SRS Checklist

Spencer Smith

August 18, 2025

This checklist is intended to help with a high-quality SRS document.

- Follows writing checklist (full checklist provided in a separate document)
  - ☐ LaTeX points
  - ☐ Structure
  - ☐ Spelling, grammar, attention to detail
  - ☐ Avoid low information content phrases (like replacing "in order to" with "to")
  - ☐ Writing style
- Follows the template, all parts present
  - ☐ Selected template is appropriate for the project
  - ☐ Unused template folders are deleted from your repo
  - ☐ File name for the SRS matches the name in the template repo
  - ☐ Table of contents
  - ☐ Pages are numbered
  - ☐ Revision history included for major revisions
  - ☐ Sections from template are all present
  - ☐ Values of auxiliary constants are given (constants are used to improve maintainability and to increase understandability)
  - ☐ Symbolic names are used for quantities, rather than literal values

- Overall qualities of documentation
  - ☐ No statement is repeated at the same level of abstraction (for instance the scope should be more abstract than the assumptions, the goal statements should be more abstract than the requirements, etc.)
  - ☐ Someone that meets the characteristics of the intended reader could learn what they need to know
  - ☐ Someone that meets the characteristics of the intended reader could verify all of the statement made in the SRS. That is, they do not have to trust the SRS authors on any information.
  - ☐ SRS is unambiguous. At least check a representative sample.
  - ☐ SRS is consistent. At least check a representative sample.
  - ☐ SRS is validatable. At least check a representative sample.
  - ☐ SRS is abstract. At least check a representative sample.
  - ☐ SRS is traceable. At least check a representative sample.
  - ☐ Literal symbols (like numbers) do not appear, instead being represented by SYMBOLIC_CONSTANTS (constants are given in a table in the Appendix)
- Reference Material
  - ☐ All units introduced are listed (searching the document can help look for other units that may be present, but not listed)
  - ☐ All symbols used in the document are listed
  - ☐ All symbols listed are used in the document
  - ☐ All abbreviations/acronyms used in the document are listed
  - ☐ All abbreviations/acronyms listed are used in the document
- Functional Requirements
  - ☐ All requirements are validatable
  - ☐ All requirements are abstract
  - ☐ Requirements are traceable to where the required details are found in the document

- Nonfunctional Requirements
  - ☐ NFRs are verifiable
  - ☐ Usability used for users and understandability used for programmers
  - ☐ Specify what you want, not how to achieve it (for instance, don't say how you will make the software maintainable via modularization, say how you will measure maintainability and your target)
  - ☐ NFRs point to the VnV plan for details as appropriate
- Requirements
  - ☐ Requirements should trace to IMs
  - ☐ Rationale is provided for assumptions, scope decisions and constraints
- Likely and Unlikely changes
  - ☐ Likely changes are feasible to hide in the design

Other checklists to consider can be found in the resources for the University of Toronto course CSC340F include:

- Checklist for Requirements Specification Reviews
- Software Requirements Checklist (JPL)

References