# **How to Write a Software Requirements Specification (SRS Document)**

REQUIREMENTS MANAGEMENT

SOFTWARE QUALITY

By [Charles Lane](https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document#authorcharleslane) and [Nico Krüger](https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document#authornicokr%C3%BCger)

Clear, concise, and executable [requirements](https://www.perforce.com/resources/alm/your-guide-successful-requirements-management) help development teams create a proper product. How do we organize and present these requirements? That's where a Software Requirements Specification (SRS) comes in. But what is an SRS, and when should you use one?

In this blog, we'll outline a typical software requirements specification, including how to define your product's purpose, describe what you're building, detail the requirements, and, finally, deliver it for approval. We'll also discuss the benefits of working through a requirement software vs. word.

* [What Is a Software Requirements Specification (SRS) Document?](https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document#what-is-a-software-requirements-specification-document)
* [Why Use an SRS Document?](https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document#why-use-an-srs-document)
* [Software Requirements Specification vs. System Requirements Specification](https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document#software-requirements-specification-vs-system-requirements-specification)
* [How to Write an SRS Document](https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document#how-to-write-an-srs-document)
* [Writing an SRS in Microsoft Word vs. Requirement Software](https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document#writing-an-srs-in-microsoft-word-vs-requirement-software)

**What Is a Software Requirements Specification (SRS) Document?**

A software requirements specification (SRS) is a document that describes what the software will do and how it will be expected to perform. It also describes the functionality the product needs to fulfill all stakeholders (business, users) needs.

An SRS can be simply summarized into four Ds:

* [Define your product's purpose.](https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document#purpose)
* [Describe what you're building.](https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document#describe)
* [Detail the requirements.](https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document#detail)
* [Deliver it for approval.](https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document#approve)

We want to DEFINE the purpose of our product, DESCRIBE what we are building, DETAIL the individual requirements, and DELIVER it for approval. A good SRS document will define everything from how software will interact when embedded in hardware to the expectations when connected to other software. An even better SRS documents also account for real-life users and human interaction.

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?**

An SRS gives you a complete picture of your entire project. It provides a single source of truth that every team involved in development will follow. It is your plan of action and keeps all your teams — from development to maintenance — on the same page (no pun intended).

This layout not only keeps your teams in sync but it also ensures that each requirement is hit. It can ultimately help you make vital decisions on your product’s lifecycle, such as when to retire an obsolete feature.

The time it takes to write an SRS is given back in the development phase. It allows for better understanding or your product, team, and the time it will take to complete.

**Software Requirements Specification vs. System Requirements Specification**

A **software requirements specification (SRS)** includes in-depth descriptions of the software that will be developed.

A **system requirements specification (SyRS)** collects information on the requirements for a system.

“Software” and “system” are sometimes used interchangeably as SRS. But, a software requirement specification provides greater detail than a system requirements specification.

*>> Need to prove compliance?*[*Here's how to create a traceability matrix >>*](https://www.perforce.com/blog/alm/how-create-traceability-matrix)

**How to Write an SRS Document**

Writing an SRS document is important. But it isn’t always easy to do.

Here are five steps you can follow to write an effective SRS document.

**1. Define the Purpose with an Outline (Or Use an SRS Template)**

Your first step is to create an outline for your software requirements specification. This may be something you create yourself. Or you may use an existing SRS template.

If you’re creating this yourself, here’s what your outline might look like:

**1. Introduction**

           1.1 Purpose

           1.2 Intended Audience

           1.3 Intended Use

           1.4 Scope

           1.5 Definitions and Acronyms

**2. Overall Description**

           2.1 User Needs

           2.2 Assumptions and Dependencies

**3. System Features and Requirements**

            3.1 Functional Requirements

            3.2 External Interface Requirements

            3.3 System Features

            3.4 Non-functional Requirements

**2. Define your Product’s Purpose**

This introduction is very important as it sets expectations that we will hit throughout the SRS.   
Some items to keep in mind when defining this purpose include:

*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. Defining this now will lead to less work in the future.

*Product Scope*

What are the benefits, objectives, and goals we intend to have for this product? This should relate to overall business goals, especially if teams outside of development will have access to the SRS.

*Definitions and Acronyms*

It’s important to define the risks in the project. What could go wrong? How do me mitigate these risks? Who is in charge of these risk items?  
For example, if the failure of a medical device would cause slight injury, that is one level of risk. Taking into account the occurrence level and the severity, we can come up with a strategy to mitigate this risk.

**3. Describe What You Will Build**

Your next step is to give a description of what you’re going to build. Is it a new product? Is it an add-on to a product you’ve already created? Is this going to integrate with another product? Why is this needed? Who is it for?

Understanding these questions on the front end makes creating the product much easier for all involved.

*User Needs*

Describe who will use the product and how. Understanding the user of the product and their needs is a critical part of the process.

Who will be using the product? Are they a primary or secondary user? Do you need to know about the purchaser of the product as well as the end user? In medical devices, you will also need to know the needs of the patient.

*Assumptions and Dependencies*

What are we assuming will be true? Understating and laying out these assumptions ahead of time will help with headaches later. Are we assuming current technology? Are we basing this on a Windows framework?  We need to take stock of these assumptions to better understand when our product would fail or not operate perfectly.

Finally, you should note if your project is dependent on any external factors. Are we reusing a bit of software from a previous project? This new project would then depend on that operating correctly and should be included.

**4. Detail Your Specific Requirements**

In order for your development team to meet the requirements properly, we MUST include as much detail as possible. This can feel overwhelming but becomes easier as you break down your requirements into categories. Some common categories are:

*Functional Requirements*

Functional requirements are essential to your product because, as they state, they provide some sort of functionality.

Asking yourself the question “does this add to my tool’s functionality?” Or “What function does this provide?” can help with this process. Within Medical devices especially, these functional requirements may have a subset of risks and requirements.

You may also have requirements that outline how your software will interact with other tools, which brings us to external interface requirements.

*External Interface Requirements*

External interface requirements are specific types of functional requirements. These are especially important when working with embedded systems. They outline how your product will interface with other components.

There are several types of interfaces you may have requirements for, including:

* User
* Hardware
* Software
* Communications

*System Features*

System features are types of functional requirements. These are features that are required in order for a system to function.

*Other Non-functional Requirements*

Non-functional 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. In the medical device industry, there are often regulations that require the tracking and accounting of safety.

[*IEEE*](https://ieeexplore.ieee.org/document/278253)*also provides guidance for writing software requirements specifications, if you’re a member.*

**5. Deliver for Approval**

We made it! After completing the SRS, you’ll need to get it approved by key stakeholders. This will require everyone to review the latest version of the document.

**Writing an SRS in Microsoft Word vs. Requirement Software**

You can write your software requirement specification in Microsoft Word. A smart way to do this is to create an SRS template that you can use as a starting point for every project.

However, even with a template, writing an SRS this way can be a painstaking process. And if a requirement changes, your SRS can fall easily out-of-date. Plus, there can be versioning issues with requirements documents in Word.

You can save time — and ensure accuracy — by writing an SRS in [Helix ALM](https://www.perforce.com/products/helix-alm) instead.

**Why Helix ALM Is Better…**

Helix ALM (which comes with a dedicated [requirements management tool](https://www.perforce.com/blog/alm/requirements-management-modern-tips-tactics-tools)) adds efficiency through your entire [requirements management](https://www.perforce.com/alm/requirements-management-modern-tips-tactics-tools) process.

By creating an SRS in Helix ALM, you’ll ensure a single source of truth on your SRS. It will be easier to do requirements reviews of your SRS. And that will help you get faster approvals — so your developers can get started.

Once you have requirements in an SRS, you can easily manage them throughout your development process.

If you’re also [writing a PRD](https://www.perforce.com/blog/alm/how-write-product-requirements-document-prd), you can link those feature requirements to the high-level requirement in the SRS. This creates traceability across your requirements process.

You can also link the requirements in your SRS to tests. This will help you ensure that the product you deliver fulfills the purpose and requirements of your SRS.

See for yourself how easy it can be to write an SRS. [Try Helix ALM free](https://www.perforce.com/products/helix-alm/free-alm-trial) — and see how an effective SRS will improve your development process. You can also watch our demo to see more functionality.

Assignment#1:

Discuss the IEEE prescribed SRS document layout/standard/guideline.

(22/Sep/2022; Thursday)