

Requirements Verification Checklist

There are two checklists included in this document. First, there is a checklist for each individual requirement or user function. *Each* requirement must have the properties in the checklist. Second, there is a checklist for the requirements document containing many requirements. *This set* of requirements must have the properties in the second checklist.

Checklist of Individual Requirements:

1. *Complete.*
The information covers all aspects of this user function.
2. *Consistent.*
The information for this user function is internally consistent. There are no conflicting statements.
3. *Correct.*
The item is free from error.
4. *Precise, unambiguous, and clear.*
The item is exact and not vague; there is a single interpretation; the meaning of each item is understood; the specification is easy to read. Use well defined terms and enough information. Terms such as “high”, “low”, “quick”, etc. are unacceptable. Define all terms and use measurable quantities.
5. *Relevant.*
The item is pertinent to the problem and its solution.
6. *Testable.*
Can the requirement be tested? If so, are the test cases defined. During program development and acceptance testing, it will be possible to determine whether the item has been satisfied.
7. *Understandable.*
Difficult language should not be used. Be clear. If formal notations are used, make sure they are easy to understand. Figures and tables are helpful, but must be explained.
8. *Expressed in the User's Language.*
All requirements shall be expressed using no design and in the users terminology.
9. *Traceable.*
The item can be traced to its origin in the problem environment. Also, you must be able to find who originated a requirement, who wrote it, what other requirements does this one impact, etc. This information is essential to make a document usable and changeable.
10. *Feasible.*
The item can be implemented with the available techniques, tools, resources, and personnel, and within the specified cost and schedule constraints

11. *Prioritized.*

The priority of this item relative other requirements has been recorded.

12. *Classified for Stability*

Is this requirement stable or volatile. This is essential to know when planning for change and when other requirements rely on a volatile requirement.

13. *Free of unwarranted design detail.*

The requirements specifications are a statement of the requirements that must be satisfied by the problem solution, and they are not obscured by proposed solutions to the problem.

The following guidelines apply to the whole requirements document and all the requirements and constraints therein.

Checklist for Full Document

14. *Complete.*

All items needed to specify the solution to the problem have been included. Make sure you cover all cases. If you have defined what to do in the normal case, make sure you cover all abnormal cases. For example, if you have defined what to do if there are funds in the account, you have to define what to do if the account is missing, if there are no funds, etc.

15. *Consistent.*

No item conflicts with another item in the specification. A good document structure may help here. Related (and possibly conflicting) requirements should be kept together.

16. *Manageable.*

The requirements are expressed in such a way that each item can be changed without excessive impact on other items.

17. *Modifiable*

Structure the requirements so they are easy to change and update.

18. *Environment and Document Scope is Defined*

The system boundary and all interfaces to the environment must be clearly defined. This will help you avoid requirements creep later in the process.

When structuring the document, also make sure that changes to the completed requirements specifications can be controlled; each proposed change can be traced to an existing requirement; and the impact of the proposed change can be assessed.

Keeping your requirements document under version control is an absolute must!

Requirements Definition Checklist

Using natural language for the requirements statements may lead to ambiguous, confusing, and contradictory requirements. Here are some methods of transforming requirements statements to reveal ambiguities, errors, and/or misunderstandings:

1. Vary the stress pattern in a sentence to reveal possible alternative meanings.
2. Put yourself in the designers' shoes. Could you commit to have this requirement implemented by next Tuesday with the information included?
3. When a term is defined explicitly somewhere, try substituting that definition in place of the term.
4. When a structure is described in words, try to sketch a picture of the structure being described.
5. When a picture describes a structure, try to redraw the picture in a form that emphasizes different aspects of the structure.
6. When there is an equation, try expressing the meaning of the equation in words.
7. When a calculation is specified or implied in words, try expressing it in an equation.
8. When a calculation is specified, work at least two examples by hand and give them as examples in the specification.
9. Look for statements that in any way imply **certainty** and then ask for proof. Words such as **always**, **everywhere**, **every all**, **none**, and **never** are useful clues to indicate unproved certainty.
10. Be on the lookout for words that are supposed to be **persuasive** and used to convince the reader of some point, such as **certainly**, **therefore**, **clearly**, **obviously**, or **as any fool can clearly see**.
11. Watch for **vague** words, such as **some**, **sometimes**, **often**, **usually**, **ordinarily**, **customarily**, **most**, or **mostly**.
12. Watch for non-committal words such as **should**, **ought to**, **preferred**, **desirable**, and **wanted**.
13. When lists are given, but not completed, make sure that there is a complete understanding of the nature of the subsequent items. Watch out for **etc.**, **and so forth**, **and so on**, or **such as**.
14. In attempting to clarify lists, as in (13), we sometimes state a rule. Be sure that the rule does not contain unstated assumptions.
15. Look for lists **without examples** or examples that are **too few** or **too similar** to each other to explicate the rule.
16. Beware of vague verbs such as **handled**, **processed**, **rejected**, **skipped**, or **eliminated**.
17. **Passive voice** constructions are also traps. Since the passive voice does not name an actor, it is easy to overlook who is doing the work.
18. Be especially on the lookout for **comparatives without referents**.
19. Pronouns are often clear to the writer and not to the reader.