CREATING AN SRS DOCUMENT

MR. ANIKET MISHRA

ALL ABOUT SRS AND ITS NECESSITY

INTRODUCTION

- SRS: Software Requirements Specification

1. Software Requirements Specification (SRS) is a document that lays out the description of the software that is to be developed as well as the intention of the software under development.
2. Software requirements specification shows what the software is supposed to do as well as how it is supposed to perform. It is written down before the actual software development work starts.
3. An SRS contains the goals and measurable outcomes of the development process.
4. With a single glance at an SRS, any stakeholder understands what the finished product should be.

WHY SRS IS IMPORTANT?

Advantages				
Software requirements specification(SRS) is important for developers because it minimizes the amount of time and effort developers have to expend to achieve desired software goals. It thus reduces development cost.	benefits the client company because the lesser the development cost, the lesser the developers will charge from the client.	If composed properly, an SRS ensures that there is less possibility of future redesigns as there is less chance of mistake on the part of developers as they have a clear idea on the functionalities and externalities of the software.	It also helps clear any communication problems between the client and the developer. Furthermore, an SRS serves to form a foundation of mutual agreement between the client and the developer (supplier).	It also serves as the document to verify the testing processes.

WHO USES THE SRS DOCUMENT?

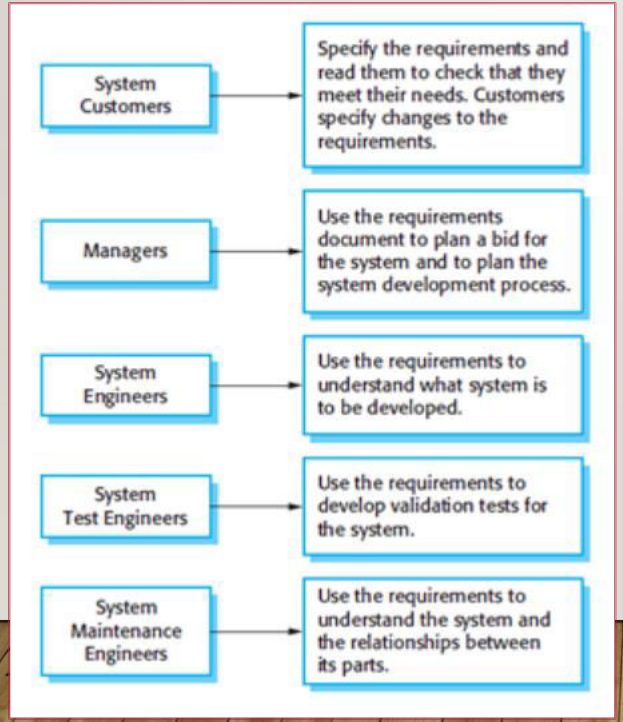

Fig 1. Users of a requirements document [SOMMERVILLE2010]

WHEN IS AN SRS SAID TO BE GOOD?
TRIES TO ADDRESS THE FOLLOWING:

IF IT

Functionality. What is the software supposed to do?

External interfaces. How does the software interact with people, the system's hardware, other hardware, and other software?

Performance. What is the speed, availability, response time, recovery time of various software functions, etc.?

Attributes. What are the portability, correctness, maintainability, security, etc. considerations?

Design constraints imposed on an implementation. Are there any required standards in effect, implementation language, policies for database integrity, resource limits, operating environment(s) etc.?

8 Characteristics of Good SRS: https://docs.google.com/document/d/1NeDqK4w0GGRO5zFPeKviqkzR65iCWQZkif7z5hGAP4A/edit?usp=sharing

TEMPLATE OF AN SRS DOCUMENT

FORMAT BY IEEE (INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS)

1. Introduction

1.1 Purpose

a) This gives the purpose of the SRS document, not the software itself.
b) It also states how much of the software is covered by the document, particularly saying whether it describes the entire software system or only a part of it.
c) It also states the intended readers of the document.

1.2 Document conventions

This covers whether the documentation follows any particular format.

CONTINUED..

1.3 Intended audience and reading suggestions

a) This segment lists the intended audience and provides reading suggestions.
b) For example, a document might be for both developers and project managers.
c) This part suggests how developers should read and also how project managers should read this document.
d) The sequence and importance of different segments might be different for developers and project managers.

CONTINUED..

1.4 Product Scope

a) This segment describes the software in brief. Its purpose, objectives etc.
b) It further states how the software ties into corporate aims and objectives.

1.5 References

This segment lists any and all websites and documents that were referred to in this document. The reference should be clear enough that the reader can access the referenced document or site after reading the reference list.

CONTINUED..

2. Overall description

2.1 Product Perspective

a) This part describes the context within which the product (software) is being built.
b) It also shows if the software is part of a product family, a replacement for an already existing member, or a completely new and unique product.
c) Also, if the SRS only describes a part of a larger system, then this part will lay out the requirements from the larger system for this part to operate effectively.
d) **A diagram should be drawn here to show the chief components of the entire system, the interlinking connections, and also the interfaces.**

CONTINUED..

2.2 Product Functions

a) A list of the major functions that product is to perform, or lets the users perform.
b) This should be readable and easily understandable by all the intended readers of the SRS.

2.3 User classes and characteristics

a) This part identifies the user classes that will use the product.
b) The classes may be categorized based on which functions they use, usage frequency, technical expertise, experience etc.
c) The relevant characteristics of each class should be given here. Also the most important user classes should be identified here.

CONTINUED..

2.4 Operating Environment

a) This segment describes the environment in which the product will operate, such as the platform (hardware) operating system version etc.
b) Also any other components the application will cohabit.

2.5 Design and Implementation Constraints

a) This segment lays out any constraints or issues that will limit options to developers.
b) Some of these might be hardware limitations, interfaces, particular technologies, parallel operations, language requirements, regulatory policies, corporate policies, security provisions etc.

CONTINUED..

2.6 User Documentation

This part shows the user manuals, tutorials etc that will be provided along with the software.

2.7 Assumptions and Dependencies

This part lists any assumptions that could affect the requirements stated in the SRS. The software would not work to the desired level if these assumptions are incorrect or change. Also listed here is any dependency the product will have on external factors.

CONTINUED..

External Interface Requirements

3.1 User Interfaces

a) This part describes the logical characteristics of each interface between the software and the user.
b) This might include screen image samples, Graphics User Interface standards, screen layout constraints, buttons and functions, keyboard shortcuts, message displays etc.
c) The software components for which user interface is need will also be defined.

3.2 Hardware Interfaces

This segment describes the logical and physical characteristics of each interface between the software and hardware components of the system.

CONTINUED..

3.3 Software Interfaces

a) This segment describes the connections between the product and any other specific software components, be it operating systems, databases libraries, etc.
b) Data items and messages going into the system and going is identified and the purpose of each is described.

3.4 Communications Interfaces

a) This part describes any communications functions required by the software, such as e-mail, web browser, network server communications protocols, electronic forms etc. Any communication standards to be used is also identified.
b) Communication security and/or encryption issues are specified.

4. System Features

This part lists the major services the product will provide. This part may be organized by use case, mode of operation, user class, object class, functional hierarchy, a combination of these etc.

An example might be:

4.1 System Feature 1 (basically here the name of the feature will be stated)

4.1.1 Description and Priority

4.1.2 Stimulus/Response Sequences

4.1.3 Functional Requirements

4.2 System Feature 2 (and so and so)

5. Other Non-functional Requirements

5.1 Performance Requirements

Requirements such as RAM requirements, CPU speed etc. Basically, these will ensure the software performs smoothly and without any problems

5.2 Safety Requirements

Requirements concerned with possible loss, damage or harm that could result from the use of the product are stated here. Any safeguards that must be implemented should also be defined. External policies or regulations stating safety issues that affect the product's design or use should be referred. Any safety certifications that must be satisfied is defined.

5.3 Security requirements

This part specifies any requirements regarding privacy and safety, as well as protection of data created or used by the product. Any user identity authentication requirements are stated here. Any safety certifications that needs to be satisfied is also defined.

5.4 Software Quality Attributes

Any additional quality characteristics of the product that might be important are specified here. Such as: adaptability, interoperability, availability, correctness, maintainability, robustness, usability, testability etc. These are written in a specific, verifiable and quantitative way.

CONTINUED..

5.5 Business Rules

Operating principles of the product are stated here. Such as which individuals can perform which functions under different circumstances.

6. Other requirements

Any requirements not stated in the SRS elsewhere are stated here. The requirements may include database requirements, legal requirements etc.

EXAMPLE OF SRS DOCUMENT

HTTPS://KRAZYTECH.COM/PROJECTS/SAMPLE-SOFTWARE-REQUIREMENTS-SPECIFICATIONSRS-REPORT-AIRLINE-DATABASE

THANKS!