SE Unit 1

What is Software Engineering?

- Software engineering is the process of analyzing user needs and designing, constructing, and testing end user applications that will satisfy these needs through the use of software programming languages.
- It is the application of engineering principles to software development.
- In contrast to simple programming, software engineering is used for larger and more complex software systems, which are used as critical systems for businesses and organizations.

Software Development Life Cycle -

- The software development life cycle (SDLC) is a framework defining tasks performed at each step in the software development process.
- It consists of a detailed plan describing how to develop, maintain and replace specific software.
- The life cycle defines a methodology for improving the quality of software and the overall development process.
- The software development life cycle is also known as the software development process.
- There are following six phases in every Software development life cycle model -
    1. Requirement gathering and analysis
    2. Design
    3. Implementation or coding
    4. Testing
    5. Deployment
    6. Maintenance

Software Requirements -

- Requirements are descriptions of the services that a software system must provide and the constraints under which it must operate.
- These requirements reflect the needs of customers for a system.
- The process of finding out, analyzing, documenting and checking these services and constraints is called requirements engineering (RE).
- In some cases, a requirement is simply a high level, abstract statement of a service that a system should provide or a constraint on a system. At the other extreme, it is a detailed, formal definition of a system function.

Types of Requirement -

1. **User requirements -**

- User requirements are statements, in a natural language plus diagrams, of what services the system is expected to provide to system users and the constraints under which it must operate.
- The readers of the user requirements are not usually concerned with how the system will be implemented and may be managers who are not interested in the detailed facilities of the system.

2. **System Requirement Specification -**

- The user should be provided with facilities to define the type of external files.
- Each external file may have an associated tool which may be applied to the file and is represented as a specific icon on the user’s display.
- Facilities should be provided for the icon representing an external file to be define by the user.
- When the user selects an icon representing an external file, the effect of that selection is to apply the tool associated with the type of external file to the file represented by the selected icon.

Functional and Non functional Requirements -

- Software system requirements are often classified as -
  1. Functional Requirements
  2. Non-Functional Requirements

1. **Functional Requirements -**

- These are statements of services the system should provide, how the system should react to particular inputs, and how the system should behave in particular situations.
- In some cases, the functional requirements may also explicitly state what the system should not do.
- The functional requirements for a system describe what the system should do.
- These requirements depend on the type of software being developed, the expected users of the software, and the general approach taken by the organization when writing requirements.
- Functional system requirements describe the system functions, its inputs and outputs, exceptions.
- These functional user requirements define specific facilities to be provided by the system.

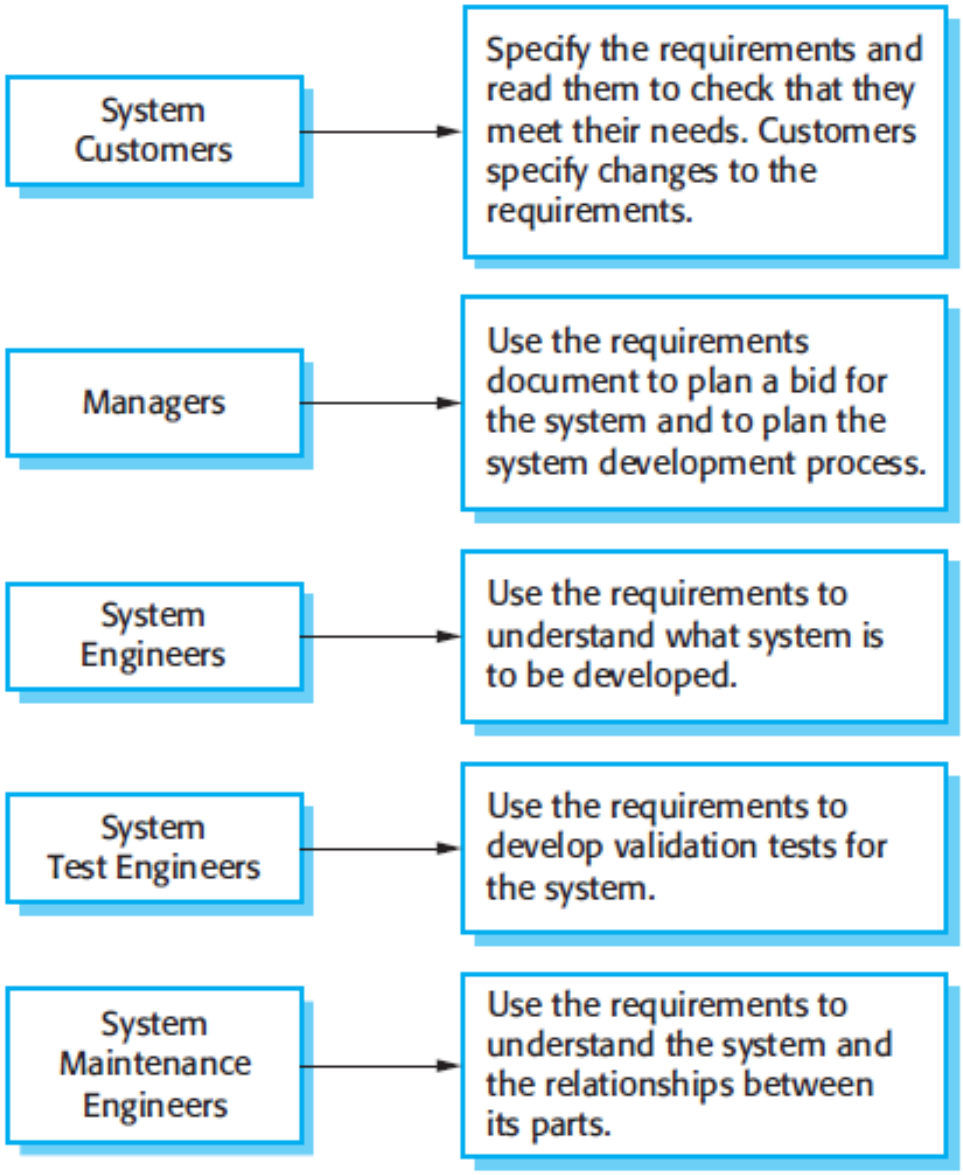
2. **Non-Functional Requirements -**

- These are constraints on the services or functions offered by the system.
- They include timing constraints, constraints on the development process, and constraints imposed by standards.
- Non functional requirements often apply to the system as a whole, rather than individual system features or services.
- Non functional requirements, as the name suggests, are requirements that are not directly concerned with the specific services delivered by the system to its users.
- They may relate to system properties such as reliability, response time, and store occupancy.

- Non functional requirements, such as performance, security, or availability, usually specify characteristics of the system as a whole.
- Non functional requirements are often more critical than individual functional requirements.

Software Requirements Document -

- The software requirements document (sometimes called the software requirements specification or SRS) is an official statement of what the system developers should implement.
- It should include both the user requirements for a system and a detailed specification of the system requirements.
- Requirements documents are essential when an outside contractor is developing the software system.
- The requirements document has a diverse set of users, ranging from the senior management of the organization that is paying for the system to the engineers responsible for developing the
  software.



Software Processes -

- The processes that deal with the technical and management issues of software development are collectively called the software process.

- **Software specification** – defining what the system should do;
- **Software design and implementation** – defining the organization of the system and implementing the system.
- **Software validation** – checking that it does what the customer wants
- **Software evolution** – changing the system in response to changing customer needs.

Software Development Process Models (Refer Notes)-

- A software process model is an abstract representation of a process. It presents a description of a process from some particular perspective.
- Generic software process models are -
    1. Waterfall Model
    2. Prototyping
    3. Iterative Development
    4. Rational Unified Development
    5. The RAD Model
    6. Time Boxing Model

Agile Model -

The word ‘agile’ means -

- Able to move your body quickly and easily.
- Able to think quickly and clearly.
- In business, ‘agile’ is used for describing ways of planning and doing work wherein it is understood that making changes as needed is an important part of the job.

- Business ‘agility’ means that a company is always in a position to take account of the market changes.

Agile Software Development -

- Agile is a software development methodology to build a software incrementally using short iterations of 1 to 4 weeks so that the development process is aligned with the changing business needs.
- Instead of a single pass development of 6 to 18 months where all the requirements and risks are predicted upfront, Agile adopts a process of frequent feedback where a workable product is delivered after 1 to 4 week iteration.