

# Software Engineering Process

Gunjan Sahni

# Software Process

To produce a software product the set of activities is used. This set is called a software process.

The fundamental objectives of a process are

- **Optimality** means the process should be able to produce high-quality software at low cost, and
- **Scalability** means that it should also be applicable for large software projects.
- **Predictability** of a process determines how accurately the outcome of following a process in a project can be predicted before the project is completed

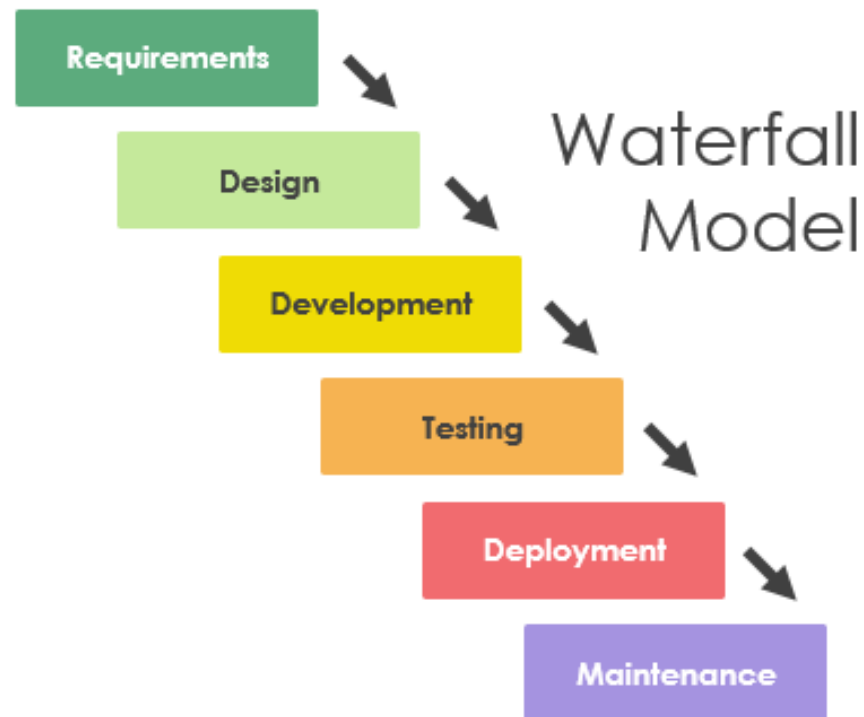
Note: testing and maintenance depend heavily on the design and coding of software, and these costs can be considerably reduced if the software is designed and coded to make testing and maintenance easier. Hence, during the early phases of the development process the prime issues should be “can it be easily tested” and “can it be easily modified”.

# Software Processes

- **Specification:** Set out the requirements and constraints on the system.
- **Design:** Produce a model of the system.
- **Manufacture:** Build the system.
- **Test:** Check the system meets the required specifications.
- **Install:** Deliver the system to the customer and ensure it is operational.
- **Maintain:** Repair faults in the system as they are discovered.

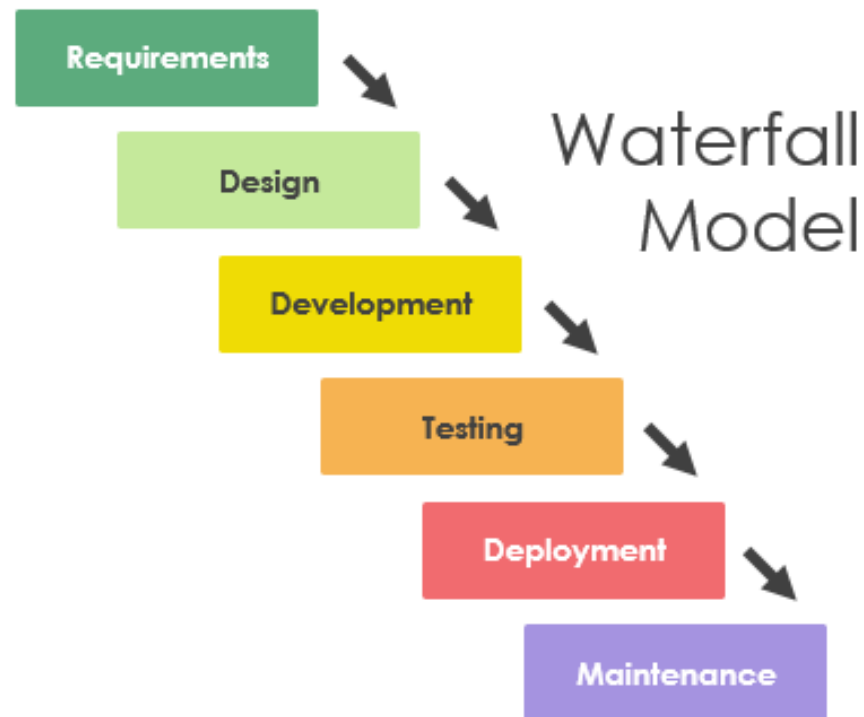
# Waterfall model

The waterfall model is a breakdown of project activities into linear sequential phases, where each phase depends on the deliverables of the previous one and corresponds to a specialisation of tasks. The approach is typical for certain areas of engineering design.



# Waterfall model

The waterfall model is a breakdown of project activities into linear sequential phases, where each phase depends on the deliverables of the previous one and corresponds to a specialisation of tasks. The approach is typical for certain areas of engineering design.



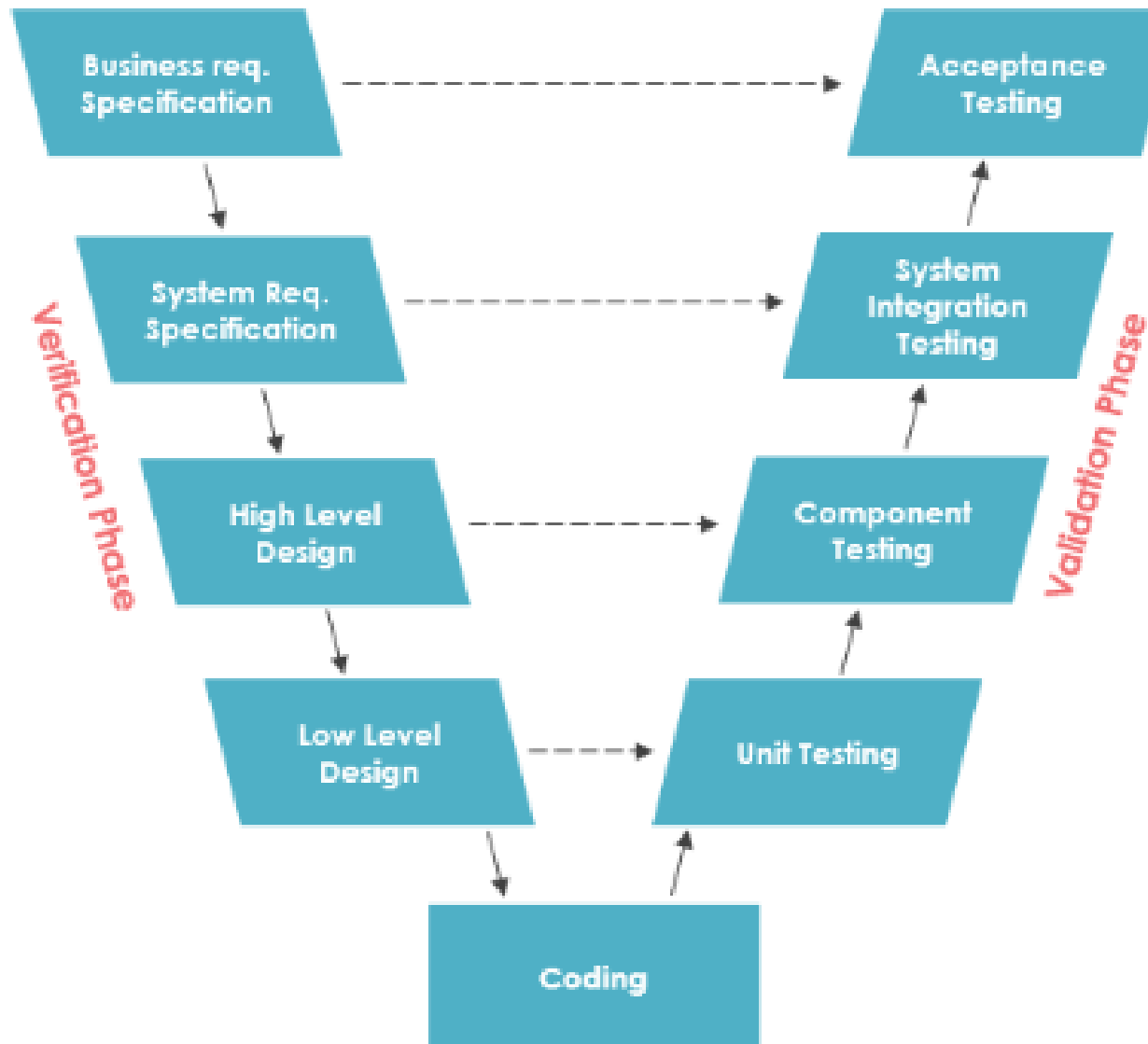
# V model

- The V-model represents a development process that may be considered an extension of the waterfall model.
- The process steps are bent upwards after the coding phase, to form the typical V shape.
- The V-Model demonstrates the relationships between each phase of the development life cycle and its associated phase of testing.
- The horizontal and vertical axes represent time or project completeness (left-to-right) and level of abstraction (coarsest-grain abstraction uppermost), respectively.

## V-Model

## Developer's Life Cycle

## Tester's Life Cycle



# Prototype Model

- Applied when detailed information related to input and output requirements of the system is not available.
- It is usually used when a system does not exist or in case of a large and complex system where there is no manual process to determine the requirements.
- It allows the users to interact and experiment with a working model of the system known as **prototype**. The prototype gives the user an actual feel of the system.
- At any stage, if the user is not satisfied with the prototype, it can be discarded and an entirely new system can be developed



# Prototype Model

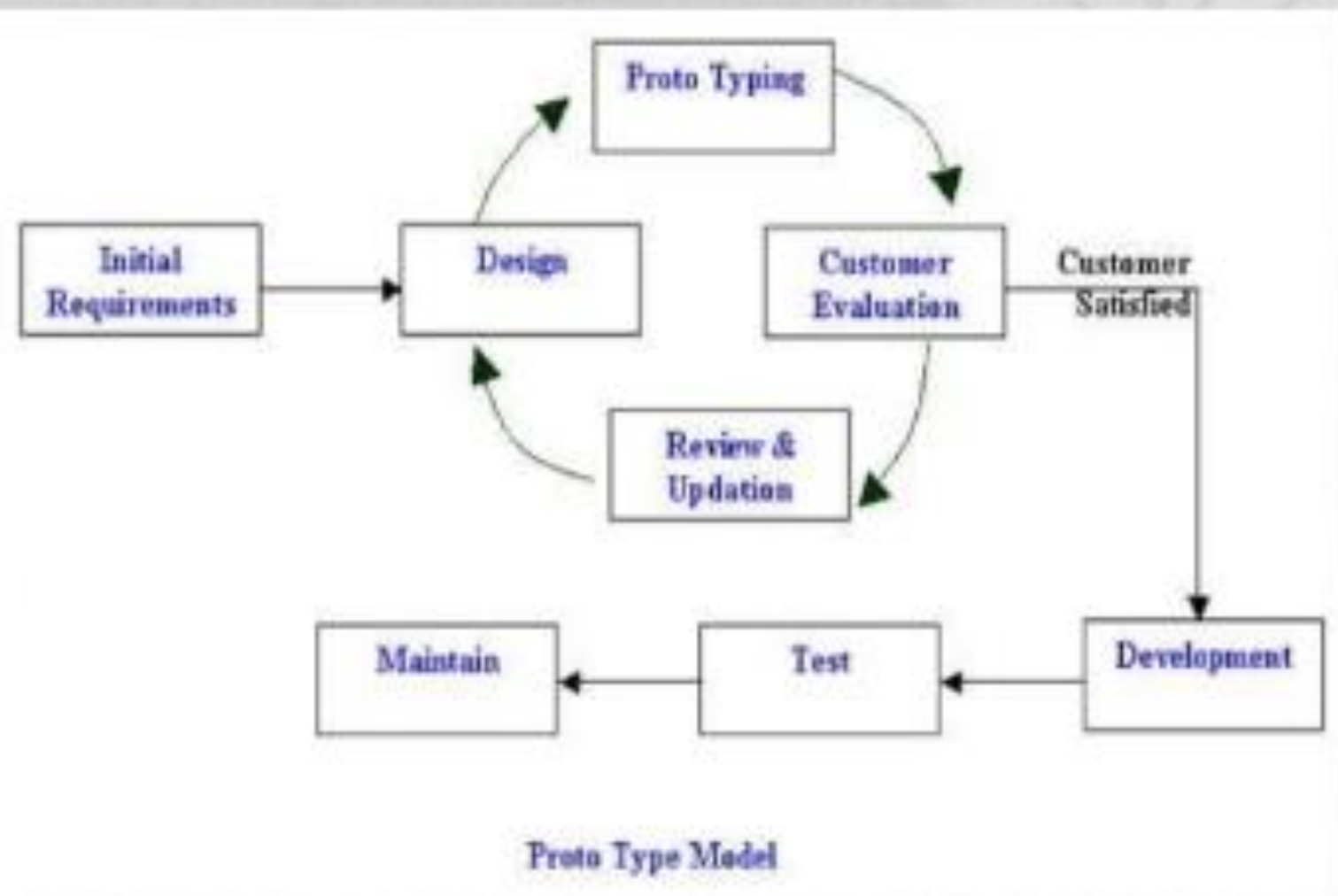
Prototype can be prepared by the approaches listed below.

- By creating main user interfaces without any substantial coding so that users can get a feel of how the actual system will appear.
- By abbreviating a version of the system that will perform limited subsets of functions.
- By using system components to illustrate the functions that will be included in the system to be developed .

# Prototype Model

- They provide feedback to the developers regarding the prototype: what is correct, what needs to be modified, what is missing, what is not needed, etc.
- Based on the feedback, the prototype is modified to incorporate some of the suggested changes that can be done easily, and then the users and the clients are again allowed to use the system.
- This cycle repeats until, in the judgment of the prototypes and analyst. Based on the feedback, the initial requirements are modified to produce that final requirements specification, which is then used to develop the production quality system.

# PROTOTYPE MODEL



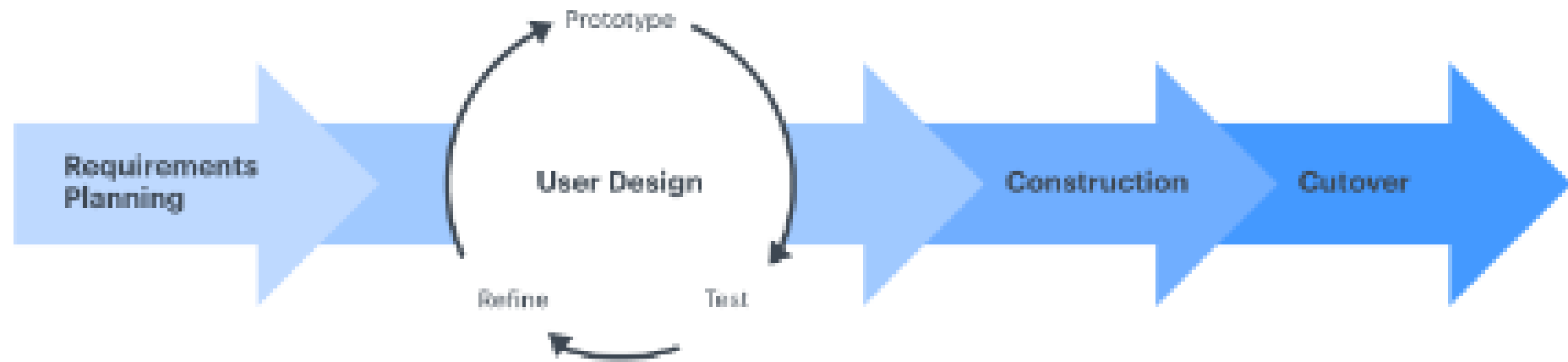
# RAD model

## (Rapid Application Development Model)

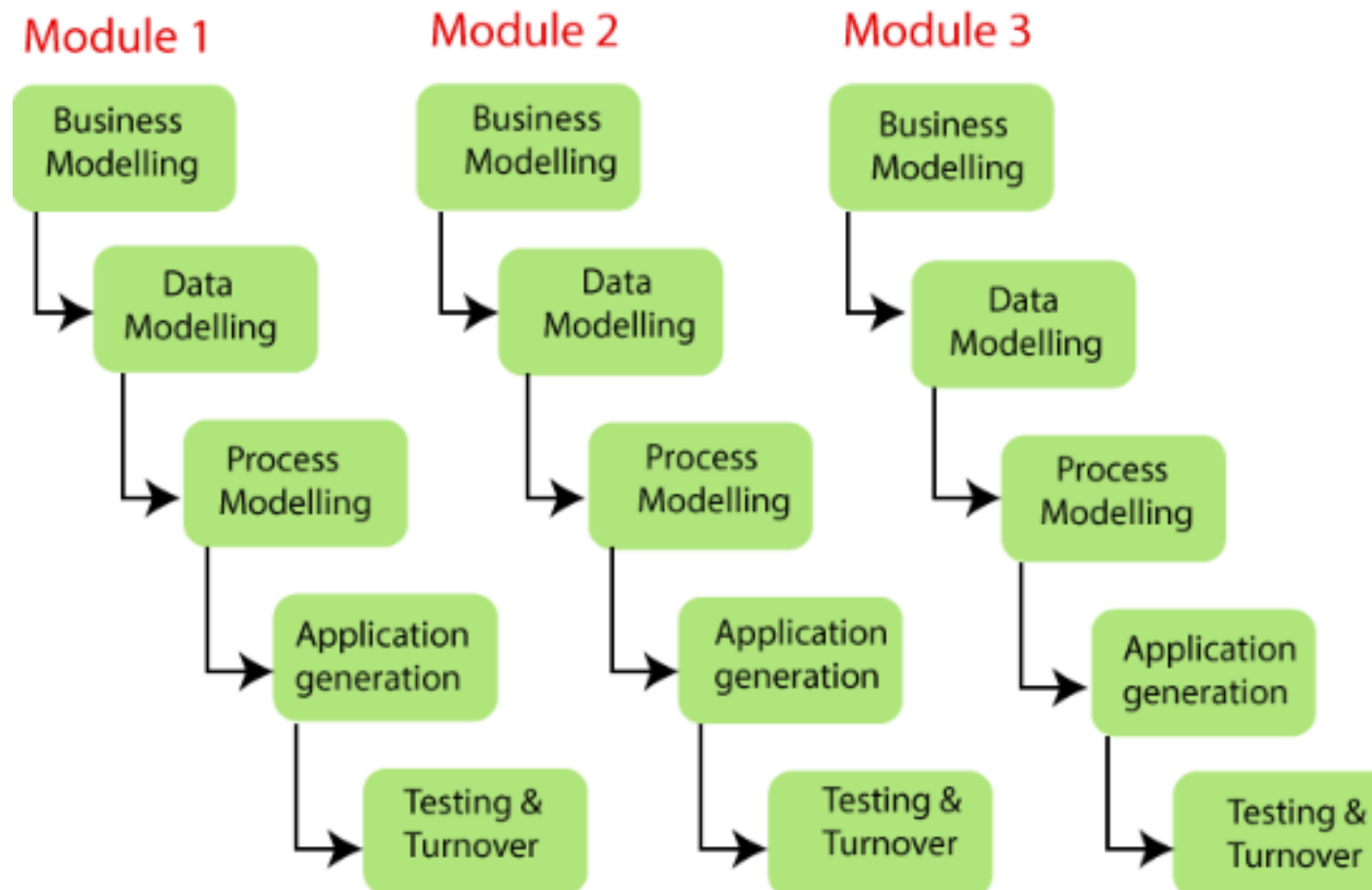
- The **RAD (Rapid Application Development)** model is based on prototyping and iterative development with no specific planning involved.
- Rapid Application Development focuses on gathering customer requirements through workshops or focus groups, early testing of the prototypes by the customer using iterative concept, reuse of the existing prototypes (components), continuous integration and rapid delivery.

- Uses minimal planning in favour of rapid prototyping. A prototype is a working model that is functionally equivalent to a component of the product.
- In the RAD model, the functional modules are developed in parallel as prototypes and are integrated to make the complete product for faster product delivery.
- Since there is no detailed preplanning, it makes it easier to incorporate the changes within the development process.
- RAD projects follow iterative and incremental model
- Have small teams comprising of developers, domain experts, customer representatives and other IT resources working progressively on their component or prototype.
- The most important aspect for this model to be successful is to make sure that the prototypes developed are reusable.

## Rapid Application Development (RAD)



# Different phase Of RAID Model



# Evolutionary model

- also known as the **successive versions model or the incremental model**.
- The software requirement is first broken down into several modules or functional unit that can be incrementally constructed and delivered
- Each evolutionary model may be developed using an iterative waterfall model of development.
- **Used** when the customer prefers to receive the product in increments so that he can start using the different features as and when they are developed rather than waiting all the time for the full product to be developed and delivered.