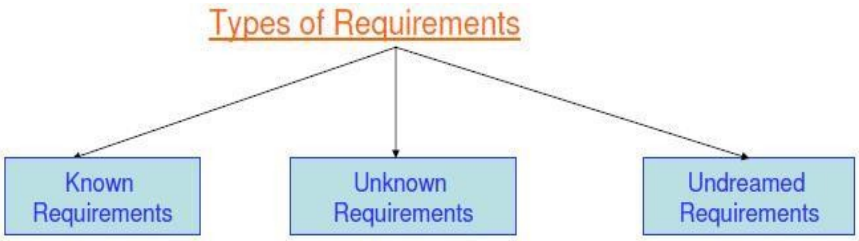
Unit-2: Software Requirement Specifications

Software Requirement

1. A condition of capability needed by a user to solve a problem or achieve an objective
2. A condition or a capability that must be met or possessed by a system ... to satisfy a contract, standard, specification, or other formally imposed document."



1. **Known Requirements:-** Something a stakeholder believes to be implemented
2. **Unknown requirements:-**Forgotten by the stakeholder because they are not needed right now or needed only by another stakeholder
3. **Undreamt requirements:-** stakeholder may not be able to think of new requirement due to limited knowledge

A Known, Unknown and Undreamt requirements may functional or non functional.

- **Functional requirements: -** describe what the software has to do. They are often called product features. It depends on the type of software, expected users and the type of system where the software is used.
- **Non Functional requirements**: - are mostly quality requirements. That stipulate how well the software does, what it has to do.These define system properties and constraints e.g. reliability, response time and storage requirements. Constraints are I/O device capability, system representations, etc.
- **User requirements:-** Statements in natural language plus diagrams of the services the system provides and its operational constraints.
- **System requirements: -** A structured document setting out detailed descriptions of the system's functions, services and operational constraints. Defines what should be implemented so may be part of a contract between client and contractor.

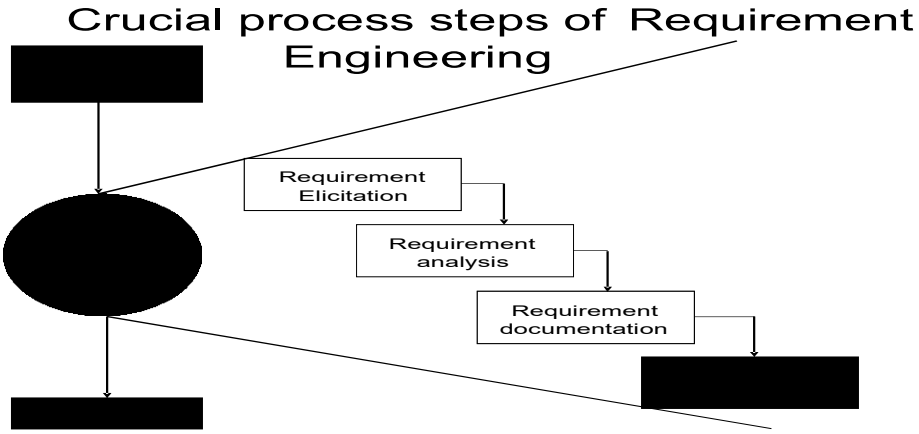
Requirements engineering Process

- The process of finding out, analyzing, documenting and checking these services and constraints called requirement engineering.
- RE produces one large document, written in a natural language, contains a description of what the system will do without how it will do
- Input to RE is problem statement prepared by the customer and the output is SRS prepared by the developer.
- The requirements themselves are the descriptions of the system services and constraints that are generated during the requirements engineering process.

Requirements engineering processes:- The processes used for RE vary widely depending on the application domain, the people involved and the organisation developing the requirements. However, there are a number of generic activities common to all processes

- Requirements elicitation
- Requirements analysis

- Requirements documentation
- Requirements review



- **Requirement Elicitation: -** known as gathering of requirement. Here requirement are identified with the help of customer and exiting system processes, if they are available.
- **Requirement Analysis: -** analysis of requirement stars with Requirement Elicitation. Requirements are analyzed in order to identify inconstancy, defects etc.
- **Requirement Documentation:-** this is the end product of requirement elicitation and analysis. Documentation is very important as it will be the foundation for the design of the software. The documentation is known as SRS.
- **Requirement Review:-**review process is carried out to improve the quality of the SRS. it may also called verification. It should be a continuous activity that is incorporated into the elicitation, analysis, documentation.
  The primary output of requirement engineering process is requirement specification (SRS).

Feasibility Study: - It is the process of evaluation or analysis of the potential impact of a propose project or program.

Five common factors (TELOS)

- **Technical feasibility:** - Is it technically feasible to provide direct communication connectivity through space from one location of globe to another location?
- **Economic feasibility:-**Are the project's cost assumption realistic?
- **Legal feasibility:** - Does the business model realistic?
- **Operational feasibility: -** Is there any market for the product?

- **Schedule feasibility: -** Are the project's schedule assumption realistic?

Elicitation: - It is also called requirement discovery.Requirements are identified with the help of customer and exiting system processes, if they are available. Requirement Elicitation is most difficult is perhaps most critical most error prone most communication intensive aspect of software development.

Various methods of Requirements Elicitation

1. Interviews

- After receiving the problem statement from the customer, the first step is to arrange a meeting with the customer.
- During the meeting or interview, both the parties would like to understand each other.
- The objective of conducting an interview is to understand the customer's expectation from the

software

Selection of stakeholder

1. Entry level personnel
2. Middle level stakeholder
3. Managers
4. Users of the software (Most important)

2. Brainstorming Sessions

- Brainstormy is a group technique that may be used during requirement elicitation to understand the requirement
- Every idea will be documented in a way that every can see it
- After brainstormy session a detailed report will be prepared and reviewed by the facilitator

3. Facilitated Application specification approach (FAST)

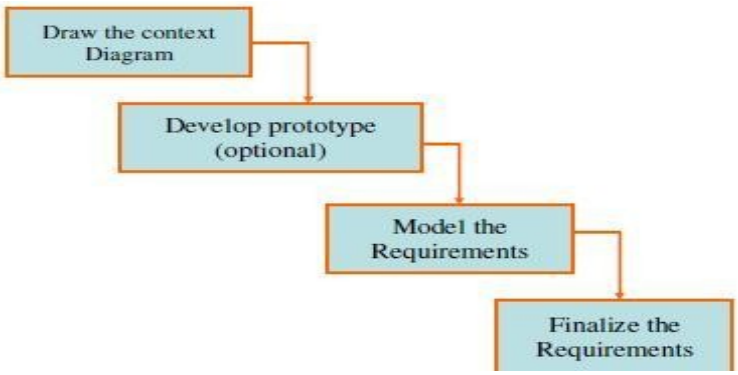
- FAST is similar to brainstormy session and the objective is to bridge the expectation gap-a difference what the developers think they are suppose to build and what the customer think they are going to get
- In order to reduce the gap a team oriented approach is developed for requirement gathering and is called FAST

4. Quality function deployment (QFT)

It incorporates the voice of the customer and converts it in to the document.

Analysis

Requirement analysis phase analyze, refine and scrutinize requirements to make consistent & unambiguous requirements.



1. Draw the context diagram

The context diagram is a simple model that defines the boundaries and interface of the proposed system.

2. Development of prototype

Prototype helps the client to visualize the proposed system and increase the understanding of requirement. Prototype may help the parties to take final decision.

3. Model the requirement

This process usually consists of various graphical representations of function, data entities, external entities and their relationship between them. It graphical view may help to find incorrect, inconsistent, missing requirements. Such models include data flow diagram, entity relationship diagram, data dictionary, state transition diagram.

4. Finalize the requirement

After modeling the requirement inconsistencies and ambiguities have been identified and corrected. Flow of data among various modules has been analyzed. Now Finalize and analyzes requirements and next step is to document these requirements in prescribed format.

Documentation

This is the way of representing requirements in a consistent format SRS serves many purpose depending upon who is writing it.

Software requirement specification

Requirements specification is a complete description of the behavior of a system to be developed. It includes a set of use cases that describe all the interactions the users will have with the software. Use cases are also known as functional requirements. In addition to use cases, the SRS also contains non-functional (or supplementary) requirements. Non- functional requirements are requirements which impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints).

Need for Software Requirement Specification (SRS)

- The problem is that the client usually does not understand software or the software development process, and the developer often does not understand the client's problem and application area.

- This causes a communication gap between the parties involved in the development project. A basic purpose of software requirements specification is to bridge this communication gap.

Characteristics of good SRS document:- Some of the identified desirable qualities of the SRS documents are the following-

Concise- The SRS document should be concise and at the same time unambiguous, consistent, and complete. An SRS is unambiguous if and only if every requirement stated has one and only one interpretation.

Structured- The SRS document should be well-structured. A well-structured document is easy to understand and modify. In practice, the SRS document undergoes several revisions to cope with the customer requirements.

Black-box view- It should only specify what the system should do and refrain from stating how to do. This means that the SRS document should specify the external behaviour of the system and not discuss the implementation issues.

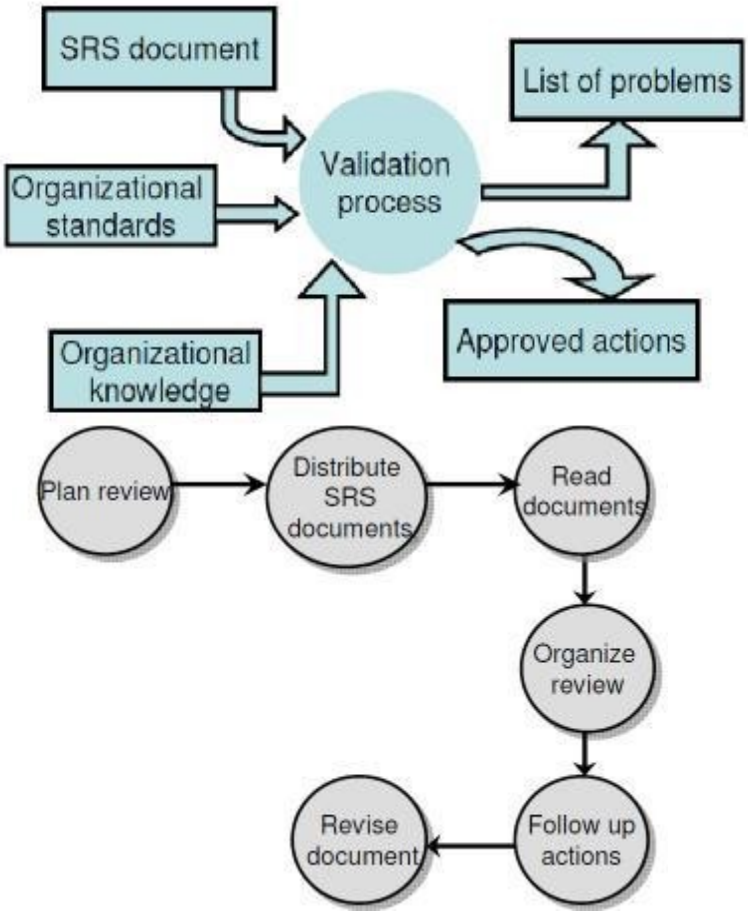
Conceptual integrity- The SRS document should exhibit conceptual integrity so that the reader can easily understand the contents.

Verifiable- All requirements of the system as documented in the SRs document should be verifiable. This means that it should be possible to determine whether or not requirements have been met in an implementation.

Requirements Validation

Requirement Validation is used for checking the document:-

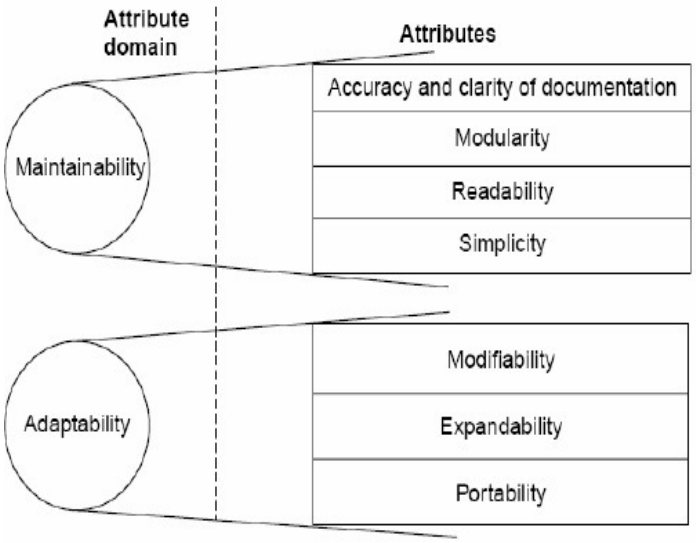
- Completeness & consistency
- Conformance to standards
- Requirements conflicts
- Technical errors
- Ambiguous requirements

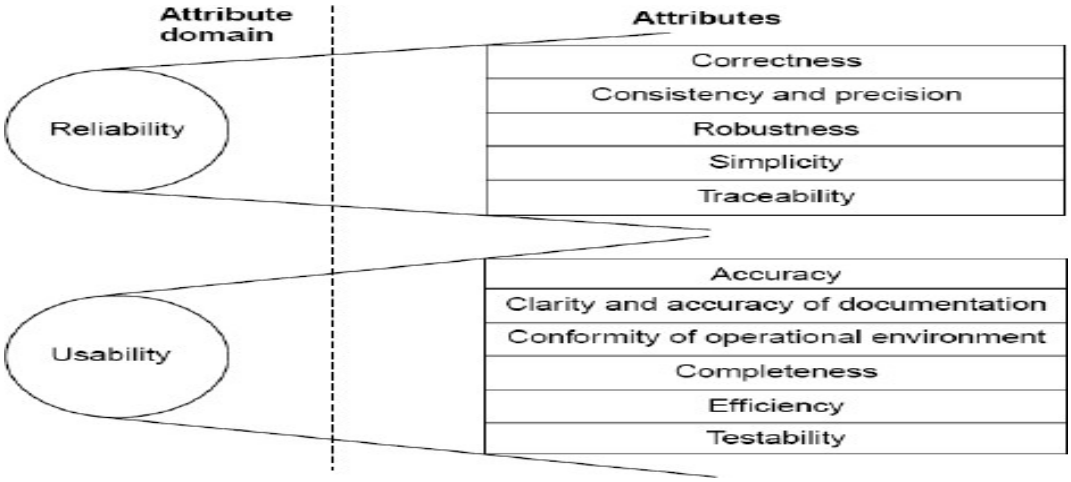


Software Quality

- The degree to which a system, component, or process meets specified requirements.
- The degree to which a system, component or process meets customer or user needs or expectations.

Software Quality attribute





1	Reliability	The extent to which a software performs its intended functions without failure.
2	Correctness	The extent to which a software meets its specifications.
3	Consistency & precision	The extent to which a software is consistent and give results with precision.
4	Robustness	The extent to which a software tolerates the unexpected problems.
5	Simplicity	The extent to which a software is simple in its operations.
6	Traceability	The extent to which an error is traceable in order to fix it.
7	Usability	The extent of effort required to learn, operate and understand the functions of the software

8	Accuracy	Meeting specifications with precision.
9	Clarity & Accuracy of documentation	The extent to which documents are clearly & accurately written.
10	Conformity of operational environment	The extent to which a software is in conformity of operational environment.
11	Completeness	The extent to which a software has specified functions.
12	Efficiency	The amount of computing resources and code required by software to perform a function.
13	Testability	The effort required to test a software to ensure that it performs its intended functions.
14	Maintainability	The effort required to locate and fix an error during maintenance phase.

2.3.1 McCall Software Quality Model

i. Product Operation

Factors which are related to the operation of a product are combined. The factors are:

- Correctness
- Efficiency
- Integrity
- Reliability
- Usability

These five factors are related to operational performance, convenience, ease of usage and its correctness. These factors play a very significant role in building customer's satisfaction.

ii. Product Revision

The factors which are required for testing & maintenance are combined and are given below:

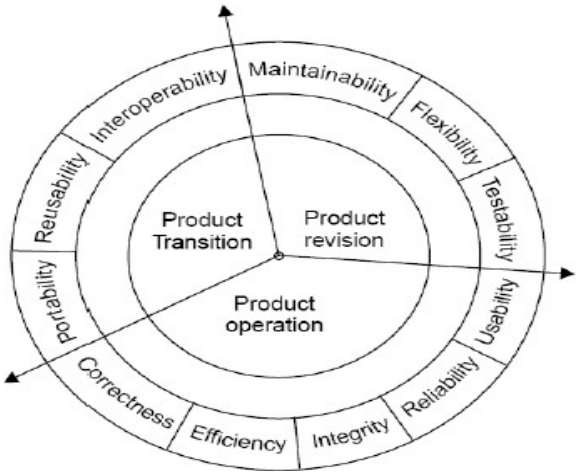
- Maintainability
- Flexibility
- Testability

These factors pertain to the testing & maintainability of software. They give us idea about ease of maintenance, flexibility and testing effort. Hence, they are combined under the umbrella of product revision**.**

iii. Product Transition

We may have to transfer a product from one platform to an other platform or from one technology to another technology. The factors related to such a transfer are combined and given below:

- Portability
- Reusability
- Interoperability



Software Quality Assurance

- It is a planned and systematic pattern of all actions necessary to provide adequate confidence that the time or product conforms to established technical requirements."
- Purpose of SQAP is to specify all the works products that need to be produced during the project, activities that need to performed for checking the quality of each of the work product
- It is interested in the quality of not only the final product but also an intermediate product.

Verification:-is the process of determine whether or not product of a given phase of software development full fill the specification established during the previous phase.

Validation:-is the process of evaluating software at the end of software development to ensure compliance with the software requirement. testing is common method of validation Software V&V is a system engineering process employing rigorous methodologies for evaluating the correctness and quality of the software product throughout the software life cycle.

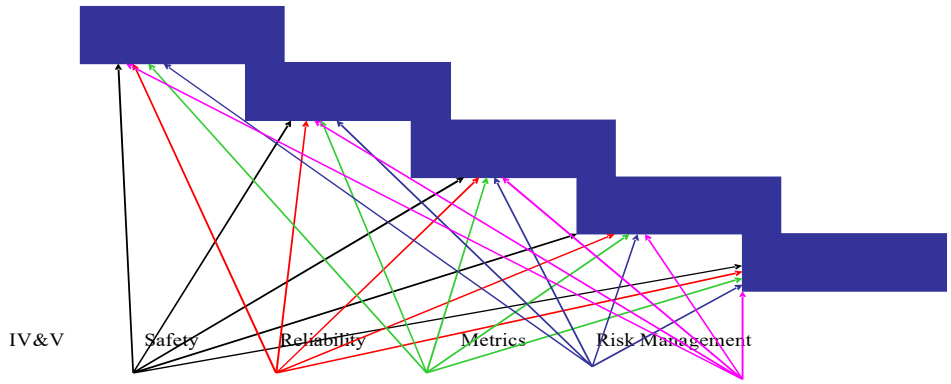
SQA Plans, Software Quality Frameworks

Quality plan structure

- Product introduction;
- Product plans;
- Process descriptions;
- Quality goals;
- Risks and risk management.

Quality plans should be short, succinct documents.If they are too long, no-one will read them.

SQA Life CYCLE or Framework of SQA



International National Organization (ISO)

An international set of standards for quality management. It is applicable to a range of organisations from manufacturing to service industries. ISO 9001 applicable to organisations which design, develop and maintain products. ISO 9001 is a generic model of the quality process that must be instantiated for each organisation using the standard.

Quality standards and procedures should be documented in an organisational quality manual. An external body may certify that an organisation's quality manual conforms to ISO 9000 standards. Some customers require suppliers to be ISO 9000 certified although the need for flexibility here is increasingly recognised.

Process of getting ISO 9000 Certification

- Application
- Pre-assessment
- Document review and adequacy of audit
- Compliance audit
- Registration
- Continued surveillance

The Capability Maturity Model (CMM)

When it is applied to an existing organization's software development processes, it allows an effective approach toward improving them. Eventually it became clear that the model could be applied to other processes. This gave rise to a more general concept that is applied to business processes and to developing people

The Capability Maturity Model involves the following aspects:

- **Maturity Levels***:* a 5-level process maturity continuum - where the uppermost (5th) level is a notional ideal state where processes would be systematically managed by a combination of process optimization and continuous process improvement.
- **Key Process Areas:** a Key Process Area (KPA) identifies a cluster of related activities that, when performed together, achieve a set of goals considered important.
- **Goals:** the goals of a key process area summarize the states that must exist for that key process area to have been implemented in an effective and lasting way. The extent to which the goals have been accomplished is an indicator of how much capability the organization has established at that maturity level. The goals signify the scope, boundaries, and intent of each key process area.
- **Common Features:** common features include practices that implement and institutionalize a key process area. There are five types of common features: commitment to Perform, Ability to Perform, Activities Performed, Measurement and Analysis, and Verifying Implementation.

CMM Level	Focus	Key Process Ares
1. Initial	Competent people	
2. Repeatable	Project management	Software project planning Software configuration management
3. Defined	Definition of processes	Process definition Training program Peer reviews
4. Managed	Product and process quality	Quantitative process metrics Software quality management
5. Optimizing	Continuous process improvement	Defect prevention Process change management Technology change management

Difference between ISO and SEI-CMM

ISO9000	CMM
ISO certification is awarded by an international standard body and can be quoted as an official document	SEI CMM assessment is purely for Internal use
Deals primarily for manufacturing industry and provisioning of services	CMM was developed specially for Software industry and therefore addresses software issues
It aims at level 3 of CMM	Goes beyond Quality Assurance and lead to TQM
Has Customer Focus as primary aim and follows procedural controls	Provide a list of Key Process Areas to proceed from lower CMM level to higher level to provide gradual Quality improvements