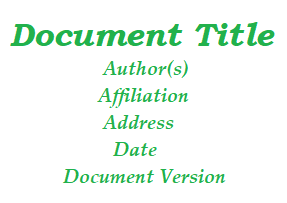
**Software Requirement Specification (SRS)**

**Software Requirement Specification (SRS)** as the name suggests, is a complete specification and description of requirements of the software that need to be fulfilled for the successful development of the software system. These requirements can be functional as well as non-functional depending upon the type of requirement. The interaction between different customers and contractors is done because it is necessary to fully understand the 

needs of customers. Depending upon information gathered after interaction, SRS is developed which describes requirements of software that may include changes and modifications that is needed to be done to increase quality of product and to satisfy customer’s demand.

# ****Introduction:****

* **Purpose of this Document –** At first, main aim of why this document is necessary and what’s purpose of document is explained and described.
* **Scope of this document –** In this, overall working and main objective of document and what value it will provide to customer is described and explained. It also includes a description of development cost and time required.
* **Overview –** In this, description of product is explained. It’s simply summary or overall review of product.

## ****General description****

In this, general functions of product which includes objective of user, a user characteristic, features, benefits, about why its importance is mentioned. It also describes features of user community.

Classification of Software Requirements

According to IEEE standard 729, a requirement is defined as follows:

* A condition or capability needed by a user to solve a problem or achieve an objective
* A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification or other formally imposed documents
* A documented representation of a condition or capability as in 1 and 2.

# Main types of software requirement can be of 3 types:

* Functional requirements
* Non-functional requirements
* Domain requirements

Non-functional requirements: These are basically the quality constraints that the system must satisfy according to the project contract.Nonfunctional requirements, not related to the system functionality, rather define how the system should perform The priority or extent to which these factors are implemented varies from one project to other. They are also called non-behavioral requirements. They basically deal with issues like:

* Portability
* Security
* Maintainability
* Reliability
* Scalability
* Performance
* Reusability
* Flexibility

Domain requirements: Domain requirements are the requirements which are characteristic of a particular category or domain of projects. Domain requirements can be functional or nonfunctional. Domain requirements engineering is a continuous process of proactively defining the requirements for all foreseeable applications to be developed in the software product line. The basic functions that a system of a specific domain must necessarily exhibit come under this category. For instance, in an academic software that maintains records of a school or college, the functionality of being able to access the list of faculty and list of students of each grade is a domain requirement.

Other common classifications of software requirements can be:

1. User requirements: These requirements describe what the end-user wants from the software system. User requirements are usually expressed in natural language and are typically gathered through interviews, surveys, or user feedback.
2. System requirements: These requirements specify the technical characteristics of the software system, such as its architecture, hardware requirements, software components, and interfaces. System requirements are typically expressed in technical terms and are often used as a basis for system design.
3. Business requirements: These requirements describe the business goals and objectives that the software system is expected to achieve. Business requirements are usually expressed in terms of revenue, market share, customer satisfaction, or other business metrics.
4. Regulatory requirements: These requirements specify the legal or regulatory standards that the software system must meet. Regulatory requirements may include data privacy, security, accessibility, or other legal compliance requirements.
5. Interface requirements: These requirements specify the interactions between the software system and external systems or components, such as databases, web services, or other software applications.