Software Processes

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

Software Processes

The four types of fundamental activities

- **Software specification** – defining what the system should do;
- **Software design and implementation** – defining the organization of the system and implementing the system.
- **Software validation** – checking that it does what the customer wants
- **Software evolution** – changing the system in response to changing customer needs.

Software Process Models

A software process model is an abstract representation of a process. It presents a description of a process from some particular perspective.

Generic software process models are:

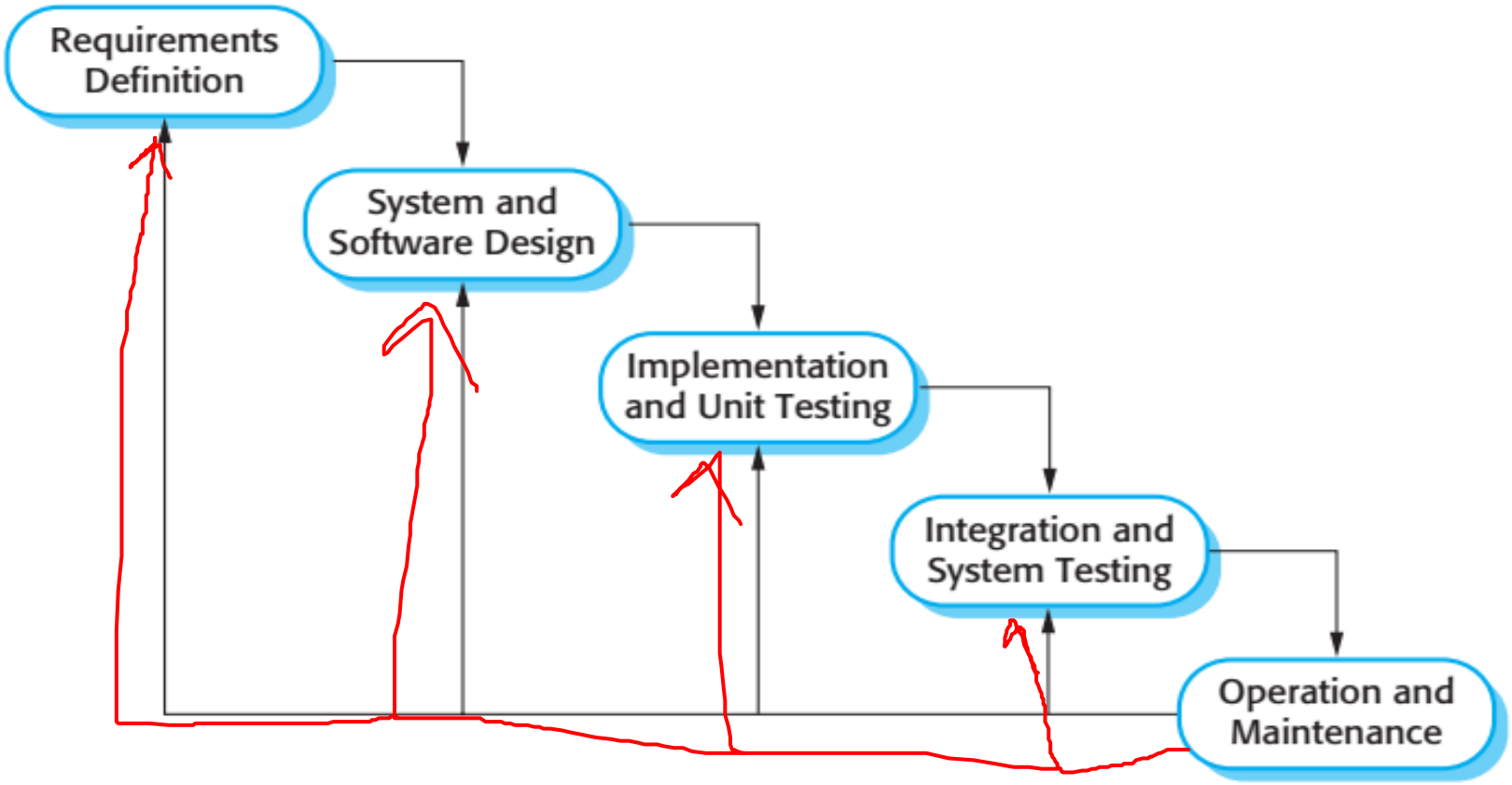
1. **The Waterfall Model**: This model has separate and distinct phases of specification and development .
2. **Evolutionary Development** : The specification and development are interleaved and an initial system is n abstract specification.
3. **Reuse-based Development or Component-Based software engineering** : The system is assembled from existing components.

Software Process Models

- **Waterfall model –** This takes the fundamental process activities of specification, development, validation and evolution and represents them as separate process phases such as requirement specification, software design, implementation, testing and so on
- **Evolutionary development** – This approach interleaves the activities of specification, development and validation. An initial system is rapidly developed from abstract specifications then refined with customer input to produce a system that specifies the customers needs
- **Component based software engineering –** This approach is based on the existence of a significant number of reusable components. The system development process focuses on integrating these components into a system rather than developing them from scratch **also called as Reuse-based Development.**

Waterfall Model

- It is the first published model of the software development process
- Because of the cascade from one phase to another this model is known as waterfall model or software life cycle



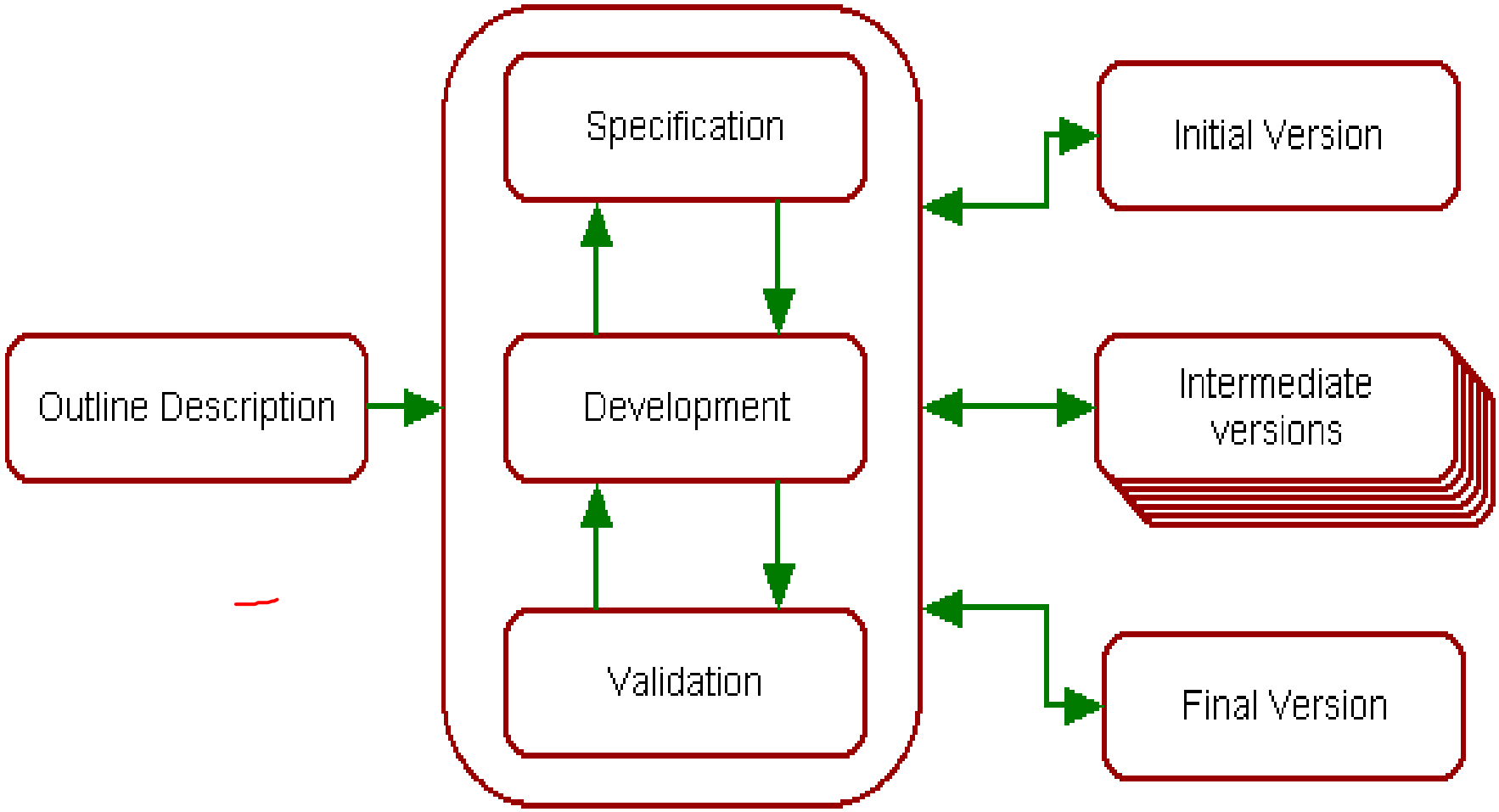
The principal stages of Waterfall model

- **Requirement analysis and definition** – The system's services, constraints and goals are established by consultation with the system users. They are then defined in detail and serve as a system specification
- **System and software design** – The system design process partitions the requirements to either software or hardware systems. It establishes an overall system architecture. Software design involves identifying and describing the fundamental software system abstractions and their relationships
- **Implementation and unit testing** – During this stage the software design is realised as a set of programs or program units. Unit testing involves verifying that each unit meets its specification
- **Integration and system testing** – The program units are integrated and tested as a complete system to ensure that the software requirements have been met. After testing the software is delivered to the customer
- **Operation and maintenance** – This is the longest life cycle phase in which the system is installed and put to use. This stage also involves correcting errors which are not discovered in earlier stages of life cycle, improving and enhancing the system services

Evolutionary Model or Prototype Model

- This approach is based on the idea of rapidly developing an initial software implementation from very abstract specifications and modifying this according to your appraisal.
- Each program version Inherits the best features from earlier versions. Each version is refined based upon feedback from yourself to produce a system which satisfies your needs.
- At this point the system may be delivered or it may be re-implemented using a more structured approach to enhance robustness and maintainability, Specification development and validation activities are concurrent with strong feedback between each.

Evolutionary Model or Prototype Model



Evolutionary Model or Prototype Model

There are two fundamental types of evolutionary development

- **Exploratory development** where the objective of the process is to work with the customer to explore their requirements and deliver a final system. The system evolves by adding new features proposed by the customer

- **Throwaway prototyping** where the objective of the evolutionary development process is to understand the customers requirement and hence develop a better requirements definition for the system. The prototype concentrates on experimenting with the customer requirements that are poorly understood

Evolutionary Model or Prototype Model

Advantages

- Exact destination of customer requirements can be known.
- It is also reducing the maintenance tasks duration because before delivery of a module, the customers are consulted.
- As the individual module takes place, so final product will contain limited number of errors.

Disadvantages

- It is only helpful for large s/w products because we can find individual modules for incremental implementation.
- It is also used when the customer is ready to receive the product.

Component-based Software-Engineering

It is based on systematic reuse where systems are integrated from existing Component or COTS (Commercial-off-the-shelf) systems.

The various process stages are:

- Component analysis
- Requirements modification
- System design with reuse
- Development and integration