AGILE-PARADIGM SHIFT IN SDLC

Session - 3

Linear Process Models

Objectives

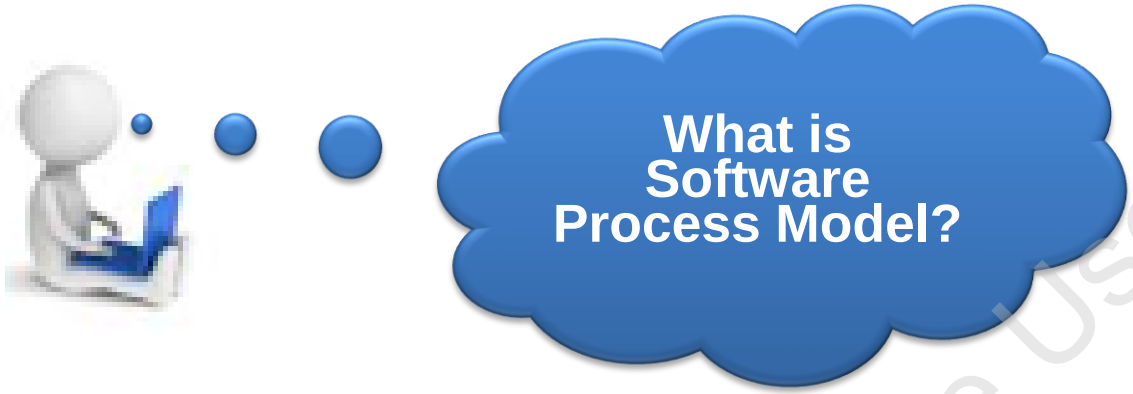
- Describe software process models
- List and describe linear process models
- Explain waterfall model
- Describe the phases in the waterfall model
- List the advantages and disadvantages of the waterfall model
- Describe the phases in the V-model
- List the advantages and disadvantages of the V-model
- Describe the shortfall of linear models

Introduction

- All software development organizations will have to carry out SDLC activities, if they have to develop a product.
- The sequence and detailed level of these activities will depend on the nature of the project and application.

Based on product generalizations, various software process models or paradigms are available.

Software Process Models



- Software process models provide a way in which we can organize our software development into a series of activities to ensure the completion, acceptance, or appreciation of our software product by our customer.
- The process models prevalent in the software industry can be broadly categorized as:

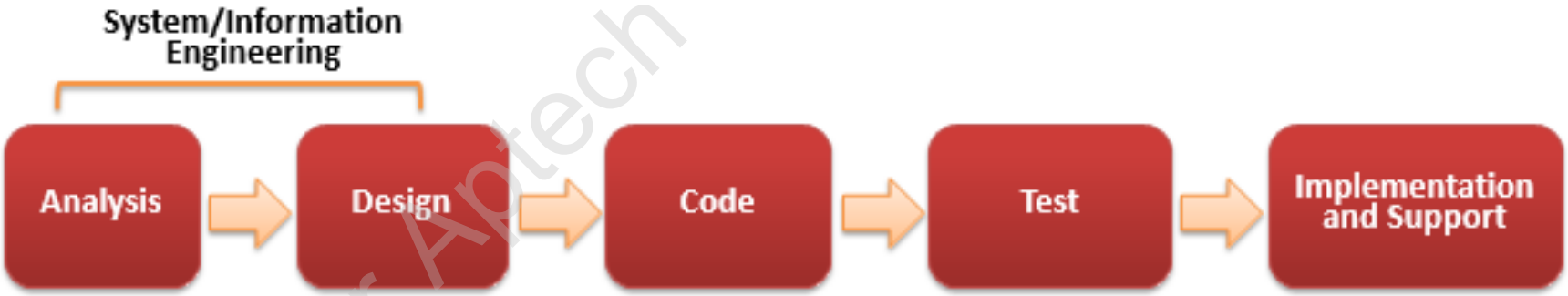
Linear Process Models

Evolutionary Process Models

Linear Process Model

Linear Process Models

- Referred to as traditional models.
- Used since early times in software development.
- Is a systematic sequential approach to software development that begins at the system level.
- Progresses through the phases of analysis, design, coding, testing, implementation, and support.
- Figure shows the sequence of activities of linear process models.



- Each phase must be completed, before you can move to next phase.

Advantages of Linear Process Models

Easy to understand and implement.

Clear demarcation between two phases.

Ensures a proper forward-looking approach to software development.

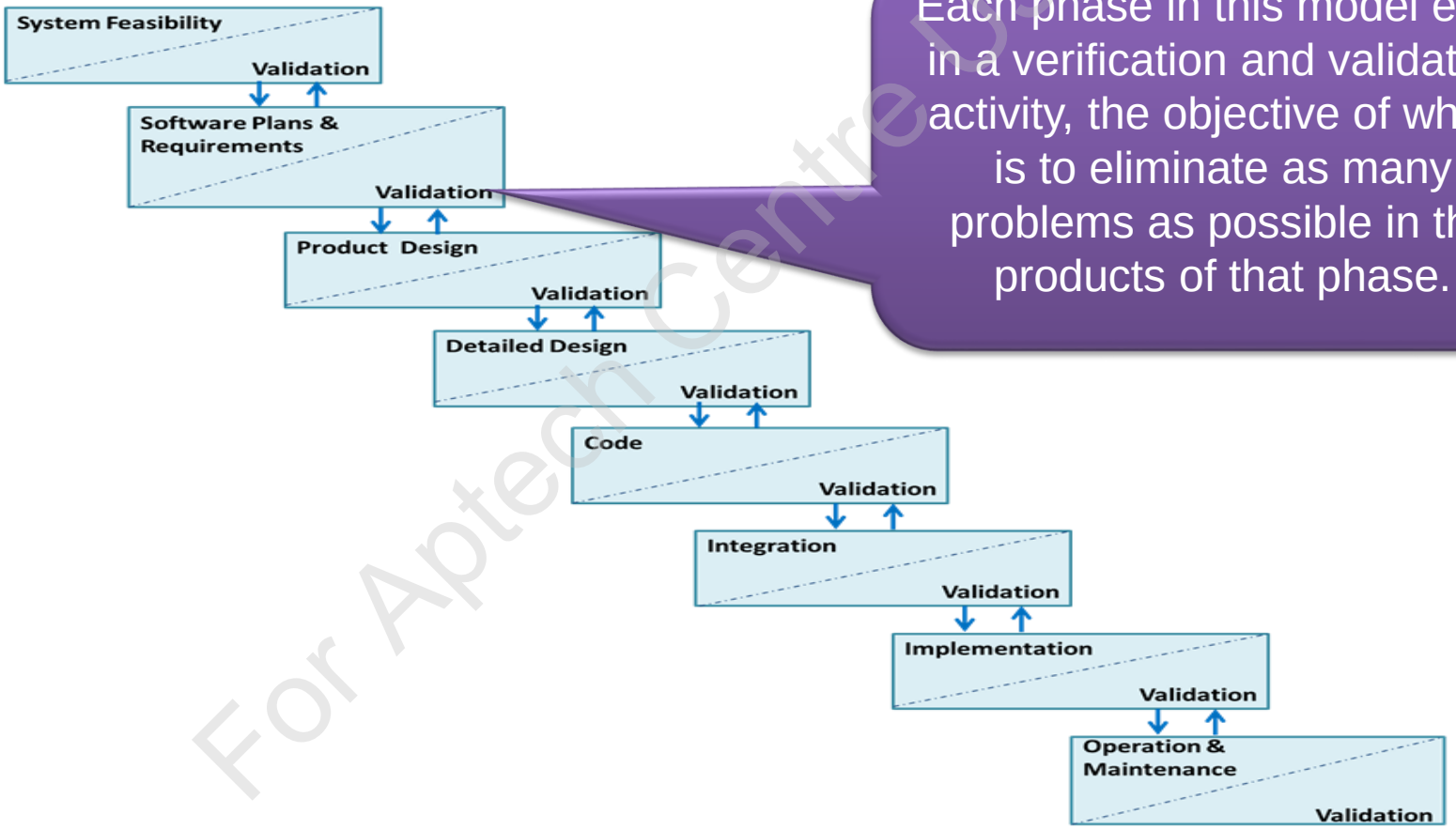
Emphasizes on requirements, before designing the solution.

Encourage close scrutiny of phase exit and entry criteria.

Waterfall Model [1-6]

The waterfall model, sometimes called the linear sequential model or 'classic life cycle'.

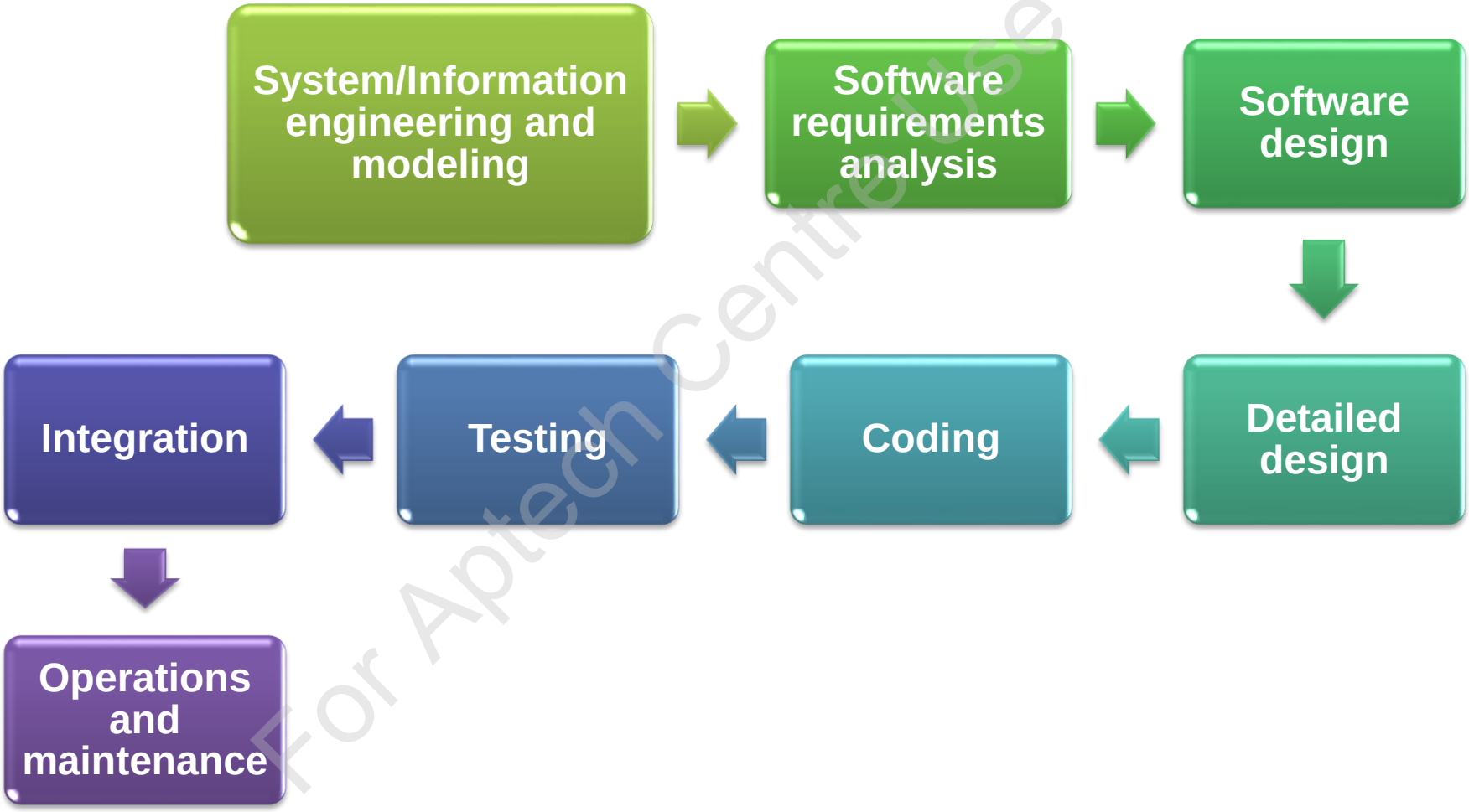
- Figure illustrates the waterfall model.



Each phase in this model ends in a verification and validation activity, the objective of which is to eliminate as many problems as possible in the products of that phase.

Waterfall Model [2-6]

- This model suggests a systematic, sequential approach to software development.



Waterfall Model [3-6]

System/Information Engineering and Modeling

- The development process has to first establish the requirements for the entire proposed system, and then identify a part of it to be delivered by the software component.
- This system view is essential when software must interface with other elements such as hardware, people, and databases.
- A feasibility of the proposed system would have to take place at this stage.

Software Requirements Analysis

- The requirements gathering process is intensified and focused specifically on software.
- Requirements for both the system and the software are documented and reviewed with the customer.

Waterfall Model [4-6]

Software Design

- It is a multi-step process, focusing on four distinct attributes of a program: data structure, software architecture, interface representations, and procedural (algorithmic) detail.
- The design process translates requirements into a representation of the software that can be assessed for quality before code generation begins.

Detailed Design

- During detailed design, the internal logic of each module specified in system design is decided.
- The logic of a module is usually specified in a high-level design description language.

Program Design Language

- PDL is one way in which the design can be communicated precisely and completely to whatever degree of detail desired by the designer.

Waterfall Model [5-6]

Coding

- The coding phase translates the design of the system, produced during the design phase, into code in a given programming language.
- The coding phase affects both the testing and maintenance phase and aims to implement the design in the best possible manner.

Testing

- The testing process focuses on the logical internals of the software, assuring that all statements have been tested, and on the functional externals, that is, conducting tests to uncover errors and ensure that defined input will produce actual results that agree with required results.
- The testing activity is carried out in two phases, the components before integration, and then the integrated product.

Integration

- Developing a strategy for integration of the components into a functioning whole requires careful planning so that the modules are available for integration when needed.

Waterfall Model [6-6]

Operations and Maintenance

- Software will undoubtedly undergo change after it is delivered to the customer (a possible exception is embedded software).
- Change will occur because errors have been encountered; the software must be adapted to accommodate changes in its external environment or because the customer requires functional or performance enhancements.
- Software maintenance reapplies each of the preceding phases to an existing program rather than a new one.

Applications of Waterfall Model

- Some situations where the use of waterfall model is most appropriate are:
  - The project requirements are clear, fixed, and very well documented.
  - The problem definition is stable.
  - There are significant resources available with the required experience to support the project development.

Waterfall Model Benefits and Drawbacks

Benefits

- Simple and easy to understand.
- Easy to arrange tasks and clearly defined stages.
- Requirement should be clear before going to next phases.
- Each phase of development has well-defined milestones.
- Works well for smaller projects.

Drawbacks

- Users can judge quality of the ready product only at the end of the development life cycle.
- Changes required at any phase may end the project development.
- Involves high level of risk and uncertainty in a project as each phase is closely linked with the earlier phase.
- It follows the 'Big-bang' approach where the entire software product is integrated at the end in the system and this may sometimes lead to challenges that may not be identified in the beginning of the project development.

V-Model [1-2]

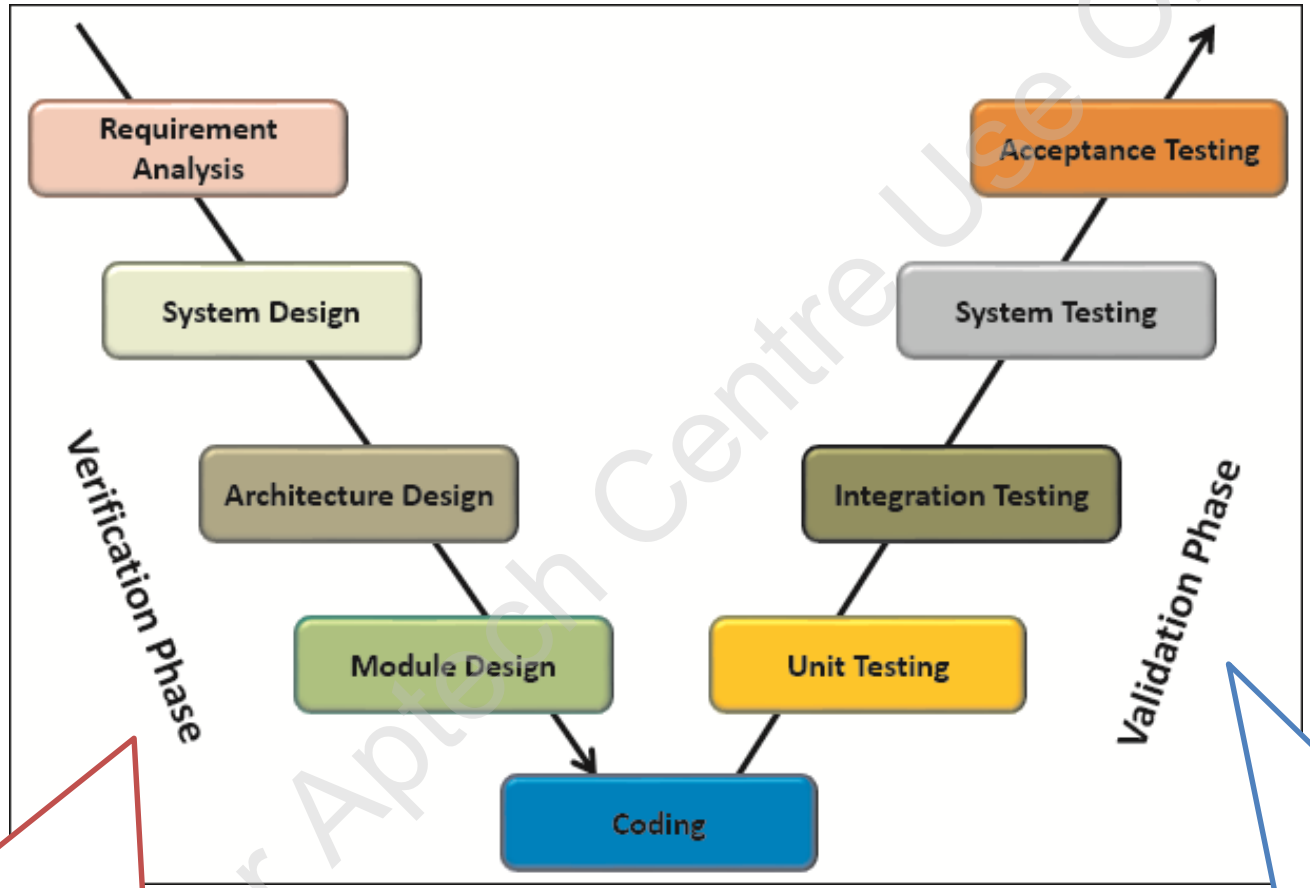
- Is an extension to the waterfall model.
- Associates each phase with a corresponding testing phase.
- Is also known as the Verification and Validation model.

Validation	• The assurance that a product, service, or system meets the needs of the customer and other identified stakeholders.
Verification	• The evaluation of whether or not a product, service, or system complies with a regulation, requirement, specification, or imposed condition.

- It is said that validation can be expressed by the query "**Are you building the right thing**?" and verification by "**Are you building it right**?".

V-Model [2-2]

- Figure shows the V-model.



The left side of the 'V' represents the decomposition of requirements and creation of system specifications.

The right side of the 'V' represents integration of parts and their validation.

Advantages and Disadvantages of V-Model

- Table lists the advantages and disadvantages of the V-model.

Advantages	Disadvantages
Simple and easy to understand.	Very rigid similar to the waterfall model.
Each phase has specific deliverables.	Little flexibility and adjusting scope is difficult and expensive.
Higher chance of success over the waterfall model due to the early development of test plans during the life cycle.	Software is developed during the implementation phase, so no early prototypes of the software are produced.
Suitable for those projects which have clear requirements.	This model does not provide a clear path for problems found during testing phases.

Shortfalls of Linear Models

- Consists of a number of dependent phases executed in a sequential order with no feedback loops.
- Late discovery of design defects, results in costly over-runs and in some cases even project cancellation.
- The acceptance test also failed to cover many off-nominal problem situations subsequently encountered in beta-testing, field-testing, hardware-software integration, or in actual mission operations.

Summary

- Software process models provide a way in which we can organize our software development into a series of activities to ensure the completion, acceptance, or appreciation of our software product by our customer.
- The process models prevalent in the software industry can be broadly categorized as Linear Process Models and Evolutionary Process Models.
- In linear process model, software development is represented by a sequence of activities or phases.
- Waterfall model suggests a systematic, sequential approach to software development that begins at the system level and progresses through analysis, design, coding, testing, and maintenance.
- V-Model is also a highly disciplined model and each phase starts only after completion of the previous phase.
- The V-Model is also known as the Verification and Validation model.
- The ideal of a software acceptance test keyed to demonstrating compliance to a set of documented requirement specifications and encountered problems as well.