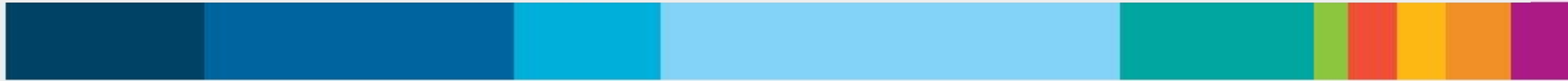


Introduction to Software / System Development Life Cycle (SDLC)



Module Overview

This module will introduce you to the key concepts of Software / System development life cycle (SDLC) and should take about 2 hours to complete.

When you complete this module, you should be able to:

- Define software engineering
- Explain the software engineering process
- List the attributes of good software
- Identify the roles and responsibilities of software team personnel
- Describe the characteristics following models with their advantages and disadvantages:
 - Waterfall model
 - Incremental model
 - Spiral model
 - V model
- List some other commonly used software development models

01 Software Engineering

Software engineering and its attributes

Software Engineering (SE) is the science and art of building significant software systems that are developed:

- On time
- Within budget
- With acceptable performance
- With correct operation

Attributes of software engineering

- Software engineering (SE) is an intellectual activity and thus human-intensive.
- Software is built to meet a certain functional goal and satisfy certain qualities.
- Software processes also must meet certain qualities.

The software process

A software process is a structured set of activities required to develop a software system.

The activities vary depending on the organization and the type of system being developed. The software process must be explicitly modeled if it is to be managed well.

These activities are specification, design and implementation, validation, and evolution.

Software specification

The functionality of the software and constraints on its operation must be defined.

Software design and implementation

The software to meet the specification must be produced.

Software validation

The software must be validated to ensure that it does what the customer wants.

Software evolution

The software must evolve to meet changing customer needs.

Need for software engineering: An analogy

1



Sally wants to construct a house of her own.

2



She contacts Newcity Builders to see if they can build the house for her.

3



Mike, the architect assigned to the project, works with Sally to understand her requirements for the house and builds a plan.

4



As the house is constructed, Mike evaluates it against the requirements.

5



Six months later Sally moves into her new house.

6



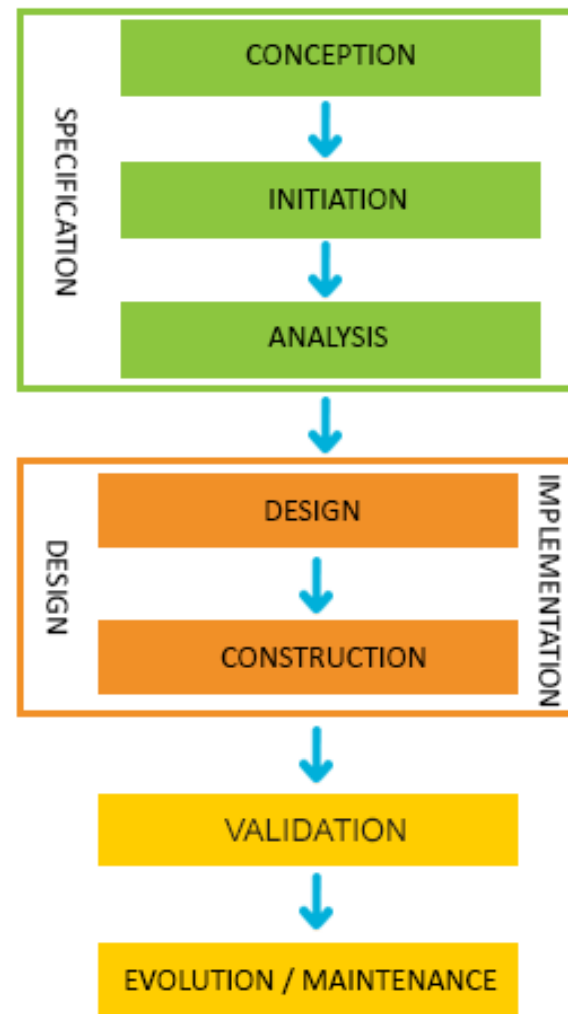
House maintenance activities down the line

Need for software engineering: An analogy (continued)

Notice that this house building project went through five stages:

- Conception: Sally conceptualizes her problem as the need to have a house of her own.
- Initiation: Riya brings in the architect to figure out a feasible solution.
- Analysis: The architect analyses the needs of Sally's family to determine their requirement
- Design: The architect draws a plan.
- Construction: The house is built.

Development of software or systems goes through these same stages.



Essential attributes of good software

Product Characteristic	Description
Maintainability	<ul style="list-style-type: none">▪ Software should be written in such a way that it can evolve to meet the changing needs of customers.▪ This attribute is critical because software change is an inevitable requirement of a changing business environment.
Dependability and Security	<ul style="list-style-type: none">▪ Software dependability includes a range of characteristics such as reliability, security, and safety.▪ Dependable software should not cause physical or economic damage in the event of system failure.▪ Malicious users should not be able to access or damage the system.
Efficiency	<ul style="list-style-type: none">▪ Software should not make wasteful use of system resources such as memory and processor cycles.▪ Efficiency therefore includes responsiveness, processing time, and memory utilisation.
Acceptability	<ul style="list-style-type: none">▪ Software must be acceptable to the users for whom it is designed.▪ It must be understandable, usable, and compatible with other systems that the users use.

Spot Quiz

02

What are the different activities of a software process?

Select the correct answers, then click **Next**.

A

Specification

B

Design & Implementation

C

Validation

D

Evolution

Spot Quiz

03

What are the different attributes of a good software?

Select the correct answers, then click **Next**.

A

Maintainability

B

Security

C

Efficiency

D

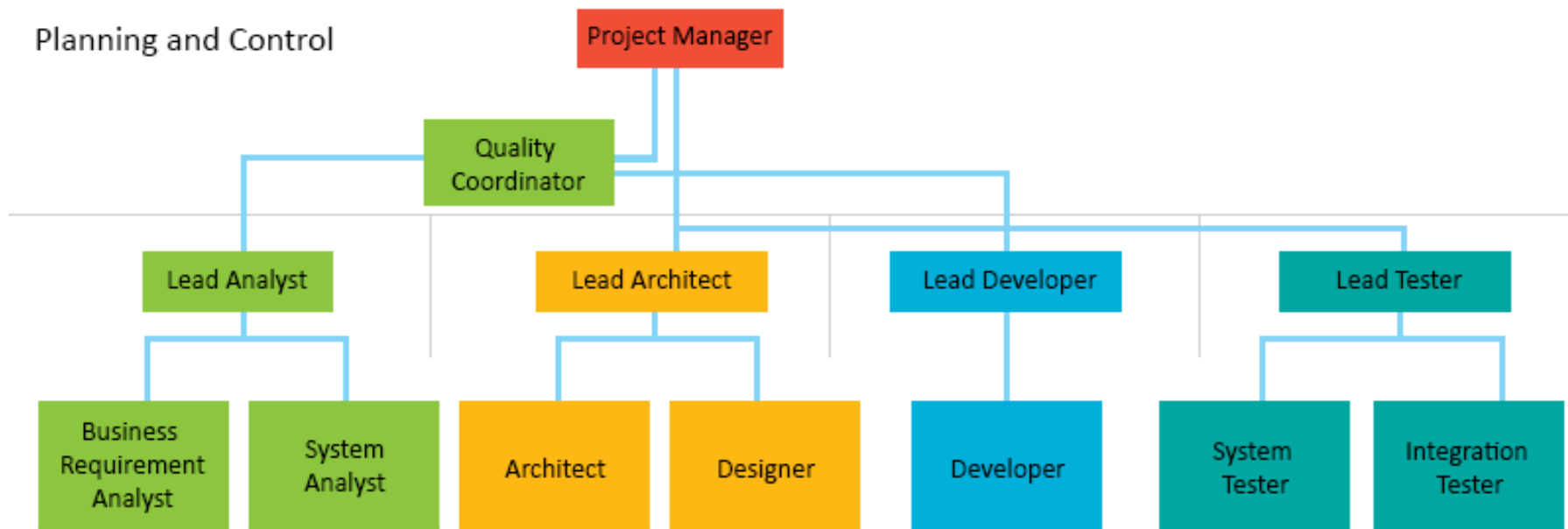
Acceptability

02 Roles and Responsibilities

Roles associated with different phases of SDLC

There are different roles and responsibilities associated with the different phases of an SDLC. The activities performed by these roles may vary from project to project but the generic ones are Project Manager, Quality Reviewer, Lead Analyst, Lead Architect, Lead Developer, and Lead Tester.

Planning and Control



Responsibilities of all roles

Let us discuss each of these roles in detail.

Click each role to learn about the associated responsibilities.

Project Manager

Quality Coordinator

Lead Analyst

Lead Architect

Project managers :

- Work with
 - Project sponsors
 - Project teams
 - Others involved in the project to meet project goals
- Every project is constrained in different ways by its:
 - Scope goals: What work will be done?
 - Time goals: How long should it take to complete?
 - Cost goals: What should it cost?

Responsibilities of all roles

Let us discuss each of these roles in detail.

Click each role to learn about the associated responsibilities.

Project Manager

Quality Coordinator

Lead Analyst

Lead Architect

Quality Coordinator:

- Conducts internal deliverable quality reviews
- Reviews and approves all deliverables

Responsibilities of all roles

Let us discuss each of these roles in detail.

Click each role to learn about the associated responsibilities.

Project Manager

Quality Coordinator

Lead Analyst

Lead Architect

Lead Analyst:

- Ensures the preparation of the Detailed Design deliverable
- Assists logical data model
- Prepares physical data model
- Coordinates system testing

Responsibilities of all roles

Let us discuss each of these roles in detail.

Click each role to learn about the associated responsibilities.

Project Manager

Quality Coordinator

Lead Analyst

Lead Architect

Lead Architect:

- Leads the Design Team
- Is responsible for all the activities in the Design Phase
- Leads the application architect and a application designer
 - The application architect is an information technology professional who designs, develops, and implements applications for computer systems
 - The application designer gathers requirements from business systems analysts and end users

Spot Quiz

04

Which of these is not the responsibility of a Project Manager?

Select the correct answer, then click **Next**.

A

Status Tracking

B

Prepare Project Plan

C

Review and Approve Deliverables

D

Manage Risks and Issues

03 Overview of the SDLC Model

A software development model / process: A framework of activities

A model is a framework which describes the activities performed in every stage of a software development project.

There are many frameworks defined and the most commonly used ones are shown below:

A purple rounded square containing the text "Waterfall Model" in white.

**Waterfall
Model**

An orange rounded square containing the text "Incremental Model" in white.

**Incremental
Model**

A green rounded square containing the text "Spiral Model" in white.

Spiral Model

A blue rounded square containing the text "V-Model" in white.

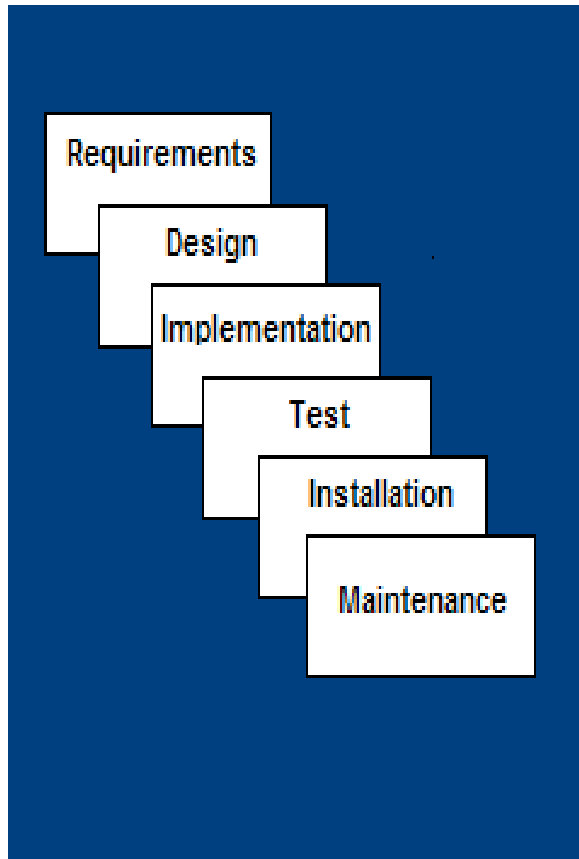
V-Model

We will learn about each framework and understand how a quality product can be developed using these models.

04 Waterfall Model

What is the Waterfall Model?

The Waterfall model is a sequential software development model in which progress is seen as flowing steadily downwards (like a waterfall) through the phases as shown below.



Requirements: Defines the required information, function, behavior, performance, and interfaces

Design: Includes data structures, software architecture, interface representations, and algorithmic details

Implementation: Includes source code, database, user documentation, and testing

Testing: Checks if all code modules work together and if the system as a whole behaves as per the specifications

When to use the Waterfall Model

The Waterfall Model must be used when:

- The requirements are known well
- It is possible to produce a stable design

Examples:

- A new version of an existing product
- Porting an existing product to a new platform

Click each tab to learn more.

Advantages

- Easy to understand, easy to use
- Provides structure to inexperienced staff
- Milestones are well understood
- Sets requirements stability
- Good for management control (plan, staff, track)

Disadvantages

When to use the Waterfall Model

The Waterfall Model must be used when:

- The requirements are known well
- It is possible to produce a stable design

Examples:

- A new version of an existing product
- Porting an existing product to a new platform

Click each tab to learn more.

Advantages

Disadvantages

- All requirements must be known upfront
- Deliverables created for each phase are considered frozen, hence inhibits flexibility
- Integration is one big bang at the end
- Little opportunity for customer to preview the system until it may be too late

Waterfall Model case study: ABC Company

ABC is a Mobile Service Provider. They do not have their own website and hence go in search of companies who can create a required application to include their website, online bill payment and online chat.

In the current requirement, ABC needs an online application where they want their customers to:

1. Pay their bills online using various modes
2. View their call summary
3. Perform account management

Let us discuss the different phases adopted in this case.

Waterfall Model case study: ABC Company (continued)

The following phases formed part of this case.

Click each button to learn more.

Requirement

ABC Business Analyst (BA), in a meeting, discusses and gets clarity on the requirements. The BA documents these requirements in the Business Requirement Document (BRD). The BRD contains information such as:

Design

- The browsers that the application should be supported in
- The modes of payment

Implementation and Testing

- The kind of user id and password required for customer sign-up (Example: User id as email id or mobile number; password as alphanumeric or numeric).

Installation

Maintenance

Waterfall Model case study: ABC Company (continued)

The following phases formed part of this case.

Click each button to learn more.

Requirement

In this phase the gathered requirements are converted into the High Level Design (HLD) and the Low Level Design (LLD). Design specifications, technical specifications, and flowcharts will be designed in this phase.

Design

- HLD has the information about the structure of login application and screen navigation process. It contains information such as the flow-graph information and how login will to appear.
- LLD includes further details specifications, of the points in the HLD, such as:
 - User Name should be the 10 digit customer mobile number.
 - Password should be alphanumeric, contain at least one upper case and one special character, possess a maximum strength value of 10 and a minimum strength value of 6.

**Implementation
and Testing**

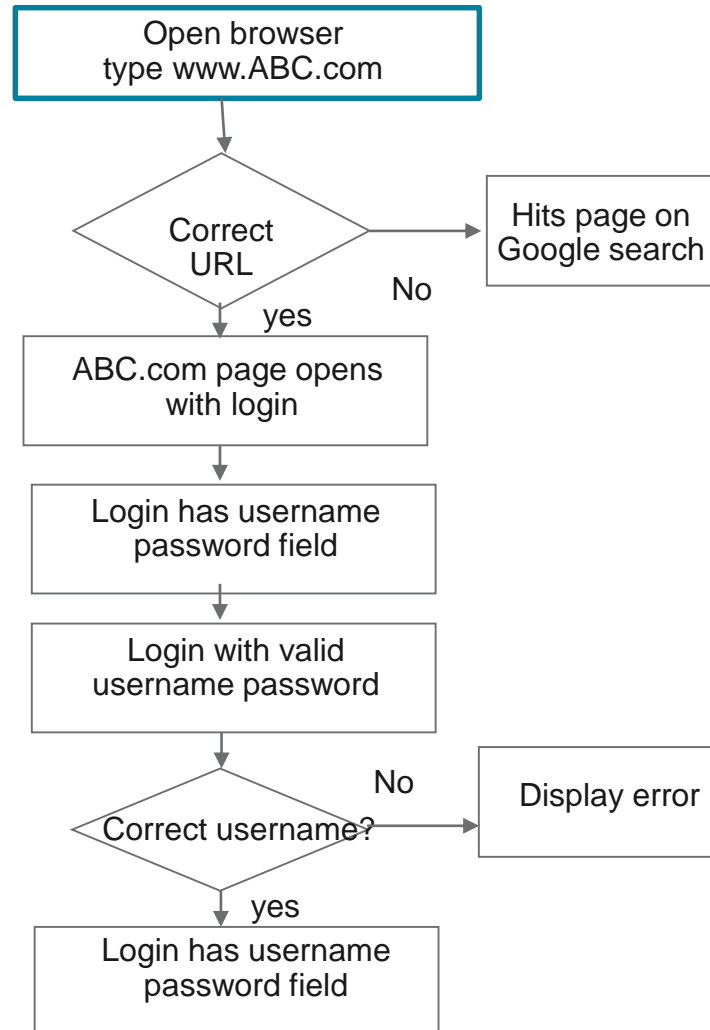
Installation

Maintenance

Click [here](#) to view the flow chart.

Waterfall Model case study: ABC Company (continued)

Flow chart for Design tab:



Waterfall Model case study: ABC Company (continued)

The following phases formed part of this case.

Click each button to learn more.

Requirement

Based on the design and language decided, all the functionality is coded and tested.

Design

**Implementation
and Testing**

Installation

Maintenance

Waterfall Model case study: ABC Company (continued)

The following phases formed part of this case.

Click each button to learn more.

Requirement

Once implemented the application is installed on the system for ABC to browse through. Documents are also prepared to guide the user.

Design

**Implementation
and Testing**

Installation

Maintenance

Waterfall Model case study: ABC Company (continued)

The following phases formed part of this case.

Click each button to learn more.

Requirement

Once the application is developed and tested, it is given to ABC. If ABC has any issues or in case of any doubts that the company may have in the future, the service provider maintains the application and resolves the issues. This helps gain customer good will.

Design

All the issues faced in this project are summarized and documented for future reference.

Implementation and Testing

Installation

Maintenance

05 Incremental Model

What is the Incremental model?

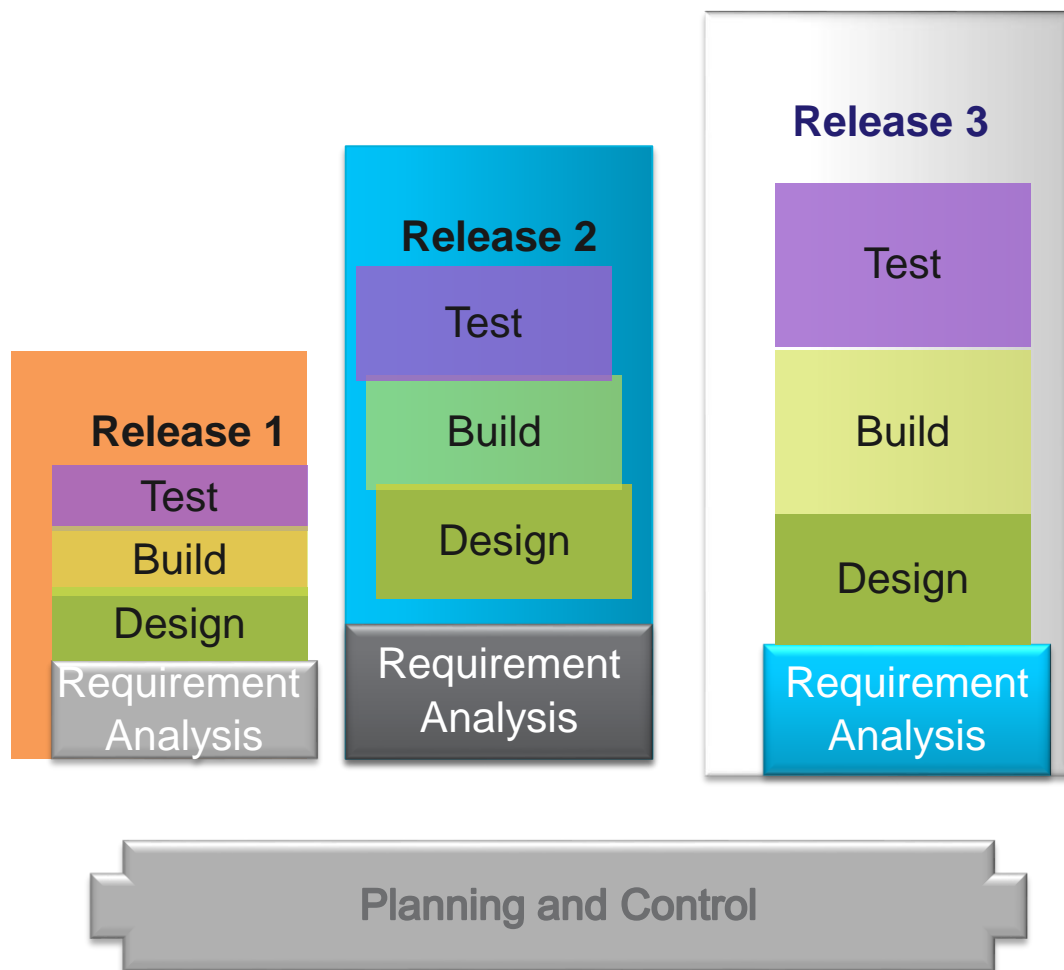
The Incremental Model is an evolution of the Waterfall Model. Here, the Waterfall Model is incrementally applied.

In the Incremental Model, there will be a series of releases, referred to as “increments”, with each increment providing more functionality to the customers.

1. After the first increment, a core product is delivered. This product can be used by the customer.
2. Based on customer feedback, a plan is developed for the next increments, and modifications are made accordingly.
3. This process continues, with increments being delivered until the complete product is delivered.

Here, the end of each cycle results in an operational product. This increases the cost of the product. Complexity increases and makes it difficult to find the origin of a defect. As a result, there are many disadvantages.

Each release delivers an operational product



We have seen that in the Incremental model, the product is designed, implemented, integrated, and tested as a series of incremental builds.

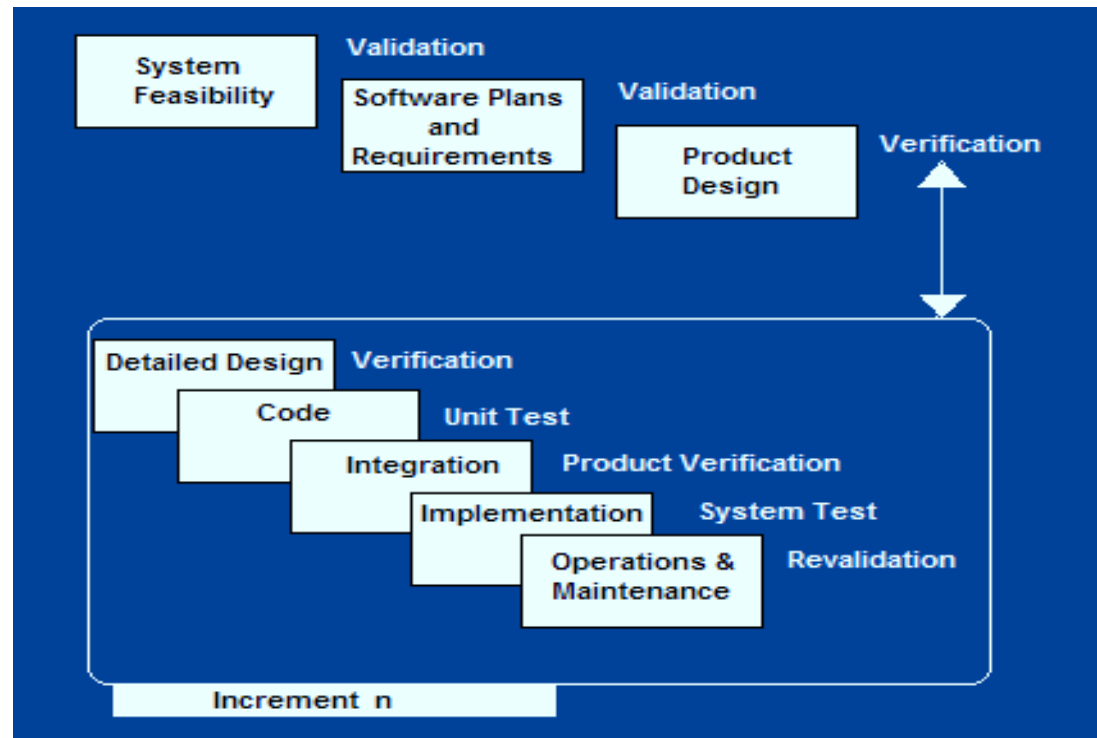
For this purpose, the software requirements are initially broken down into various serviceable modules.

Features are then implemented in an increment and the last product is achieved after all the useful units are implemented in the development process.

The LLD, Implementation, Build, and Testing are incremented

Steps in Incremental Model:

1. Construct a partial implementation of a total system.
2. Then slowly add increased functionality.
3. The Incremental Model prioritizes requirements of the system and then implements them in groups.
4. Each subsequent release of the system adds function to the previous release, until all the designed functionality has been implemented.



When to use the Incremental Model?

The Incremental Model must be used when:

- There are some high risk features and goals.
- There is early realization of benefits.
- Most of the requirements are known up-front but are expected to evolve over time.
- There is a need to get basic functionality to the market early.
- There is a project with new technology.

Click each tab to learn more.

Advantages

- Developing high-risk or major functions first solves the issue of what to develop first
- Each release delivers an operational product so customers can start using product samples faster
- Customer can respond to each build and provide feedback for each build
- Initial product delivery is faster and this reduces the time constraint in product delivery

Disadvantages

When to use the Incremental Model?

The Incremental Model must be used when:

- There are some high risk features and goals.
- There is early realization of benefits.
- Most of the requirements are known up-front but are expected to evolve over time.
- There is a need to get basic functionality to the market early.
- There is a project with new technology.

Click each tab to learn more.

Advantages

- Requires early definition of a complete and fully functional system to allow for the definition of increments
- Needs good planning and design
- Total cost of complete system is high

Disadvantages

Incremental Model case study: ABC Company

ABC is a Mobile Service Provider. They do not have their own website and hence go in search of companies who can create a required application to include their website, online bill payment and online chat.

In the current requirement, ABC needs an online application where they want their customers to:

1. Pay their bills online using various modes
2. View their call summary
3. Perform account management

Let us discuss the different phases adopted in this case.

Incremental Model case study: ABC Company (continued)

In the Incremental Model, the approach of creating the application for ABC Company is in phases.

Phase 1: Static application show items to the end user.

Phase 2: Login functionality is applied to the static application.

Phase 3: Users can buy items online.

Phase 4: To resolve any issues, users can chat with the representative.

Each phase in the Incremental Model is a complete Waterfall Model with requirement, design, implement, testing, and integration phases.

Spot Quiz

05

We can use the Waterfall Model when_____ .

Select the correct answer, then click **Next**.

A

There are high risk features

B

There is early realization of benefit

C

Requirements are stable

D

None of the above

Spot Quiz

06

The Incremental Model is the Waterfall Model applied for series of releases

Select the correct answer, then click **Next**.

A

True

B

False

06 V Model

What is the V model?

The Verification and Validation model commonly known as the V model is considered to be an extension of the Waterfall model.

Instead of moving down in a linear way, the process steps are bent upwards after the coding phase, to form the typical shape of the letter V. The V model demonstrates the relationship between each phase of the development life cycle and its associated phase of testing. The V model also provides guidance for the planning and realization of projects.

The following objectives are intended to be achieved by a project execution:

Click each objective to learn more.

- **Minimization of Project Risks**
- **Improvement and Guarantee of Quality**
- **Reduction of Total Cost over the Entire Project and System Life Cycle**
- **Improvement of Communication between all Stakeholders**

Minimization of Project Risks:

The V-Model improves project transparency and project control by specifying standardized approaches and describing the corresponding results and responsible roles. It permits an early recognition of planning deviations and risks and improves process management, thus reducing the project risk.

What is the V model?

The Verification and Validation model commonly known as the V model is considered to be an extension of the Waterfall model.

Instead of moving down in a linear way, the process steps are bent upwards after the coding phase, to form the typical shape of the letter V. The V model demonstrates the relationship between each phase of the development life cycle and its associated phase of testing. The V model also provides guidance for the planning and realization of projects.

The following objectives are intended to be achieved by a project execution:

Click each objective to learn more.

- **Minimization of Project Risks**
- **Improvement and Guarantee of Quality**
- **Reduction of Total Cost over the Entire Project and System Life Cycle**
- **Improvement of Communication between all Stakeholders**

Improvement and Guarantee of Quality:

As a standardized process model, the V-Model ensures that the results to be provided are complete and have the desired quality. Defined interim results can be checked at an early stage. Uniform product contents will improve readability, understandability, and verifiability.

What is the V model?

The Verification and Validation model commonly known as the V model is considered to be an extension of the Waterfall model.

Instead of moving down in a linear way, the process steps are bent upwards after the coding phase, to form the typical shape of the letter V. The V model demonstrates the relationship between each phase of the development life cycle and its associated phase of testing. The V model also provides guidance for the planning and realization of projects.

The following objectives are intended to be achieved by a project execution:

Click each objective to learn more.

- **Minimization of Project Risks**
- **Improvement and Guarantee of Quality**
- **Reduction of Total Cost over the Entire Project and System Life Cycle**
- **Improvement of Communication between all Stakeholders**

Reduction of Total Cost over the Entire Project and System Life Cycle:

The effort for the development, production, operation, and maintenance of a system can be calculated, estimated, and controlled in a transparent manner by applying a standardized process model. The results obtained are uniform and easily retraced. This reduces the acquirer's dependency on the supplier and the effort for subsequent activities and projects.

What is the V model?

The Verification and Validation model commonly known as the V model is considered to be an extension of the Waterfall model.

Instead of moving down in a linear way, the process steps are bent upwards after the coding phase, to form the typical shape of the letter V. The V model demonstrates the relationship between each phase of the development life cycle and its associated phase of testing. The V model also provides guidance for the planning and realization of projects.

The following objectives are intended to be achieved by a project execution:

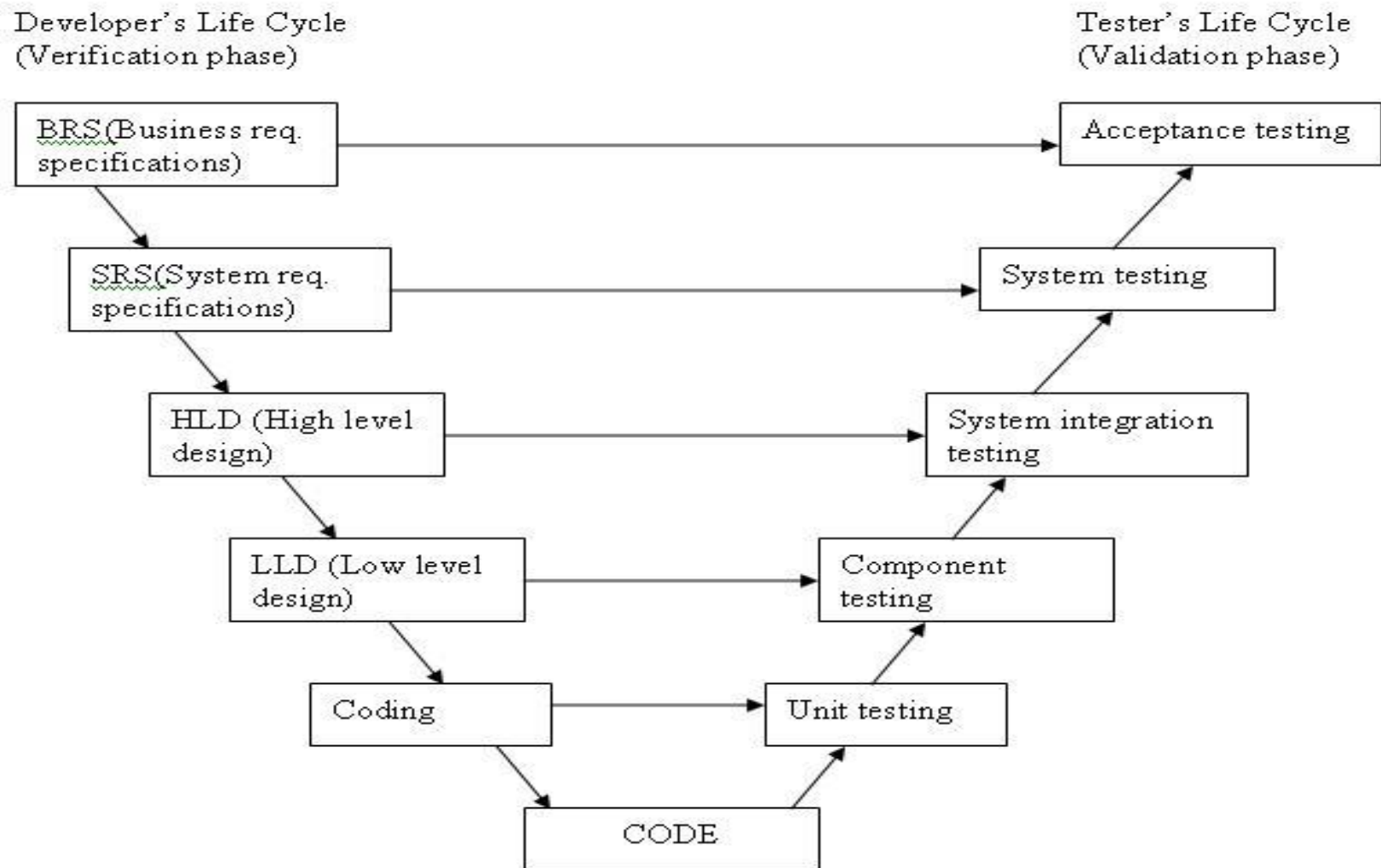
Click each objective to learn more.

- **Minimization of Project Risks**
- **Improvement and Guarantee of Quality**
- **Reduction of Total Cost over the Entire Project and System Life Cycle**
- **Improvement of Communication between all Stakeholders**

Improvement of Communication between all Stakeholders:

The standardized and uniform description of all relevant elements and terms is the basis for the mutual understanding between all stakeholders. Thus, the frictional loss between user, acquirer, supplier, and developer is reduced.

The V Model flowchart



Phase and activities of the V Model

Phases	Development Activities	Testing Activities
Planning and Control	Preparation such as Project Plan, Configuration Plan, and Test Plan	
Requirements	Define Business Scenarios	Define System Acceptance Criteria
	Prepare Functional Specs	Identify test scenarios
Design	High Level Design	Draft Test Cases for Functional and Integration Testing
	Low Level Design	Draft Test Cases for Unit Testing
Develop / Build	Coding	Execute Unit Testing
	Bug Fixing	
	Product Build	Testing Builds
Test		All Testing activities excluding Unit Testing
Deploy		Operation Readiness Testing
		Deploy Beta Version
		Deploy Complete Version

Verification methods used in the V Model

The following verification methods are used:

Stage Containment

- Performing verification and a validation at every stage, will improve the efficiency of the product.

Reviews

- Any thing that we can read can be reviewed. By executing this activity the Bugs can be reduced.

Entry and Exit

- Defining an entry and exit criteria for every software module will help the developers and tester to get better clarity on its implementation.

When to use the V Model

The V Model is an excellent choice for systems requiring high reliability such as hospital patient control applications. Use the V Model

- When all requirements are known up-front
- In systems that can be modified to handle changing requirements beyond the analysis phase
- When the solution and technology are known

Click each tab to learn more.

Advantages

- Emphasizes planning for verification and validation of the product in early stages of product development
- Each deliverable must be testable; there is more emphasis on the Testing phase
- Easy to use.

Disadvantages

When to use the V Model?

The V Model is an excellent choice for systems requiring high reliability such as hospital patient control applications. Use the V Model

- When all requirements are known up-front
- In systems that can be modified to handle changing requirements beyond the analysis phase
- When the solution and technology are known

Click each tab to learn more.

Advantages

Disadvantages

- Concurrent events cannot be easily handled.
- Dynamic changes in requirements cannot be easily handled.
- Risk analysis activities are not included.

Spot Quiz

07

What are the different verification methods followed in the V Model?

Select the correct answers, then click **Next**.

A

Stage Containment

B

Reviews

C

Entry and Exit Criteria

07 Spiral Model

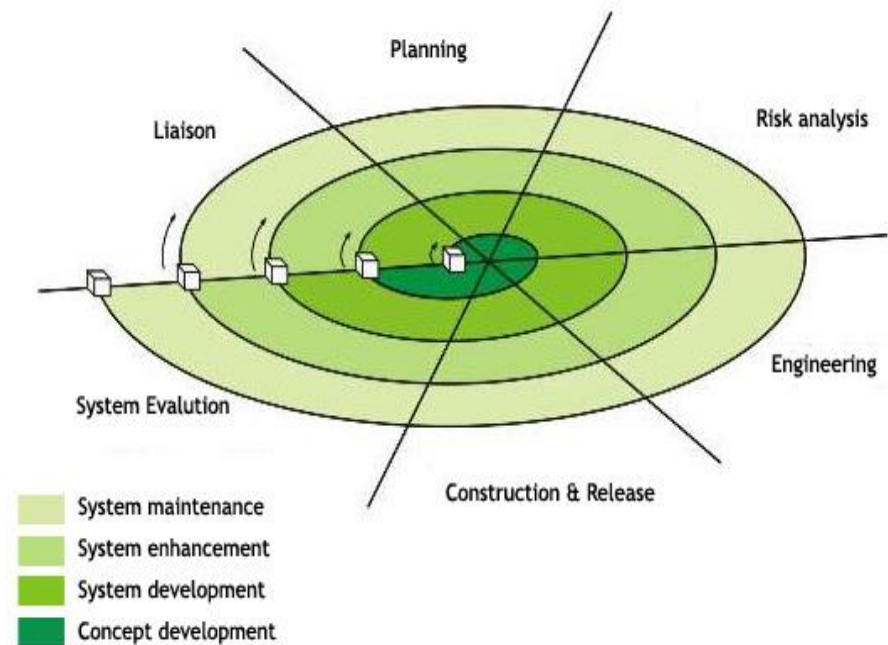
About the Spiral Model

The Spiral Model is a recent model that has been proposed by Boehm. As the name suggests, the activities in this model can be organized like a spiral. The spiral has many cycles.

The Spiral Model combines the idea of iterative development with systematic, controlled aspects of the Waterfall Model.

It allows for incremental releases of the product, or incremental refinement through each time around the spiral.

The Spiral Model also explicitly includes risk management within software development. Identifying major risks, both technical and managerial, and determining how to lessen the risk helps keep the software development process under control.



When to use the Spiral Model?

The Spiral Model must be used when:

- Medium to high-risk projects are being dealt with
- Costs and risk evaluation is important
- Requirements are complex
- Users are unsure of their needs

Click each tab to learn more.

Advantages

- The spiral life cycle model allows for elements of the product to be added in when they become available or known.
- This model provides early indication of insurmountable risks, without much cost.
- The Spiral Model forces early user involvement in the system development effort.
- This model resolves critical operational and / or technical issues early in the life cycle.

Disadvantages

When to use the Spiral Model?

The Spiral Model must be used when:

- Medium to high-risk projects are being dealt with
- Costs and risk evaluation is important
- Requirements are complex
- Users are unsure of their needs

Click each tab to learn more.

Advantages

Disadvantages

- The time spent for evaluating risks is too large for small or low-risk projects.
- The time spent in planning, resetting objectives, doing risk analysis and prototyping may be excessive.
- The model is complex.
- Risk assessment expertise is required.

08 Other Models

Other software development models

The following are some of the other models used for software development:

- Agile Software Development
- Rapid Application Development
- IBM Rational Unified Process

Spot Quiz

08

Which model can be used for implementation when requirements are known upfront?

Select the correct answer, then click **Next**.

A

V-Model

B

Waterfall Model

C

Either A or B

D

Neither A nor B

Reference links

If you would like to learn more about SDLC visit the links below:

- http://www.tutorialspoint.com/sdlc/sdlc_quick_guide.htm
- http://en.wikipedia.org/wiki/Software_development_process

Module summary

Congratulations! Now that you have successfully completed this module, you should be able to:

- Define software engineering.
- Explain the software engineering process.
- List the attributes of good software.
- Identify the roles and responsibilities of software team personnel.
- Explain the following models with their advantages and disadvantages:
 - Waterfall Model
 - Incremental Model
 - Spiral Model
 - V-Model
- List other models used for software development.