Software Requirements Specification

Software Engineering (IT314)

Lab – 04 (21th February, 2022)

Course Instructor: Prof. Jayprakash Lalchandani

Index

- What is Software Requirements Specification ?
- Why to use SRS document ?
- Software Requirements Specification vs. System Requirements Specification
- How to write an SRS documents
- Writing an SRS in MS. Word vs. Requirements Software

What is Software Requirements Specification ?

- 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.
- A good SRS document will define everything from how software will interact when embedded in hardware to the expectations when connected to other software.
- SRS documents also account for real-life users and human interaction.

What is Software Requirements Specification ?

- An SRS can be simply summarized into four Ds:

- Define your product’s purpose
- Describe what you’re building
- Detail the requirements
- Deliver it for approval.

Why to use 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
- This layout not only keeps your teams in sync but it also ensures that each requirement is hit.
- It allows for better understanding or your product, team, and the time it will take to complete.

SRS vs SyRS

- **SRS**

- Software Requirements Specification
- SRS includes in-depth descriptions of the software that will be developed.
- SRS provides greater detail than a SyRS.

- **SyRS**

- System requirements specification
- SyRS collects information on the requirements for a system.

How to write an SRS documents

1. **Define the Purpose With an Outline**
2. **Define your Product’s Purpose**
3. **Describe What You Will Build**
4. **Detail Your Specific Requirements**
5. **Deliver for Approval**

1. Define the Purpose With an Outline

- Your first step is to create an outline for your software requirements specification.

1. Introduction
2. Overall Description
3. System Features and Requirements

1. Define the Purpose With an Outline

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

2. Define your Product’s Purpose

- It sets expectations that we will hit throughout the SRS.

1. Intended Audience and Intended Use
2. Product Scope
3. Definitions and Acronyms

1. Intended Audience and Intended Use

- Define who in your organization will have access to the SRS
- 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.

2. Product Scope

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

3. Definitions and Acronyms

- Define who in your organization will have access to the SRS
- 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.

3. Describe What You Will Build

- This next step is to give a description of what you're going to build.
- Questions on the front end makes creating the product much easier for all involved.
- 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?

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?

Assumptions and Dependencies

Assumptions :

- 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.

Dependencies

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

4. Detail Your Specific Requirements

- In order for your development team to meet the requirements properly, we must include as much detail as possible.
- It becomes easier as you break down your requirements into categories. Some common categories are:

1. Functional Requirements
2. Non-functional Requirements
3. External Interface Requirements
4. System Features

1. Functional Requirements

- Essential to your product
- They state, they provide some sort of functionality.
- functional requirements may have a subset of risks and requirements.
- Does this add to my tool's functionality?
- What function does this provide?
- You may also have requirements that outline how your software will interact with other tools, which brings us to external interface requirements.

2. Nonfunctional Requirements

- It can be just as important as functional ones.
- 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.

Requirements including:

- Performance
- Safety
- Security
- Quality

3. External Interface Requirements

- These are especially important when working with embedded systems.
- They outline how your product will interface with other components.

Requirements including:

- User
- Hardware
- Software
- Communications

4. System Features

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

5. Deliver for Approval

- 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.

SRS in MS. Word vs. Requirements Software

- You can write your SRS in Microsoft Word.

- You can save time and ensure accuracy by writing an SRS in SRS TOOL.