

1

IT4490 - SOFTWARE DESIGN AND CONSTRUCTION

**2. SOFTWARE DEVELOPMENT PROCESS**

---



1

2

**References**

[1] ISO/IEC FDIS 12207, *Systems and software engineering — Software life cycle processes*.

2

3

**Content**

1. Software Life Cycle Process
2. Software Implementation Process
3. Software Development Models

3

4

**Content**

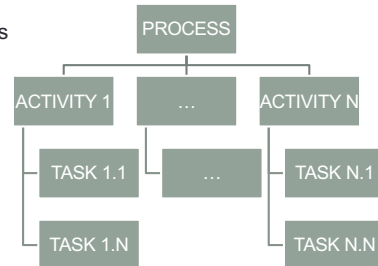
⇒ 1. Software Life Cycle Process

2. Software Implementation Process
3. Software Development Models

4

## 1. Software Life Cycle Process

- “ISO/IEC 12207:2008, Systems and software engineering — Software life cycle processes”
  - The latest and International Standard Software Development Process
- “The life cycle begins with an idea or a need that can be satisfied wholly or partly by software and ends with the retirement of the software.”
- Standard implementation works
  - hierarchically as processes



5

## What are International Standards?

- ◆ In ISO, all industry standards, including Information Technology, are developed.
- ◆ In the field of Information Technology, in ISO/IEC JTC1, international standards are developed.
- ◆ ISO/IEC JTC1 has 32 principal member bodies which develop the international standards, and 44 observer member bodies.
- ◆ Some abbreviation
  - ISO: International Organization for Standard
  - IEC: International Electrotechnical Commission
  - JTC1: Joint Technical Committee

6

## Why International Standards?

- ◆ Standards are important, especially in ICT
  - Basis of common understanding such as frameworks, and terminology / definitions
  - TBT Agreement of WTO recommends the use of ISO Standards for governmental purchase in affiliate countries
  - Based on some standards, the certifications can be got, and they make some appeal points in international transaction

WTO: World Trade Organization

TBT Agreement: Agreement on Technical Barriers to Trade

7

## Software Life Cycle Process

- “This International Standard groups the activities that may be performed during the life cycle of a software system into seven process groups”<sup>[1]\*</sup>:
  1. Agreement Processes: 2 processes
  2. Organizational Project-Enabling Processes: 5 processes
  3. Project Processes: 7 processes
  4. Technical Processes: 11 processes
  5. **Software Implementation Processes: 6 processes**
    - Purpose: “to produce a specified system element implemented as a software product or service”<sup>[1]\*\*</sup>.
  6. Software Support Processes: 8 processes
  7. Software Reuse Processes: 3 processes

[1]\*: clause 5.2.1; pp. 13, [1]\*\*: clause 7.1.1.1; pp. 57,

8

9

## Content

1. Software Life Cycle Process
- ⇒ 2. Software Implementation Process
3. Software Development Models

9

10

## 2. Software Implementation Process

**Software Implementation Process** includes the following lower-level processes:

1. **Software Requirements Analysis Process**
2. **Software Architecture Design Process**
3. **Software Detailed Design Process**
4. **Software Construction Process**
5. **Software Integration Process**
6. **Software Qualification Testing Process**

10

11

### 2.1. Software Requirements Analysis process

- Purpose: “to establish the requirements of the software elements of the system” [1]
- Main items written on the brief requirement description
  - **System environmental conditions** under which the software is to perform.
  - The **functional requirements** and the **interface requirements**.
  - **Data definition and database requirements**.
  - Some **non-functional requirement** items such as reliability, usability, time efficiency
  - **Qualification requirements**: The requirements are used as criteria or conditions to qualify a software product as complying with its specifications.

[1]: Session 7.1.2.1; pp. 59

11

12

### 2.2. Software Architectural Design process

- Purpose: “to provide a design for the software that implements and can be verified against the requirements” [1]
- Software architecture is designed from the software requirements
- Main items
  - a top-level structure of the software and the software components which constructs the software
  - a top-level design for the interfaces external to the software and between the software components
  - a top-level design for the database

[1]: Session 7.1.3.1; pp. 61

12

## 2.3. Software Detailed Design process

- Purpose: “to provide a design for the software that implements and can be verified against the requirements and the software architecture and is sufficiently detailed to permit coding and testing” [1]
- A detailed design for each software components are developed. In the detailed design, the following items are developed:
  - **each component** is refined into **software units** that can be coded, compiled, and tested
  - **the interfaces external to the software item, between the software components, and between the software units**

[1]: Session 7.1.4.1; pp. 62

13

## 2.4. Software Construction process

- Purpose: “to produce executable software units that properly reflect the software design” [1]
- Main items to be developed:
  - Each software unit and database
  - Test procedure and test data for software unit and database
  - Unit tests and database test
- Evaluate code and test results considering this criteria: traceability to requirements, external consistency with requirements, internal consistency, test coverage of units, appropriateness of coding methods and standards used, feasibility of integration, testing, operation, and maintenance

[1]: Session 7.1.5.1; pp. 63

14

## 2.5. Software Integration process

- Purpose: “to combine the software units and software components, producing integrated software items, consistent with the software design, that demonstrate that the functional and non-functional software requirements are satisfied on an equivalent or complete operational platform” [1]
- Main tasks
  - An integration plan, including test requirements, test procedure, and test cases/data.
  - Integration of software units/components
  - Program/software/integration test

[1]: Session 7.1.6.1; pp. 64

15

## 2.6. Software Qualification Testing Process

- Purpose: “to confirm that the integrated software product meets its defined requirements” [1].
- Qualification testing in accordance with the qualification requirements for the software item is conducted
  - Tests, test cases, and test procedures
- The implementer supports audit(s) to conform the software meets to the qualification requirements
  - If it is successful completion of the audits, the implementer prepare the deliverable software product for System Construction process

[1]: Session 7.1.7.1; pp. 66

16

## Summary

- “Software Life Cycle Process – SLCP” is the **international standard** processes focused on the development and support of Application Software.
- SLCP can be used as a **common language** among the stakeholders such as acquirers and suppliers. They can communicate or order the software development using SLCP. For example, we can say “To order **the software detailed design process or later software implementation processes** of new library system”.

17

## Learning points

What are the main factors for the system development success?

1. The Software to be developed meets to **functional** requirements → ?
2. To keep appointed date of **delivery** → ?
3. Meets the required **quality** such as Reliability, Usability, Performance, Maintainability → ?
4. Necessary to provide **maintenance** activity during the system operation period → ?

18

## Content

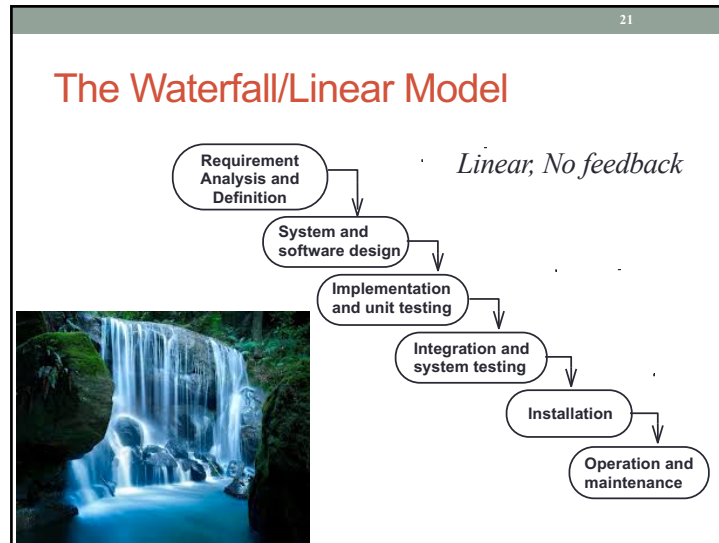
1. Software Life Cycle Process
2. Software Implementation Process
- 3. Software Development Models

19

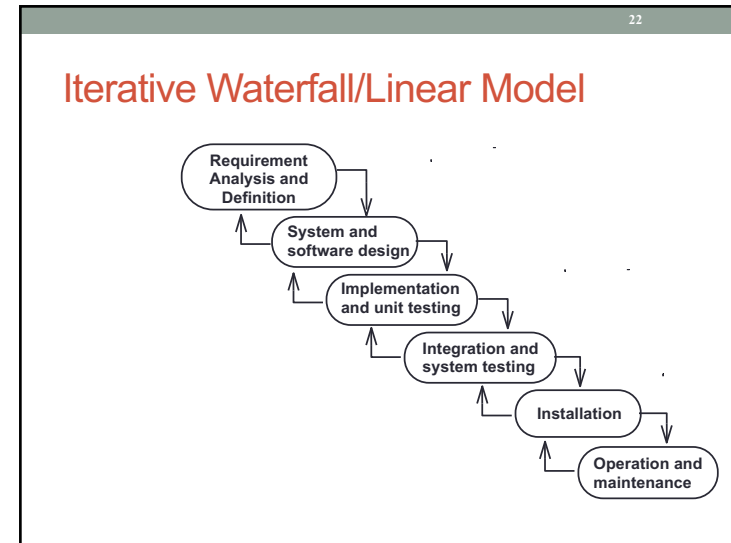
## Software Development Models

- Waterfall model
- Iterative model
- Prototype model
- Spiral model
- Agile methodology

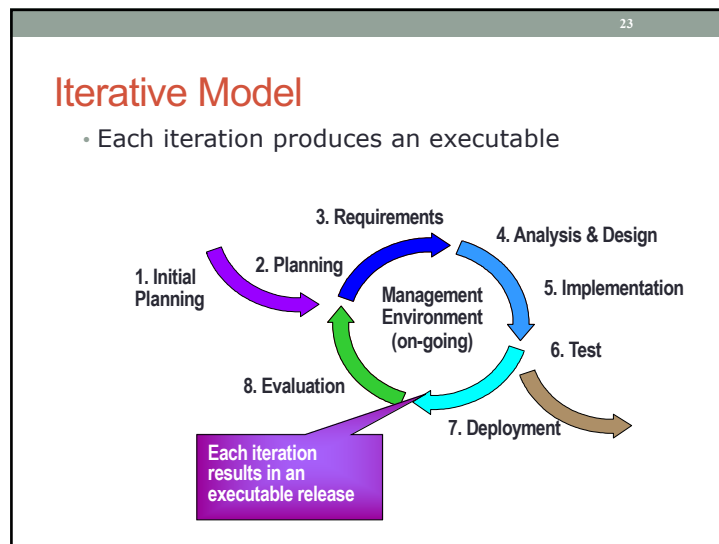
20



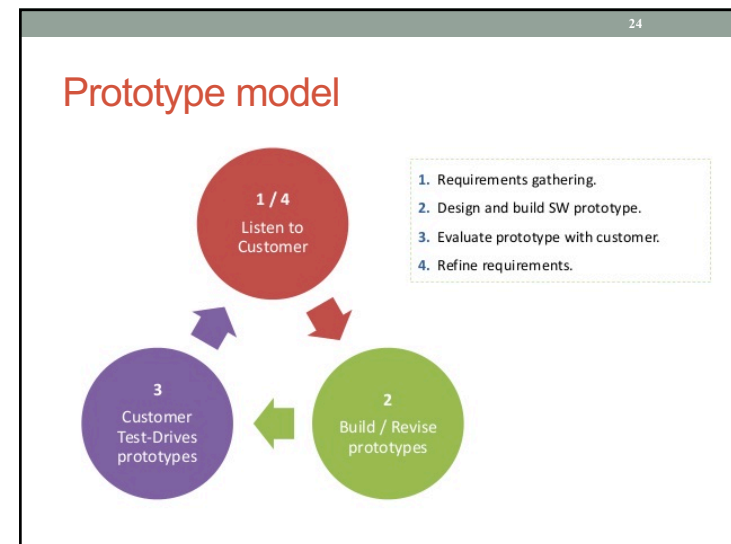
21



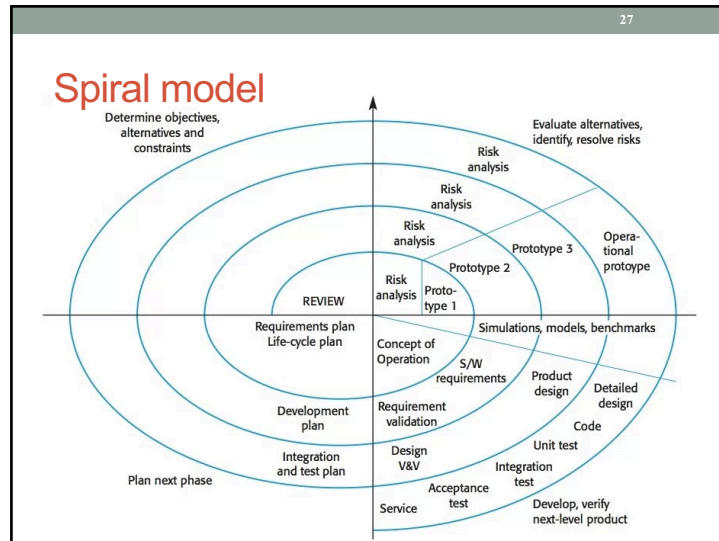
22



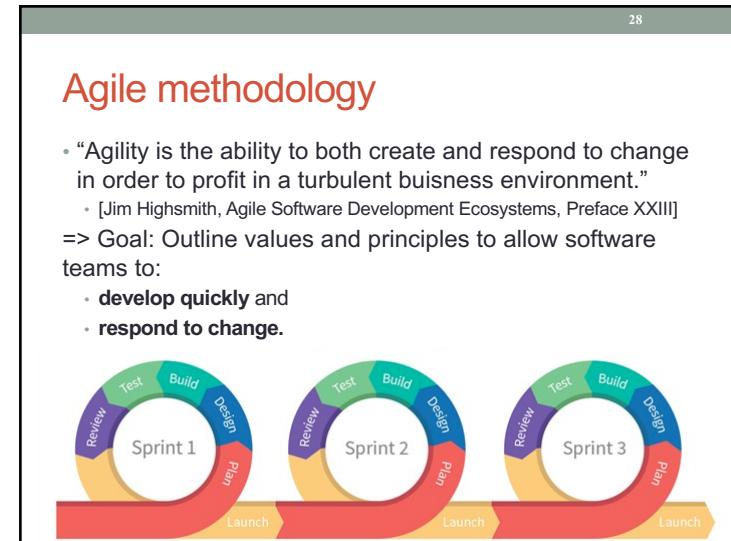
23



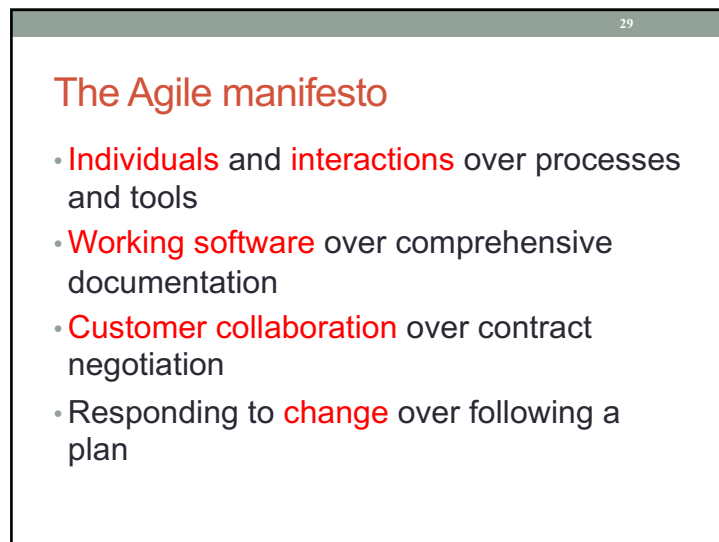
24



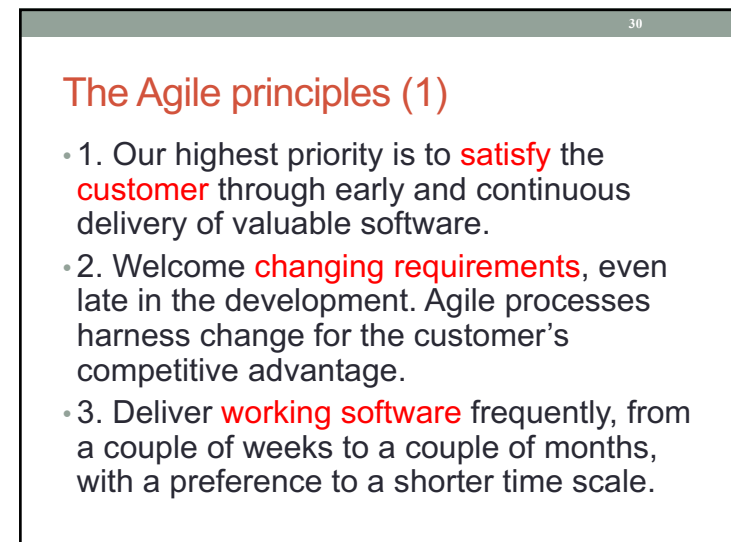
27



28



29



30

## The Agile principles (2)

- 4. **Business** people and **developers** must work together daily throughout the project.
- 5. Build projects around motivated individuals. Give them the environment and support their need, and **trust** them to get the job done.
- 6. The most efficient and effective method of conveying information to and within a development team is **face-to-face conversation**.

31

## The Agile principles (3)

- 7. **Working software** is the primary measure of progress.
- 8. Agile processes promote **sustainable** development.
- 9. The sponsors, developers, and users should be able to maintain a **constant pace**.
- 10. Continuous attention to technical excellence and good design enhances agility.

32

## The Agile principles (4)

- 11. **Simplicity** – the art of maximizing the amount of work not done – is essential.
- 12. The best architectures, requirements, and designs emerge from **self-organising teams**.
- 13. At regular intervals, the team **reflects** on how to become more effective, then tunes and **adjusts** its behaviour accordingly.

33