WIKIPEDIA

Software requirements specification

A **software requirements specification** (**SRS**) is a description of a software system to be developed. It lays out functional and non-functional requirements, and may include a set of use cases that describe user interactions that the software must provide.

Software requirements specification establishes the basis for an agreement between customers and contractors or suppliers (in market-driven projects, these roles may be played by the marketing and development divisions) on what the software product is to do as well as what it is not expected to do. Software requirements specification permits a rigorous assessment of requirements before design can begin and reduces later redesign. It should also provide a realistic basis for estimating product costs, risks, and schedules.[1] Used appropriately, software requirements specifications can help prevent software project failure.[2]

The software requirements specification document enlists enough and necessary requirements that are required for the project development.[3] To derive the requirements, the developer needs to have clear and thorough understanding of the products to be developed or being developed. This is achieved and refined with detailed and continuous communications with the project team and customer till the completion of the software.

The SRS may be one of a contract deliverable Data Item Descriptions[4] or have other forms of organizationally-mandated content.

Contents

Structure

An example organization of an SRS is as follows:[5]

- Purpose
  - Definitions
  - Background
  - System overview
  - References
- Overall description

  - Product perspective
    - System Interfaces
    - User interfaces
    - Hardware interfaces
    - Software interfaces
    - Communication Interfaces
    - Memory Constraints
  - Design constraints
    - Operations
    - Site Adaptation Requirements
  - Product functions
  - User characteristics
  - Constraints, assumptions and dependencies
- Specific requirements
  - External interface requirements
  - Functional requirements
  - Performance requirements
  - Logical database requirement
  - Software System attributes
    - Reliability
    - Availability
    - Security
    - Maintainability
    - Portability.
  - functional requirements
    - functional partitioning
    - functional description
    - control description
  - Environment characteristics
    - Hardware
    - peripherals
    - people
  - others..

Goals

The Software Requirements Specification (SRS) is a communication tool between stakeholders and software designers. The specific goals of the SRS are:

- Facilitating reviews
- Describing the scope of work
- Providing a reference to software designers (i.e. navigation aids, document structure)
- Providing a framework for testing primary and secondary use cases
- Including features to customer requirements

- Providing a platform for ongoing refinement (via incomplete specs or questions)

Requirements smells

Following the idea of code smells, the notion of *requirements smells* has been proposed to describe issues in requirements specification where the requirement is not necessarily wrong but could be problematic. In particular, the requirements smell :[6]

- is an indicator for a quality problem of a requirements artifact.
- does not necessarily lead to a defect and, thus, has to be judged by the context.
- has a concrete location in the requirements artifact itself, e.g. a word or a sequence.
- has a concrete detection mechanism (which can be automatic or manual and more or less accurate).

Examples of requirements smells are *Subjective Language*, *Ambiguous Adverbs and Adjectives*, *Superlatives* and *Negative Statements*.[6] Several of these smells can also be automatically detected with tools like *Requirements Scout (https://www.qualicen.de/en/products/scout.html)*.

See also

- Concept of operations
- Requirements engineering
- Software Engineering Body of Knowledge (SWEBOK)
- Design specification
- Specification (technical standard)
- Formal specification
- Abstract type

References

1. Bourque, P.; Fairley, R.E. (2014). "Guide to the Software Engineering Body of Knowledge (SWEBOK)" (http://www.computer.org/portal/web/swebok/v3guide). IEEE Computer Society. Retrieved 17 July 2014.
2. "Software requirements specification helps to protect IT projects from failure" (https://belitsoft.com/php-development-services/software-requirements-specification-helps-protect-it-projects-failure). Retrieved 19 December 2016.
3. Pressman, Roger (2010). *Software Engineering: A Practitioner's Approach*. Boston: McGraw Hill. p. 123. ISBN 9780073375977.
4. "DI-IPSC-81433A, DATA ITEM DESCRIPTION SOFTWARE REQUIREMENTS SPECIFICATION (SRS)" (http://www.everyspec.com/DATA-ITEM-DESC-DIDs/DI-IPSC/DI-IPSC-81433A_3709/). everyspec.com. 1999-12-15. Retrieved 2013-04-04.
5. Stellman, Andrew & Greene, Jennifer (2005). *Applied software project management*. O'Reilly Media, Inc. p. 308. ISBN 0596009488.
6. Femmer, Henning; Méndez Fernández, Daniel; Wagner, Stefan; Eder, Sebastian (2017). "Rapid quality assurance with Requirements Smells" (https://dx.doi.org/10.18419/opus-8479). *Journal of Systems and Software*. **123**: 190–213. doi:10.1016/j.jss.2016.02.047 (https://doi.org/10.1016%2Fj.jss.2016.02.047).

External links

- *830-1984 — IEEE Guide to Software Requirements Specifications*. 1984. doi:10.1109/IEEESTD.1984.119205 (https://doi.org/10.1109%2FIEEESTD.1984.119205). ISBN 0-7381-4418-5.
- *830-1993 — IEEE Recommended Practice for Software Requirements Specifications*. 1994. doi:10.1109/IEEESTD.1994.121431 (https://doi.org/10.1109%2FIEEESTD.1994.121431). ISBN 0-7381-4723-0.
- *830-1998 — IEEE Recommended Practice for Software Requirements Specifications*. 1998. doi:10.1109/IEEESTD.1998.88286 (https://doi.org/10.1109%2FIEEESTD.1998.88286). ISBN 0-7381-0332-2.
- *29148-2011 - Systems and software engineering — Life cycle processes — Requirements engineering* (http://ieeexplore.ieee.org/servlet/opac?punumber=6146377). 2011. pp. 1–94. doi:10.1109/IEEESTD.2011.6146379 (https://doi.org/10.1109%2FIEEESTD.2011.6146379). ISBN 978-0-7381-6591-2.("This standard replaces IEEE 830-1998, IEEE 1233-1998, IEEE 1362-1998 - http://standards.ieee.org/findstds/standard/29148-2011.html")
- Leffingwell, Dean; Widrig, Don (2003). *Managing Software Requirements: A Use Case Approach* (2nd ed.). Addison-Wesley. ISBN 032112247X.
- Gottesdiener, Ellen (2009). *The Software Requirements Memory Jogger: A Desktop Guide to Help Business and Technical Teams Develop and Manage Requirements*. Addison-Wesley. ISBN 157681114X.
- Wiegers, Karl; Beatty, Joy (2013). *Software Requirements, Third Edition*. Microsoft Press. ISBN 9780735679665.
- "IEEE SRS Template - rick4470/IEEE-SRS-Tempate" (https://github.com/rick4470/IEEE-SRS-Tempate). Retrieved 27 Dec 2017.

Retrieved from "https://en.wikipedia.org/w/index.php?title=Software_requirements_specification&oldid=819897873"

This page was last edited on 11 January 2018, at 22:09.