Savitribai Phule Pune University

S.Y.B.B.A. (CA)

Course Code: CA-303

Software Engineering

Chapter 1

Introduction To Software Engineering and Process Models

What is Software Engineering ?...

Software engineering is defined as a process of analyzing user requirements and then designing, building, and testing software application which will satisfy those requirements.

Definition of Software

Software is a set of instructions, data or programs used to operate computers and execute specific tasks.

Example : Word Processor, Adobe Photoshop, Adobe Acrobat Reader etc.

The Four Categories of Software

- Programming software
- System software
- Application software
- Malicious software (malware)

Programming software

Programming software is a set of tools to developers in writing programs. The various tools available are compilers, linkers, debuggers, interpreters and text editors.

Different programming language editors, debuggers, compilers and IDEs
are examples of programming software.

System software

System software serves as a base for application software. System software includes device drivers, operating systems (OSs), compilers, disk formatters, text editors and utilities helping the computer to operate more efficiently.

Application software

Application software is intended to perform certain tasks. Examples of application software include office suites, gaming applications, database systems and educational software.

Malicious software (malware)

Malicious software is intentionally developed to damage computers and/or disrupt other software

Nature of Software Engineering

In **software engineering**, there is a well developed science, **computer** science, that covers, among other things, concepts of programming languages, algorithms, data structures, and important aspects of hardware systems and systems **software**. Many of the subject areas of **computer** science deal with **software** products.

Changing Nature of Software

The software is instruction or computer program that when executed provide desired features, function, and performance

Characteristic of software:

There is some characteristic of software which is given below:

- **Functionality**
- **Reliability**
- **Usability**
- **Efficiency**
- **Maintainability**
- **Portability**

Changing Nature of Software :

Nowadays, seven broad categories of computer software present continuing challenges for software engineers .which is given below:

- **System Software:**
- **Application Software:**
- **Engineering and Scientific Software:**
- **Embedded Software:**
- **Product-line Software:**
- **Web Application:**
- **Artificial Intelligence Software:**

1) System Software:

System software is a collection of programs which are written to service other programs.

2) Application Software

Application software is defined as programs that solve a specific business need.

3) Engineering and Scientific Software:

This software is used to facilitate the engineering function and task.

4) Embedded Software

Embedded software resides within the system or product and is used to implement and control feature and function for the end-user and for the system itself.

5) Product-line Software:

Designed to provide a specific capability for use by many different customers, product line software can focus on the limited and esoteric marketplace or address the mass consumer market.

6) Web Application:

It is a client-server computer program which the client runs on the web browser. In their simplest form, Web apps can be little more than a set of linked hypertext files that present information using text and limited graphics.

7) Artificial Intelligence Software:

Application within this area includes robotics, expert system, pattern recognition, artificial neural network, theorem proving and game playing.

Software Process.

A **software process** (also knows as **software** methodology) is a set of related activities that leads to the production of the **software**. These activities may involve the **development** of the **software** from the scratch, or, modifying an existing system.

Any software process must include the following four activities:

1) **Software specification** (or requirements engineering): Define the main functionalities of the software and the constrains around them. **software requirements specifications can help prevent software project failure.** The software requirements specification document lists sufficient and necessary requirements for the project development.

2) **Software design and implementation:** The software is to be designed and programmed.

3) **Software verification and validation:** The software must conforms to it's specification and meets the customer needs.

4) **Software evolution** (software maintenance): The software is being modified to meet customer and market requirements changes.

Software Process Framework

Framework is a Standard way to build and deploy applications. ***Software Process Framework is a foundation of complete software engineering process***. Software process framework includes all set of umbrella activities. **It also includes number of framework activities that are applicable to all software projects.**

A generic process framework encompasses five activities which are given below one by one:

Communication:
In this activity, heavy communication with customers and other stakeholders, requirement gathering is done.

Planning:
In this activity, we discuss the technical related tasks, work schedule, risks, required resources etc.

Modeling:
Modelling is about building representations of things in the 'real world'.In modelling activity, a product's model is created in order to better understanding and requirements.

Construction:
In software engineering, construction is the application of set of procedures that are needed to assemble the product. In this activity, we generate the code and test the product in order to make better product.

Deployment:
In this activity, a complete or non-complete products or software are represented to the customers to evaluate and give feedback. on the basis of their feedback we modify the products for supply better product.

Umbrella Activities

Software project tracking and control: When plan, tasks, models all have been done then a network of software engineering tasks that will enable to get the job done on time will have to be created.

Formal technical reviews: This includes reviewing the techniques that has been used in the project.

Software quality assurance: This is very important to ensure the quality measurement of each part to ensure them.

Software configuration management: Software configuration management (SCM) is a set of activities designed to control change by identifying the work products that are likely to change, establishing relationships among them, defining mechanisms for managing different versions of these work products.

Document preparation and production: All the project planning and other activities should be hardly copied and the production get started here.

Re-usability management: This includes the backing up of each part of the software project they can be corrected or any kind of support can be given to them later to update or upgrade the software at user/time demand.

Measurement & Metrics: This will include all the measurement of every aspects of the software project.

Risk management: Risk management is a series of steps that help a software team to understand and manage uncertainty. It’s a really good idea to identify it, assess its probability of occurrence, estimate its impact, and establish a contingency plan that— ‘should the problem actually occur’.

Prescriptive Process Models

The name 'prescriptive' is given because the model prescribes a set of activities, actions, tasks, quality assurance and change the mechanism for every project.

There are three types of prescriptive process models. They are:

1. The Waterfall Model
2. Incremental Process model
3. RAD model

1. The Waterfall Model

- The waterfall model is also called as 'Linear sequential model' or 'Classic life cycle model'.
- In this model, each phase is fully completed before the beginning of the next phase.
- This model is used for the small projects.
- In this model, feedback is taken after each phase to ensure that the project is on the right path.
- Testing part starts only after the development is complete.

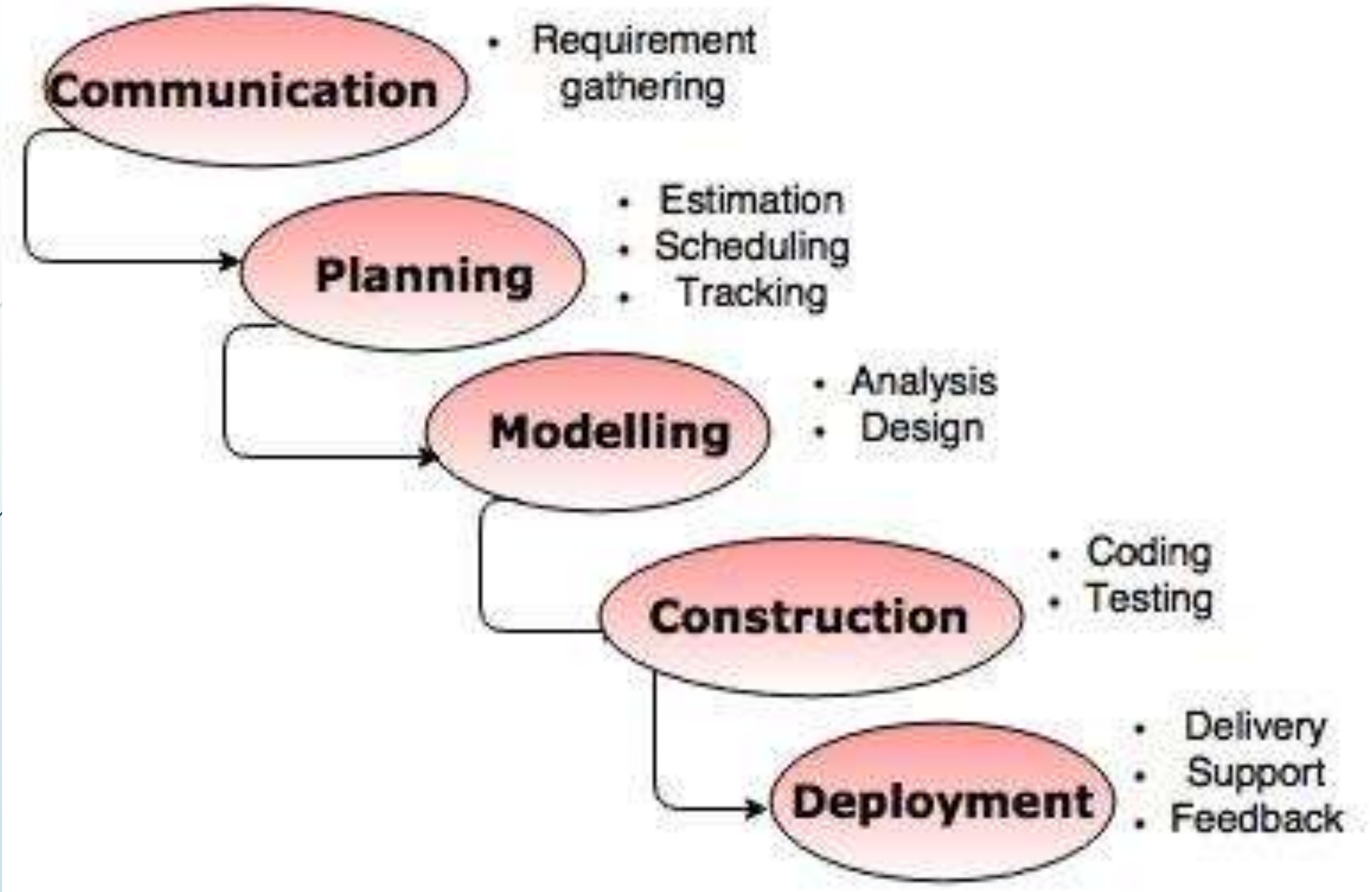
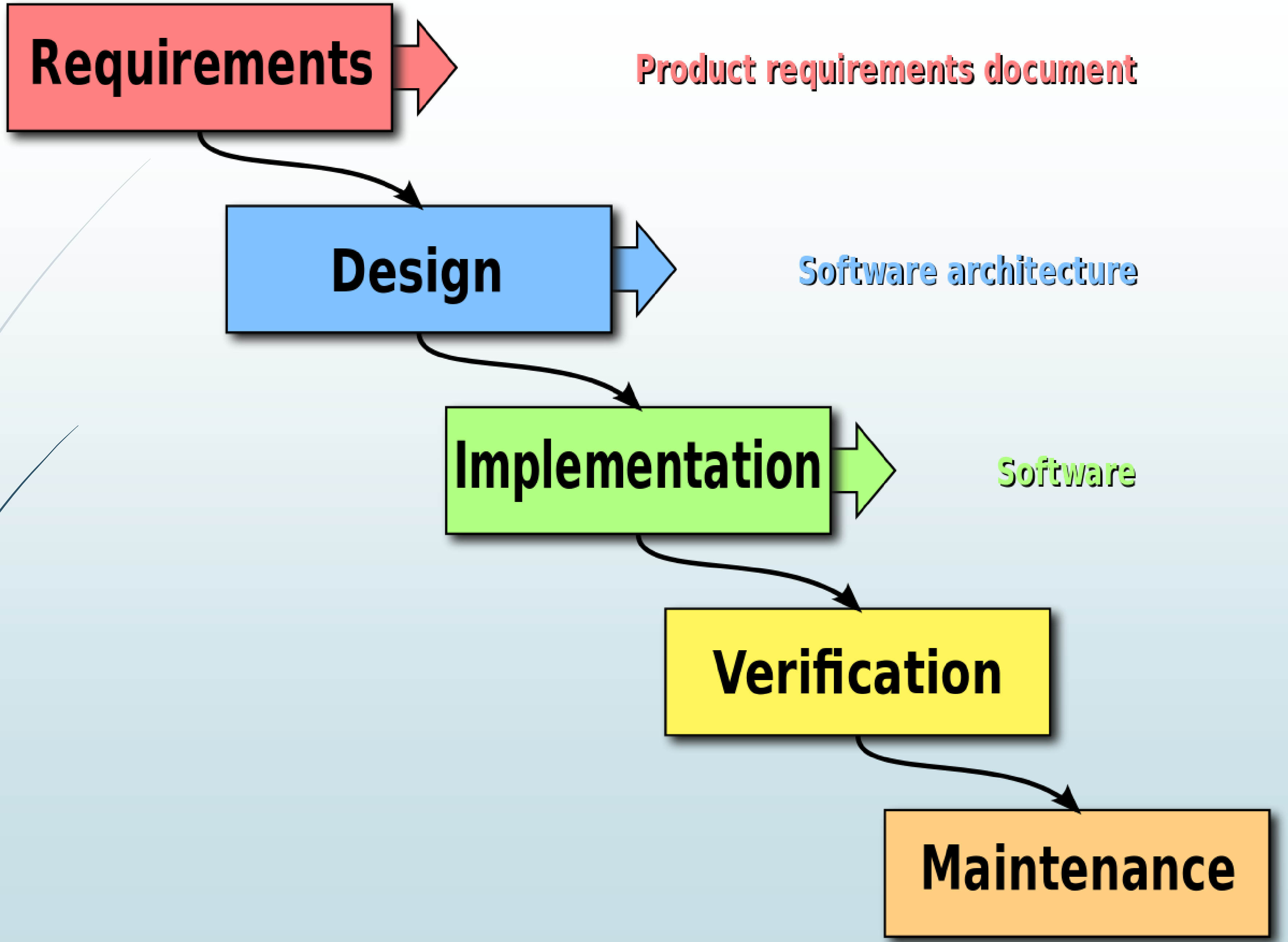


Fig. - The Waterfall model


Requirements
Product requirements document
Design
Software architecture
Implementation
Software
Verification
Maintenance

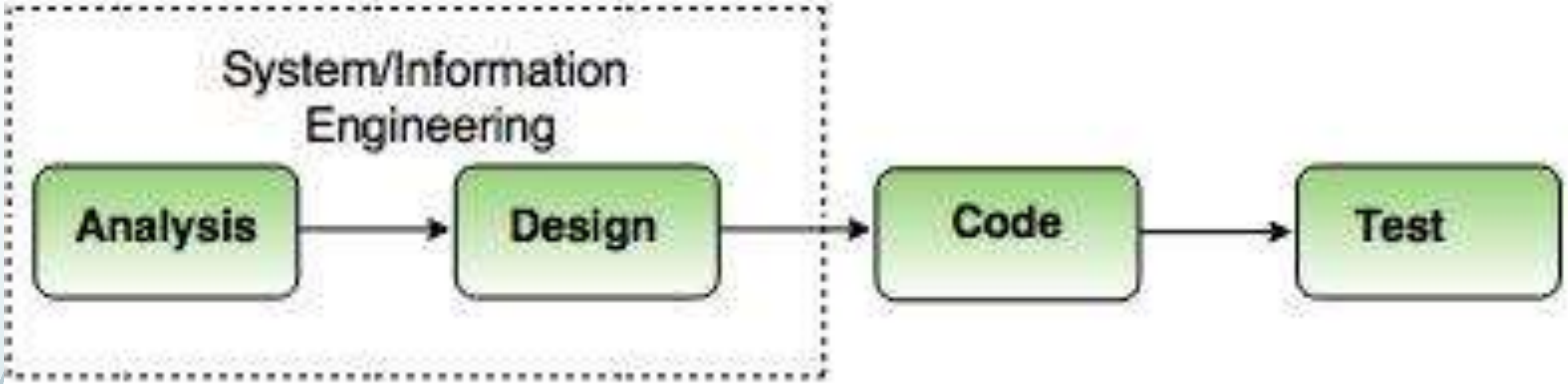


Fig. - The linear sequential model

Advantages of waterfall model

- **The waterfall model is simple and easy to understand, implement, and use.**
- **All the requirements are known at the beginning of the project, hence it is easy to manage.**
- **It avoids overlapping of phases because each phase is completed at once.**
- **This model works for small projects because the requirements are understood very well.**
- **This model is preferred for those projects where the quality is more important as compared to the cost of the project.**

Disadvantages of the waterfall model

- **This model is not good for complex and object oriented projects.**
- **It is a poor model for long projects.**
- **The problems with this model are uncovered, until the software testing.**
- **The amount of risk is high.**

2. Incremental Process model

- The incremental model ***combines the elements of waterfall model and they are applied in an iterative fashion(repetitive).***
- The first increment in this model is generally a core product.
- Each increment builds the product and submits it to the customer for any suggested modifications.
- The ***next increment implements on the customer's suggestions and add additional requirements in the previous increment.***
- ***This process is repeated until the product is finished.***
- For example, the ***word-processing software is developed using the incremental model.***

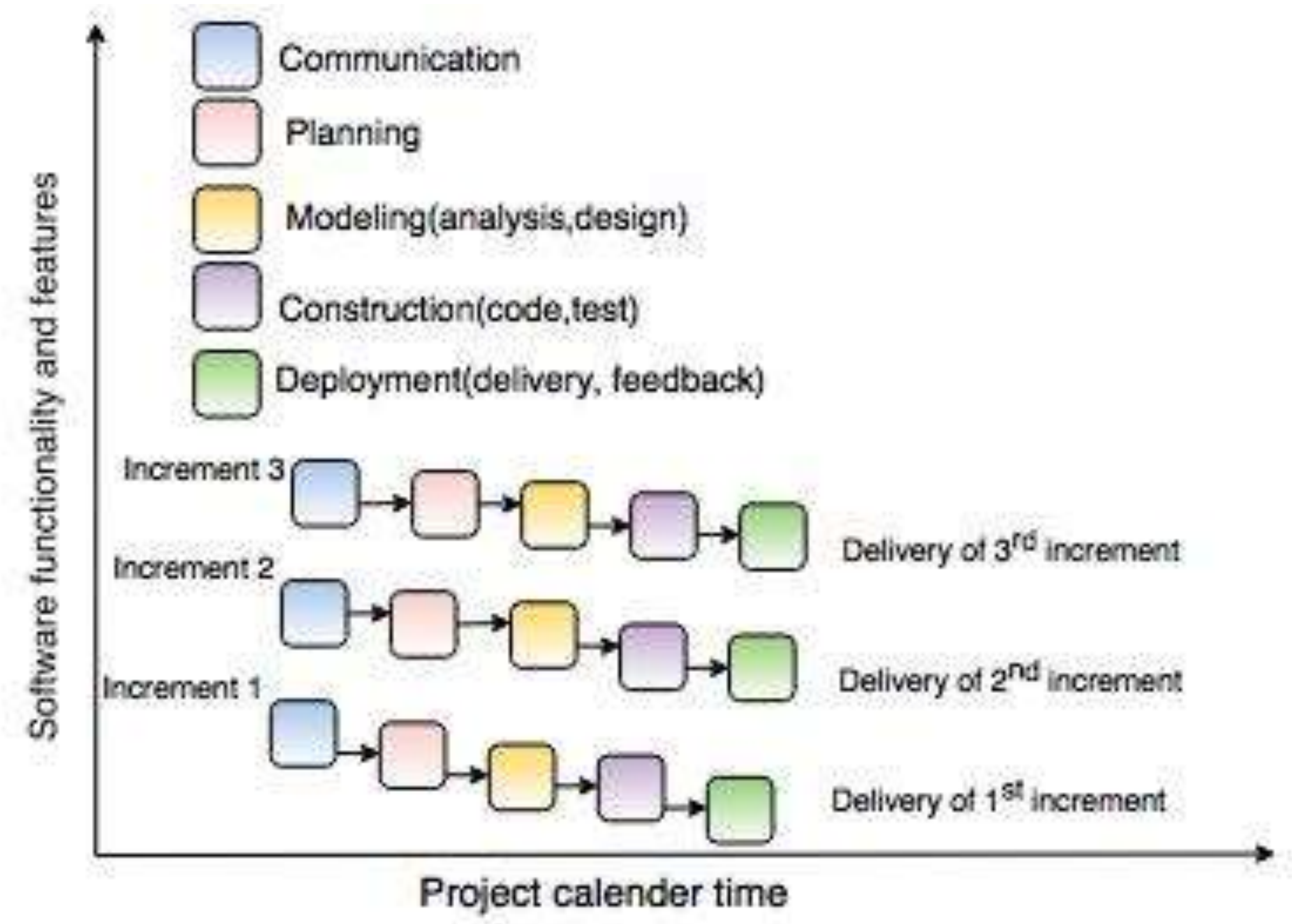
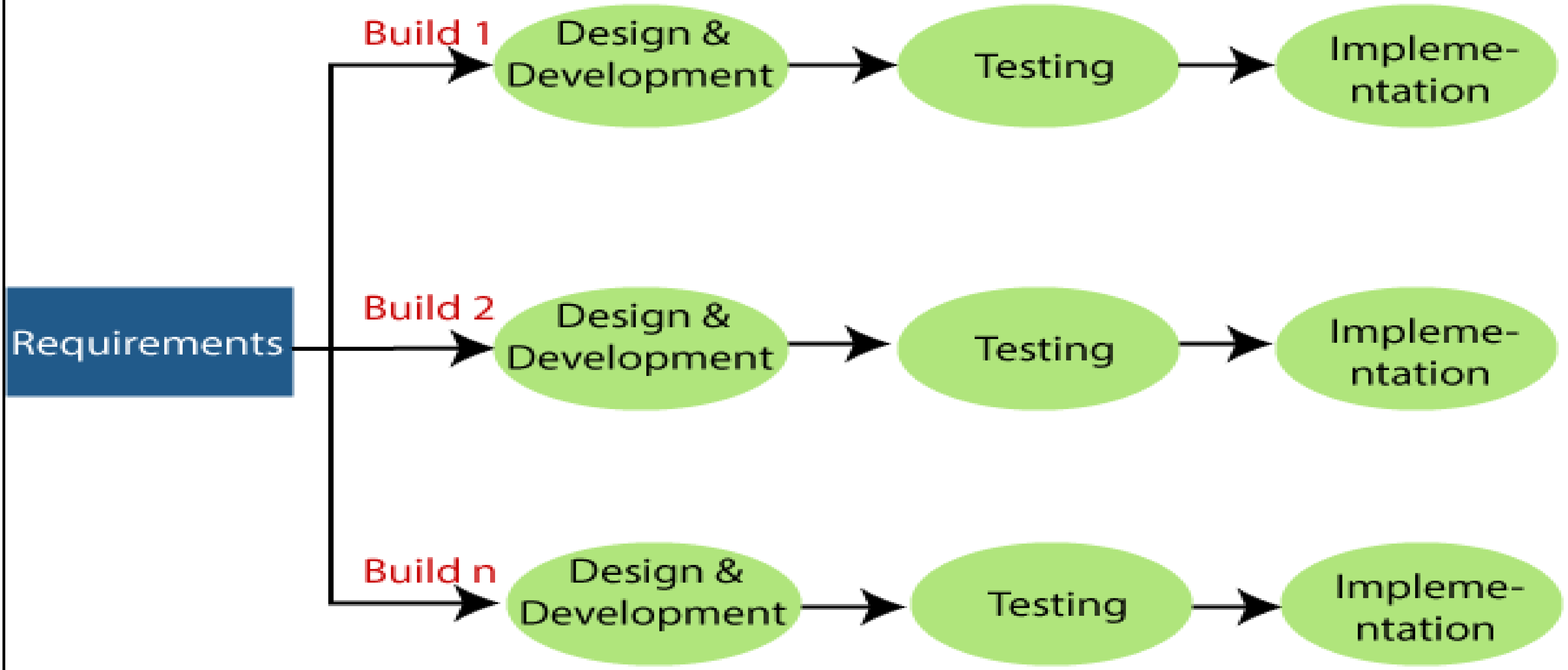

Communication
Planning
Modeling(analysis,design)
Construction(code,test)
Deployment(delivery, feedback)
Increment 3
Increment 2
Increment 1
Delivery of 3rd increment
Delivery of 2nd increment
Delivery of 1st increment
Software functionality and features
Project calender time

Fig. - Incremental Process Model

Incremental Model



Advantages of incremental model

- This model is flexible because the cost of development is low and initial product delivery is faster.
- It is easier to test and debug during the smaller iteration.
- The working software generates quickly and early during the software life cycle.
- The customers can respond to its functionalities after every increment.

Disadvantages of the incremental model

- The cost of the final product may cross the cost estimated initially.
- This model requires a very clear and complete planning.
- The planning of design is required before the whole system is broken into small increments.
- The demands of customer for the additional functionalities after every increment causes problem during the system architecture.

Evolutionary Process Models

- ✓ *Evolutionary models are iterative type models.*
- ✓ *They allow to develop more complete versions of the software.*

Following are the evolutionary process models.

1. The prototyping model
2. The spiral model
3. Concurrent development model

1. The Prototyping model

- Prototype model is a set of general objectives for software.
- It does not identify the requirements like detailed input, output.
- It is software working model of limited functionality.
- In this model, working programs are quickly produced.

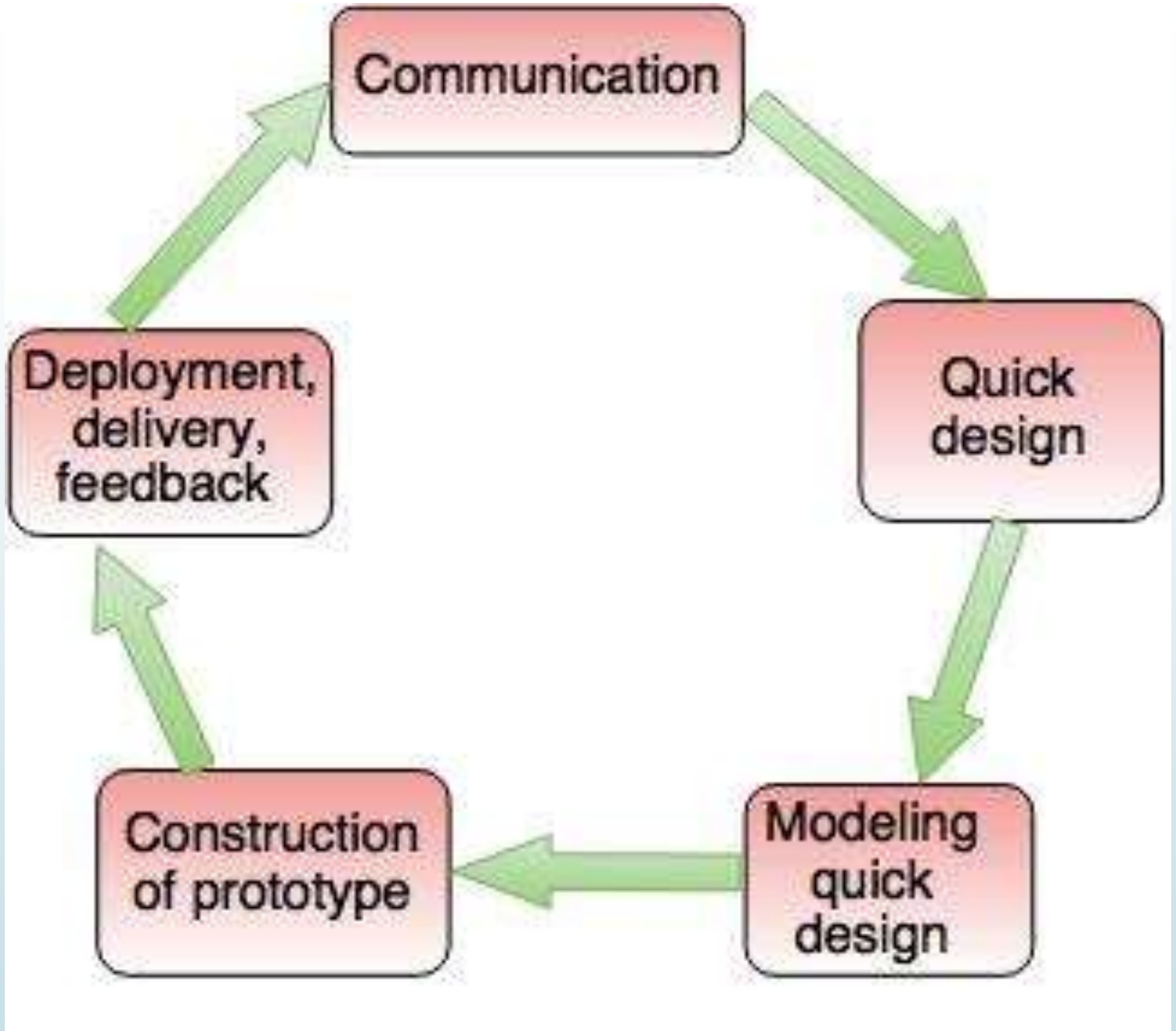


Fig. - The Prototyping Model

The different phases of Prototyping model are:

1. Communication

In this phase, developer and customer meet and discuss the overall objectives of the software.

2. Quick design

- Quick design is implemented when requirements are known.
- It includes only the important aspects like input and output format of the software.
- It focuses on those aspects which are visible to the user rather than the detailed plan.
- It helps to construct a prototype.

3. Modeling quick design

- This phase gives the clear idea about the development of software because the software is now built.
- It allows the developer to better understand the exact requirements.

4. Construction of prototype

The prototype is evaluated by the customer itself.

5. Deployment, delivery, feedback

If the user is not satisfied with current prototype then it refines according to the requirements of the user.
The process of refining the prototype is repeated until all the requirements of users are met.
When the users are satisfied with the developed prototype then the system is developed on the basis of final prototype.

Advantages of Prototyping Model

- Prototype model need not know the detailed input, output, processes, adaptability of operating system and full machine interaction.
- In the development process of this model users are actively involved.
- The development process is the best platform to understand the system by the user.
- Errors are detected much earlier.
- Gives quick user feedback for better solutions.
- It identifies the missing functionality easily. It also identifies the confusing or difficult functions.

Disadvantages of Prototyping Model:

- The client involvement is more and it is not always considered by the developer.
- It is a slow process because it takes more time for development.
- Many changes can disturb the rhythm of the development team.
- It is a thrown away prototype when the users are confused with it.

Concurrent Models

- The communication activity has completed in the first iteration and exits in the awaiting changes state.
- The modeling activity completed its initial communication and then go to the underdevelopment state.
- If the customer specifies the change in the requirement, then the modeling activity moves from the under development state into the awaiting change state.
- The concurrent process model activities moving from one state to another state.

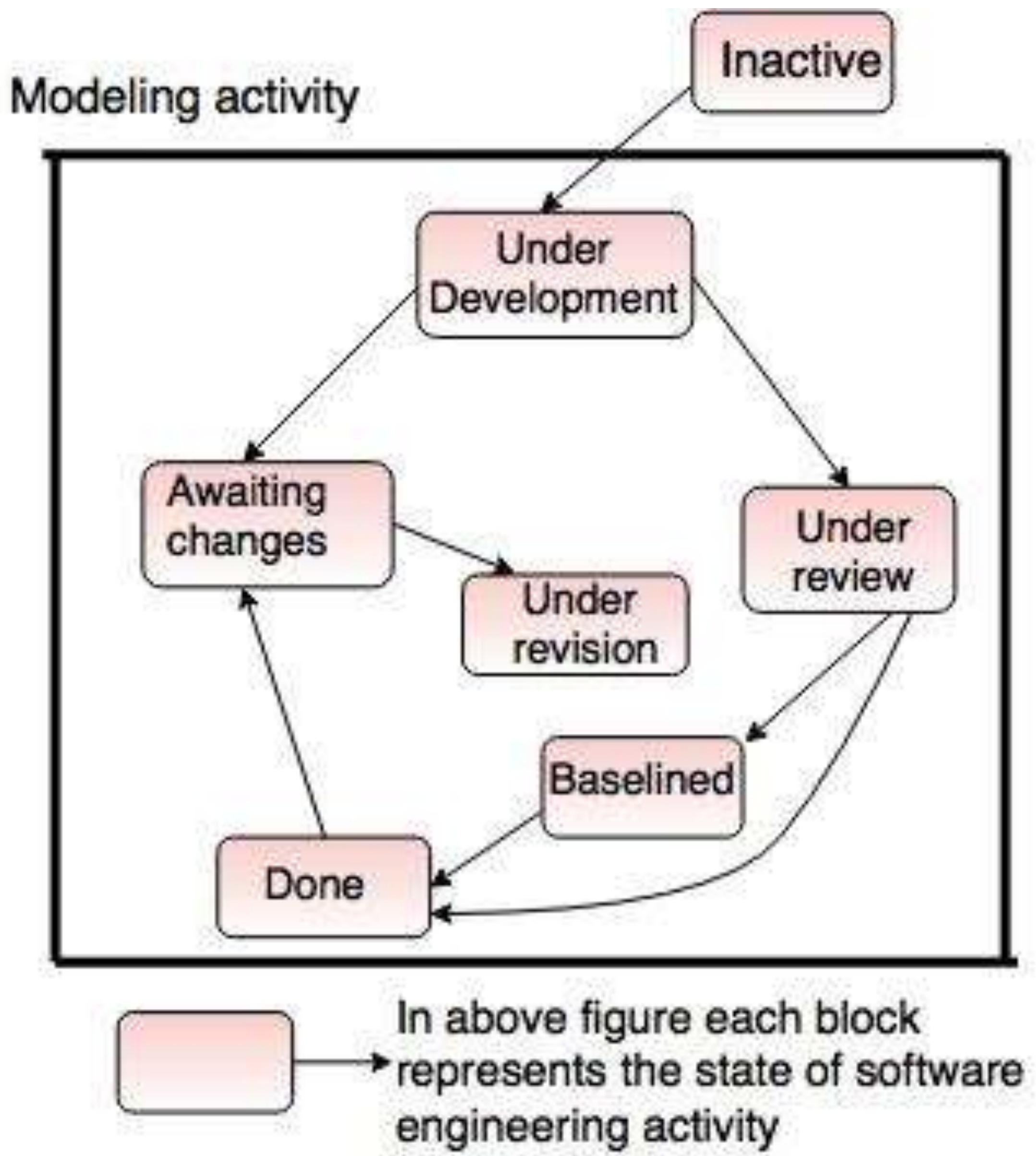
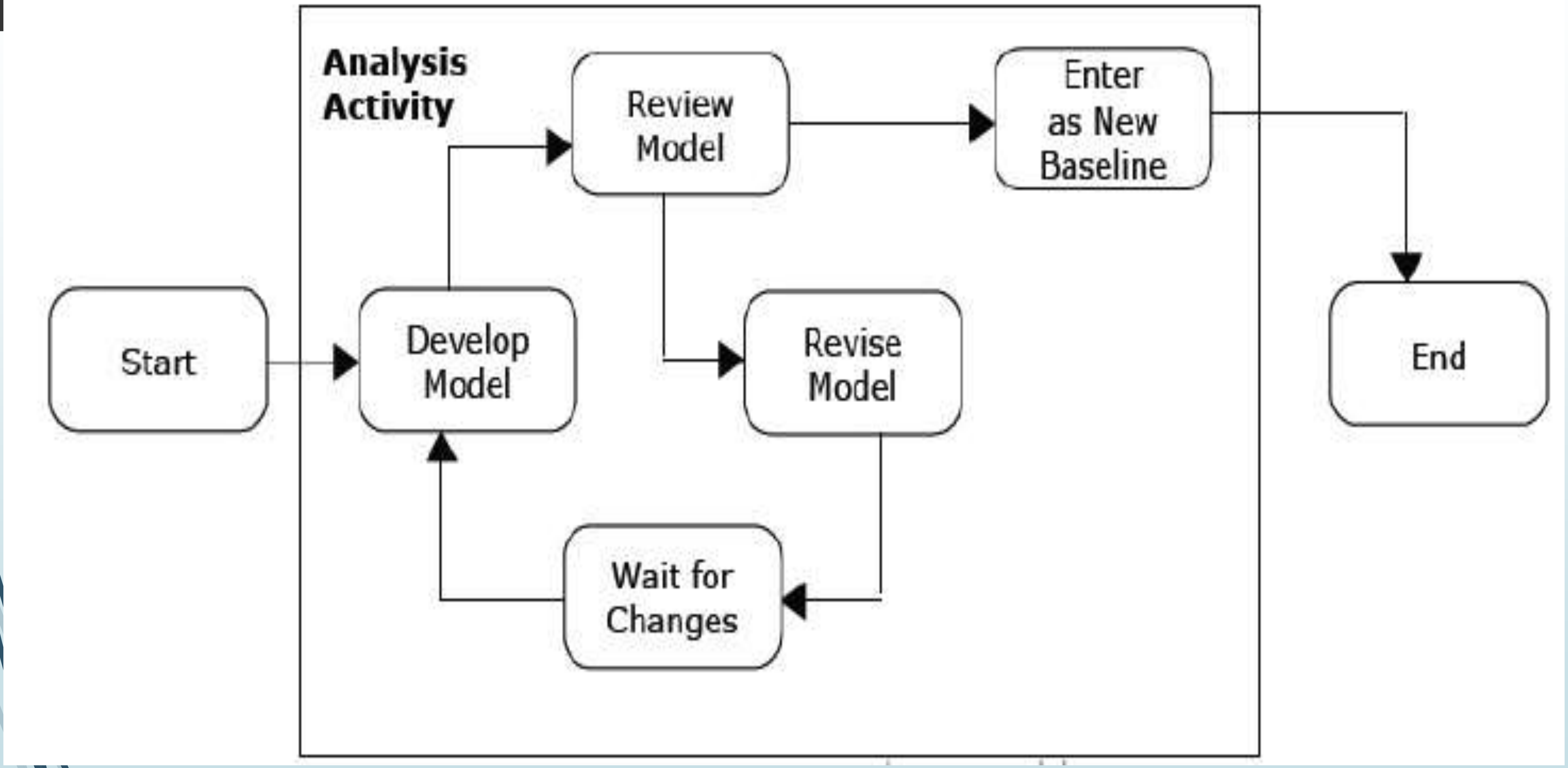
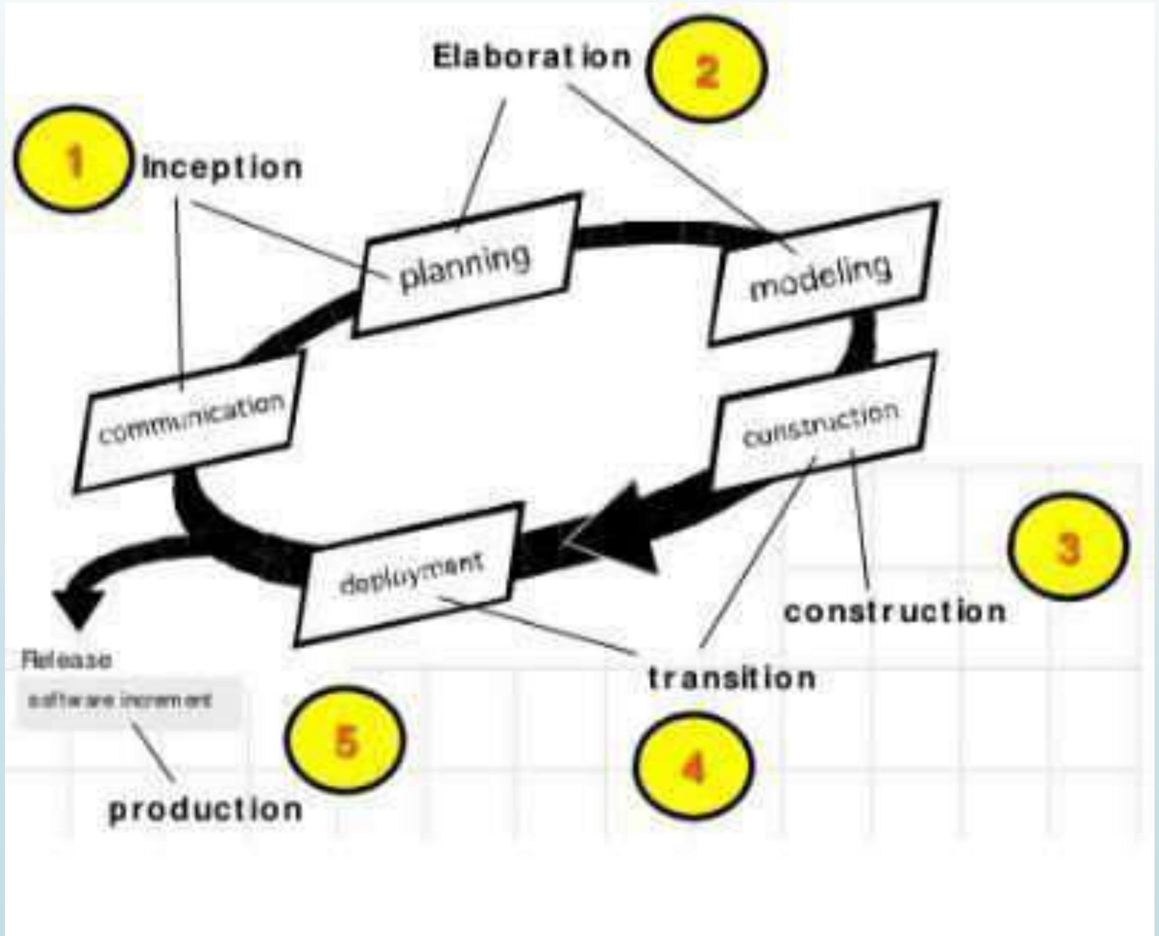


Fig. - One element of the concurrent process model


Analysis Activity
Start
Develop Model
Review Model
Enter as New Baseline
Revise Model
Wait for Changes
End

The Unified Process (UP)

The Unified Process (UP), or Unified Software Development Process, is a iterative and incremental software development framework from which a customized process can be defined. The framework contains many components and has been modified a number of times to create several variations.



Unified Process Model Phases

1.Inception: The inception phase is similar to the requirements collection and analysis stage of the waterfall model of software development. In this phase, you'd **collect requirements from the customer and analyze the project's feasibility, its cost, risks, and profits.**

2.Elaboration: In this phase, you'd be expanding upon the activities undertaken in the inception phase. **The major goals of this phase include creating fully functional requirements (use-cases) and creating a detailed architecture for fulfillment of the requirements**. You'd also prepare a business case document for the customer.

3.Construction: In this phase, **you'd be writing actual code and implementing the features for each iteration**. You'd be rolling out the first iteration of the software depending on the key use-cases that make up the core functionalities of the software system.

4.Transition: In this phase, **you'd be rolling out the next iterations to the customer and fixing bugs for previous releases**. You would also deploy builds of the software to the customer.

Chapter End

Thank you……