SRS and CA Checklist

Spencer Smith

June 12, 2023

This checklist is specific to the Smith et al template (Smith and Lai, 2005; Smith et al., 2007) for documenting requirements for scientific software, but many of the points can be abstracted and applied to other templates.

- Follows writing checklist (full checklist provided in a separate document)
  - ☐ LaTeX points
  - ☐ Structure
  - ☐ Spelling, grammar, attention to detail
  - ☐ Avoid low information content phrases
  - ☐ Writing style
- Follows the template, all parts present
  - ☐ 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.
- ☐ Terminology, definitions, symbols, TMs and DDs can be given without derivation, except possibly for a source (citation), but all GDs and IMs should be derived/justified. At least check a representative sample for this criteria.
- ☐ 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)
  - ☐ Units listed are each used at least once (manually searching the document is a quick way to check this)
  - ☐ The names of units named after people are in lower-case
  - ☐ All symbols used in the document are listed in the table of symbols
  - ☐ All symbols listed in the table of symbols are used in the document
  - ☐ All abbreviations/acronyms used in the document are listed in the table of abbreviations/acronyms

- ☐ All abbreviations/acronyms listed in the table of abbreviations/acronyms are used in the document
- ☐ If a domain specific notation will be used, it has been defined in the mathematical notation section

- Introduction
  - ☐ Introductory blurb focuses on the problem domain
  - ☐ Introductory blurb Includes a "roadmap"
  - ☐ "Purpose of the Document" discusses the documentation's purpose, not the program's purpose
  - ☐ Scope of the requirements is an abstract version of the assumptions. Every item of the scope should be reflected in at least one assumption.
  - ☐ Characteristics of the intended reader are not confused with the user characteristics
  - ☐ Characteristics of the intended reader are unambiguous

- General System Description
  - ☐ System context includes a figure showing the relation between the software system and external entities
  - ☐ If the software will depend on other software, such as other libraries, this is part of the system context. Try to keep the libraries generic, unless specific libraries are needed, which will mean software constaints are also specified.
  - ☐ User characteristics are unambiguous (for instance, don't just say the user will know physics, say they will know Newtonian mechanics as typically covered in the first year of an engineering or science degree)
  - ☐ User characteristics are specific
  - ☐ System constraints have an appropriate rationale (a constraint without a reason for that constraint is likely making the SRS less abstract than it should be)

- Problem Description

- ☐ Each item of the physical system is identified and labelled
- ☐ Goal statements are abstract
- ☐ Goal statements use a minimal amount of technical language, understandable by non-domain experts

- Solution Characteristics Specification
  - ☐ Each assumption is "atomic" (no explicit or implicit "ands")
  - ☐ Assumptions are a refinement of the scope
  - ☐ Each assumption is referenced at least once in the document
  - ☐ If an assumption is listed as being referenced by another chunk (T, IM etc), that other chunk should explicitly invoke the assumption in the describing text or derivation
  - ☐ A link exists between each chunk and anything that references it
  - ☐ If the "Ref. By" field is filled in, the entities (model, definition, assumption) listed explicitly include a reference to the original entity (model, definition, assumption).
  - ☐ The rationale is given for assumptions that require justification
  - ☐ The derivation of all GDs as refinements from other models is clear
  - ☐ The derivation of all IMs as refinements from other models is clear
  - ☐ All DD are used (referenced) by at least one other model
  - ☐ The IMs remain abstract
  - ☐ All of the inputs for an IM are used in some way to define the output for the IM
  - ☐ Input data constraints are given, with a rationale where appropriate
  - ☐ Properties of a correct solution are given (or explicitly left blank)
  - ☐ Equations are balanced with respect to units of all terms

- Functional Requirements
  - ☐ IMs and (possibly) TMs and GMs are referenced as appropriate by the requirements. It is a sign that the IMs are not set correctly if there is one or more IMs that are not referenced by any of the 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
- Likely and Unlikely changes
  - ☐ Likely changes are feasible to hide in the design
- Traceability Matrices
  - ☐ Traceability matrix is complete

References

W. Spencer Smith and Lei Lai. A new requirements template for scientific computing. In J. Ralyté, P. Ågerfalk, and N. Kraiem, editors, *Proceedings of the First International Workshop on Situational Requirements Engineering Processes – Methods, Techniques and Tools to Support Situation-Specific Requirements Engineering Processes, SREP'05*, pages 107–121, Paris, France, 2005. In conjunction with 13th IEEE International Requirements Engineering Conference.

W. Spencer Smith, Lei Lai, and Ridha Khedri. Requirements analysis for engineering computation: A systematic approach for improving software reliability. *Reliable Computing, Special Issue on Reliable Engineering Computation*, 13(1):83–107, February 2007.