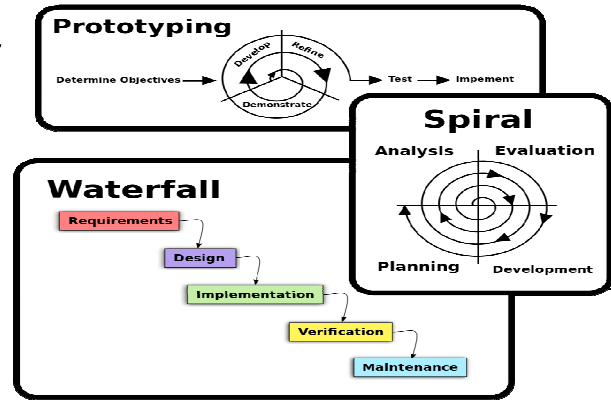
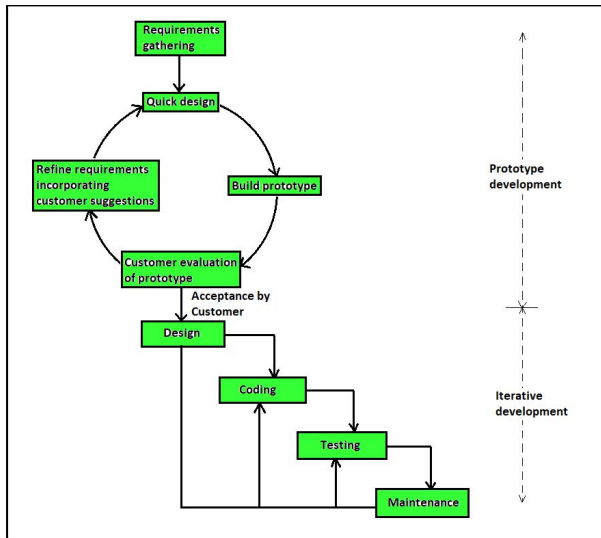


Software Engineering



Umbrella Activities

- **Software Development life cycle** are complemented by a number of **umbrella activities**. In general, umbrella activities are applied throughout a software project and help a software team manage and control progress, quality, change, and risk. Typical umbrella activities include:

Umbrella Activities

- **Software project tracking and control**
- **Risk management**
- **Software quality assurance**
- **Technical reviews**
- **Measurement**
- **Software configuration management**
- **Reusability management**
- **Work product preparation and production**

Umbrella Activities

1. ***Software project tracking and control:*** allows the software team to assess progress against the project plan and take any necessary action to maintain the schedule.
2. ***Risk management:*** assesses risks that may affect the outcome of the project or the quality .
3. ***Software quality assurance:*** defines and conducts the activities required to ensure software quality.
4. ***Technical reviews:*** assesses software engineering work products in an effort to uncover and remove errors before they are propagated to the next activity.
5. ***Measurement:*** defines and collects process, project, and product measures that assist the team in delivering software that meets customer needs.

Umbrella Activities

6. ***Software configuration management:*** manages the effects of change throughout the software process.
7. ***Reusability management:*** defines criteria for work product reuse (including software components) and establishes mechanisms to achieve reusable components.
8. ***Work product preparation and production:*** encompasses the activities required to create work products such as models, documents, logs, forms, and lists.

Software Process Models

- ***Process*** is a framework for the tasks that are required to build high-quality product.
- ***A process in software engineering*** is an adaptable approach that enables the software team doing the work to pick and choose the appropriate set of activities, actions, and tasks that are required to build high-quality software.
- ***A (software/system) process model*** is a description of the sequence of activities carried out in an software project, and the relative order of these activities.
- ***Software Process Models*** provides a fixed generic framework that can be tailored to a specific project. Always each project has specific parameters will include: Size, (person-years), Budget, Duration.

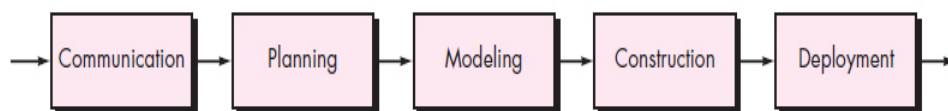
Generic Process Framework

1. ***Communication:*** Involves communication among the customer and other stake holders; encompasses requirements gathering
2. ***Planning:*** Establishes a plan for software engineering work; addresses technical tasks, resources, work products, and work schedule
3. ***Modelling (Analyze, Design):*** Encompasses the creation of models to better under the requirements and the design
4. ***Construction (Code, Test):*** Combines code generation and testing to uncover errors
5. ***Deployment:*** Involves delivery of software to the customer for evaluation and feedback

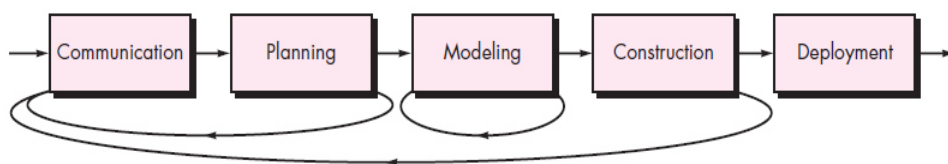
Software Process Models

- The framework activities organized with respect to sequence and time and there are different types of process flow.
- **Linear process flow** executes each of the activities **in sequence**.
- **Iterative process flow** **repeats one or more of the activities** before proceeding to the next.
- **Evolutionary process flow** executes the activities in a **circular manner**. Each circuit leads to a more complete version of the software.
- **parallel process flow** executes **one or more activities in parallel** with other activities.

Software Process Models

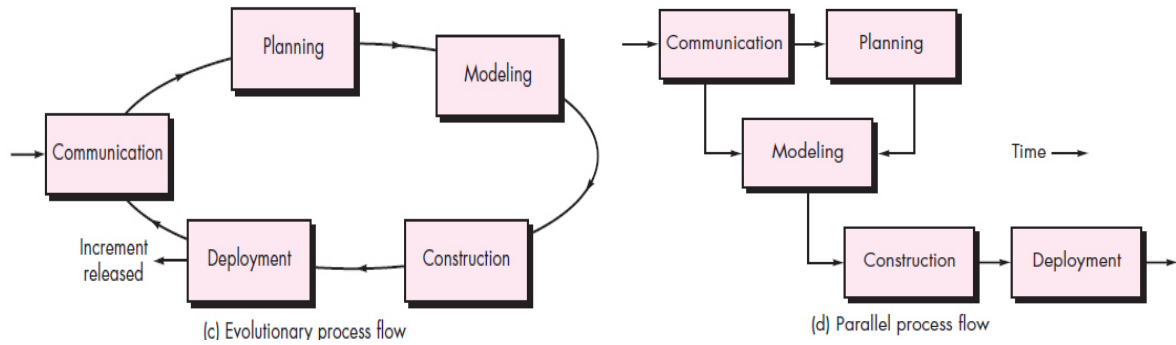


(a) Linear process flow



(b) Iterative process flow

Software Process Models



Software Process Models

1. Prescriptive Process Models

I. *The Waterfall Model*

II. *The Prototype Model*

III. *Evolutionary Software Process Models*

1. *The Spiral Model*

2. *The Incremental Model*

2. Specialized Process Models:

I. *Component-Based Development.*

Prescriptive Process Models

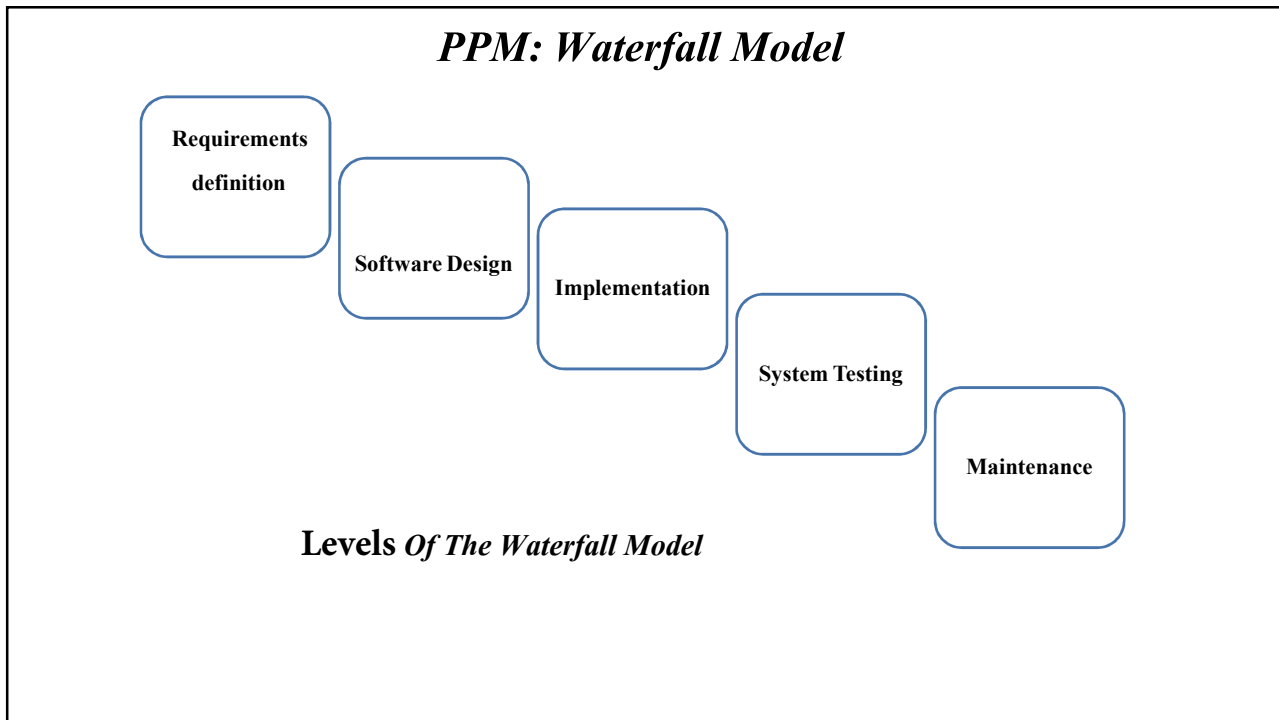
- *The process models were originally proposed to bring **order to the chaos** of software development.*
- *There are various software development approaches defined and designed which are used/employed during development process of software. and each describes their own solution to developing valid software including: waterfall, spiral and agile.*
- *Each process model follows a particular life cycle in order to ensure success in process of software development.*

11

PPM: Waterfall Model





- ***Waterfall Model** is also called the **Linear Sequential Model**, described by Royce in 1970.*
- ***Waterfall Model** was first Process Model to be introduced and followed widely in Software Engineering to ensure success of the project.*
- *The waterfall model **suggests a systematic, sequential approach** to software development that begins with **customer specification of requirements** and progresses through planning, modeling, construction, deployment and ongoing support of the completed software.*



PPM: Waterfall Model

- In the ***Waterfall Model*** when the **first phase is finished then only the second phase will start and so on.**

 This model does not work smoothly if there are some issues left at the previous step. The sequential nature of model does not allow us go back and **undo** or **redo** our **actions**.

 This model is best suited when developers already have designed and developed similar software in the past and are aware of all its domains.

PPM: Waterfall Model

1. **Requirement Specifications (analysis and definition):** Requirements are set of **functionalities and constraints** that the end-user (who will be using the system) expects from the system.
2. **Software Design:** System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture.
3. **Implementation:** On receiving system design documents, the work is divided in modules/units and actual coding is started. The system is first developed in small programs called units, which are integrated in the next phase.

PPM: Waterfall Model

4. **Testing:** The system is first divided in units which are developed and tested for their functionalities. These units are integrated into a complete system during Integration phase and tested to check if all modules/units coordinate between each other and the system as a whole behaves as per the specifications.
5. **Maintenance:** Generally, problems with the system developed (which are not found during the development life cycle) **come up after its practical use starts**, **so** the **issues** related to the system are solved **after** deployment of the system.

PPM: Waterfall Model

- ***The Waterfall Model Advantages***

1. Testing is **inherent** to every phase of the waterfall model.
2. It is an enforced **disciplined** approach.
3. It is **documentation driven**, that is, documentation is produced at every stage.
4. Easy to understand and implement.
5. Widely used and known (in theory!)
6. Fits other engineering process models: civil, mech etc.
7. Reinforces good habits: define-before- design, design before-code.

PPM: Waterfall Model

- ***The Waterfall Model disadvantages***

1. The main drawback of the waterfall model **is the difficulty of accommodating change after the process is underway. In principle, a phase has to be complete before moving onto the next phase.**
2. **Not all requirements are received at once,** this affects the system development process and its success in negative aspects.
3. The problems with one phase **are never solved completely during that phase,** this results in badly structured system as not all the problems (related to a phase) are solved during the same phase.

The End