Declarative Requirements

Requirements may be represented in many ways - as high-level business needs (e.g., "increase revenue"), or as functional requirements in context (e.g., as through a use case, process flow, or user story), or as individual declarative statements (e.g., "the system shall..."). The following standards address this last category of requirements.

Most of the material in this page is adapted from Writing High Quality Requirements, an article by Karl E. Wiegers, author of the book, More about Software Requirements. The article is available from Jama Software here.

Standards for a Declarative Requirement

"The system shall permit the user to select up to three sort columns and redisplay the list in sorted order."

This example of a requirement might be found in a Request for Proposal specification for spreadsheet software. The example represents a "declarative" requirement - one in which the requirement is explicitly declared. Note the use of "shall". Note as well how the requirement stands on its own - it has little or no context or relationship with other stated requirements.

The following standards may assist you in developing high-quality declarative requirements.

1. Use either "shall" or "must" as the declarative verb uniformly throughout the requirements document.
2. Avoid use of "should", "may", or "could" as the declarative verb. Instead of using a shall-should-may convention, prioritize each requirement (e.g., "high", "medium", or "low").
3. Write requirements using active verbs (e.g., "When the user cancels an order, the system shall log the event") and avoid use of the verb "is" (e.g., "When the user cancels an order, the event is logged").
4. Write requirements in the format of condition, result, and qualifier (e.g., "When the user cancels an order, the system shall log the event and display a confirmation message to the user"). Note that a condition phrase or qualifier phrase may not be necessary in every requirement.
5. Group requirements logically.
6. Use a decision tree or a decision table when documenting requirements involving complex logic. Refer to Decision Analysis Using Decision Tables and Business Rules (page 22 and following) for an introduction to decision tables.
7. Restate any negative requirements as a positive requirement; restate any "double negative" requirements (e.g., "The on-line application process shall not allow unauthorized users" re-stated as "Only authorized users shall access the on-line application").
8. Include enough information so that the requirement may stand on its own (e.g., the circumstances are unclear for the following requirement: "The system shall generate an error message." A better requirement: "When the address State or Zip Code is invalid, the system shall generate an error message.").
9. Resolve symmetrical omissions in requirements. "The system shall allow the user to expand the Bill of Materials item list" is more completely stated as "The system shall allow the user to expand and contract the Bill of Materials item list".
10. Address all requirement boundaries. Complex business rules may fail to address all possibilities, particularly if the condition which invokes the rule is complex. Consider: "If the Address State value is "WI", the System shall automatically apply the appropriate state tax." What if the Address State value is not "WI", or if the value is "wi" instead? Address boundary conflicts and overlaps as well.
11. Define any term that may be unfamiliar to the reader in the glossary.
12. Use pronouns (e.g., he, she, it, and "this" and "that") carefully. Using these words often reduces clarity in the requirement.
13. Be precise. For example, including the term "audit/history table" may make the requirement unclear. Will the table be used for "audit" or for "history", or for both? Does "audit" serve to qualify "history table"?
14. Watch for and eliminate use of similar sounding words (e.g., "affect" and "effect"). A list is available at Common Errors in English Usage.
15. Avoid use of adverbs which are open to interpretation by the reader. These words typically end in "ly" such as "reasonably", "timely", and "nearly".

Best Practice

Ask a peer to read the requirements you've written. Even the best business analyst serves his or her customers best by asking for someone to review the work.