CSE 3411

System Analysis and Design (SAD)

•••

SRS

Software Requirement Specifications

SRS

- A complete specification and description of requirements of the software that need to be fulfilled for the successful development of the software system.
- Generated as output of Requirements Analysis.
- A formal report that enables the customers to review whether it is according to their requirements.

Structural Components of an SRS

- Purpose – Main aim of why the document is necessary
- Scope – Overall working and main objective of document and what value it will provide to customer is described
- Overview – Description of product is explained

Structural Components of an SRS

- Functional Requirements

  — Specify the expected behavior of the system - which outputs should be produced from the given inputs.

  — Include technical details, calculations, data processing, etc.
- Interface Requirements

  — How software program communicates with each other or users either in form of any language, code, or message.

  — Shared memory, data streams, etc.

Structural Components of an SRS

- Performance Requirements

  — How a software system performs desired functions under specific conditions

  — Required time, required memory, maximum error rate, etc.

- Design Constraints

  — Limitations or restrictions are specified

  — Particular algorithm, hardware and software limitations, etc.

- Preliminary Schedule & Budget

Structural Components of an SRS

- Non-Functional Attributes
  — Attributes for better performance
  — Reusability, Application Compatibility, Data Integrity, Scalability, etc.
  — Operational Requirements (portability, maintainability)
  — Performance Requirements (speed, capacity)
  — Security Requirements (digital certificate, access control, reliability)
  — Cultural and Political Requirements (social, language, legal, unstated norms etc.)

Uses of an SRS

- Project manager base their plans and estimates of schedule, effort and resources on it.
- Development team require it for developing product according to the need.
- Test plans are generated by testing group based on the describe external behaviour.
- Maintenance and support staff need it to understand what the software product is supposed to do.

Uses of an SRS

- Customers rely on it to know that product they can expect.
- As a contract between developer and customer.
- Publications group writes documents, manuals, etc.
- Training personnel can use it to help develop educational material.

Validation of an SRS Document

- It is extremely important to detect errors in the SRS document before going to other phases.

- The major objective of SRS validation is to ensure Completeness and Correctness.

Validation of an SRS Document

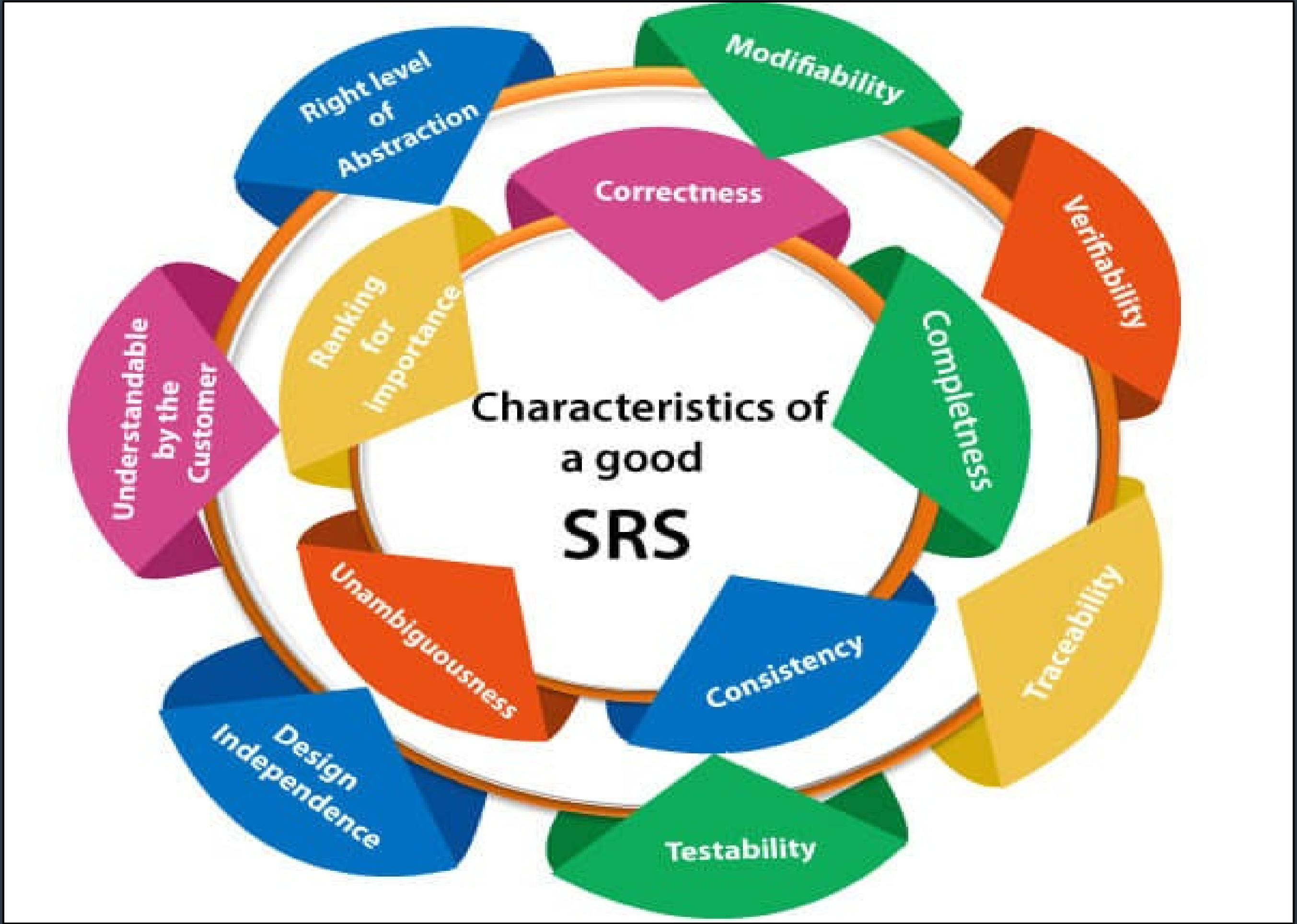
- In validation checking phase, we have to avoid the following errors:

1. Omission: Some user requirement is not included in the SRS. This error directly affects the external completeness of the system.

2. Inconsistency: Due to contradictions in requirements or incompatibility of state requirements.

3. Incorrect Fact: Some facts recorded in the SRS are not correct.

4. Ambiguity: Some requirements have multiple meanings.


Characteristics of a good SRS
Correctness
Modifiability
Right level of Abstraction
Verifiability
Ranking for importance
Understandable by the Customer
Completeness
Unambiguousness
Consistency
Traceability
Design Independence
Testability

Characteristics of Good SRS

An SRS should be:

- Correct
- Complete
- Consistent
- Unambiguous
- Modifiable
- Traceable
- Verifiable
- Testable
- Comprehensive
- Feasible

Characteristics of a Good SRS Document

- Correctness :

  —— SRS is said to be perfect if it covers all the needs that are truly expected from the system.

- Description: The SRS should accurately represent the system that needs to be built.
- Example: If the requirement states "The system shall generate monthly sales reports", but the actual need is weekly reports, it's incorrect.

Characteristics of a Good SRS Document

- Completeness :

  —— Contains all essential requirements, whether relating to goals & objectives, functionality, performance, design, constraints, attributes, or external interfaces.

  —— Contains definition of responses of the software to all realizable classes of input data in all available categories of situations.

  — Involves all stakeholders.

- Description: All possible scenarios, inputs, and outputs should be included.
- Example: For a login feature, the SRS should specify valid login, invalid login, forgot password, and account lock scenarios.

Characteristics of a Good SRS Document

- Consistency :

  —— The use of standard terms and definitions.

  —— The consistent application of business rules in all functionalities.

- Description: No two requirements should contradict each other.
- Example: One requirement says “Password must be at least 8 characters” and another says “Password must be exactly 6 characters” → inconsistency.

Characteristics of a Good SRS Document

- Unambiguity :

  — Every fixed requirement has only one interpretation.

- Description: Each requirement should have only one clear interpretation.
- Example: Writing “System should load quickly” is ambiguous; instead use “System should load within 3 seconds”.

Characteristics of a Good SRS Document

- Modifiability :

  —— Should be capable to quickly obtain changes to the system to some extent.

- Description: Requirements should be easy to update without major rewriting.
- Example: If a requirement document is structured with numbered IDs, changing REQ-12: Allow PayPal payments to REQ-12: Allow Stripe payments is easy.

Characteristics of a Good SRS Document

- Traceability :

  —— The origin of each of the requirements is clear and it facilitates the referencing of each condition in future development.

- Description: Each requirement should be linked to its origin and trackable through development.
- Example: REQ-15: Add search filter can be traced to a user story from the client meeting and later to its implementation in code and test case.

Characteristics of a Good SRS Document

- Verifiability :

—— There exists some cost-effective process that can check whether the final product meets the requirements.

- Description: Each requirement should be testable by inspection, analysis, or testing.
- Example: “System should support 500 concurrent users” is verifiable; “System should be user-friendly” is not directly verifiable.

Characteristics of a Good SRS Document

- Testability :

  — Should be written in such a method that it is simple to generate test cases and test plans from the report.

- Description: Each requirement should allow the creation of tests to confirm its implementation.
- Example: “The system shall process 1000 transactions per second” is testable by performance testing; but “The system shall be efficient” is not directly testable.

Characteristics of a Good SRS Document

- Comprehensibility :

  — Written in such a way that the customer understands.

- Description: The SRS should be written in clear, simple, and structured language so that the customer (and all stakeholders) can easily understand it.
- Example: Instead of writing "The system shall provide authentication facilities employing SHA-256 cryptographic hashing", use "The system shall allow users to log in securely with a password".

Characteristics of a Good SRS Document

- Feasibility :

  — Confirmed on technical and operational feasibility.

- Description: The requirements should be realistic and achievable within technical, financial, and time constraints.
- Example: “System should load 1 million records in 0.01 seconds on a normal laptop” is not feasible, but “System should load 1 million records in under 3 seconds on a server with 16GB RAM” is feasible.

- ❏ https://www.geeksforgeeks.org/softwarequirement-specification-srs-format/
- ❏ https://www.javatpoint.com/software-requirement-specifications

References