SOFTARWE ENGINEERING

UNIT-I

Software is a program or set of programs containing instructions which provide desired functionality. And Engineering is the processes of designing and building something that serves a particular purpose and find a cost effective solution to problems.

Dual Role of Software:

1. As a product –
- It delivers the computing potential across network of Hardware.
- It enables the Hardware to deliver the expected functionality.
- It acts as information transformer because it produces, manages, acquires, modifies, displays, or transmits information.

2. As a vehicle for delivering a product –

- It provides system functionality (e.g., payroll system)
- It controls other software (e.g., an operating system)
- It helps build other software (e.g., software tools)

Objectives of Software Engineering:

1. Maintainability –
   It should be feasible for the software to evolve to meet changing requirements.
2. Correctness –
   A software product is correct, if the different requirements as specified in the SRS document have been correctly implemented.
3. Reusability –
   A software product has good reusability, if the different modules of the product can easily be reused to develop new products.
4. Testability –
   Here software facilitates both the establishment of test criteria and the evaluation of the software with respect to those criteria.
5. Reliability –
   It is an attribute of software quality. The extent to which a program can be expected to perform its desired function, over an arbitrary time period.
6. Portability –
   In this case, software can be transferred from one computer system or environment to another.

7. Adaptability –
   In this case, software allows differing system constraints and user needs to be satisfied by making changes to the software.

Software is the set of instructions in the form of programs to govern the computer system and to process the hardware components. To produce a software product the set of activities is used. This set is called a software process.

Components of Software:

There are three components of the software:

1. **Program:**
   A computer program is a list of instructions that tell a computer what to do.
2. **Documentation:**
   Source information about the product contained in design documents, detailed code comments, etc.
3. **Operating Procedures:**
   Set of step-by-step instructions compiled by an organization to help workers carry out complex routine operations.

There are four basic key process activities:

1. **Software Specifications:**
   In this process, detailed description of a software system to be developed with its functional and non-functional requirements.
2. **Software Development**:
   In this process, designing, programming, documenting, testing, and bug fixing is done.
3. **Software Validation**:
   In this process, evaluation software product is done to ensure that the software meets the business requirements as well as the end users needs.
4. **Software Evolution**:
   It is a process of developing software initially, then timely updating it for various reasons.

Software Crisis:

1. **Size and Cost**:
   Day to day growing complexity and expectation out of software. Software are more expensive and more complex.
2. **Quality:**
   Software products must have good quality.
3. **Delayed Delivery:**
   Software takes longer than the estimated time to develop, which in turn leads to cost shooting up.

Software Process Model:

A software process model is an abstraction of the actual process, which is being described. It can also be defined as a simplified representation of a software process. Each model represents a process from a specific perspective. Basic software process models on which different type of software process models can be implemented:

1. **A workflow Model:**
   It is the sequential series of tasks and decisions that make up a business process.
2. **The Waterfall Model:**
   It is a sequential design process in which progress is seen as flowing steadily downwards. Phases in waterfall model:
   - (i) Requirements Specification
   - (ii) Software Design
   - (iii) Implementation
   - (iv) Testing

3. **Dataflow Model:**
   It is diagrammatic representation of the flow and exchange of information within a system.
4. **Evolutionary Development Model:**
   Following activities are considered in this method:
   - (i) Specification
   - (ii) Development
   - (iii) Validation

5. **Role / Action Model:**
   Roles of the people involved in the software process and the activities.

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. Modeling:

- Modeling 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. 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.

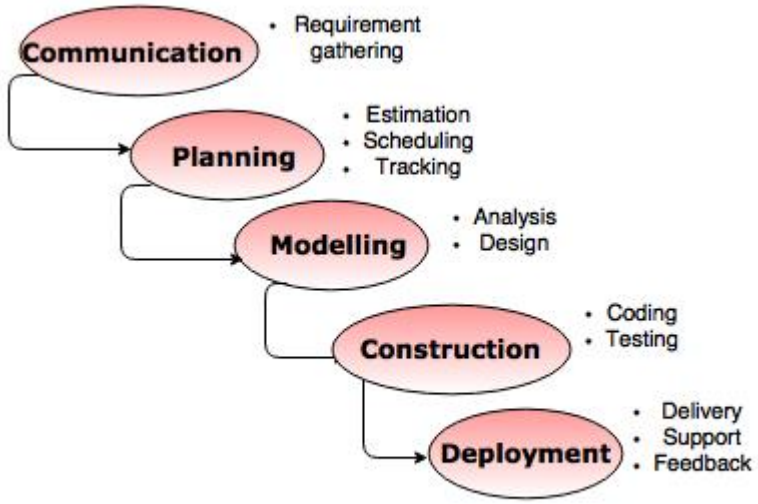

Fig. - The Waterfall model

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:

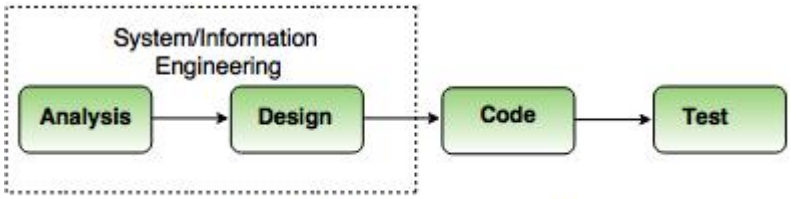

Fig. - The linear sequential model

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.

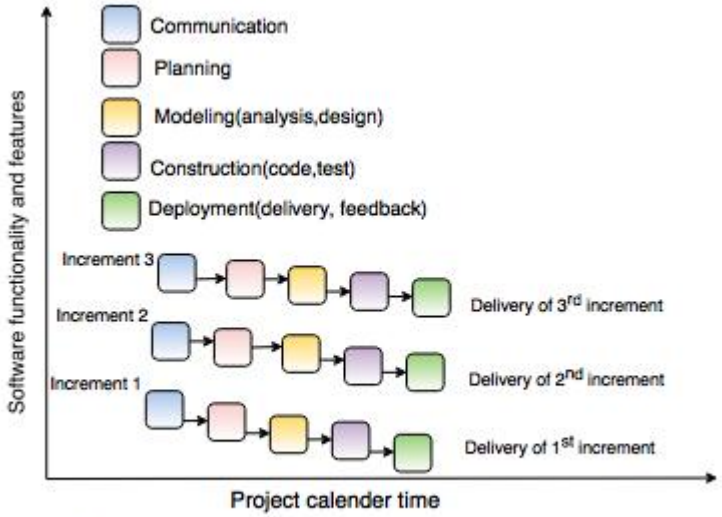

Fig. - Incremental Process Model

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.

3. 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 consist of following 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 are fully tested. Hence, most of the programming components are already tested.

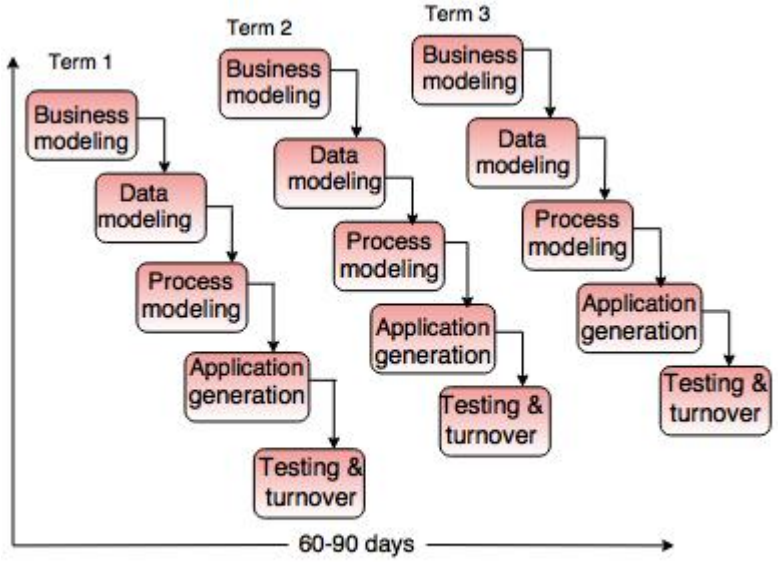

Fig. - RAD Model

Agile principles

- The highest priority of this process is to satisfy the customer.
- Acceptance of changing requirement even late in development.
- Frequently deliver a working software in small time span.
- Throughout the project business people and developers work together on daily basis.
- Projects are created around motivated people if they are given the proper environment and support.
- Face to face interaction is the most efficient method of moving information in the development team.
- Primary measure of progress is a working software.
- Agile process helps in sustainable development.
- Continuous attention to technical excellence and good design increases agility.
- From self organizing teams the best architecture, design and requirements are emerged.
- Simplicity is necessary in development.

Extreme Programming (XP)

- The Extreme Programming is commonly used agile process model.
- It uses the concept of object-oriented programming.
- A developer focuses on the framework activities like planning, design, coding and testing. XP has a set of rules and practices.

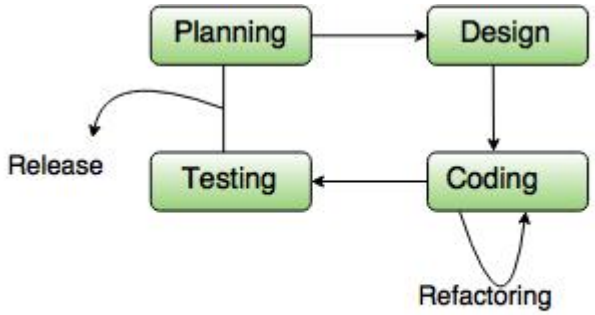

Fig. - The Extreme Programming Process

XP values

Following are the values for extreme programming:

1. Communication

- Building software development process needs communication between the developer and the customer.
- Communication is important for requirement gathering and discussing the concept.

2) Simplicity

The simple design is easy to implement in code.

3. Feedback

Feedback guides the development process in the right direction.

4. Courage

In every development process there will always be a pressure situation. The courage or the discipline to deal with it surely makes the task easy.

5. Respect

Agile process should inculcate the habit to respect all team members, other stake holders and customer.

The XP Process

The XP process comprises four framework activities:

1. Planning

- Planning starts with the requirements gathering which enables XP team to understand the rules for the software.
- The customer and developer work together for the final requirements.

2. Design

- The XP design follows the 'keep it simple' principle.
- A simple design always prefers the more difficult representation.

3. Coding

- The coding is started after the initial design work is over.

- After the initial design work is done, the team creates a set of unit tests which can test each situation that should be a part of the release.
- The developer is focused on what must be implemented to pass the test.
- Two people are assigned to create the code. It is an important concept in coding activity.

4. Testing

- Validation testing of the system occurs on a daily basis. It gives the XP team a regular indication of the progress.
- 'XP acceptance tests' are known as the customer test.

Scrum

- Scrum is an agile software development method.
- Scrum principles are consistent with the agile platform that are used to guide development activities within a process.
- It includes the framework activities like requirement, analysis, design, evolution and delivery.
- Work tasks occur within a process pattern in each framework activity called as 'sprint'.
- Scrum highlights the use of a set of software process pattern that are effective for the projects with tight timelines, changing requirements and business criticality.
- Scrum consists of the use of a set of software process patterns.

Each process patterns defines a set of development actions which are as follows:

Backlog

- A prioritized list of project requirements or features that provide business value for the customer.
- Items can be added to the backlog at any time.
- The product manager accesses the backlog and updates priorities, as required.

Sprints

- It consists of work units that are required to achieve a requirement defined in the backlog.
- Changes are not introduced during the sprints. It allows team members to work in short-term but in the stable environment.

Scrum meeting

- The short meetings are held daily by the scrum team.
- The key questions are asked and answered by all team members.
- Daily meetings guide to 'knowledge socialization' and that encourages a self-organizing team structure.

Demos

- Deliver the software increment to the customer. Using which the customer evaluates and demonstrates the functionalities.

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.

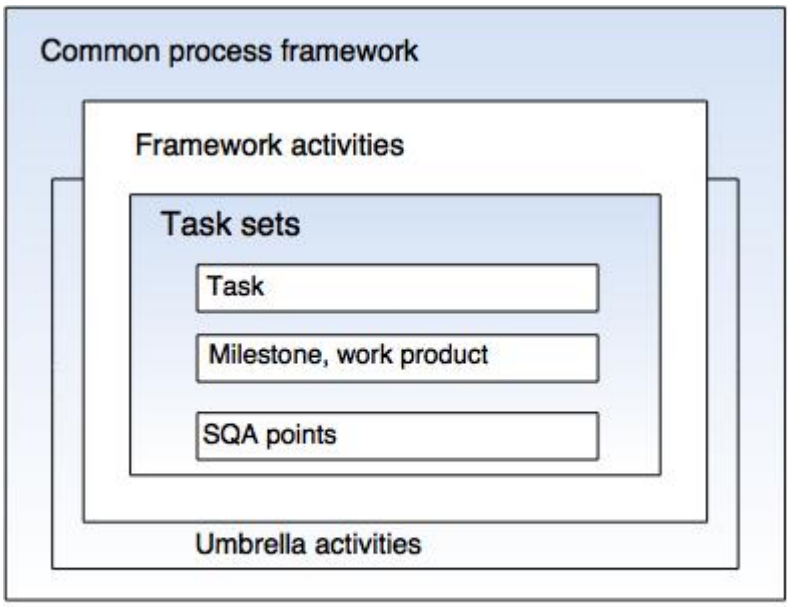

Fig.- A software process framework

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

- Measurement 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.