

# Vietnam and Japan Joint ICT HRD Program

ITSS Software Development

## **Chapter 1. Software Development Process**

Nguyen Thi Thu Trang

# Content

1. Business Application System (BA System)
2. Software Life Cycle Process
3. Construction of a Software Development Process
4. Software Development Models

# Content

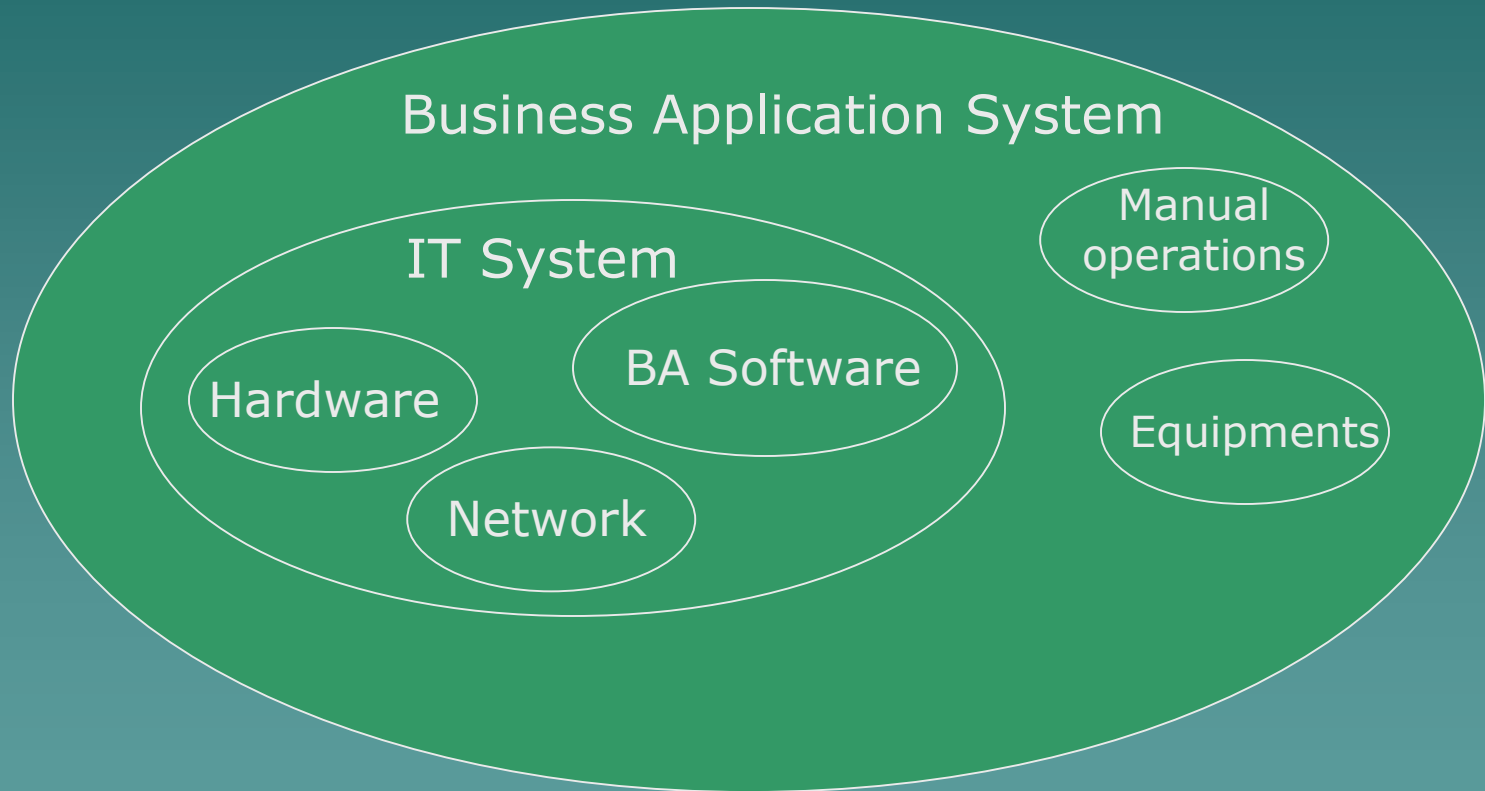


- 2. Software Life Cycle Process
- 3. Construction of a Software Development Process
- 4. Software Development Models

# 1.1. What is a Business Application System?

- ◆ “Whole system to accomplish the business”
  - Example: Personnel System, Payment System...
  - In this course, Computerized **BA System** is treated.
- ◆ The **BA System** includes:
  - hardware, networks, software\*, and manual operations
    - ◆ software\*: Includes platform software (OS...), middle software, and “**Business Application Software**”

# A big picture of Business Application System



# Importance of new IT System

- ◆ “There is No new business without new IT system”
- ◆ IT system enables organizations to react, respond, serve, store, retrieve, disseminate, and control their new valuable asset that is information.

## 1.2. Scope of this course

- ◆ “**Business Application Software**” is the software to carry out the business
- ◆ Major work of the Application Specialists (AS) is development of “**BA Software**”.
- ◆ “**BA Software**” is built up on two ways:
  - One is to develop the software from scratch
  - The other way is to construct base on package software

# 1.2. Scope of this course (2)

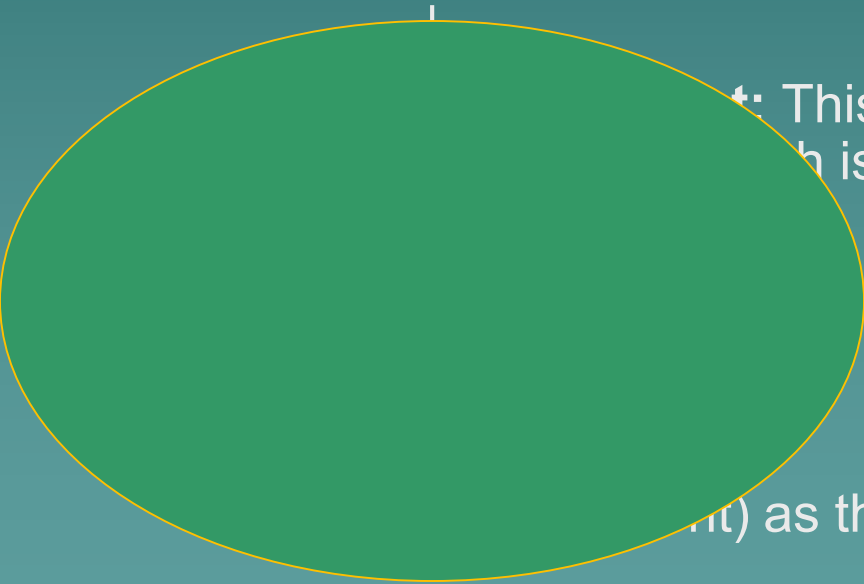
- ◆ In this course, the following things will be learned considering the development **BA Software** from scratch
  - Development Process, Development Technologies, Quality Management, Project Management for Business Application Software are overviewed.
  - On a case study, students go through an experience of **Business Application Software** development from requirements analysis to design.
  - Testing and Quality management in real project are overviewed



# 1.3. Life Cycle of BA System

- ◆ In general, “Life Cycle” means “from birth until death”
- ◆ In the case of business application system, the Life Cycle means from the conception to the retirement such as

## 1. System Planning (conception)



†: This includes development of business which is one the component of the system.

(it) as the end of maintenance

# 1.3.1. Life Cycle “Process” of System and Software development

1. Analysis and Design of Business Application System
  - **System Requirements Analysis process**
    - ◆ Stake holder's requirements are extracted and defined.
  - **System Architectural Design process**
    - ◆ Envisage whole system structure is designed and the system components such as hardware, software, network, manual operation.
2. Development of Business Application Software as a component of the System
  - Software Requirements Analysis, Software Architectural Design and so on.
  - ➔ **Software Development processes** will be mentioned in detail later.

# 1.3.1. Life Cycle “Process” of System and Software development (2)

## 3. Integration and Construction of Business Application System

### – System Integration process/System Qualification Testing process

- ◆ Integrate the components including the business application software into a system and System Qualification Testing is conducted.
- ◆ If the test is successful completion, Software developer can start to prepare the software deliverables

## 1.3.2. Life Cycle “Process” of System and Software operation/maintenance

4. Preparation of Business Application System Operation
  - **Software Installation Process/ Software Acceptance Process**
    - ◆ The software product is installed in the target environment and acceptance review and testing of the software product are conducted.
    - ◆ Depending on the contract, software developer has to assist the activities.
5. Operation of Business Application System is started.
  - **Software Operation Process**
    - ◆ New Business based on the new system starts then!!!
6. System and Software maintenance starts at the same time as operation
  - **Software Maintenance Process**
    - ◆ Depending on the contract, software developer has to assist the activities.
  - ➔ When the System and the software operation are terminated, it is the system/software is disposed, the maintenance is also terminated.

## 1.4. Learning points from the Life Cycle

“The customer’s business success depends on the system development success.”

Business Application Software is one of the major components of the Business Application System.

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 → ?

# Content

1. Business Application System (BA System)

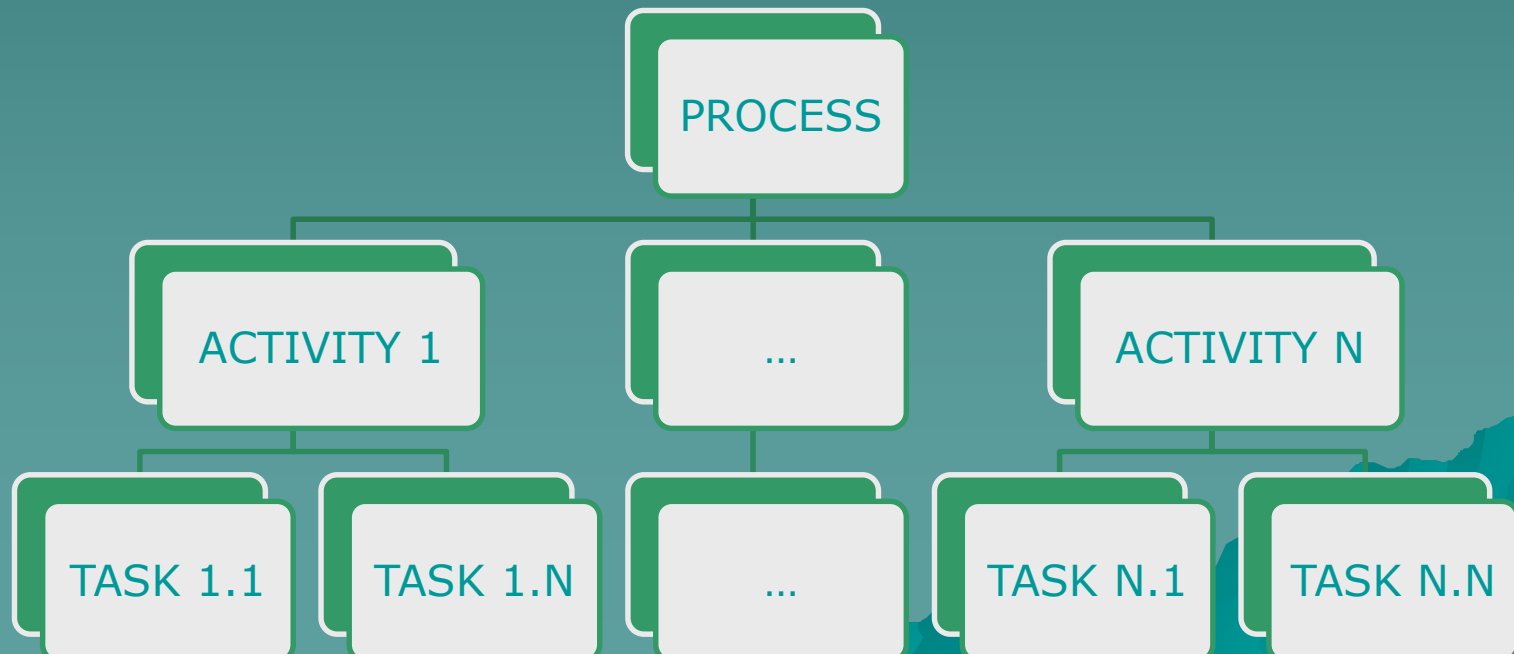


3. Construction of a Software Development Process

4. Software Development Models

# 2.1. Software Life Cycle Process Overview

- ◆ Based on “ISO/IEC12207 System and Software Engineering - Software Life Cycle Processes”
- ◆ “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.”
- ◆ In this standard, standard implementation works are clarified and defined hierarchically as processes



# International Standards

- ◆ In ISO, all industry standards, excluding 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




# Why International Standards?

- ◆ All time Standards are important. However, in the field of information technology, importance of International Standards are growing now
  - In international transaction, international standards provide the 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 (Vietnam became 150<sup>th</sup> member of WTO in January 2007, Japan is also member of WTO).
  - 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

## 2.1. Software Life Cycle Process Overview

- ◆ This International Standard groups the activities that may be performed during the life cycle of a software into 7 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.  Processes
  6. Software Support Processes: 8 processes
  7. Software Reuse Processes: 3 processes

[1]: Session 5.2.1; pp. 13

## 2.2. Software Implementation Process overview

System Requirements Analysis Process and System Architectural Design Process are achieved just before Software Implementation Process .

**Software Implementation Process** includes the following 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**

## 2.3. Software Requirements Analysis process

- ◆ Purpose: “to establish the requirements of the software elements of the system” [1]
- ◆ In this process, the following items written on the brief requirement description should be analyzed and defined:
  - **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. → Non-functional requirements will be explained later!
  - **Qualification requirements**: The requirements are used as criteria or conditions to qualify a software product as complying with its specifications.

## 2.3. Software Requirements Analysis process (2)

- ◆ Software implementer should analyze and make an internal review [1]
  - Consistency and traceability to requirements and design
  - Testability
  - Feasibility of software design, operation and maintenance
- ◆ Software implementer should conduct the common review with ordering party and confirm the result of Requirements analysis and the software development plan
  - Following a successful evaluation and review, the software requirements should be approved, baselined and communicated to all affected parties.
  - Subsequent changes to the software requirements baseline should be evaluated for cost, schedule and technical impact.

## 2.4. Software Architectural Design process

- ◆ Purpose: “to provide a design for the software that implements and can be verified against the requirements” [1]
- ◆ In this process, a software architecture is designed from the software requirements.
- ◆ In the software architecture, the following items are define:
  - **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**

## 2.4. Software Architectural Design process (2)

- ◆ All the requirements for the software item are allocated to its software components and further refined to facilitate detailed design
- ◆ The implementer should evaluate the architecture of the software item and the interface and database designs considering internal external consistency and feasibility of detailed design, operation, and maintenance.

## 2.5. 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]
- ◆ In this process, 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.**



## 2.5. Software Detailed Design process (2)

- ◆ The detailed design of the interfaces must permit coding without the need for further information
- ◆ All the software requirements are allocated from the software components to software units

## 2.6. Software Construction process

- ◆ Purpose: “to produce executable software units that properly reflect the software design” [1]
- ◆ The following items are developed:
  - **Each software unit and database**
  - **Test procedure and test data** for software unit and database
  - **Unit tests and database test** are also carried out.
- ◆ The implementer shall evaluate software code and test results considering internal external consistency, test coverage of units and, traceability to the requirements and design of the software.

## 2.7. 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

## 2.7. Software Integration process (2)

- ◆ **An integration plan** to integrate the software units and software components into the **software** is developed. The plan includes **test requirements, test procedure, and test cases/data**.
- ◆ **The software units and software components are integrated.** And the **program/software test** is achieved.
- ◆ Tests, test cases, and test procedures for **Software Qualification Testing** are developed and documented. It is ensured that the integrated software is ready for Software Qualification Testing

## 2.8. Software Qualification Testing

- ◆ 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

## 2.9. Summary

### ◆ System Construction Process

- Following the Software Implementation Process, **System Construction Process**, which consists of the following processes, are achieved
- Software Developer may assist the following process depending on contract
  - ◆ System Integration Process
  - ◆ System Qualification Testing
  - ◆ Software Installation Process
  - ◆ Software Acceptance Support Process

## 2.9. Summary (2)

- ◆ “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”.

# Content

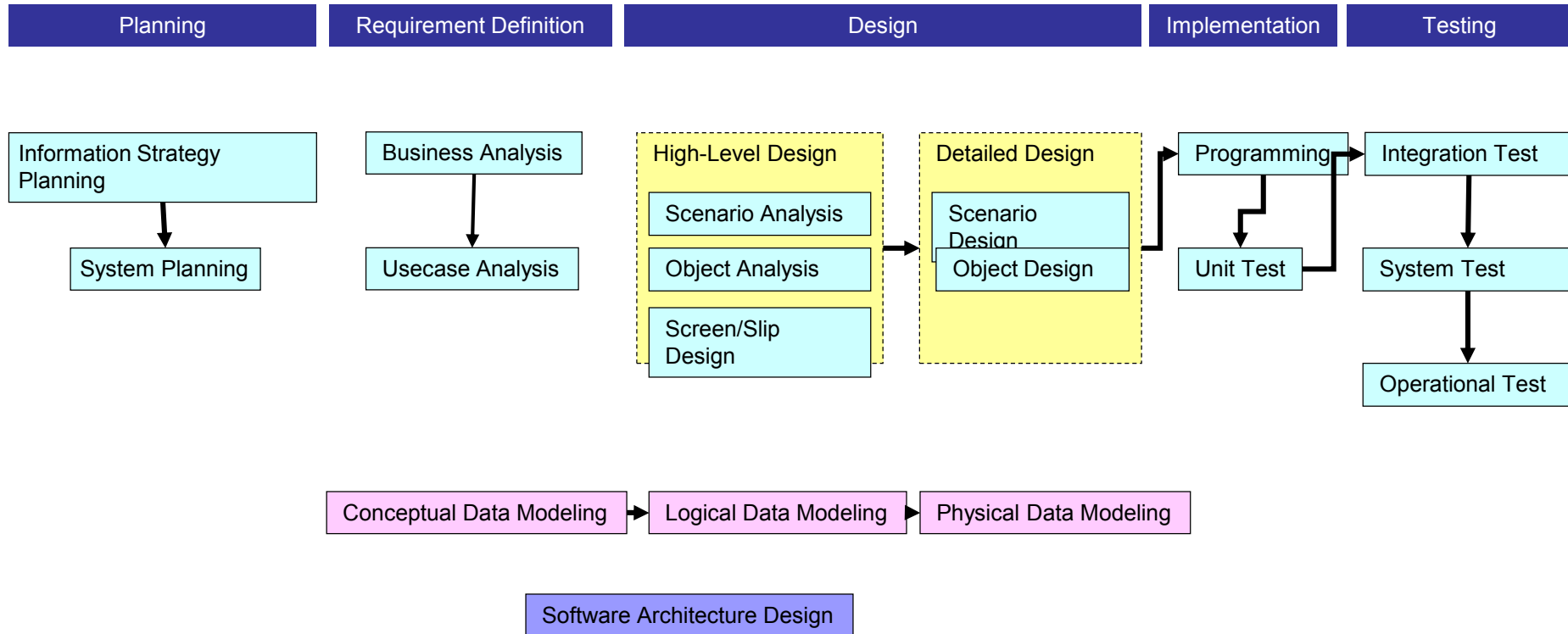
1. Business Application System (BA System)
2. Software Life Cycle Process



- 
- 
- 
4. Software Development Models



# Relation of Software development activities



# 3.1. Overview

- ◆ When a business application software is developed in organization such as a company, the development works are achieved and managed in accordance with “the organizational software development process/procedure” which should be based on SLCP.
- ◆ The software development process/procedure depends on type of software developed and type of technology adapted.
- ◆ Therefore, in this part, we try to construct the software development process/procedure for our case study “Web-based application system”.

# 3.1. Overview (2)

- ◆ What type of the business application software is developed?
  - What software do we plan to develop?  
Business Requirements, System Requirements, and Software Requirements are basic information
- ◆ What process model and technology should be adapted?
  - Development Process Model
  - Development Language
  - Analysis and Design Method
  - Software Application Framework
  - High Quality Design method and Testing Method

## 3.2. Business Application Software

- ◆ “What type of the business application software is developed?”
  - Some major parts of the Business Requirements, and the System Requirements are already described in System Requirements Analysis process and System Architecture Design process
  - Some parts of Software Requirements is already described also.

## 3.2. Business Application Software (2)

- ◆ The following pieces of information are mainly needed for constructing the development process/procedure of our business application the “Course Registration System for University”
  - Web-based Application system with Database
  - Main users are lecturers and students.
  - Particulars mentioned below, conform to the university system standards.
    - ◆ Platform and network: Windows, Ether-net of university internal LAN
    - ◆ System administration user management: Administrators of university management systems are managed.
    - ◆ User Interface: Internet Explorer based UI for the main user.
    - ◆ Reliability, Usability, Security, Performance, Maintenance

## 3.3. Process model and technology

“What the process model and the technology should be adapted?”

### ◆ 1. “Development Process Model”

- Waterfall model?, Incremental model?
- Or some other model?

### ◆ Discussion points:

- Merit of Waterfall model is easy to managed new or comparative large system. Weakness is difficult to control to change specification and design.
  - Merit of Incremental model is comparatively easy to managed to change specification and design
  - What about “agile”? Is it a process model or a method?
- Here we choose the waterfall model!

## 3.3. Process model and technology (2)

“What the process model and the technology should be adapted?”

### ◆ 2. “Development Language”

- Java!!
- Or any other candidate for Web-based Application?

### ◆ Discussion points

- What do acquirers say? Do they require a certain language?
  - Suitable for Web-base system?
  - How many programmers and maintenance people are needs?
  - Development environments is fine?
  - Framework is available and proven?
- Here we choose Java!!

## 3.3. Process model and technology (3)

“What the process model and the technology should be adapted?”

### ◆ 3. “Analysis and Design Method”

- Structured Design
- Objected-Oriented or any other methods?

### ◆ Discussion points

- What do acquirers say? Do they require a certain method?
- What language is used in this project? For Cobol based dev.???
- Are there many successful project the method used. Proven?
- What method is familiar with? I am expert for structured design!!
- Environment and framework are available and proven?
- Here we choose Object-Oriented method!!



## 3.3. Process model and technology (4)

“What the process model and the technology should be adapted?”

### ◆ 4. “Software Application Framework”

- Should we use any application frameworks?

### ◆ Discussion points

- What merits can we get using a application framework?
  - What about “Struts”? Any other candidates? MVC model design?
  - Proven? What about development efficiency?
  - What about maintainability?
  - What do acquirers say? Do they require a certain method?
- Here we choose Struts!!

## 3.3. Process model and technology (5)

“What the process model and the technology should be adapted?”

- ◆ 5. **“High Quality Design method and Testing Method”**
    - As for Reliability, Usability, Security, Performance, Maintenance, to be explained and discussed later.
- We decided to adopt “Upper Stream Quality built-in Method” and “Systematic Testing Method”!!

## 3.3. Process model and technology (6)

- ◆ Summary: We decide to adopt the model, language, and method:
  - Process model: waterfall model
  - Programming language: Java
  - Analysis and design method: Object Oriented method
  - Application framework: Struts
  - Quality design method: “Upper Stream Quality built-in method”
  - Testing method: “Systematic Testing method”

To be constructed **software** based on the choices mentioned above

## 3.4. Software Development Process

- ◆ Software Requirement Analysis process
- ◆ Software Architectural Design process
- ◆ Software Detailed Design process
- ◆ Software Construction process
- ◆ Software Integration process
- ◆ Software Qualification Testing

## 3.4.1. Software Requirement Analysis process

- ◆ Following items written on the brief requirements description are analyzed and defined
  - System environmental conditions are confirmed and defined.
  - The functional and the interface requirements external to the software are analyzed and defined using Use Case Diagram and Use Case Scenarios.
  - Data definition and database requirements are also defined in UCS.
  - Non-functional requirement are analyzed and defined → To be explained later

## 3.4.1. Software Requirement Analysis process (2)

- ◆ Software developers conduct an internal review in the following view
  - External consistency, and tractability to brief requirements description
  - Internal consistency within requirements description
  - Implementation feasibility and the software development plan
- ◆ Developers should conduct the common review with acquirers and confirm the result of Requirements analysis and the software development plan

## 3.4.2. Software Architectural Design process

- ◆ In this process, a software architecture is designed based on the business, system, and software requirements such as the followings
    - Objects, such as MVC\* classes, are extracted using OOSE method, and analysis model is defined using Class Diagram
    - Communications external to the software and between the class objects are analyzed and defined using Sequence Diagram
    - The top-level design for the database is also included in the class and sequence diagram as “entity object”
- \*) The software application frame work is based in MVC modeling

## 3.4.2. Software Architectural Design process (2)

- ◆ All software requirements should be allocated to its software components such as class object and further refined to facilitate detailed design
- ◆ The developer should evaluate the software architecture, the interface and database designs considering internal external consistency and feasibility of detailed design and also non-functional requirements.



## 3.4.3. Software Detailed Design process

- ◆ In this process, a detailed design of each software components such as class object are developed
  - Each class object is refined into software **units** that can be coded, compiled, and tested
  - The interfaces external to the software, and between the class objects are also refined as the same granularity level as the units
  - Basic Data model is described using Entity-Relationship Diagram and refined into implementation data model by normalizations

### 3.4.3. Software Detailed Design process (2)

- ◆ All software requirements should be allocated to software components such as class object and further refined into detailed class object and detailed E-R model
- ◆ The developer should evaluate detailed design of the software, the interface and database designs considering internal external consistency and feasibility of construction and also non-functional requirements

## 3.4.4. Software Construction process

- ◆ In this process, **executable** software **units** that properly reflect the software design are produced
- ◆ The following items are developed
  - Each software unit is coded in Java language accordance with coding rule standards
  - The database is implemented based on E-R model
  - Test procedure and test data for software unit and database
  - Unit tests and database test are also carried out
- ◆ The developer evaluates software code and test results considering internal external consistency, test coverage of units and, traceability to the requirements and design of the software

## 3.4.5. Software Integration process

- ◆ An integration plan to integrate the software units and software components into the **software** is developed. The plan includes test requirements, test procedure, and use case-based test cases/data
- ◆ The software units and software components are integrated in the windows environment. And the program/software test is achieved
- ◆ Tests, test cases, and test procedures for **Software Qualification Testing** are developed and documented. The testing plan is developed based on software requirements described in Use Case and Use case scenario

## 3.4.6. Software Qualification Testing

- ◆ In this process, it is confirm that the integrated software product meets its defined requirements
- ◆ Qualification testing in accordance with the **qualification requirements** for the software item is conducted
- ◆ The developer 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

## 3.4.7. Summary

- ◆ We made discussions and developed the software development process for the Case Study “Course registration system”
- ◆ However, we have one more issue to realize the software development process
- ◆ In next chapter, **Quality including non-functional qualities** are introduced and refine the our software development process

# Content

1. Business Application System (BA System)
2. Software Life Cycle Process
3. Construction of a Software Development Process

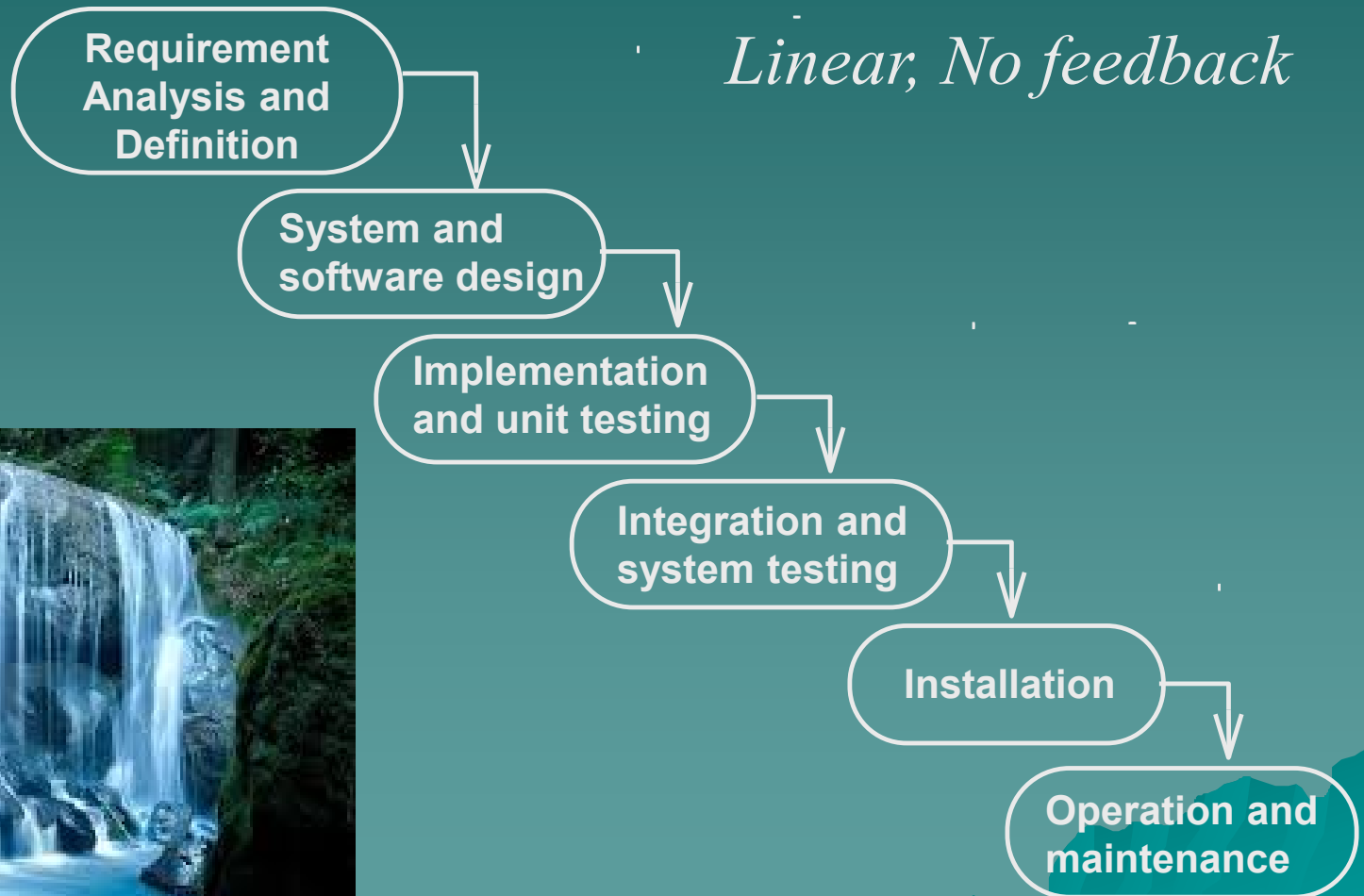


# Software Development Models

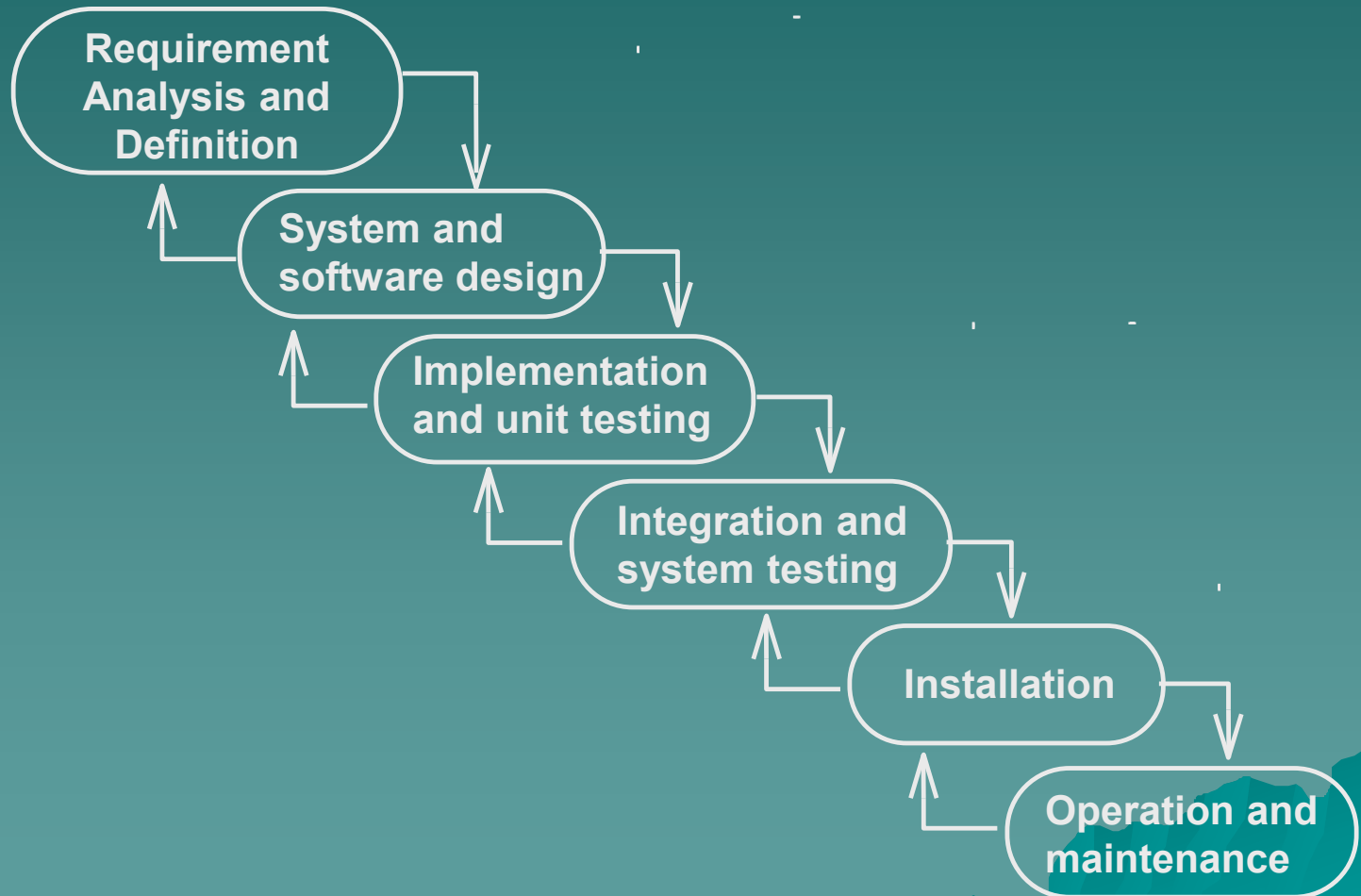
- ◆ Waterfall model
- ◆ Iterative model
- ◆ Prototype model
- ◆ Spiral model
- ◆ Agile methodology



# The Waterfall/Linear Model



# Iterative Waterfall/Linear Model

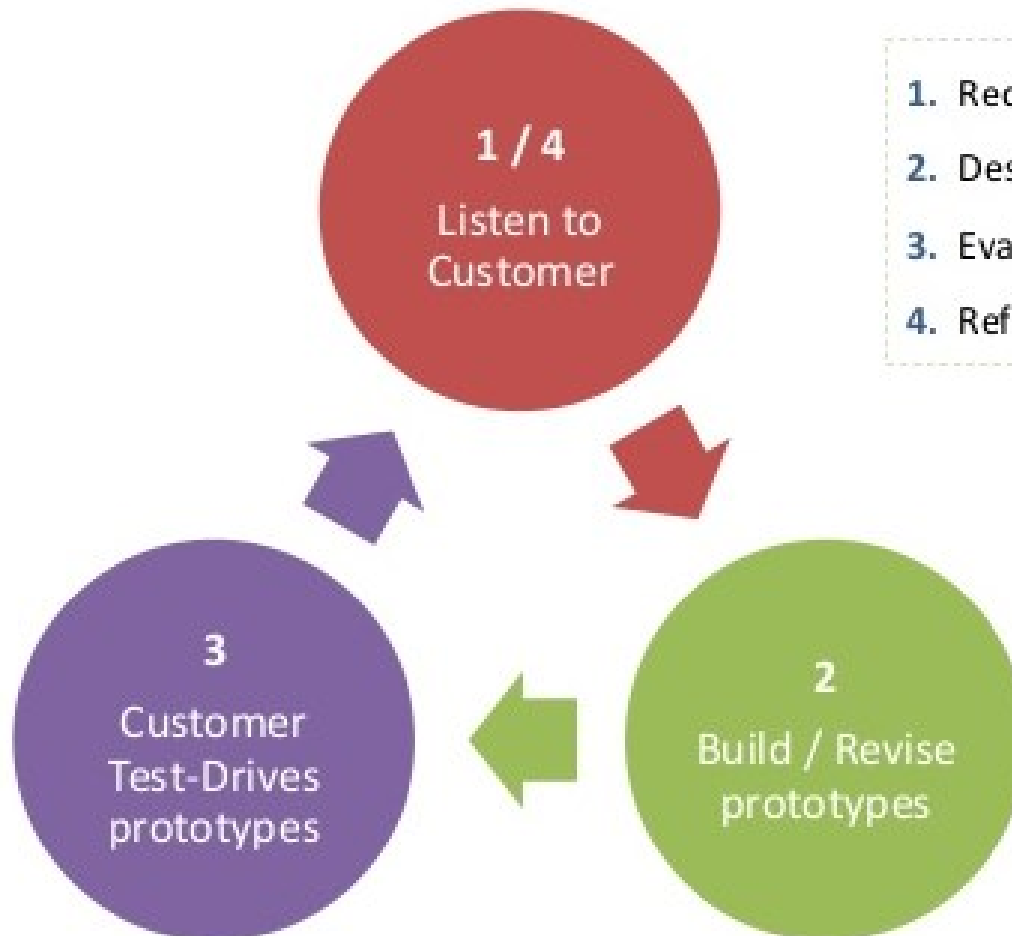


# Iterative Model

- ◆ Each iteration produces an executable

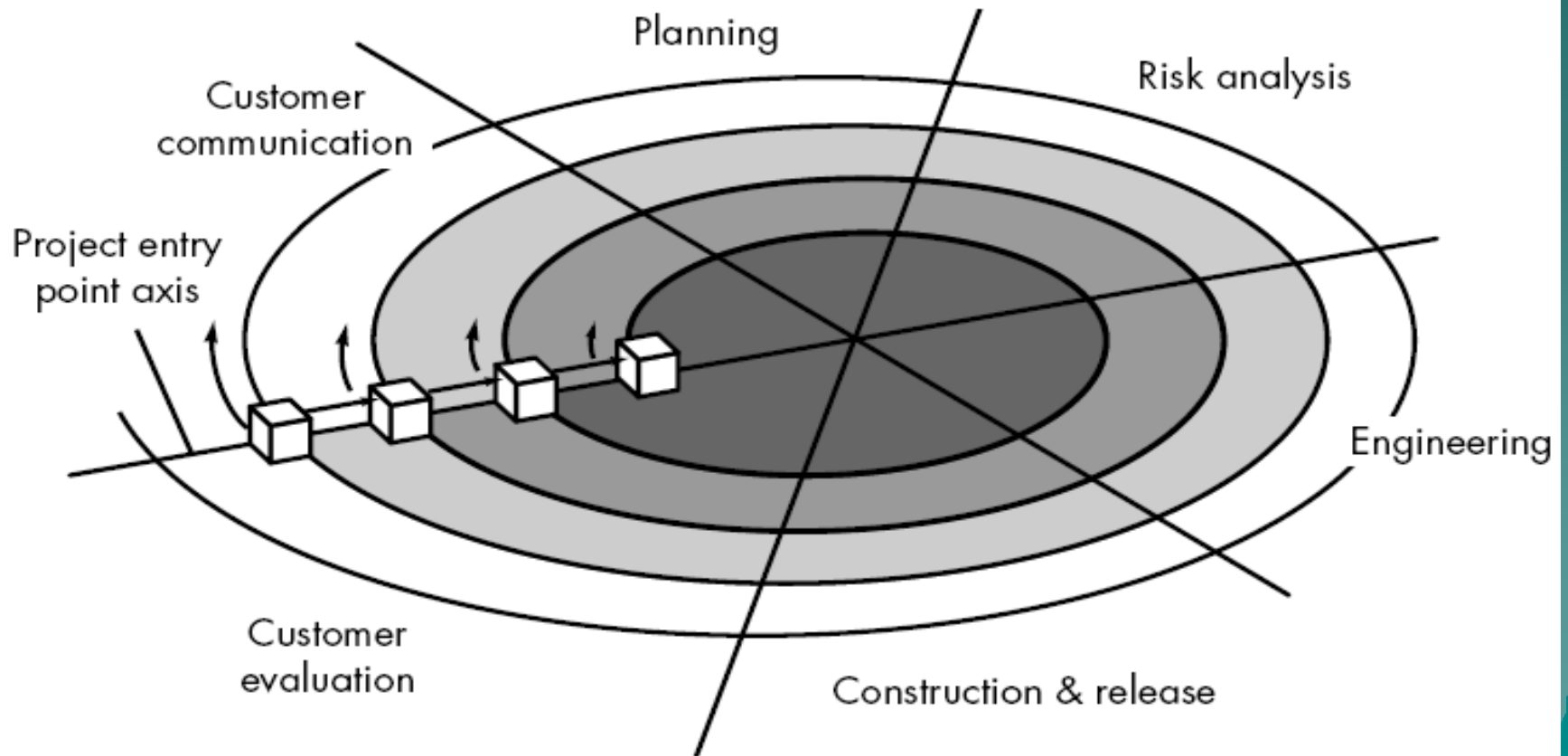


# Prototype model



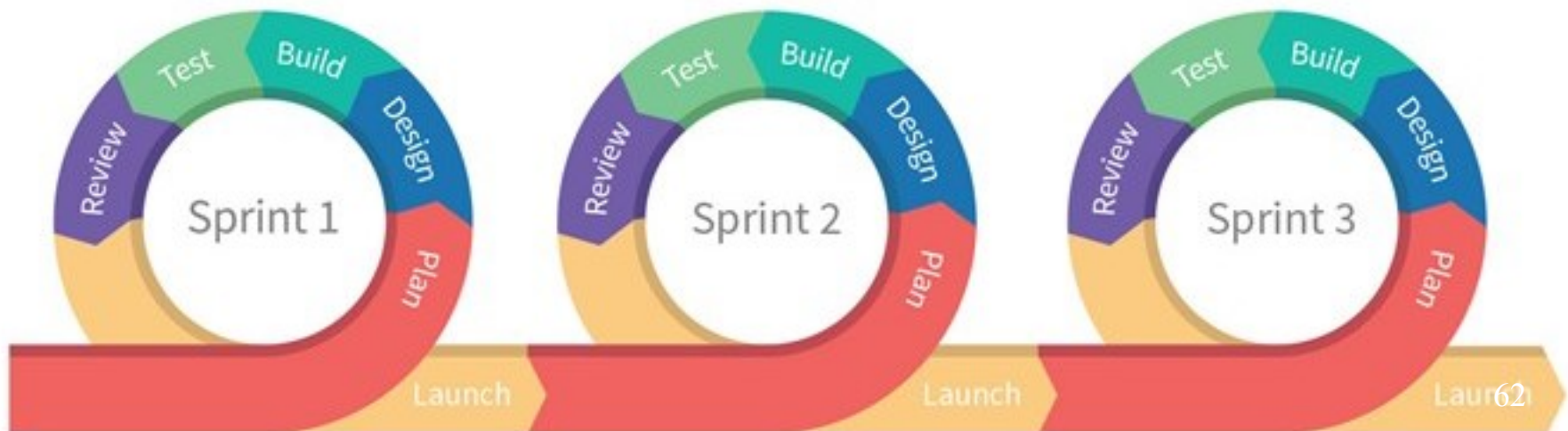
1. Requirements gathering.
2. Design and build SW prototype.
3. Evaluate prototype with customer.
4. Refine requirements.

# Spiral model



# Agile methodology

- “Agility is the ability to both create and respond to change in order to profit in a turbulent business environment.”
    - [Jim Highsmith, Agile Software Development Ecosystems, Preface XXIII]
- => Goal: Outline values and principles to allow software teams to:
- **develop quickly** and
  - **respond to change.**



# The Agile manifesto

- ◆ Individuals and interactions over processes and tools
- ◆ Working software over comprehensive documentation
- ◆ Customer collaboration over contract negotiation
- ◆ Responding to change over following a plan

# The Agile principles (1)

- ◆ 1. Our highest priority is to **satisfy** the **customer** through early and continuous delivery of valuable software.
- ◆ 2. Welcome **changing requirements**, even late in the development. Agile processes harness change for the customer's competitive advantage.
- ◆ 3. Deliver **working software** frequently, from a couple of weeks to a couple of months, with a preference to a shorter time scale.



# 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**.

# 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 indefinitely**.
- ◆ 10. Continuous attention to technical excellence and good design enhances agility.

# 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.

# References

- [1] ISO/IEC FDIS 12207, *Systems and software engineering — Software life cycle processes*.
- [2] ISO/IEC 9126-1:2001, *Software engineering – Product quality – Part 1: Quality model*.

# Software documents

Planning

Requirement Definition

Design

Implementation

Testing

