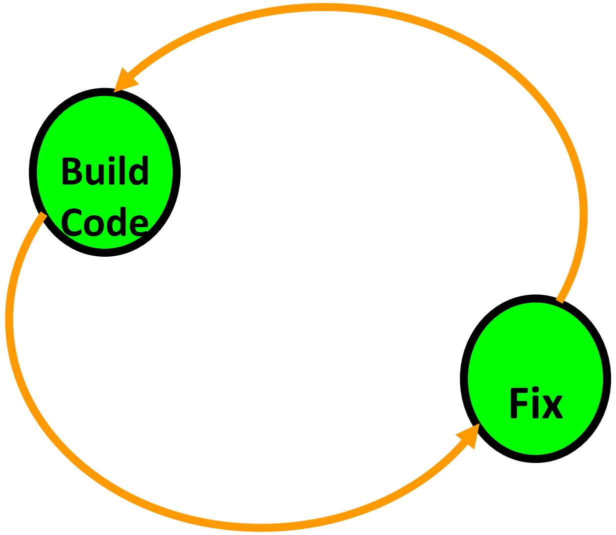
Software Life Cycle Models

- The goal of Software Engineering is to provide models and processes that lead to the production of well-documented maintainable software in a manner that is predictable.
- “The period of time that starts when a software product is conceived and ends when the product is no longer available for use. The software life cycle typically includes a requirement phase, design phase, implementation phase, test phase, installation and check out phase, operation and maintenance phase, and sometimes retirement phase”.

Build & Fix Model

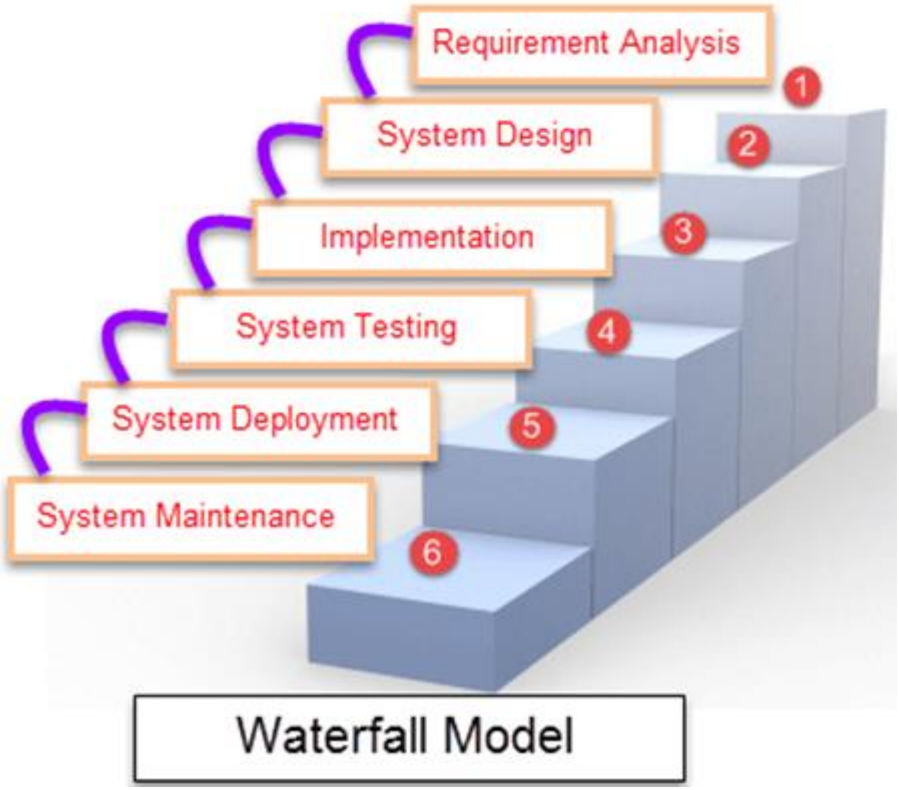
- Product is constructed without specifications or any attempt at design
- Adhoc approach and not well defined
- Simple two phase model
- Suitable for small programming exercises of 100 or 200 lines
- Unsatisfactory for software for any reasonable size
- Code soon becomes unfixable & unenhanceable
- No room for structured design
- Maintenance is practically not possible



WATERFALL MODEL

This model is named “waterfall model” because its diagrammatic representation resembles a cascade of waterfalls.

WATERFALL MODEL is a sequential model that divides software development into pre-defined phases. Each phase must be completed before the next phase can begin with no overlap between the phases. Each phase is designed for performing specific activity during the SDLC phase.



WATERFALL MODEL

Requirement Gathering stage: During this phase, detailed requirements of the software system to be developed are gathered from client

•**Design Stage:** Plan the programming language, for Example Java, PHP, .net

•or database like Oracle, MySQL, etc.

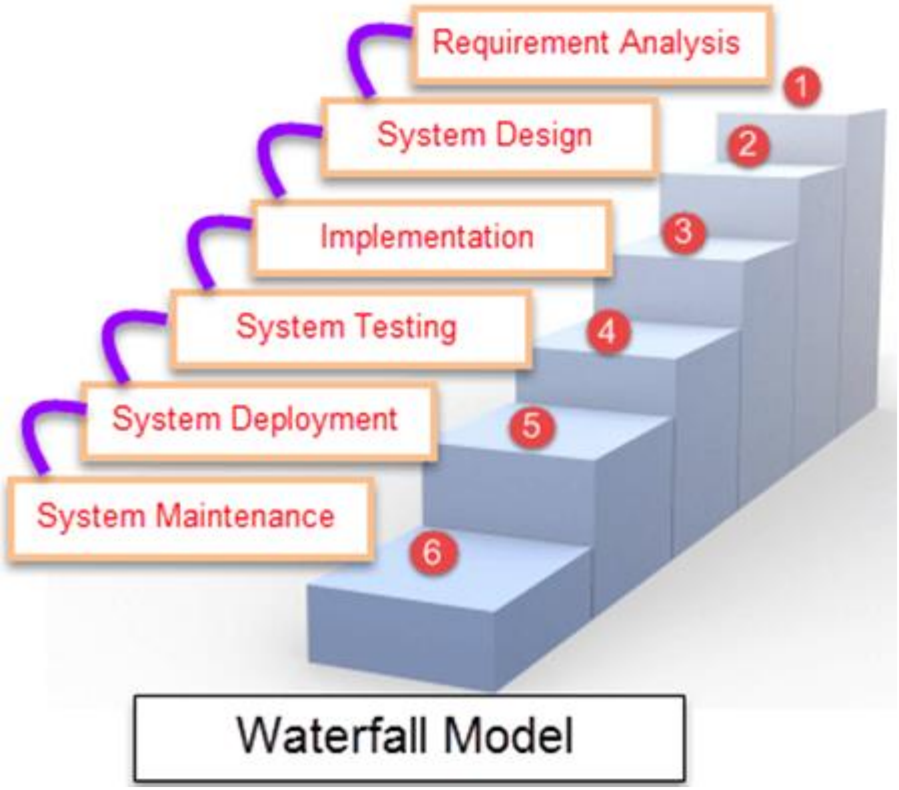
•Or other high-level technical details of the project

Built Stage: After design stage, it is built stage, that is nothing but coding the software

Test Stage: In this phase, you test the software to verify that it is built as per the specifications given by the client.

Deployment stage: Deploy the application in the respective environment

Maintenance stage: Once your system is ready to use, you may later require change the code as per customer request



When to use SDLC Waterfall Model

- Waterfall model can be used when Requirements are not changing frequently
- Application is not complicated and big
- Project is short
- Requirement is clear
- Environment is stable
- Technology and tools used are not dynamic and is stable
- Resources are available and trained

Problems of waterfall model

- It is difficult to define all requirements at the beginning of a project
- This model is not suitable for accommodating any change
- A working version of the system is not seen until late in the project’s life
- It does not scale up well to large projects.
- Real projects are rarely sequential.

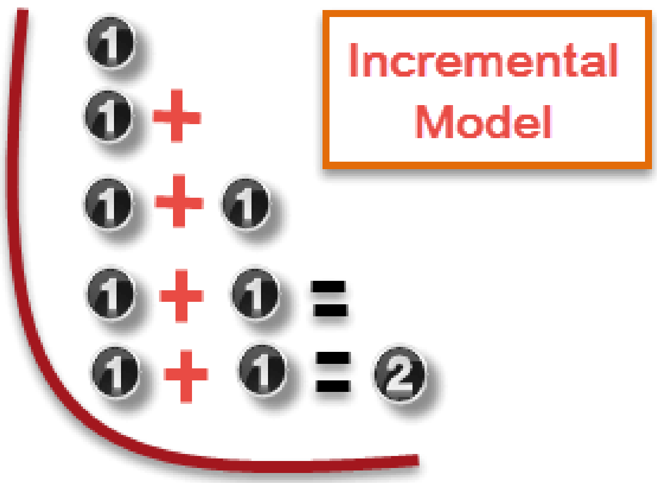
Incremental Model/Iterative Model

Incremental Model is a process where the project is divided into small subsets known as **increments** that are implemented individually.

This model comprises several iterations where each iteration produces an increment.

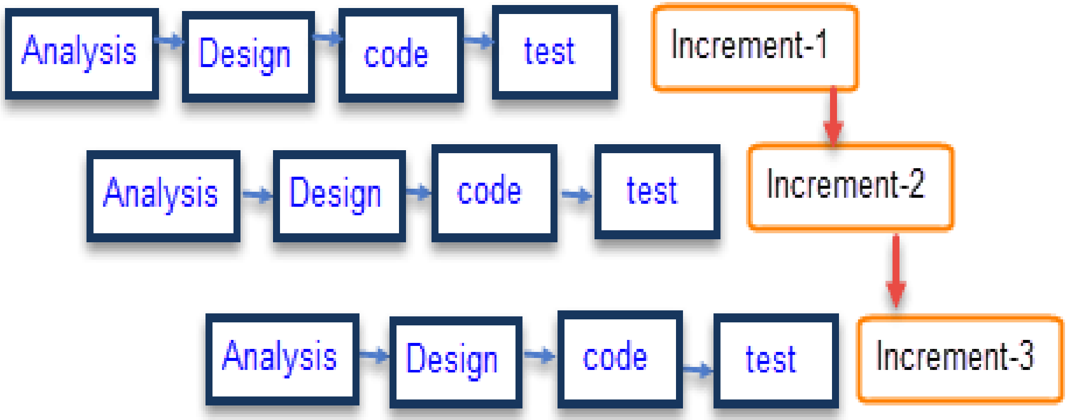
These increments are identified in the beginning of the development process and the entire process from requirements gathering to delivery of the product is carried out for each increment.

The basic idea of this model is to start the process with requirements and iteratively enhance the requirements until the final software is implemented.



Incremental Model

Each iteration passes through the **requirements, design, coding and testing phases**. And each subsequent release of the system adds function to the previous release until all designed functionality has been implemented.



Incremental Model

Incremental Model

- The system is put into production when the first increment is delivered. The first increment is often a core product where the basic requirements are addressed, and supplementary features are added in the next increments.
- Once the core product is analyzed by the client, there is plan development for the next increment.
- This plan determines the modifications (features or functions) of the product in order to accomplish user requirements. The iteration process, which includes the delivery of the increments to the user, continues until the software is completely developed.

Incremental Model

Characteristics of an Incremental module includes

- System development is broken down into many mini development projects
- Partial systems are successively built to produce a final total system
- Highest priority requirement is tackled first
- Once the requirement is developed, requirement for that increment are frozen

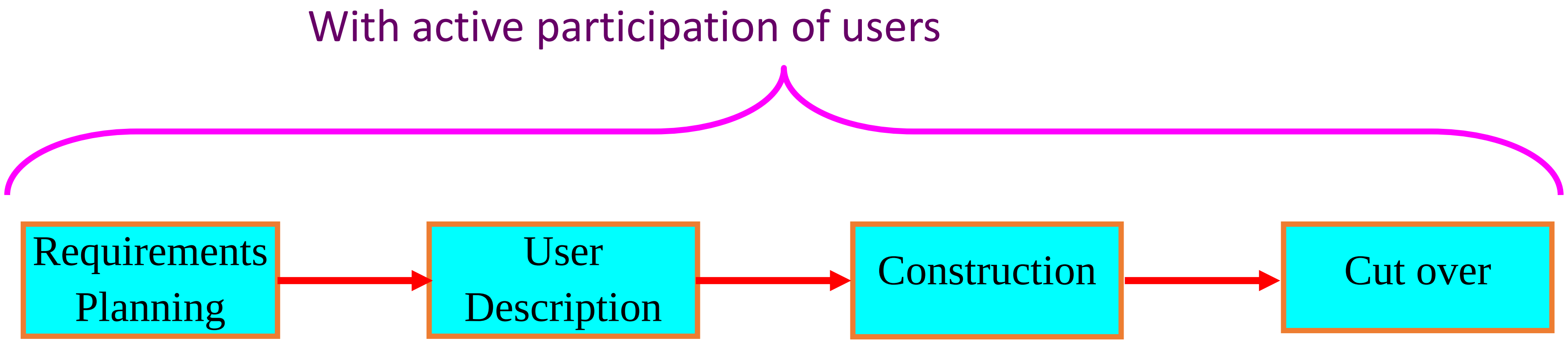
Incremental Model

When to use Incremental models?

- Requirements of the system are clearly understood
- When demand for an early release of a product arises
- Popular particularly when we have to quickly deliver a limited functionality system.
- When software engineering team are not very well skilled or trained

The Rapid Application Development (RAD) Model

- Developed by IBM in 1980
- User participation is essential
- Build a rapid prototype
- Give it to user for evaluation & obtain feedback
- Prototype is refined
-



The Rapid Application Development (RAD) Model

- A software project can be implemented using this model if the project can be broken down into small modules wherein each module can be assigned independently to separate teams.
- It is an incremental model, the Application will be developed in a set of divided timelines called iterations, each iteration results in a deliverable product at the end.
- These modules can finally be combined to form the final product.
- Development of each module involves the various basic steps as in waterfall model i.e analyzing, designing, coding and then testing, etc. as shown in the figure.
- Another striking feature of this model is a short time span i.e the time frame for delivery(time-box) is generally 60-90 days

The Rapid Application Development (RAD) Model

- This model consists of 4 basic phases:

1. **Requirements Planning –**
   It involves the use of various techniques used in requirements elicitation like brainstorming, task analysis, form analysis, user scenarios, FAST (Facilitated Application Development Technique), etc. It also consists of the entire structured plan describing the critical data, methods to obtain it and then processing it to form final refined model.
2. **User Description –**
   This phase consists of taking user feedback and building the prototype using developer tools. In other words, it includes re-examination and validation of the data collected in the first phase. The dataset attributes are also identified and elucidated in this phase.

The Rapid Application Development (RAD) Model

1. **Construction –**
   In this phase, refinement of the prototype and delivery takes place. It includes the actual use of powerful automated tools to transform process and data models into the final working product. All the required modifications and enhancements are too done in this phase
2. **Cutover –**
   All the interfaces between the independent modules developed by separate teams have to be tested properly. The use of powerfully automated tools and subparts makes testing easier. This is followed by acceptance testing by the user.

When to use RAD Methodology?

- When a system needs to be produced in a short span of time (2-3 months)
- When the requirements are known
- When the user will be involved all through the life cycle
- When technical risk is less
- When there is a necessity to create a system that can be modularized in 2-3 months of time
- When a budget is high enough to afford designers for modeling along with the cost of automated tools for code generation