Software Process Model (SPM)

Software:

A group of computer programs that can perform some particular task in our computer is called software. It makes computer hardware functional by instructing them to carry out some operation. It is also called the logical part of our computer.

Software Project:

A Software Project is the complete procedure of software development from requirement collection to testing and maintenance, carried out in a specified period of time to achieve the intended software product.

SDLC (System Development Life Cycle):

SDLC is a systematic process used to develop, maintain and replace the existing software systems. It involves a series of steps to make the development process organised. It is needed to deliver high quality software that meets the users requirements.

The different phases of SDLC are:

1. **System Study:**
   It is the very first phase of SDLC, where the problems with the current system are studied in depth, getting possible solutions. It is also called preliminary investigation. Feasibility study is also carried out at the end of this phase.
2. **System Analysis:**
   It is a phase in which data are collected from various sources determining system requirements. It determines what system is going to be developed.
3. **System Design:**
   It is a phase in which a blueprint of a proposed system is designed by the system designers based on system specifications. It is one of the most challenging phases of SDLC.
4. **System Development:**
   It is a phase in which the system developers translate the design of the system into programming codes. It is the phase where actual coding is done.
5. **System Testing:**
   It is a process of checking whether the components of the developed system contain any errors or not and meet the user's requirements or not.
   The process of fixing errors occurred in our system is called debugging.
6. **System Implementation:**
   It is a process of carrying out a developed system into working condition. After system implementation, the users need to move from the older system to the new system. And the process of moving or switching from an old system to a new system is called system conversion.
7. **System Maintenance and Review:**
   System maintenance is a process of correcting and upgrading the existing system. It is necessary to fix any bugs on the system and modify it to meet the changing requirements of the users.
   Review means taking feedback from the users.

System Analyst:

System analyst is a person who is involved in analysing, designing, implementing and evaluating computer based systems to support decision making and effective operations of an organisation. System analyst should have the ability to study the system, analyze, design and develop it.

Major Roles/Responsibilities of System Analyst:

The major roles or responsibilities of system analyst are:

1. **Defining requirements:**
   It involves understanding user's requirements by interviewing them, finding out what information is being used in the current system and what are their expectations from a proposed system.
2. **Prioritizing requirements:**
   It is not possible to satisfy all the needs or requirements of the users. So, a system analyst needs to prioritize the requirements collected from the users.

3. **Collecting data, fact and opinion of user:**
   The system analyst must collect the necessary data and views from the users. The system analyst must develop the system through the active and willing cooperation of the users.
4. **Solving problems:**
   A system analyst must study the problem in depth and suggest the possible solutions to the management.
5. **Evaluating system:**
   A system analyst must evaluate a system after it has been used for a specific period of time by taking user's feedbacks.

Feasibility Study:

It is the process of analyzing whether the proposed system is feasible to develop or not. It is carried out after system study, once the system goal is set. It also determines the strengths and limitations of the system before its development.

The different types of feasibility are as follows:

1. **Technical Feasibility:**
   It is a feasibility study that determines whether a proposed system can be developed and implemented using existing technology or not.
2. **Economic Feasibility:**
   It is a feasibility study that determines whether a proposed system can be developed under the allocated budget or not. It determines the cost and benefits of the proposed system.
3. **Operational Feasibility:**
   It is a feasibility study that determines whether the available human resource can operate the proposed system or not.
4. **Schedule(Time) Feasibility:**
   It is a feasibility study that determines whether a proposed system can be completed within the defined time or not. It is concerned with time constraints and system deadlines.
5. **Social Feasibility:**
   It is a feasibility study that determines whether a proposed system will be acceptable to the people or not.
6. **Legal Feasibility:**
   It is a feasibility study that determines whether a proposed system is according to the law of the country or not.

Requirement Collection Methods:

The requirements needed to develop any software project can be collected through various techniques. Some of them are:

1. Interview
2. Survey
3. Focused Group
4. Prototyping
5. Brainstorming

Models of SDLC:

The different models of software development life cycle are:

1. Waterfall Model
2. Prototype Model
3. Agile

Waterfall Model:

It is the simplest and most popular model of SDLC in which development takes place in linear or sequential manner. It is easy to use and understand. The development proceeds to the next phase only if the previous phase is completed and we never turn back to the previous phase.

Advantages:

- Simple, easy to use and understand.
- No chances of phases overlapping.
- Better for smaller projects where requirements are known.

Disadvantages:

- Requirements must be known in advance.
- Cannot add the dynamic requirements of users.
- Not good for complex and ongoing projects.

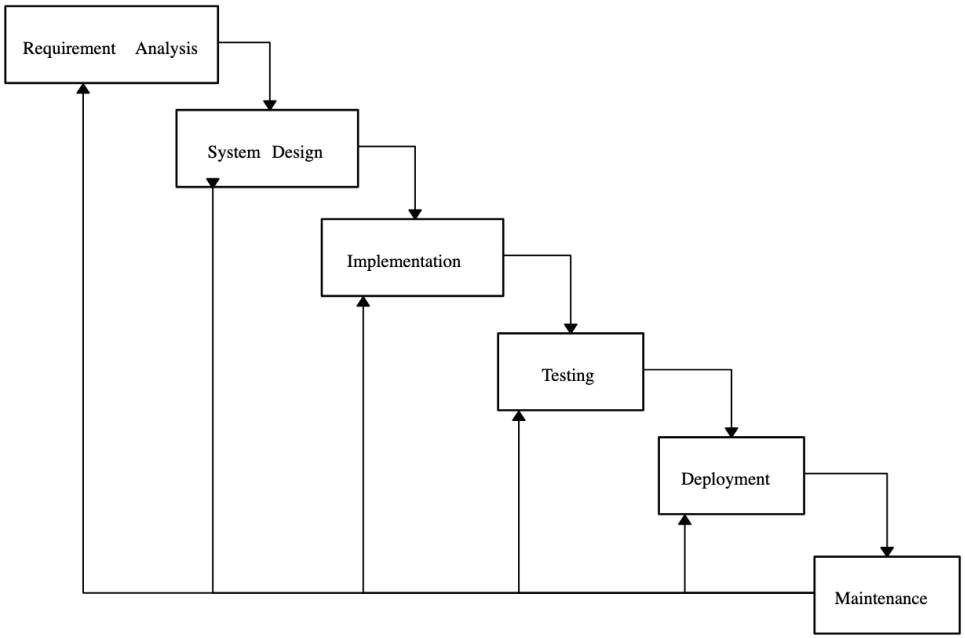

Fig: Waterfall Model

Prototyping Model:

It is a model of SDLC in which a prototype is developed instead of developing the fully working model. A prototype is an early approximation of the final system. It is very useful where the requirements are not known in advance during system development.

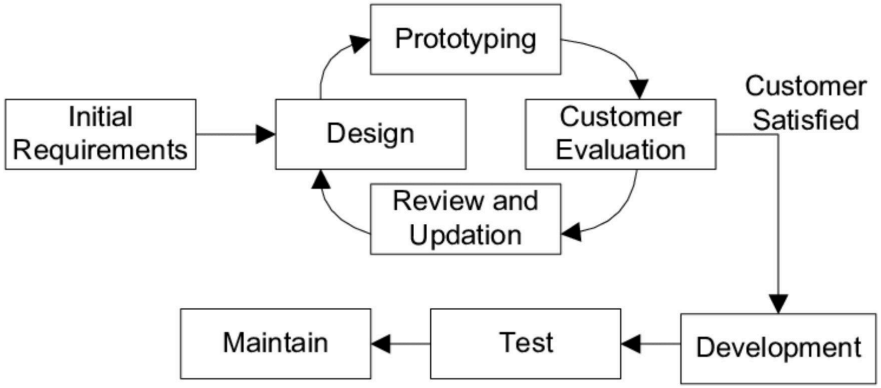

Fig: Prototyping Model

Advantages:

- Improved communication between system analyst and user.
- Do not require to know all requirements in advance.
- Easy to implement the changing requirements of the user.

Disadvantages:

- It is more costly.
- Difficult to determine the number of iterations during prototyping.

Agile:

Agile is a software development model that focuses on creating working software quickly, collaborating with customers frequently, and being able to adapt to changes easily. It is especially beneficial for projects that are complex or have uncertain requirements. It is the most popular development model followed by most of the IT companies to develop software projects.

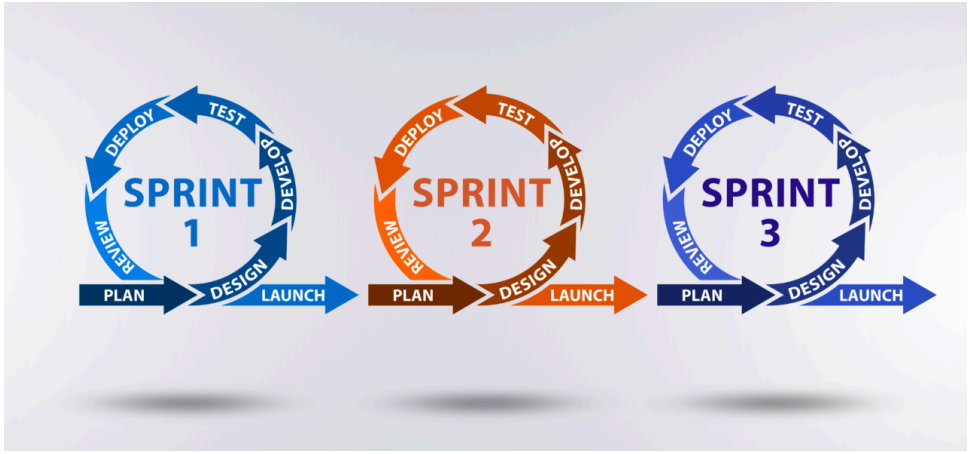

Fig: Agile Development

Advantages:

- Increased flexibility.
- Improved Communication.
- Increased customer satisfaction.

Disadvantages:

- Lack of documentation.

- Needs a high level of collaboration.
- Complex projects can be lengthy.

Documentation:
The process of keeping records of all the system information during the system development process is called documentation. A proper documentation is necessary to operate the system, modify the system and maintenance of the system in the future.
There are 2 types of documentations as follows:

1. **Internal Documentation:**
   It is a type of documentation written by the system developers or programmers during the development process. It is useful to modify, upgrade and maintain the software system. It is not visible to the general users.
   It contains the description of program codes, their functionalities, algorithms, flowcharts, etc.

2. **External Documentation:**
   It is a type of documentation written for the general users to help them to operate the software system. It is also called the users manual and contains the instructions to execute the program or the system effectively.

Qualities (Attributes) of Good Software:
A software should have the following characteristics to become a good software.

1. **Usability:**
   A software should be easy to use and understand for the users (i.e., user-friendly).
2. **Reliability:**
   It means a software should be error free and can perform consistently and accurately under different conditions.
3. **Performance:**
   A software should have the ability to handle large amount of data or incoming traffic without degrading its performance.
3. **Maintainability:**
   It should be easy to change or maintain a software for any kinds of users using its documentation.
4. **Security:**
   A software should be able to block unauthorized access to data and functions, keeping them safe from different malicious attacks.
5. **Scalability:**
   It should handle the increasing workload and extend easily to meet the changing requirements of the users.
6. **Portability:**
   It refers to an ability of a software to perform the same function across different platforms or environments.