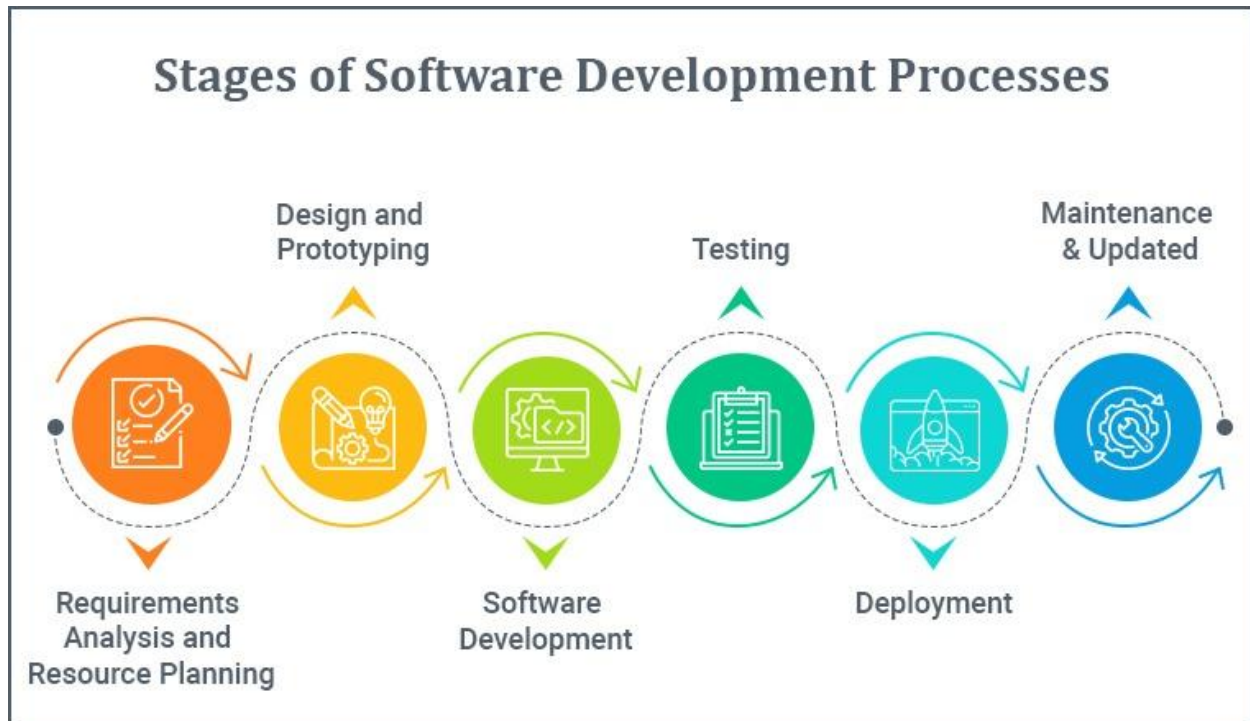


Software Processes

The term **software** specifies to the set of computer programs, procedures and associated documents (Flowcharts, manuals, etc.) that describe the program and how they are to be used.

A software process is the set of activities and associated outcome that produce a software product. Software engineers mostly carry out these activities. These are four key process activities, which are common to all software processes. These activities are:



1. **Software specifications:** The functionality of the software and constraints on its operation must be defined.
2. **Software development:** The software to meet the requirement must be produced.
3. **Software validation:** The software must be validated to ensure that it does what the customer wants.
4. **Software evolution:** The software must evolve to meet changing client needs.

The Software Process Model

A software process model is a specified definition of a software process, which is presented from a particular perspective. Models, by their nature, are a simplification, so a software process model is an abstraction of the actual process, which is being described. Process models may

contain activities, which are part of the software process, software product, and the roles of people involved in software engineering. Some examples of the types of software process models that may be produced are:

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.

Need for Process Model:

The software development team must decide the process model that is to be used for software product development and then the entire team must adhere to it. This is necessary because the software product development can then be done systematically. Each team member will understand what is the next activity and how to do it. Thus process model will bring the definiteness and discipline in overall development process. Every process model consists of definite entry and exit criteria for each phase. Hence the transition of the product through various phases is definite.

If the process model is not followed for software development then any team member can perform any software development activity, this will ultimately cause a chaos and software project will definitely fail without using process model, it is difficult to monitor the progress of software product. Thus process model plays an important rule in software engineering.

Advantages or Disadvantages:

There are several advantages and disadvantages to different software development methodologies, such as:

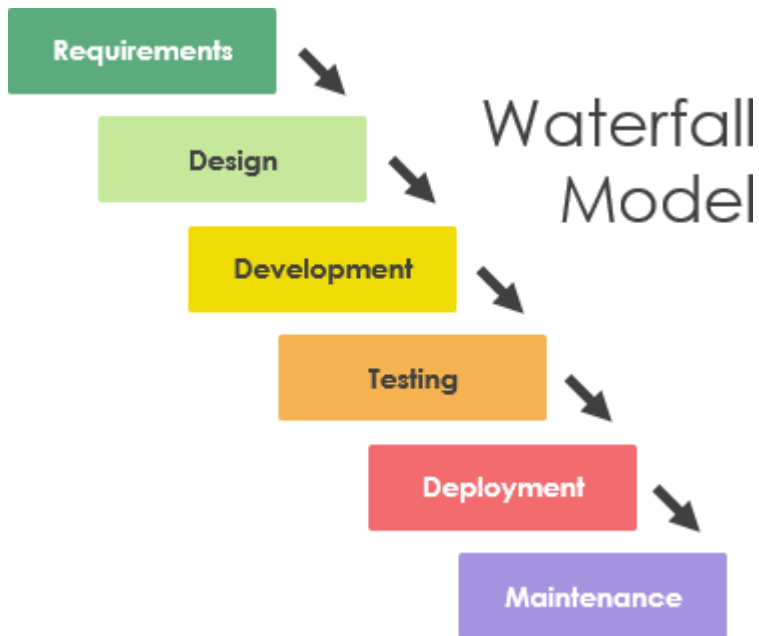
Waterfall:

Advantages:

1. Clear and defined phases of development make it easy to plan and manage the project.
2. It is well-suited for projects with well-defined and unchanging requirements.

Disadvantages:

- Changes made to the requirements during the development phase can be costly and time-consuming.
- It can be difficult to know how long each phase will take, making it difficult to estimate the overall time and cost of the project.
- It does not have much room for iteration and feedback throughout the development process.



Agile:



Fig. Agile Model

The meaning of Agile is swift or versatile. "Agile process model" refers to a software development approach based on iterative development. Agile methods break tasks into smaller iterations, or parts do not directly involve long term planning. The project scope and requirements are laid down at the beginning of the development process. Plans regarding the number of iterations, the duration and the scope of each iteration are clearly defined in advance.

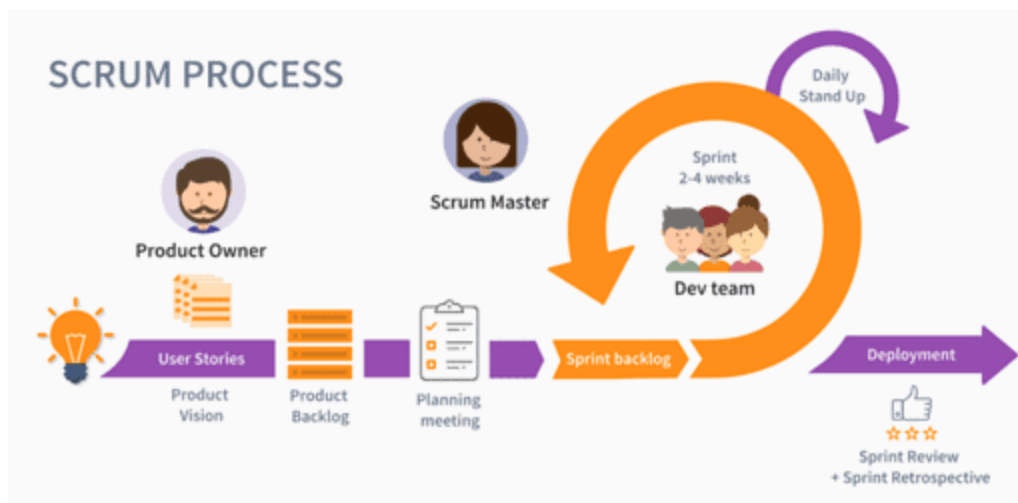
Advantages:

1. Flexible and adaptable to changing requirements.
2. Emphasizes rapid prototyping and continuous delivery, which can help to identify and fix problems early on.
3. Encourages collaboration and communication between development teams and stakeholders.

Disadvantages:

1. It may be difficult to plan and manage a project using Agile methodologies, as requirements and deliverables are not always well-defined in advance.
2. It can be difficult to estimate the overall time and cost of a project, as the process is iterative and changes are made throughout the development.

Scrum:



Advantages:

1. Encourages teamwork and collaboration.
2. Provides a flexible and adaptive framework for planning and managing software development projects.
3. Helps to identify and fix problems early on by using frequent testing and inspection.

Disadvantages:

1. A lack of understanding of Scrum methodologies can lead to confusion and inefficiency.
2. It can be difficult to estimate the overall time and cost of a project, as the process is iterative and changes are made throughout the development.

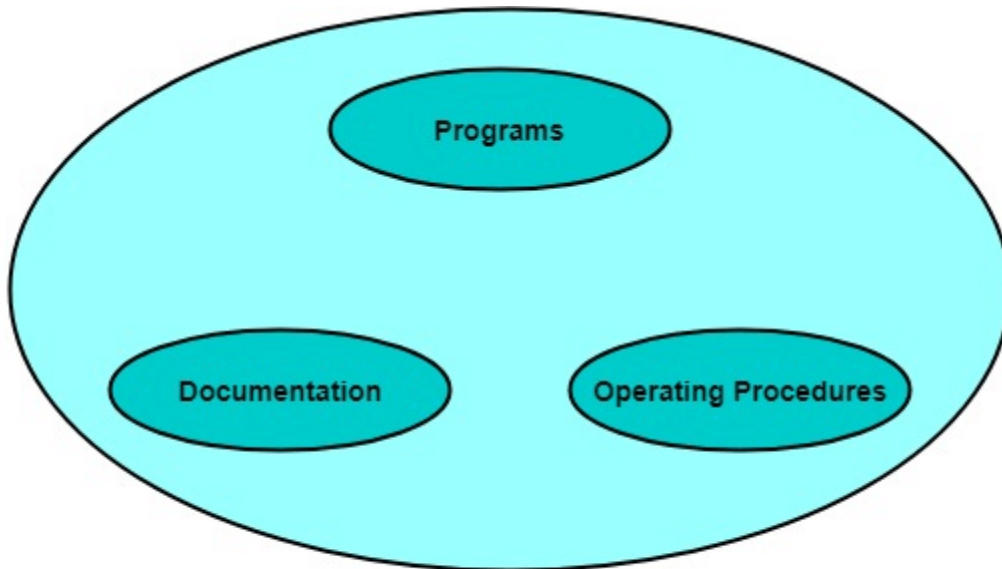
Software Crisis

1. **Size:** Software is becoming more expensive and more complex with the growing complexity and expectation out of software. For example, the code in the consumer product is doubling every couple of years.
2. **Quality:** Many software products have poor quality, i.e., the software products defects after putting into use due to ineffective testing technique. For example, Software testing typically finds 25 errors per 1000 lines of code.
3. **Cost:** Software development is costly i.e. in terms of time taken to develop and the money involved. For example, Development of the FAA's Advanced Automation System cost over \$700 per lines of code.
4. **Delayed Delivery:** Serious schedule overruns are common. Very often the software takes longer than the estimated time to develop, which in turn leads to cost shooting up. For example, one in four large-scale development projects is never completed.

Program vs. Software

Software is more than programs. Any program is a subset of software, and it becomes software only if documentation & operating procedures manuals are prepared.

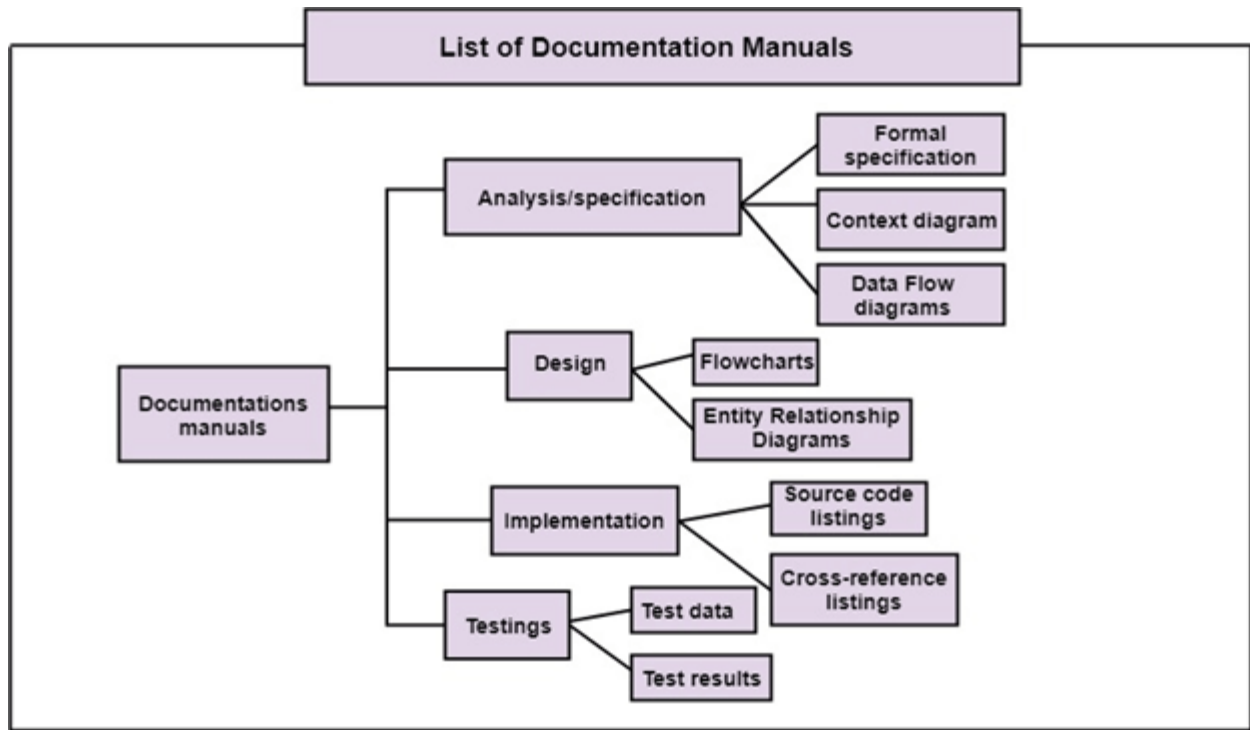
There are three components of the software as shown in fig:



Software= Program + Documentation + Operating Procedures

Fig:Components of Software

- 1. Program:** Program is a combination of source code & object code.
- 2. Documentation:** Documentation consists of different types of manuals. Examples of documentation manuals are: Data Flow Diagram, Flow Charts, ER diagrams, etc.



3. Operating Procedures: Operating Procedures consist of instructions to set up and use the software system and instructions on how react to the system failure. Example of operating system procedures manuals is: installation guide, Beginner's guide, reference guide, system administration guide, etc.

