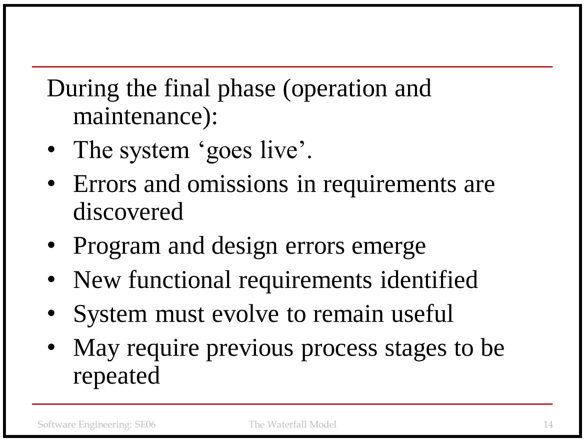
Software Engineering: SE06 The Waterfall Model 12



Freezing the requirements specification can result in a system that does not provide the required functionality.

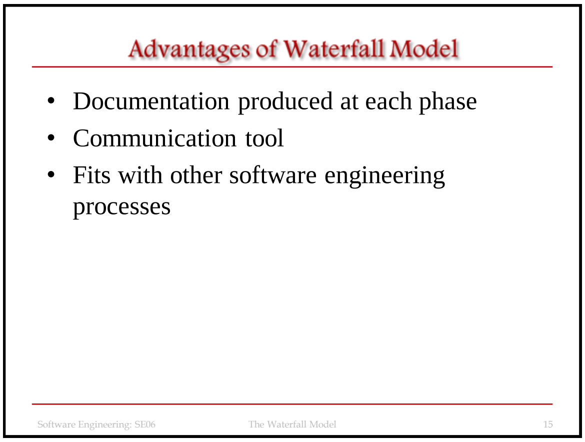
Some functions/services may be overlooked.

May also result in a badly structured system – design problems tend to be resolved by implementation tricks



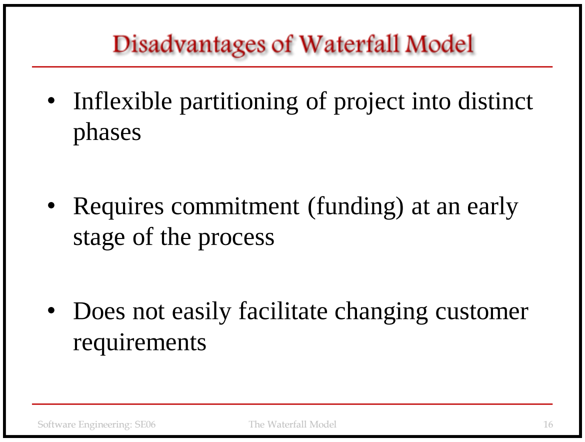
During the final phase (operation and maintenance):

- The system ‘goes live’.
- Errors and omissions in requirements are discovered
- Program and design errors emerge
- New functional requirements identified
- System must evolve to remain useful
- May require previous process stages to be repeated



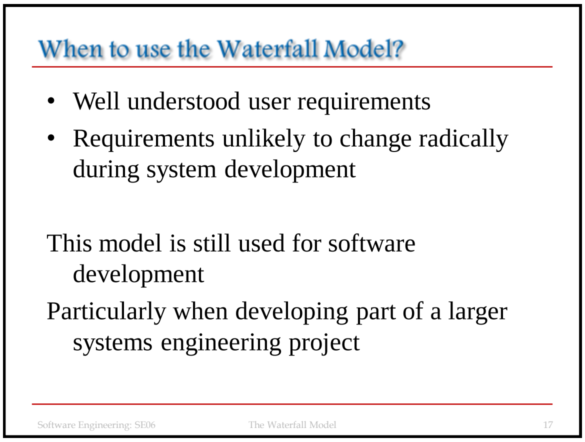
Advantages of Waterfall Model

- Documentation produced at each phase
- Communication tool
- Fits with other software engineering processes



Disadvantages of Waterfall Model

- Inflexible partitioning of project into distinct phases
- Requires commitment (funding) at an early stage of the process
- Does not easily facilitate changing customer requirements



When to use the Waterfall Model?

- Well understood user requirements
- Requirements unlikely to change radically during system development

This model is still used for software development

Particularly when developing part of a larger systems engineering project