



[Downloads \(/downloads\)](/downloads)

[Integrations \(/integrations\)](/integrations)

[Blog \(/blog\)](/blog)

[Company](#) ▼

[Contact](#) ▼

 [Language](#) ▼

[Perforce continues to operate as usual to support you >>](https://www.perforce.com/covid-19-update)

[PERFORCE](https://www.perforce.com/covid-19-update)



October 23, 2018

How to Write a Software

Requirements Specification (SRS Document)

REQUIREMENTS MANAGEMENT | APPLICATION LIFECYCLE MANAGEMENT

By [Nico Krüger](#)

Clear requirements help development teams create the right product. And a software requirements specification (SRS) helps you lay the groundwork for product development.

We'll define what this is, when you'd use one, and five steps to writing an SRS Document.

At a glance, this is how to write a requirements document:

1. Make an outline.
2. Define the purpose of your product.
3. Describe what you're building.
4. Detail the requirements.
5. Get it approved.

What Is a Software Requirements Specification (SRS) Document?

An 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.

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.

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. Create 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 Nonfunctional 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 and Acronyms

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 upfront, 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

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 and Dependencies

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.

Finally, you should note if your project is dependent on any external factors. This might include software components you're reusing from another project.

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.

External Interface Requirements

External interface requirements are types of functional requirements. They're important for embedded systems. And 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 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.

IEEE (<https://ieeexplore.ieee.org/document/278253>) also provides guidance for writing software requirements specifications, if you're a member.

5. Get Approval for the SRS

Once you've completed the SRS, you'll need to get it approved by key stakeholders. And everyone should be reviewing the latest version of the document.

Writing an SRS in Microsoft Word vs. Helix ALM

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](/products/helix-requirements-management) (</products/helix-requirements-management>)) adds efficiency through your entire [requirements management](/alm/requirements-management-modern-tips-tactics-tools) (</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) (<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](/products/helix-alm/free-alm-trial) (</products/helix-alm/free-alm-trial>) — and see how an effective SRS will improve your development process. You can also sign up to watch our next live demo to see more functionality.

WRITE AN SRS IN HELIX ALM
([HTTPS://WWW.PERFORCE.COM/PRODUCTS/HELIX-REQUIREMENTS-MANAGEMENT/FREE-TRIAL](https://www.perforce.com/products/helix-requirements-management/free-trial))

CATCH THE NEXT DEMO
([HTTPS://WWW.PERFORCE.COM/PRODUCTS/HELIX-ALM/LIVE-DEMO](https://www.perforce.com/products/helix-alm/live-demo))

Nico Krüger

Chief Technology Officer, Helix
ALM, Perforce Software

Nico Krüger is CTO for Helix ALM. He specializes in technical guidance and product development efficiency for global organizations. An expert in software development, he is dedicated to helping customers drive quality product strategy.

Recommended Posts

[\(/blog/alm/10-symptoms-inefficient-requirements-management\)](/blog/alm/10-symptoms-inefficient-requirements-management)

December 18, 2019

10 Symptoms of Inefficiently Managing Requirements
(/blog/alm/10-symptoms-inefficient-requirements-management)

REQUIREMENTS
MANAGEMENT,
TRACEABILITY

[\(/blog/alm/requirements-management-modern-tips-tactics-tools\)](/blog/alm/requirements-management-modern-tips-tactics-tools)

March 21, 2019

Requirements Management: Modern Tips, Tactics, & Tools
(/blog/alm/requirements-management-modern-tips-tactics-tools)

REQUIREMENTS
MANAGEMENT,
APPLICATION
LIFECYCLE
MANAGEMENT

[\(/blog/alm/how-automate-helix-alm-remote-teams\)](/blog/alm/how-automate-helix-alm-remote-teams)

May 12, 2020

How to Automate Helix ALM for Remote Teams
(/blog/alm/how-automate-helix-alm-remote-teams)

APPLICATION
LIFECYCLE
MANAGEMENT

Helix Core Apps	Application	Events & Webinars	Request Support	Our Team	Core/Liv
(/Products/Helix-	Lifecycle	(/Resources/Events)	(/Support/Request-	(/Company/Management-	Helix ALI
Core-Apps)	Management	Recorded Webinars	Support)	Team)	(Https://,
Helix Plugins	(/Solutions/Application-	(/Resources/Recorded-	Video Tutorials		Alm/Live
(/Plugins-	Development-	Webinars)	(/Support/Video-	Our Culture	Hansoft
Integrations)	Lifecycle-	Blog (/Blog)	Tutorials)	(Https://Www.Perforce.Com	(Http://
	Management)	Subscribe	Perforce	Culture)	Overview
ENTERPRISE AGILE	Agile Project	(Http://Info.Perforce.Com/Subscription-	Center)	Careers (/Careers)	Subscrip
PLANNING	Management		(Https://Community.Perforce.Com/S/)	Press (/Press)	(Http://
Hansoft	(Https://Www.Perforce.Com/Products/Hansoft)	Agile-	Training	Contact (/Contact-	Custom
	Scrum-Project-		(/Support/Training)	Us)	Login
DEV COLLABORATION	Management-Tool)		Consulting		(Http://
Helix TeamHub	Backlog		(/Support/Consulting-	<u>PARTNERS</u> >	Licensing
(/Products/Helix-	Management		Support Plans		(/Suppo
Teamhub)	(/Solutions/Product-		(/Support/Support-	Integrations	Server-R
Helix Swarm	Backlog-		Plans)	(/Integrations)	Educatic
(/Products/Helix-	Management-Tool)		Maintenance	Resellers	Licenses
Swarm)	Project Portfolio		Support	(/Partners/Reseller-	(/Educat
Helix4Git	Management		(Https://Www.Perforce.Com/Partners)	Maintenance	Licenses
(/Products/Helix4git)	(/Solutions/Lean-		Support)	Partner Portal	How To I
	Project-Portfolio-		EOL Schedule	(/Partner/Login)	Buy)
DEVELOPMENT LIFECYCLE	Management-		(Https://Www.Perforce.Com/Maintenance-		
MANAGEMENT	Software)		Support#Eol_schedule)		
Helix ALM Suite:	Audit & Compliance		Release Notes		
(/Products/Helix-	(/Solutions/Audit-		(/Support/Release-		
Alm)	And-Compliance)		Notes)		
- Helix RM	DevOps		Download		
(/Products/Helix-	(/Solutions/Devops)		(/Downloads)		
Requirements-	Static Analysis	Copyright © 2020	Open Case		
Management)	(Https://Www.Perforce.Com/Solutions/Static-	Perforce Software	Status		
- Helix IM	Analysis)	Inc. All rights	OrgId=00DA00000009M15&PortalId=060F0000		
(/Products/Helix-		reserved.			
Issue-Management)		Sitemap			
- Helix TCM	Game Development	(/sitemap) Terms			
(/Products/Helix-	(/Solutions/Game-	of Use (/terms-			
Test-Case-	Development)	use) Privacy			
Management)	Life Sciences	Policy (/privacy-			
Surround SCM	(/Solutions/Life-	policy)			
(/Products/Surround-					

