



# CSE291- Introduction to Software Engineering (FALL 2022)

---

Lecture 3

## **Software Process and Process Models**

# Outline

- Software Process
- Software Development Life Cycle
- Plan-driven and Agile Processes
- Software Process models
  - Waterfall Model
  - V Model

# The Software Process

A **structured set of activities** required to develop and maintain 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.

# Software Development Activities

## **Construction**

Some of the major construction activities are listed below.

- Requirement Gathering
- Design Development
- Coding
- Testing

## **Management**

Some of the major management activities are listed below.

- Project Planning and Management
- Configuration Management
- Software Quality Assurance
- Installation and Training

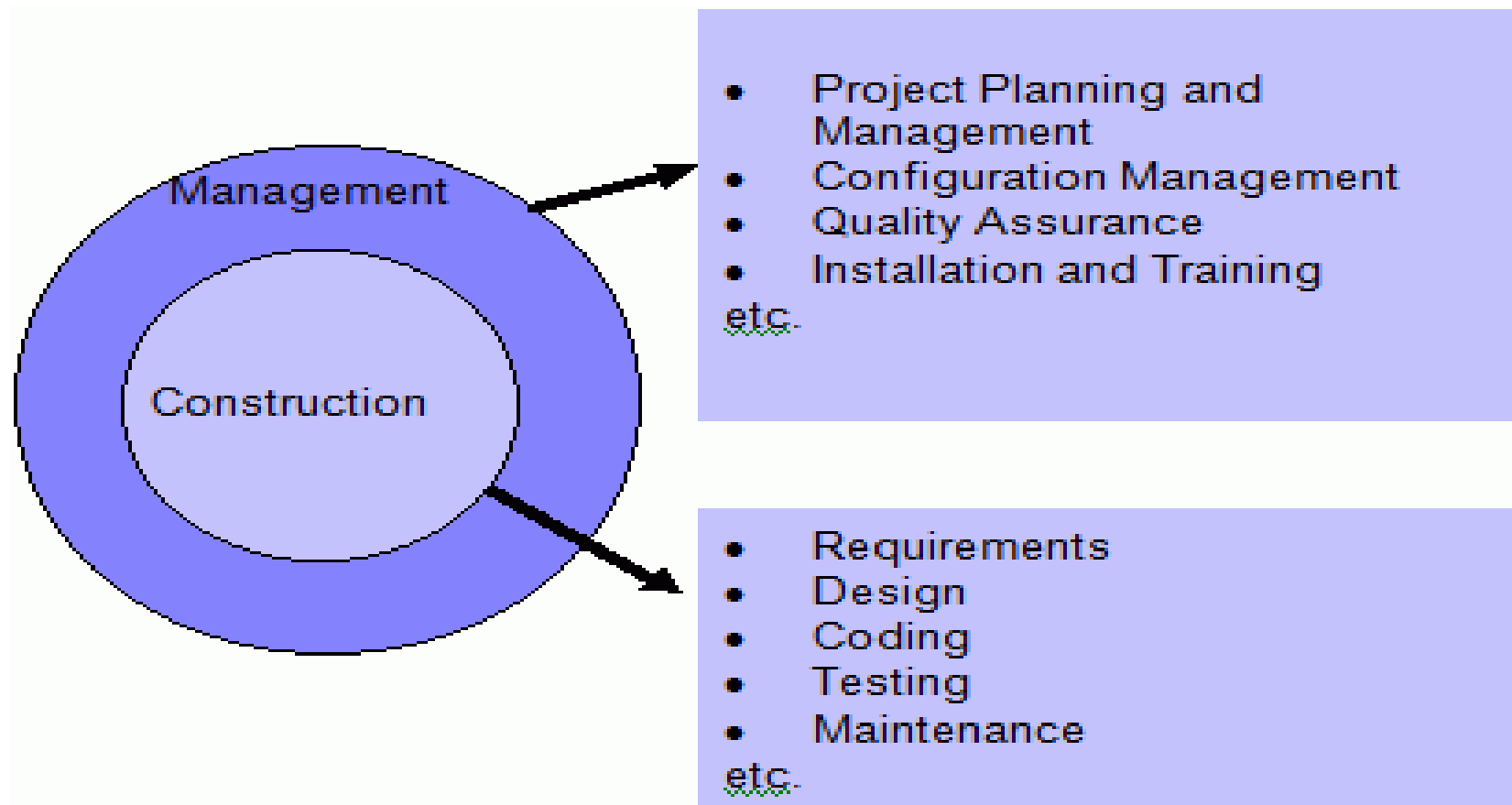


Figure 1: Development Activities

# SDLC(Software Development life cycle)

SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing

- How to develop
- Maintain

The life cycle also defines a methodology for improving the quality of software and the overall development process.

# SDLC-Stages

A typical Software Development life cycle consists of the following stages:

1. Planning and Requirement Analysis
2. Defining Requirements
3. Designing the product architecture
4. Building or Developing the Product
5. Testing the Product
6. Deployment in the Market and Maintenance

# SDLC-Stage 1

## ***Planning and Requirement Analysis***

- The most important and fundamental stage in SDLC.
- It is performed by the senior members of the team with inputs from the customer, the sales department, market surveys and domain experts in the industry.
- This information is then used to plan the basic project approach.
- Identification of the risks associated with the project is also done in the planning stage.
  - Define the various technical approaches that can be followed to implement the project successfully with minimum risks.



# SDLC-Stage 2

## ***Defining Requirements***

- Once the requirement analysis is done the next step is to clearly define and document the product requirements.
- Get them approved from the customer or the market analysts.
- This is done through 'SRS' – Software Requirement Specification document which consists of all the product requirements to be designed and developed during the project life cycle.

# SDLC-Stage 3

## ***Designing the product architecture :***

- Based on the requirements specified in SRS, usually more than one design approach for the product architecture is proposed and documented in a DDS - Design Document Specification.
- This DDS is reviewed by all the important stakeholders.
- The best design approach is selected for the product.
- A design approach clearly defines all the architectural modules of the product along with its communication and data flow representation with the external and third party modules (if any).

# SDLC-Stage 4

## ***Building or Developing the Product***

- The actual development starts and the product is built.
- The programming code is generated as per DDS during this stage.
- Developers have to follow the coding guidelines defined by their organization and programming tools .
- The programming language is chosen with respect to the type of software being developed.

# SDLC-Stage 5

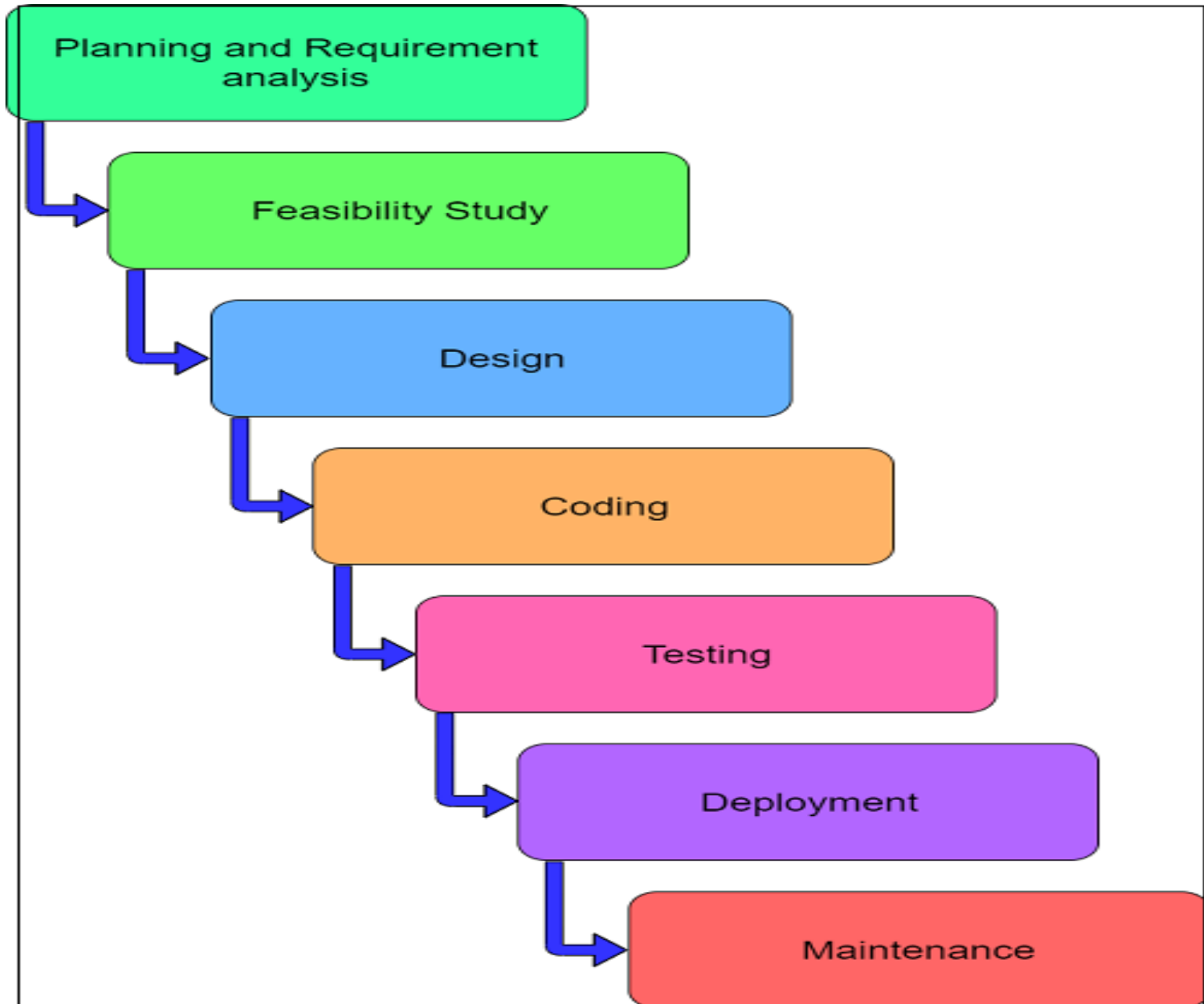
## ***Testing the Product***

- This stage is usually a subset of all the stages as in the modern SDLC models.
- The testing activities are mostly involved in all the stages of SDLC.
- Where products defects are reported, tracked, fixed and retested, until the product reaches the quality standards defined in the SRS.

# SDLC-Stage 6

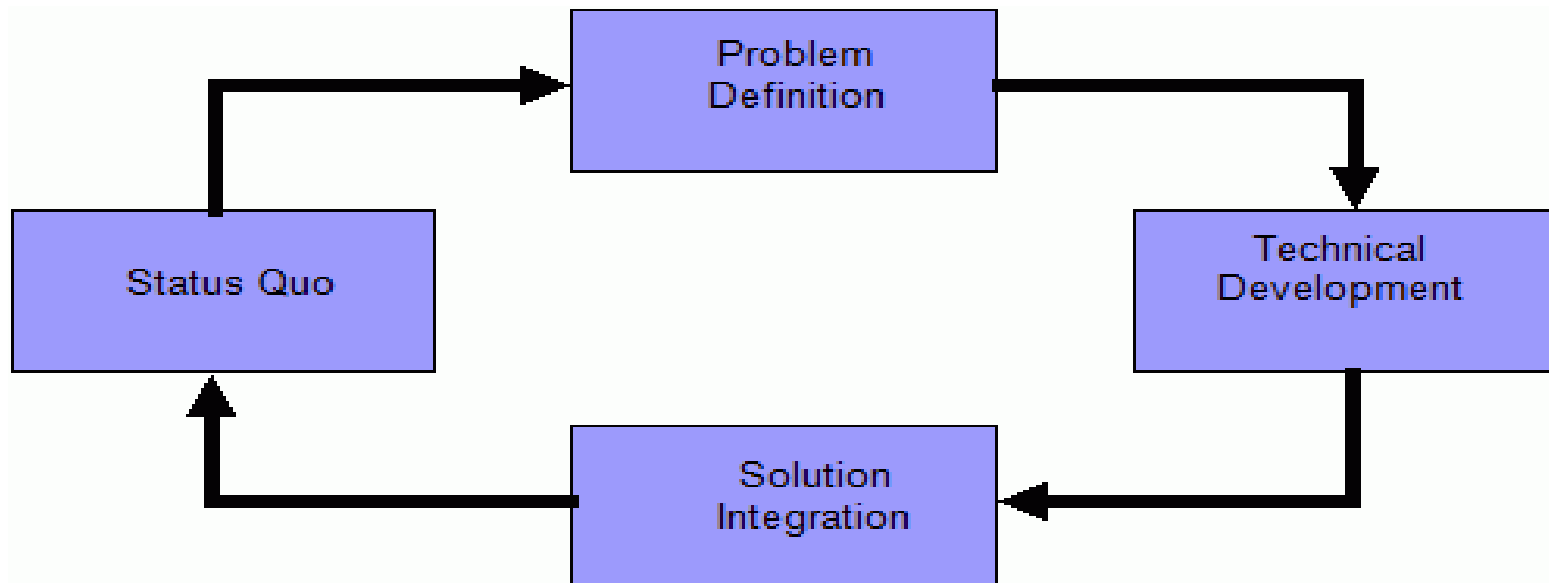
## ***Deployment in the Market and Maintenance***

- Once the product is tested and ready to be deployed it is released formally in the appropriate market.
- Sometime product deployment happens in stages as per the organizations' business strategy.
- The product may first be released in a limited segment and tested in the real business environment (UAT- User acceptance testing).
- Then based on the feedback, the product may be released as it is or with suggested enhancements in the targeting market segment.



# Software Development Loop

- Software development activities could be performed in a cycle called **software development loop** which is shown in Figure below
- The software developed through this process has the property that this could be evolved and integrated easily with the existing systems.



# Software Development Loop

**Problem Definition:** In this stage, Determine what is the problem against which we are going to develop software. Here we try to completely comprehend the issues and requirements of the software system to build.

**Technical Development:** This is the stage where a new system is actually developed that solves the problem defined in the first stage.

**Solution Integration:** All existing system(s) integrate with our new system at this stage.



# Software Development Loop

**Status Quo:** After going through the previous three stages successfully, when we actually deployed the new system at the user site then that situation is called status quo. But once we get new requirements then we need to change the status quo.

- After getting new requirements we perform all the steps in the software development loop again.

# 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.

# Ad hoc Software Development

- Developing software without planning for each phase, and without specifying tasks, deliverables, or time constraints.
- Relies entirely on the skills and experience of the individuals performing the work.
- The software process may change as work progresses.

# Case Study : SaudiTech

## What could go wrong in a software development project?

SaudiTech is a software company. It has a team of **25 programmers**. Ali has recently established a dental clinic in Riyadh and asked SaudiTech to develop a management system for his dental clinic.

Ali's request was to develop a system to electronically manage patient records and perform administrative functions similar to 'what is being used in other clinics' but within a budget of **500 Saudi riyals** and delivered within **3 months**.

SaudiTech adopts an ad hoc approach (**Developing software without planning for each phase, and without specifying tasks, deliverables, or time constraints**) to software development and has not worked on large scale projects before.

## What problems do you anticipate in this project?

# How can we overcome problems with ad hoc software development?

## ➤ Problems:

- Difficult to distinguish between tasks → important tasks may be ignored.
- Inconsistent schedules, budgets, functionality and product quality.
- Delayed problem discovery → more costly to fix.

**Solution?      Software Process Model**

- Software Process Models provide guidelines to organize **how** software process activities should be performed and **in what order**.

# 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.
  - Plan drives everything!
- 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.
- **There are no right or wrong software processes.**

# Software Process/SDLC Models

- Following are the most important and popular SDLC models followed in the industry:
  - Waterfall model.
  - V Model
  - Evolutionary development.
  - Component-Based development model (CBSE).
  - Process iteration(incremental and spiral Models)

The other related methodologies are Agile and RAD Model – Rapid Application Development

# Waterfall Model

- First published model
- Because of the cascade from one phase to another, this model is known as 'Waterfall Model'.
- It is also referred to as linear-sequential life cycle model.
- Each phase must be completed before the next phase can begin and there is no overlapping in the phases.



# Waterfall Model

The sequential phases in Waterfall model are:

**Requirement Gathering and analysis:** All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.

- System's services
- Constraints
- Goals

**System Design:** The requirement specifications from first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture.

# Waterfall Model

**Implementation and unit testing:** With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.

**Integration and Testing:** All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

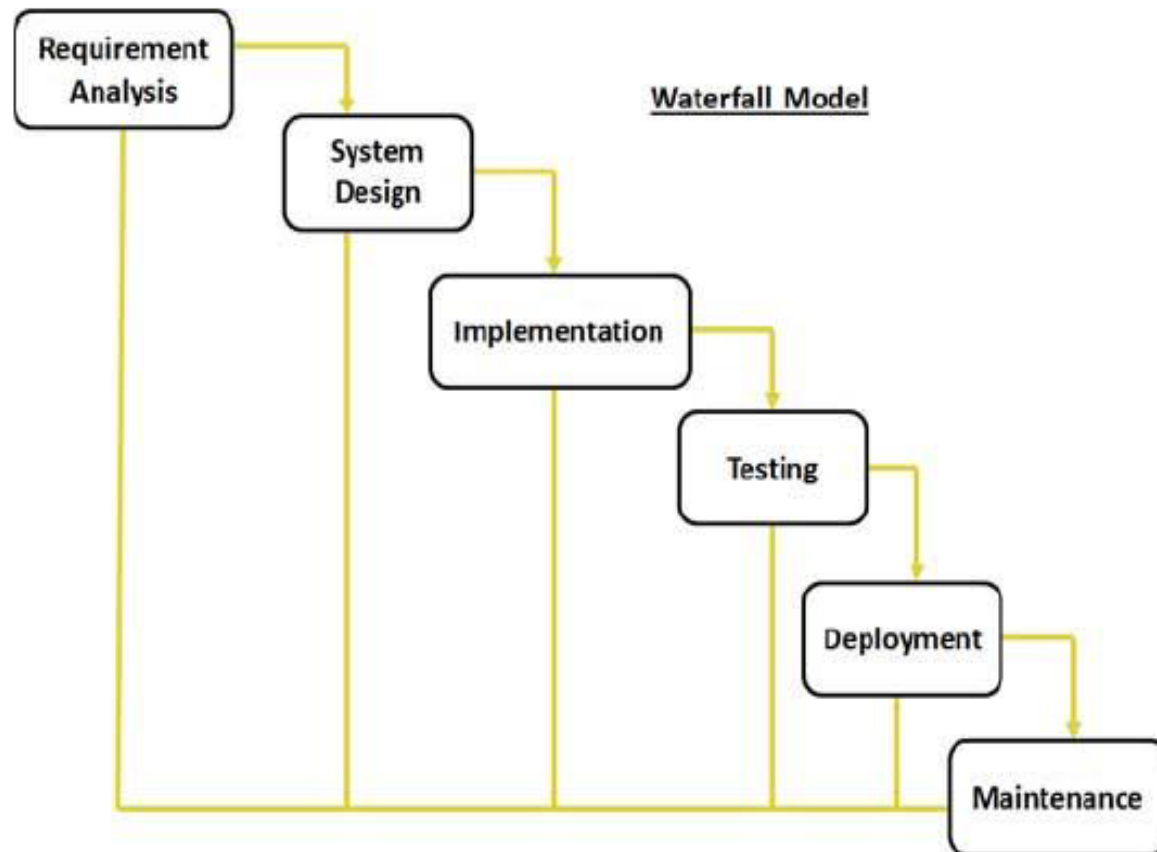
# Waterfall Model

**Deployment of system:** Once the functional and non functional testing is done, the product is deployed in the customer environment or released into the market.

**Maintenance:** There are some issues which come up in the client environment. To fix those issues patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

# Waterfall Model

Following is a diagrammatic representation of different phases of waterfall model.



# Waterfall Model

## **Applicability:**

- Requirements are very well understood, clear and fixed
- Product definition is stable
- Technology is understood

# Waterfall Model

## **Advantages:**

- Documentation - at each phase
- Phases are processed and completed one at a time.
- Easy to arrange tasks.

## **Problems:**

- Inflexible partitioning of the project into distinct stages makes it difficult to respond to changing customer requirements.

# V Model

- A variation of the waterfall model
- Development phases are presented in a V-shaped graph.
- It is also known as **Verification and Validation model**.
- For every **single** phase in the development cycle there is a directly associated testing phase.

# Verification and Validation

- ***Verification***

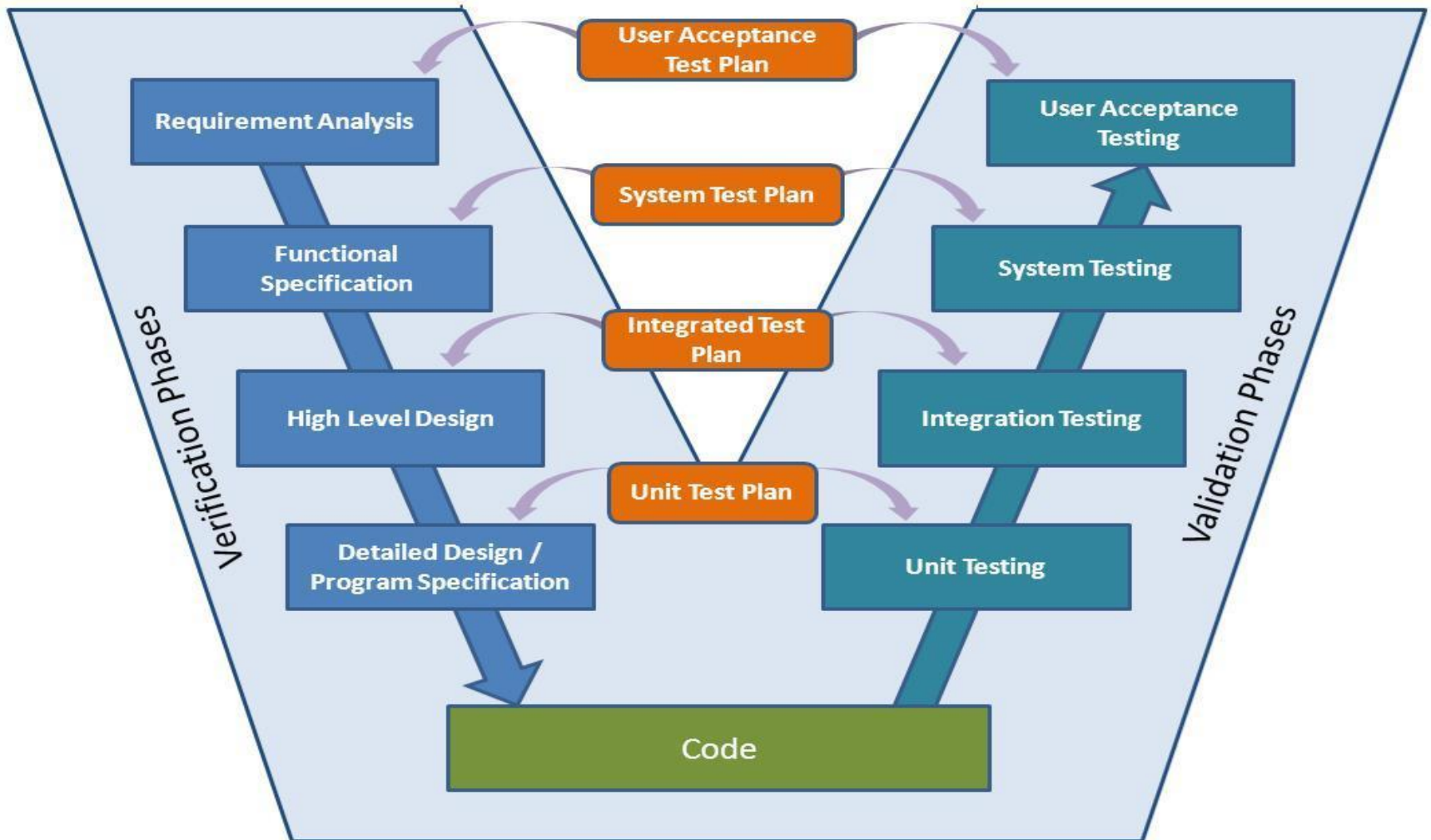
- Does the product meet system specifications?
- Have you built the product right?

- ***Validation***

- Does the product meet user expectations?
- Have you built the right product?



# V Model



# V Model Verification Phases

Following are the Verification phases in V-Model

## **Business Requirement Analysis:**

- This phase involves detailed communication with the customer to understand his expectations and exact requirement.
- The acceptance **test** design planning is done at this stage as business requirements can be used as an input for acceptance testing.

## **System Design:**

- System design would comprise of understanding and detailing the complete hardware and communication setup for the product under development.
- **System test plan** is developed based on the system design.

# V Model Verification Phases

## Architectural Design:

- System design is broken down further into modules taking up different functionality.
- The data transfer and communication between the internal modules and with the outside world (other systems) is clearly understood and defined in this stage.
- With this information, integration tests can be designed and documented during this stage.

## Module Design:

- In this phase the detailed **internal design** for all the system modules is specified
- It is important that the design is compatible with the other modules in the system architecture and the other external systems.
- Unit tests can be designed at this stage

# V Model

## Coding Phase

- The actual coding of the system modules designed in the design phase is taken up in the Coding phase.
- The best suitable **programming** language is decided based on the system and architectural requirements.
- The coding is performed based on the coding guidelines and standards.
- The code goes through numerous code reviews and is optimized for best performance before the final build is checked into the repository.

# V Model Validation Phases

Following are the Validation phases in V-Model:

## **Unit Testing:**

- Unit tests designed in the module design phase
- Unit testing is the testing at code level and helps to eliminate bugs at an early stage

## **Integration Testing:**

- Integration testing is associated with the architectural design phase.
- Integration tests are performed to test the coexistence and communication of the internal modules within the system.

# V Model Validation Phases

## **System Testing:**

- System tests check the entire system functionality and the communication of the system under development with external systems.
- Most of the software and hardware compatibility issues can be uncovered during system test execution.

## **Acceptance Testing:**

- Acceptance testing is associated with the business requirement analysis phase and involves testing the product in user environment.
- It also discovers the non functional issues such as load and performance defects in the actual user environment.

# V Model

## Advantages:

- Testing activities like planning, test designing, happens well before coding.
- Higher chances of success over the waterfall model.
- Works well for projects where requirements are easily understood.

## Problems:

- Difficult to handle changes in requirements throughout the life cycle

# V Model

## Applicability:

- The V-shaped model should be used for projects where requirements are clearly defined and fixed.
- The V-Shaped model should be chosen when technical resources are available with needed technical expertise.
- High confidence of customer is required for choosing the V-Shaped model approach.



# Conclusion

- What is a process model?
- Importance of Process Model
- Waterfall Model
- V Model

# Chapter Reading

- ***Chapter 2: Software Processes***

Software Engineering by Ian Sommerville