Software requirements specification

From Wikipedia, the free encyclopedia

A **software requirements specification** (SRS) is a description of a software system to be developed, laying out functional and non-functional requirements, and may include a set of use cases that describe interactions the users will have with the software.

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]

The software requirements specification document enlists enough and necessary requirements that are required for the project development.[2] To derive the requirements we need 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[3] or have other forms of organizationally-mandated content. An example organization of an SRS is as follows:[4]

- Introduction
  - Purpose
  - Definitions
  - System overview
  - References
- Overall description
  - Product perspective
    - System Interfaces
    - User Interfaces
    - Hardware interfaces
    - Software interfaces
    - Communication Interfaces
    - Memory Constraints
    - Operations
    - Site Adaptation Requirements
  - Product functions
  - User characteristics
  - Constraints, assumptions and dependencies

- Specific requirements
    - External interface requirements
    - Functional requirements
    - Performance requirements
    - Design constraints
        - Standards Compliance
    - Logical database requirement
    - Software System attributes
        - Reliability
        - Availability
        - Security
        - Maintainability
        - Portability
    - Other requirements

Contents

Write specifications to be readable and reviewable

One of the main values of writing specifications is to have them reviewed by stakeholders and to allow the stakeholders to provide feedback. Therefore, specifications should be written in such a way that they can easily be read and reviewed.

Some of the questions to ask yourself about readability is:

- Does the specification contain a high-level description of the scope of the work?
- Does the formatting allow a reader to easily navigate the specification?
- Does the formatting allow a reader to easily understand the primary versus secondary cases?
- Does the specification contain links to related features?
- Does the specification call out questions which still need to be addressed?

See also

- Concept of operations
- Requirements engineering
- Software Engineering Body of Knowledge (SWEBOK)

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. Pressman, Roger (2010). *Software Engineering: A Practitioner's Approach*. Boston: McGraw Hill. p. 123. ISBN 9780073375977.
3. "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.
4. Stellman, Andrew and Greene, Jennifer (2005). *Applied software project management*. O'Reilly Media, Inc. p. 308. ISBN 0596009488.

External links

- *830-1984 — IEEE Guide to Software Requirements Specifications*. 1984. doi:10.1109/IEEESTD.1984.119205 (https://dx.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://dx.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://dx.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://dx.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.

- Software Requirements Specification Writing Service (http://developex.com/srs-writing-service)

Retrieved from "http://en.wikipedia.org/w/index.php?title=Software_requirements_specification&oldid=648604784"

Categories: Software requirements | Software documentation | IEEE standards

- This page was last modified on 3 April 2015, at 14:20.
-