How to Write a Software Requirements Specification (SRS Document)

What Is a Software Requirements Specification Document?

A software requirements specification (SRS) is a document that describes what the software will do and how it will be expected to perform.

An SRS describes the functionality the product needs to fulfill all stakeholders (business, users) needs.

A typical SRS includes:

- A purpose
- An overall description
- Specific requirements

The best SRS documents define how the software will interact when embedded in hardware — or when connected to other software. Good SRS documents also account for real-life users.

Why Use an SRS Document?

A software requirements specification is the basis for your entire project. It lays the framework that every team involved in development will follow.

It's used to provide critical information to multiple teams — development, quality assurance, operations, and maintenance. This keeps everyone on the same page.

Using the SRS helps to ensure requirements are fulfilled. And it can also help you make decisions about your product's lifecycle — for instance, when to retire a feature.

Writing an SRS can also minimize overall development time and costs. Embedded development teams especially benefit from using an SRS.

1. SRS Outline

1. Introduction

1.1 Purpose

1.2 Intended Audience

1.3 Intended Use

1.4 Product Scope

1.5 Risk Definitions

2. Overall Description

2.1 User Classes and Characteristics

2.2 User Needs

2.3 Operating Environment

2.4 Constraints

2.5 Assumptions

3. Requirements

    3.1 Functional Requirements

    3.2 Non Functional Requirements

Once you have your basic outline, you're ready to start filling it out.

2. Start with a Purpose

The introduction to your SRS is very important. It sets the expectation for the product you're building.

So, start by defining the purpose of your product.

Intended Audience and Intended Use

Define who in your organization will have access to the SRS — and how they should use it. This may include developers, testers, and project managers. It could also include stakeholders in other departments, including leadership teams, sales, and marketing.

Product Scope

Describe the software being specified. And include benefits, objectives, and goals. This should relate to overall business goals, especially if teams outside of development will have access to the SRS.

Definitions

It’s smart to include a risk definition. Avoiding risk is top-of-mind for many developers — especially those working on safety-critical development teams.

Here’s an example. If you’re creating a medical device, the risk might be the device fails and causes a fatality.

By defining that risk up front, it’s easier to determine the specific requirements you’ll need to mitigate it.

3. Give an Overview of What You’ll Build

Your next step is to give a description of what you’re going to build. Is it an update to an existing product? Is it a new product? Is it an add-on to a product you’ve already created?

These are important to describe upfront, so everyone knows what you’re building.

You should also describe why you’re building it and who it’s for.

User Needs and User Classes

User needs — or user classes and characteristics — are critical. You’ll need to define who is going to use the product and how.

You’ll have primary and secondary users who will use the product on a regular basis. You may also need to define the needs of a separate buyer of the product (who may not be a primary/secondary user). And, for example, if you’re building a medical device, you’ll need to describe the patient’s needs.

Assumptions

There might be factors that impact your ability to fulfill the requirements outlined in your SRS. What are those factors?

Are there any assumptions you’re making with the SRS that could turn out to be false? You should include those here, as well.

4. Detail Your Specific Requirements

The next section is key for your development team. This is where you detail the specific requirements for building your product.

Functional Requirements

Functional requirements are essential to building your product.

If you’re developing a medical device, these requirements may include infusion and battery. And within these functional requirements, you may have a subset of risks and requirements.

Write the functional requirements in **User Story** format (with confirmation part).

Nonfunctional Requirements

Nonfunctional requirements can be just as important as functional ones.

These include:

- Performance
- Safety
- Security
- Quality

The importance of this type of requirement may vary depending on your industry. Safety requirements, for example, will be critical in the medical device industry.