1cLab summary2

Paul Contreras-Ortiz

04/14/2021

IEEE Recommended Practice for Software Requirements Specifications is an article written by the IEEE-SA Standards Board, and sponsored by the Software Engineering Standards Committee of the IEEE Computer Society. They provide an introduction that is not a part of the actual article, but it provides brief information on what a good SRS (software requirements specifications) should provide to the users. The introduction is followed by a list of participants to acknowledge their work in this practice, as well as a list of members that were on the balloting committee.

The scope of the article briefly discusses why this practice is recommended for specifying requirements of software and provides examples of other products it can be applied to like in-house or commercial. It also provides an example of a much more complex system like those found in medical equipment. Complex systems will require far more analysis to address any issues that are not included in this article. The scope is followed by a reference page to inform the reader that this practice is influenced by over a dozen publications made by the IEEE.

A couple definitions that are worth knowing are for the following words: contract, customer, supplier, and user. They are useful in reading a clause in the article, the Considerations for producing a good SRS. In this clause there are many subclauses that cover various topics. For example, the Nature of the SRS subclause covers what should be addressed with the customer. Information like functionality, performance, attributes within the SRS, design constraints, but also what should not be included, like design or project requirements.

A few of the subclauses are short in information but the information is important in clarifying what an SRS is and how and it should function. Subclause 4.3 Characteristics of a good SRS consists of adjectives or phrases that are used to describe what an SRS should be. Words like correct, unambiguous, complete, consistent, modifiable, and traceable are a few of the words that define a well-built SRS. A few subclauses that follow subclause 4.3 clarify all these words like correct and unambiguous and provide more information as to what it means for an SRS to be correct or ambiguous.

Subclauses like 4.3.4.1 cover how an SRS is internally consistent and what can disrupt it. A couple conflicts that can disrupt an SRS are conflict between two specified actions that is illogical, another is requirement characteristics being mixed and unclear like stating that turtles cannot swim but then further on stating that they can swim. Other subclauses provide opposite information, such as 4.3.5, 4.3.5.1 and 4.3.5.2 that revolve around stability. Stability in an SRS is having requirements that are identifiable, these three clauses cover several methods in identifying requirements.

Subclause 5, the Parts of an SRS consists of other subclauses that describe the many parts in an SRS. It provides an outline that can be followed, but does not have to be, it serves as an example of what an SRS consists of. A couple of the parts described are the introduction, a section of the SRS that contains five subsections the purpose, the scope, the overview, definitions, acronyms, and references. The five subsections are then discussed and elaborated thoroughly through their respective subclause, just like the characteristics of an SRS are elaborated through their own subclause.

The Annex A serves as a canvas filled with SRS templates, that show the many ways SRS sections can be organized. For example, some of the templates are organized by user class, by object, by different mode versions, by feature, and by stimulus. They are a visual aid that will help in structuring SRS sections. Annex B informs the reader of the guidelines for compliance of or with IEEE/EIA. It offers tables that can be used to understand the requirements for an SRS, requirements that were supported in other documents.

Discussion paragraph:

This article applies to the field of computer science because it covers how to handle and write an SRS before releasing it to customers. Such as styling it appropriately for the user to read; Including the necessary components describing what should be included and what should be left out and the reasons why. Other subclauses provided like 4.3.2.2 cover that the SRS should be written in a requirements specification language that can detect lexical, syntactical, and semantical errors. 4.3.2.1 covers methods used in detecting ambiguity so it can be corrected.