Software Process

Presented by Dr. Rubeena Doomun

Learning Objectives

- Understand the concepts of software processes and software process models.
- be familiar with three generic software process models and when they might be used.
- be familiar with the fundamental process activities of software requirements engineering, software development, testing, and evolution.

Software Process

A software process is the set of activities and associated outcome that produce a software product.

Software engineers mostly carry out these activities.

These are four key process activities, which are common to all software processes.

01. **Software specifications:**
The functionality of the software and constraints must be defined.

02. **Software development:**
The software to meet the requirement must be produced.

03. **Software validation:**
The software must be validated to ensure that it does what the customer wants.

04. **Software evolution:**
The software must evolve to meet changing client needs.

Software Process Model

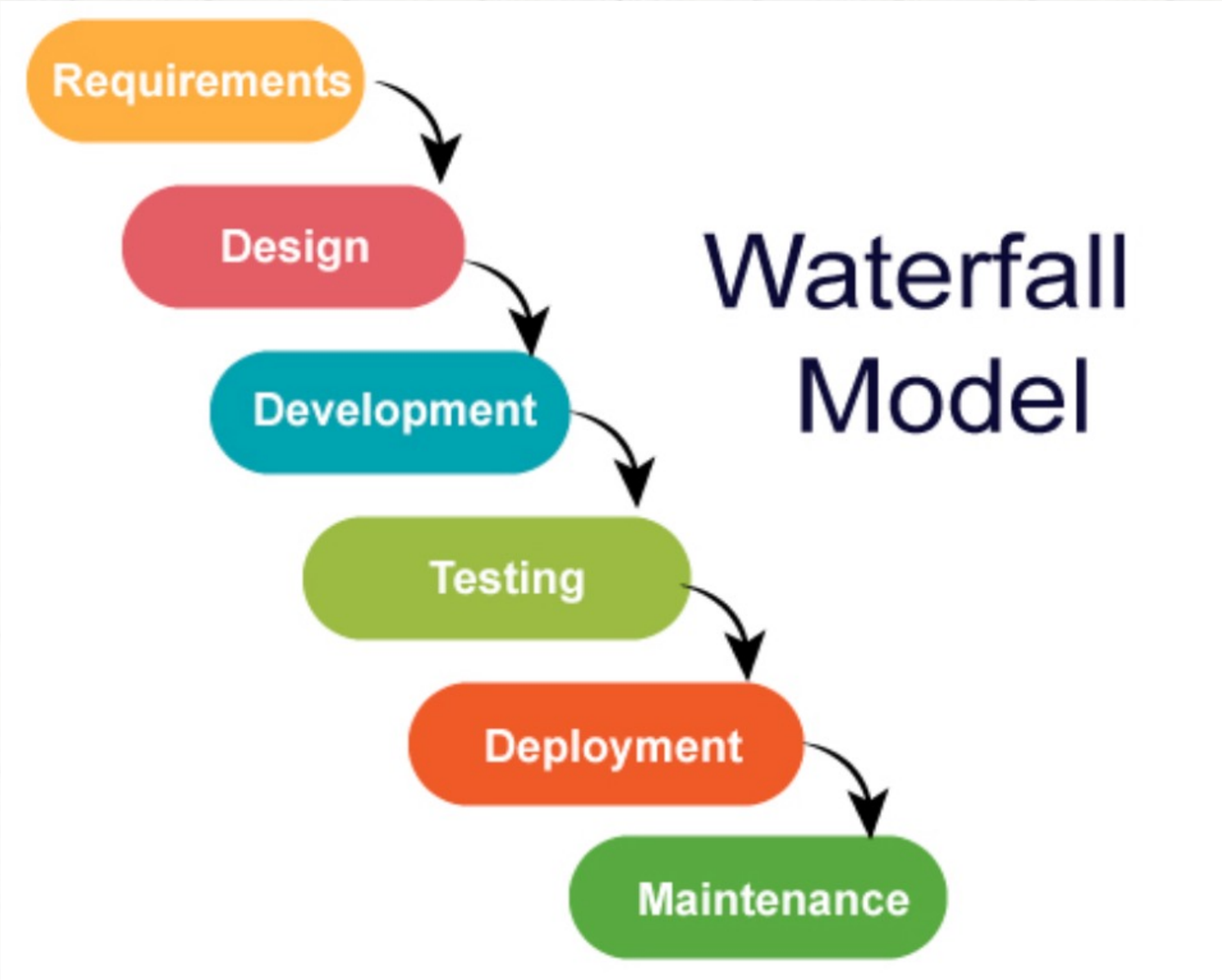
A software process model is a specified definition of a software process, which is presented from a particular perspective.

A software process model is an abstraction of the actual process, which is being described.

Process models may contain activities, which are part of the software process, software product, and the roles of people involved in software engineering.

Kahoot! Time

Waterfall Model



Why choose waterfall model?

•When the requirements are constant and not changed regularly.

•A project is short.

•Where the tools and technology used is consistent and is not changing.

•When resources are well prepared and are available to use.

Advantages

Waterfall model

This model is simple to implement also the number of resources that are required for it is minimal.

The requirements are simple and explicitly declared; they remain unchanged during the entire project development.

The start and end points for each phase is fixed, which makes it easy to cover progress.

The release date for the complete product, as well as its final cost, can be determined before development.

It gives easy to control and clarity for the customer due to a strict reporting system.

Disadvantages

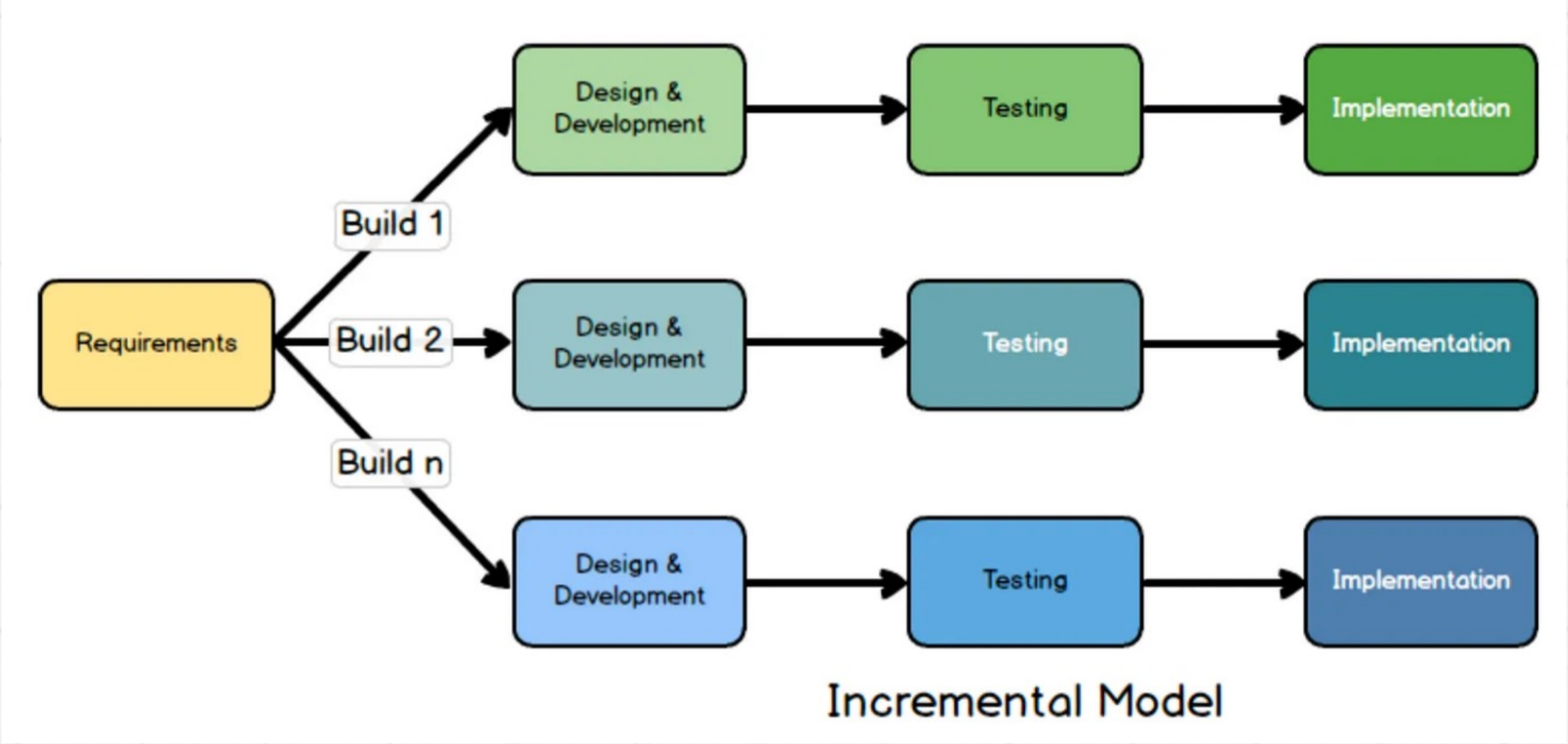
Waterfall model

This model cannot accept the changes in requirements during development.

It becomes tough to go back to the phase. For example, if the application has now shifted to the coding phase, and there is a change in requirement, It becomes tough to go back and change it.

Since the testing done at a later stage, it does not allow identifying the challenges and risks in the earlier phase, so the risk reduction strategy is difficult to prepare.

Incremental Model



Why choose incremental model?

- When the requirements are superior.
- A project has a lengthy development schedule.
- When software team are not very well skilled or trained.
- When the customer demands a quick release of the product.
- Prioritized requirements can be developed first.

Advantages

Incremental model

Errors are easy to be recognized.

Easier to test and debug.

Early feedback.

Simple to manage risk because it is handled during its iteration.

Deliver functionality early.

Regular change tends to destroy the software's structure unless time and money are spent on refactoring to improve it.

Well defined module interfaces are needed.

Disadvantages

Incremental model

Software modifications become more difficult and expensive to incorporate.

Needs a proper planning.

Reuse Oriented Model

Reuse Oriented Model (ROM), also known as reuse-oriented development (ROD), it can be steps of the software development for specific duration in which software is redesigned through creating a sequence of prototypes known as models, every system is derived from the previous one with constant series of defined rules.

Reuse software engineering is based on guidelines and principles for reusing the existing software.

Why choose software reuse?

•Less effort: Software reuse requires less effort because many components use in the system are ready made components.

•Time-saving: Re-using the ready made components is time saving for the software team.

•Reduce cost: Less effort, and time saving leads to the overall cost reduction.

•Increase software productivity: when provided with ready made components, then the focus will be on new components that are not available.

•Utilize fewer resources: Software reuse save many sources just like effort, time, money etc.

•Leads to a better quality software: Software reuse save time and can consume our more time on maintaining software quality and assurance.

Advantages

Reuse oriented model

Reduce total cost of software development.

Efficient process

Risk factor is very low.

Save time and less effort.

It does not always worked as a practice in its true form.

Compromises in requirements may lead to a system that does not fulfill requirement of user.

Disadvantages

Reuse oriented model

Sometimes using old system component, that is not compatible with new version of component, this may lead to an impact on system evolution.

System integration needs to be part of the development process.

Kahoot! Time

Thank you