Software Engineering Introduction

- Computer software is a product or program code developed by software engineers.
- The applications of computer software are: Telecommunication, military, medical sciences, online shopping, office products, IT industry etc.
- A Software consists of data and the related documents.
- The software is the key element in all computer based systems and products.
- The main purpose behind software engineering is to give a framework for building a software with best quality.

Software engineering definitions

- The establishment and use of sound engineering principles in order to obtain economical software that is reliable and works efficiently on real machines.
- Software engineering is a systematic and disciplined approach towards the development of the software operation and maintenance.
- Software engineering is an engineering branch associated with the development of software product using well-defined scientific principles, methods and procedures.

Characteristics of a software

- Software should achieve a good quality in design and meet all the specifications of the customer.
- Software does not wear out i.e. it does not lose the material.
- Software should be inherently complex.
- Software must be efficient i.e. the ability of the software to use system resources in an effective and efficient manner.
- Software must be integral i.e. it must prevent from unauthorized access to the software or data.

Software engineering - Layered technology

- Software engineering is a fully layered technology.
- To develop a software, we need to go from one layer to another.
- All these layers are related to each other and each layer demands the fulfilment of the previous layer.

Software Engineering Layers

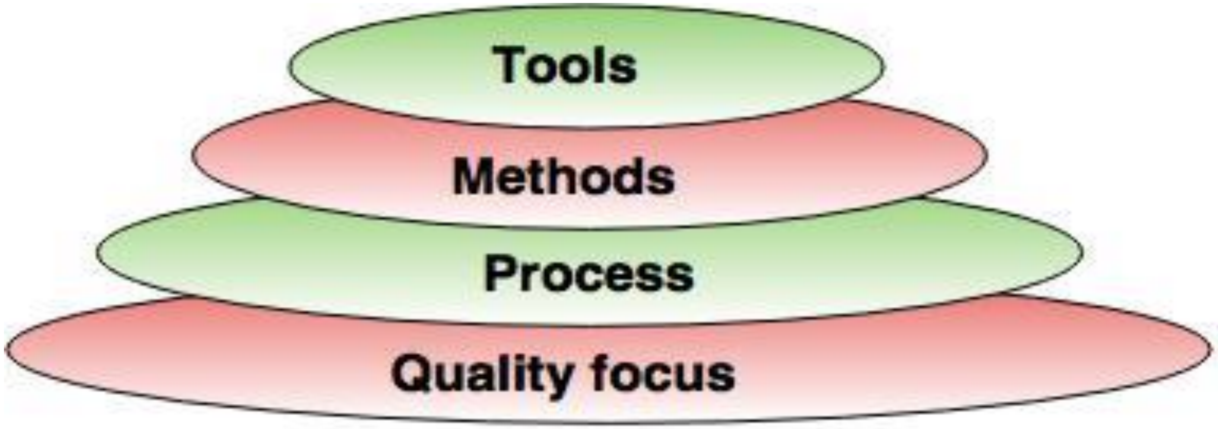


Fig. - Software Engineering Layers

The layered technology consists of:

1. Quality focus

   The characteristics of good quality software are: Correctness of the functions required to be performed by the software.

   Maintainability of the software

   Integrity i.e. providing security so that the unauthorized user cannot access information or data.

   Usability i.e. the efforts required to use or operate the software.

2. Process **It** is the base layer or foundation layer for the software engineering.

   The software process is the key to keep all levels together.

   It defines a framework that includes different activities and tasks.

   In short, it covers all activities, actions and tasks required to be carried out for software development.

3. **Methods** The method provides the answers of all 'how-to' that are asked during the process.

   It provides the technical way to implement the software.

   It includes collection of tasks starting from communication, requirement analysis, analysis and design modelling, program construction, testing and support.

4. **Tools** The software engineering tool is an automated support for the software development.

   The tools are integrated i.e the information created by one tool can be used by the other tool.

For example: The Microsoft publisher can be used as a web designing tool.

Software Process Framework

- The process of framework defines a small set of activities that are applicable to all types of projects.
- The software process framework is a collection of task sets.
- Task sets consist of a collection of small work tasks, project milestones, work productivity and software quality assurance points.

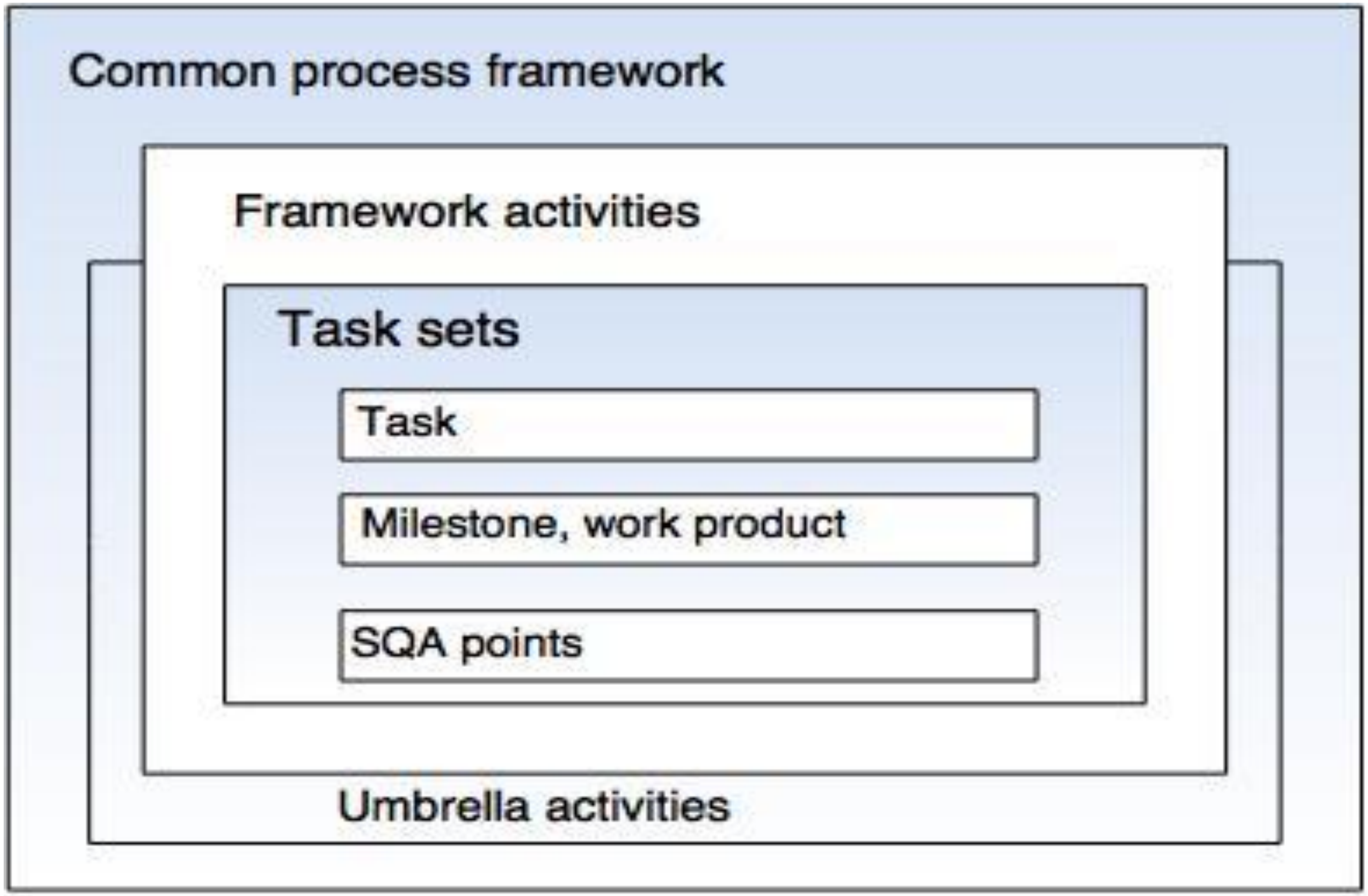


Fig.- A software process framework

Umbrella activities

Typical umbrella activities are:

1. **Software project tracking and control** In this activity, the developing team accesses project plan and compares it with the predefined schedule.

   If these project plans do not match with the predefined schedule, then the required actions are taken to maintain the schedule.

2. **Risk management** Risk is an event that may or may not occur.

   If the event occurs, then it causes some unwanted outcome. Hence, proper risk management is required.

3. **Software Quality Assurance (SQA)** SQA is the planned and systematic pattern of activities which are required to give a guarantee of software quality. **For example,** during the software development meetings are conducted at every stage of development to find out the defects and suggest improvements to produce good quality software.

4. **Formal Technical Reviews (FTR)** FTR is a meeting conducted by the technical staff.

   The motive of the meeting is to detect quality problems and suggest improvements.

   The technical person focuses on the quality of the software from the customer point of view.

5. **Measurement** It consists of the effort required to measure the software.

The software cannot be measured directly. It is measured by direct and indirect measures.

Direct measures like cost, lines of code, size of software etc.

Indirect measures such as quality of software which is measured by some other factor. Hence, it is an indirect measure of software.

6. **Software Configuration Management (SCM)**It manages the effect of change throughout the software process.

7. **Reusability management** It defines the criteria for reuse the product.

The quality of software is good when the components of the software are developed for certain application and are useful for developing other applications.

8. **Work product preparation and production It** consists of the activities that are needed to create the documents, forms, lists, logs and user manuals for developing a software.

Generic Process Model

A software process is a collection of various activities.

There are five generic process framework activities:

1.Communication:

The software development starts with the communication between customer and developer.

2. Planning:

It consists of complete estimation, scheduling for project development and tracking

3. **Modelling**: **Modelling** consists of complete requirement analysis and the design of the project like algorithm, flowchart etc.

The algorithm is the step-by-step solution of the problem and the flow chart shows a complete flow diagram of a program.

4. **Construction**: **Construction** consists of code generation and the testing part.

Coding part implements the design details using an appropriate programming language.

Testing is to check whether the flow of coding is correct or not.

Testing also check that the program provides desired output.

5. **Deployment: Deployment** step consists of delivering the product to the customer and take feedback from them.

If the customer wants some corrections or demands for the additional capabilities, then the change is required for improvement in the quality of the software.

Prescriptive Process Models

The following framework activities are carried out irrespective of the process model chosen by the organization.

1. Communication
2. Planning
3. Modeling
4. Construction
5. Deployment

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.**

  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.

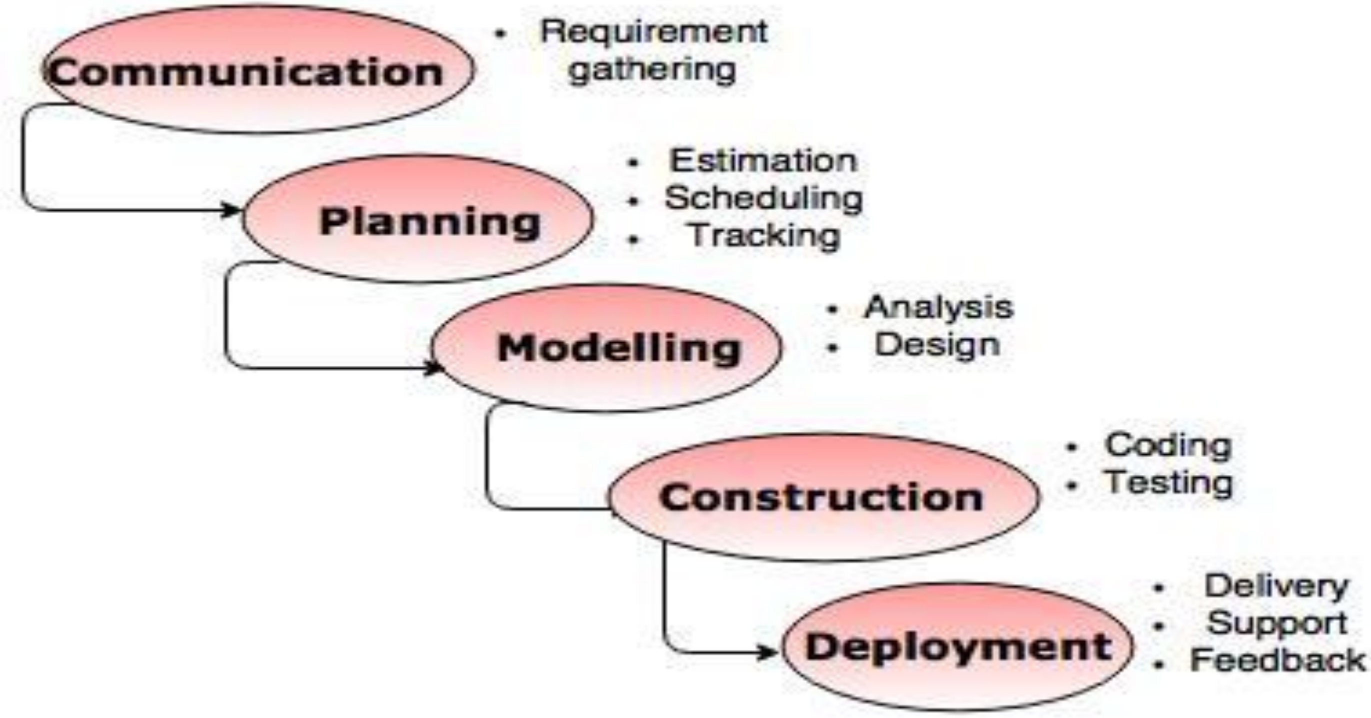


Fig. - The Waterfall model

NOTE: The description of the phases of the waterfall model is same as that of the process model.

- **An alternative design for 'linear sequential model' is as follows:**

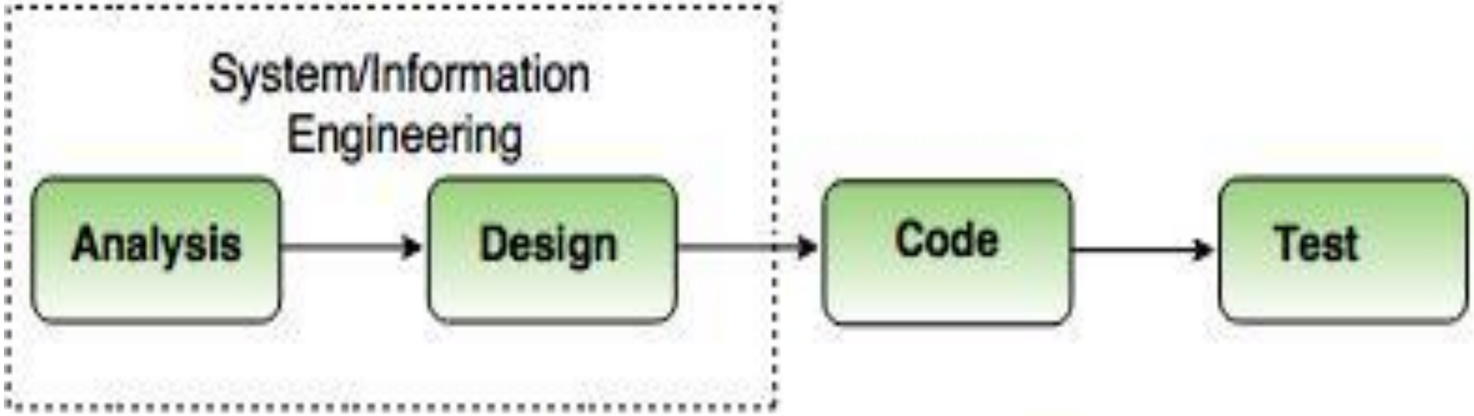


Fig. - The linear sequential model

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.
- 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.

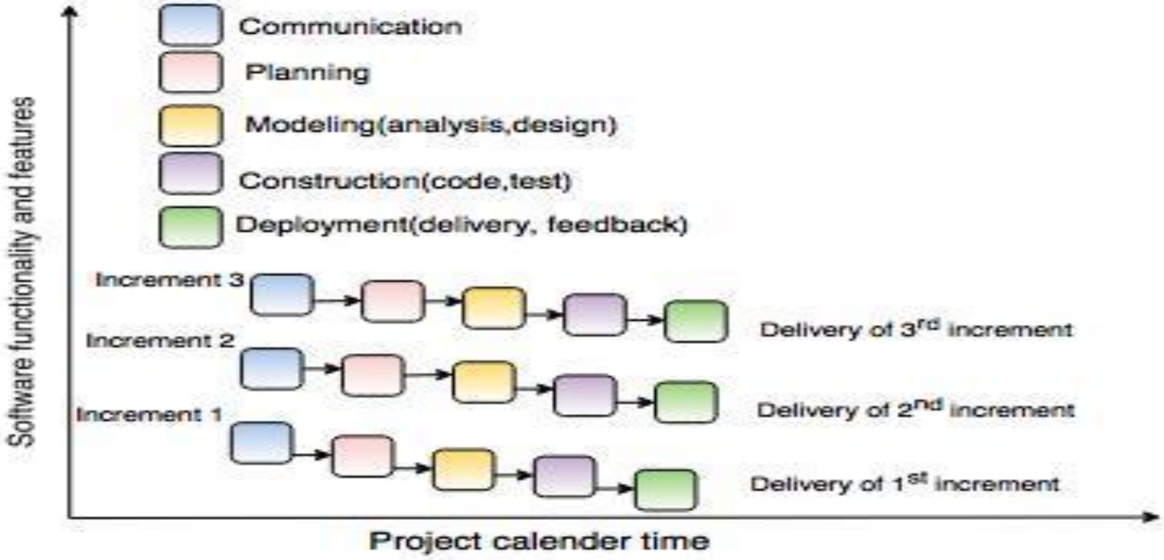

Communication
Planning
Modeling(analysis,design)
Construction(code,test)
Deployment(delivery, feedback)
Increment 3
Increment 2
Increment 1
Delivery of 3rd increment
Delivery of 2nd increment
Delivery of 1st increment
Software functionality and features
Project calender time

Fig. - Incremental Process Model

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.

RAD model

- RAD is a Rapid Application Development model.
- Using the RAD model, software product is developed in a short period of time.
- The initial activity starts with the communication between customer and developer.
- Planning depends upon the initial requirements and then the requirements are divided into groups.
- Planning is more important to work together on different modules.

The RAD model phases:

1. **Business Modeling** Business modeling consist of the flow of information between various functions in the project.

   For example what type of information is produced by every function and which are the functions to handle that information.

   A complete business analysis should be performed to get the essential business information.

2. **Data modeling** The information in the business modeling phase is refined into the set of objects and it is essential for the business.

   The attributes of each object are identified and define the relationship between objects.

3. **Process modeling** The data objects defined in the data modeling phase are changed to fulfil the information flow to implement the business model.

   The process description is created for adding, modifying, deleting or retrieving a data object.

4. **Application generation** In the application generation phase, the actual system is built.

   To construct the software the automated tools are used.

5. **Testing and turnover** The prototypes are independently tested after each iteration so that the overall testing time is reduced.

   The data flow and the interfaces between all the components are Fully tested.

Hence, most of the programming components are already tested.

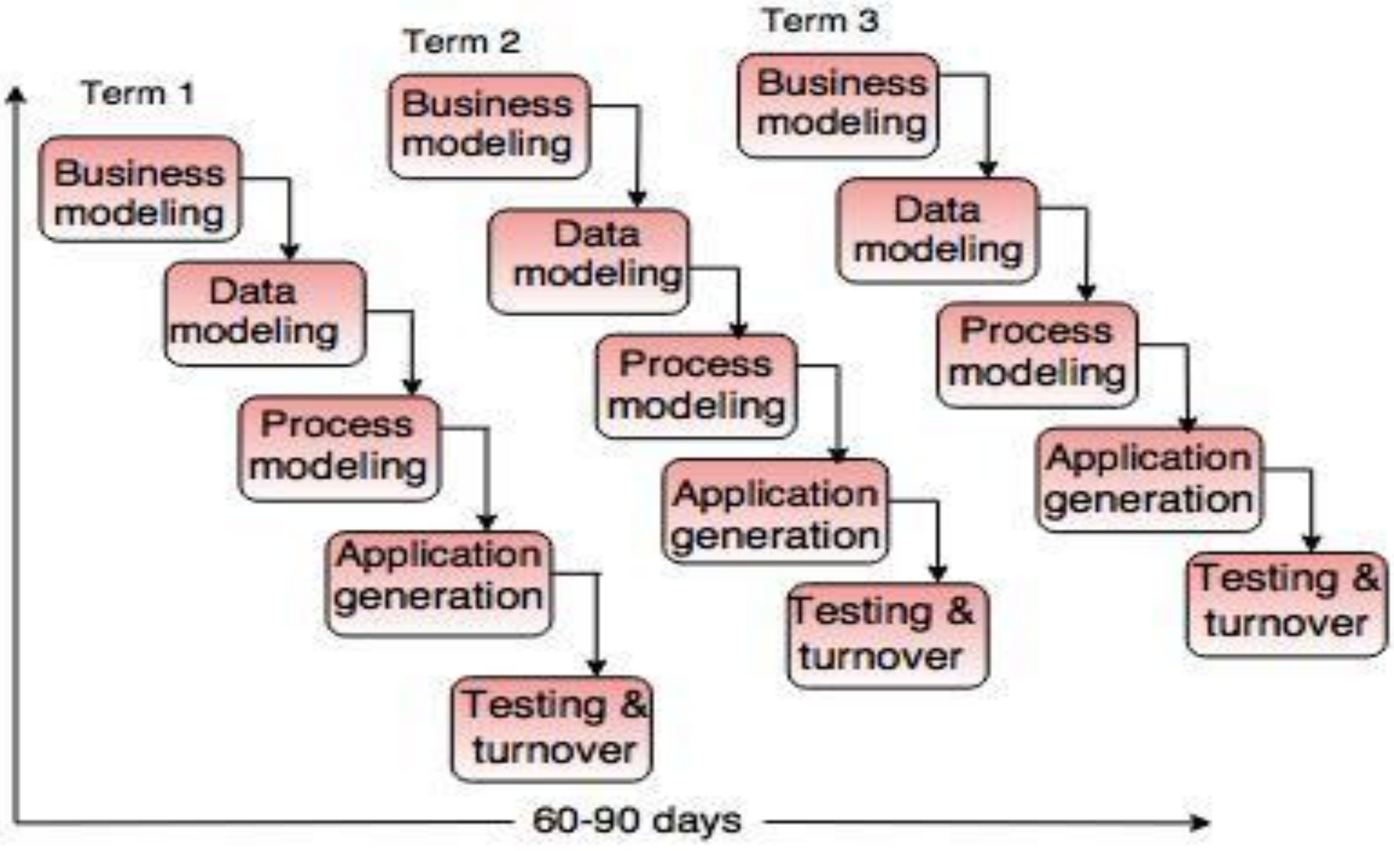


Fig. - RAD Model

Evolutionary Process Models in Software Engineering

- 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 is defined as first or preliminary form using which other forms are copied or derived.
- 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.

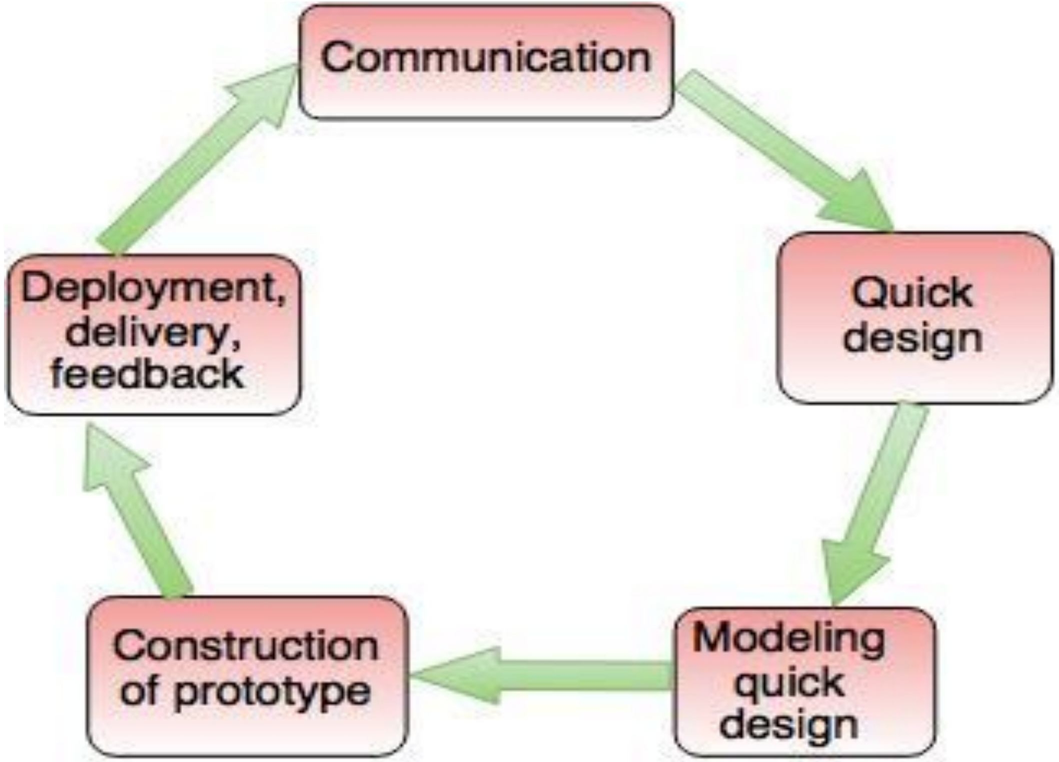


Fig. - The Prototyping Model

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.

2. The Spiral model

- Spiral model is a risk driven process model.
- It is used for generating the software projects.
- In spiral model, an alternate solution is provided if the risk is found in the risk analysis, then alternate solutions are suggested and implemented.
- It is a combination of prototype and sequential model or waterfall model.
- In one iteration all activities are done, for large project's the output is small.

The framework activities of the spiral model are as shown in the following figure.

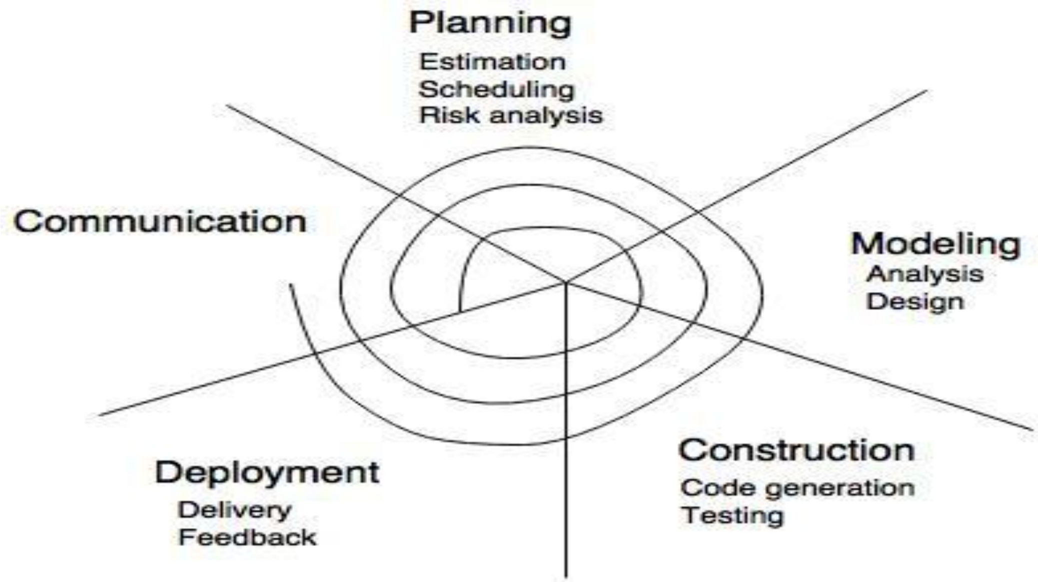


Fig. - The Spiral Model

NOTE: The description of the phases of the spiral model is same as that of the process model.

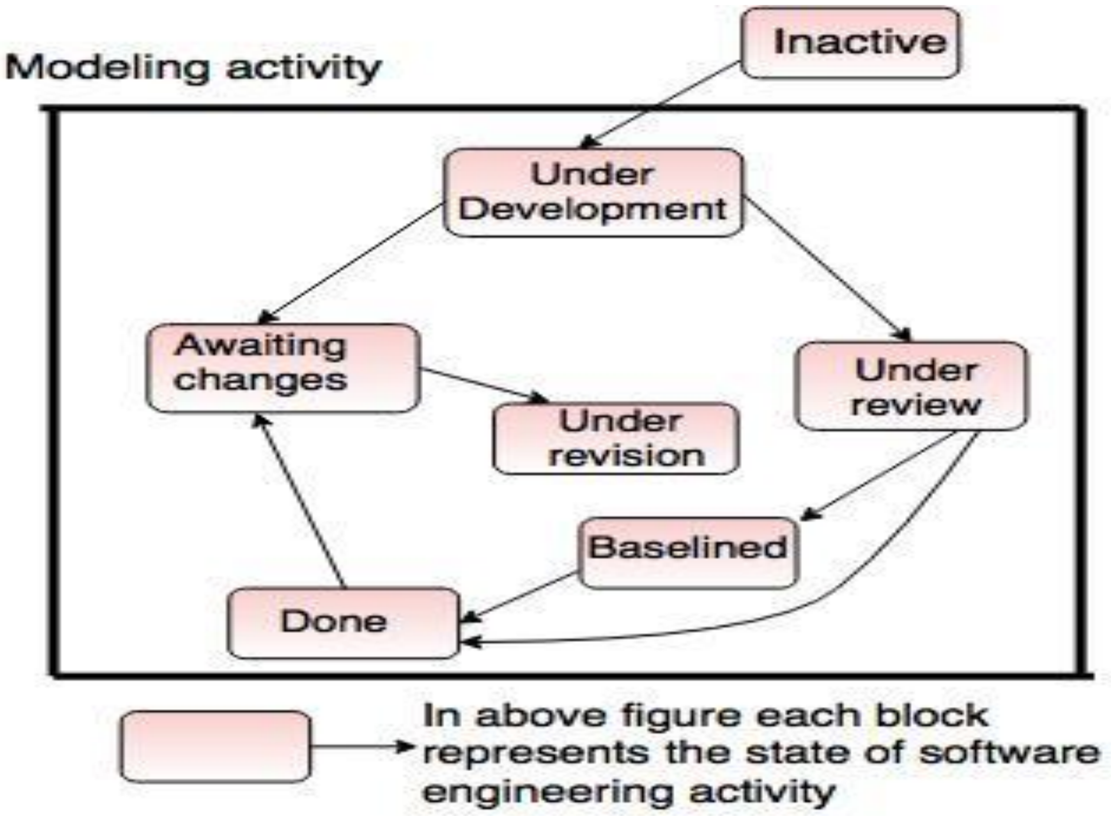
Advantages of Spiral Model

- It reduces high amount of risk.
- It is good for large and critical projects.
- It gives strong approval and documentation control.
- In spiral model, the software is produced early in the life cycle process.
- **Disadvantages of Spiral Model**It can be costly to develop a software model.
- It is not used for small projects.

3. The concurrent development model

- The concurrent development model is called as concurrent model.
- 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.
-

Concurrent Process Model



In above figure each block represents the state of software engineering activity

Fig. - One element of the concurrent process model

Advantages of the concurrent development model

- This model is applicable to all types of software development processes.
- It is easy for understanding and use.
- It gives immediate feedback from testing.
- It provides an accurate picture of the current state of a project.
- **Disadvantages of the concurrent development model**It needs better communication between the team members. This may not be achieved all the time.
- It requires to remember the status of the different activities.