

## SRS Document: The What, the Why, and the How

By Ladan Jafari

### What is a software requirements specification (SRS) document?

Software requirements specification (SRS) document in software engineering, or a black-box specification, is a comprehensive description of a software system. It determines what features a system must have and how its features must function.

*Check out [this forum](#) if you're wondering why an SRS document is also called a black-box specification!*

Since SRS documents designate a software project's features in detail, the tasks can be broken down and turned into epics and user stories. By defining user stories and assigning story points, each task's effort is estimated and is used to define the project's scope, timeline, and costs.

*Read about [software development effort estimation](#) in detail in another blog post.*

### What does an SRS document contain?

A typical SRS document lays out all software requirements in detail and sometimes even includes a set of use cases that describe the software's necessary user interactions.

It describes a software project's purpose, contains an overall description of its features, and specifies its requirements.

SRS documents generally include three types of software requirements:

1. **Functional requirements** which contain the actions a system must undertake
2. **Non-functional requirements** that define the performance attributes of a software system
3. **Domain requirements** which are system constraints from the domain of the operation

A well written SRS document must have certain qualities, so it doesn't cause mistakes in the development process. These qualities include:

- Correctness

An SRS document is correct only if the stated requirements are met by the software.

There is no set procedure to ensure an SRS's correctness, but it can be compared with other superior specifications or documents to assure that they're in-line.

Users can also determine if the SRS reflects their needs through feedback.

- Unambiguousness

Since an SRS document is the basis of a software development project, all its statements must be clear, concise, and to-the-point.

There should be no amphiboly, vague adverbs, words that imply multiple meanings, etc.

A glossary at the end of the document that explains each term will be helpful as well.

- Completeness

A complete SRS contains all the necessary software requirements and responses to input data (whether valid or invalid).

- Consistency

As we already mentioned, SRS must be in-line with other higher-level documents, such as system requirements specification.

An SRS document must also be consistent within itself, meaning that it shouldn't change the information it's stating throughout itself.

- Ranking for importance and/or stability

All software requirements are not equally important; some are essential, while some are less vital add-ins.

An SRS document must rank the software's features by importance and/or stability so the developing teams can complete each task in the right order.

- Verifiability

All the statements in the document must be provable, and that's when the software can be checked if it meets the requirements.

This factor rests solely on an SRS's unambiguity; ambiguous statements cannot be verified.

- Modifiability

Since the software development process is prone to changes, software requirements can change significantly.

An SRS needs to be structurally and stylistically flexible so it can be easily modified when necessary.

- Traceability

Each software requirement's origin must be clear, and facilitating references in the future documents must be simple; this means that each feature's life-cycle must be detectable.

Why is writing an SRS document important?

Writing an SRS is necessary for creating startups and launching new software products.

Even though they tend to avoid planning their project in detail because they assume it takes too much time and resources, startups may undergo irreparable consequences without sufficient planning.

SRS documents create a framework for development teams and help them clearly define what they need to develop.

Defining software requirements in an SRS also lays the groundwork for other teams such as quality assurance, operations, and maintenance.

A detailed SRS coordinates between these different teams as well and helps to ensure all software requirements are fulfilled.

The teams are sure to run in circles without a detailed plan; errors come up, deadlines will be missed, cost overruns occur, and the projects derail and even threaten the company's existence.

Since writing as SRS delineates everything that needs to be done and leaves no ambiguity, it mitigates project deviations and time and cost overruns.

### **How is an SRS document prepared?**

The initial step in preparing an SRS document is using an already existing template or writing a more personalized outline.

Even though the first option seems more reasonable, creating a modified and customized version of the SRS will lead to better results because it will lay out all the business' unique needs and requirements.

Companies can use the best of both worlds; they can use standard templates and add their project's required features.

One of the organizations that offers a standard SRS template is IEEE (Institute of Electrical and Electronics Engineers). IEEE is a professional association that develops and defines electronics and computer science standards.

An SRS document must contain the following sections, and its information must be organized in the following order according to IEEE standards:

1. Introduction
  - 1.1 Purpose
  - 1.2 Scope
  - 1.3 Definitions, acronyms, and abbreviations
  - 1.4 References
  - 1.5 Overview
2. Overall description
  - 2.1 Product perspective

2.2 Product functions

2.3 User characteristics

2.4 Constraints

2.5 Assumptions and dependencies

3. Specific requirements (functional, non-functional, and domain)

4. Appendixes and index

*If you want to learn more about SRS documents, check out [IEEE's Recommended Practice for Software Requirements Specifications \(pdf\)](#).*

Writing an SRS is a collaborative effort, though it has its difficulties.

Since an SRS document covers almost all of a project's aspects and concerns all the teams involved in it, everyone participates in its preparation process. From technical writers and business analysts to software and quality assurance engineers and investors and customers.

All the team members partake in preparing an SRS document through various techniques such as brainstorming, interviewing, mind mapping, etc.