Guidelines for Good Requirements Writing with Examples

Technical Report · May 2014

DOI: 10.13140/RG.2.1.5135.8168

1 author:

Naoufel Boulila
Siemens, Munich. Germany
45 PUBLICATIONS **74** CITATIONS

SEE PROFILE

Guidelines for Writing Good Requirements

Dr. Naoufel Boulila (naoufel.boulila@siemens.com)

Siemens Corporate Technology, Munich, 2015

naoufel.boulila@siemens.com

Guidelines for Writing Good Requirements

Goal

The goal of these guidelines is to provide few non-exhaustive rules to consider in writing requirements. In combination with the quality aspects of the single requirements, these guidelines provide a basic ground to achieve quality requirement specifications.

Outline

1. Use simple direct sentences
2. Use a limited vocabulary
3. Identify the type of user who needs the requirement
4. Focus on stating results
5. Define verifiable criteria
6. Avoid ambiguity
7. Avoid making multiple requirements
8. Avoid building in let-out clauses
9. Avoid designing the system
10. Avoid mixing requirements and design
11. Avoid mixing requirements and plan
12. Avoid speculation
13. Avoid playing on ambiguous requirements
14. Avoid using vague, indefinable terms
15. Avoid rambling
16. Avoid expressing possibilities
17. Avoid wishful thinking
18. Avoid Negative Specification

naoufel.boulila@siemens.com

Guidelines for Good Requirements 1

Use simple direct sentences

Every requirement should be a single active sentence, as short as possible – but no shorter

Example

The pilot shall be able to view the airspeed

naoufel.boulila@siemens.com

Guidelines for Good Requirements 2

Use a limited vocabulary

Write simple subset of English avoiding terms that many confuse non-technical or foreign readers

Example

- **Not good**: The airline shall be able to reconfigure conventional global business/global traveler seating in less than half a day (see FAA rules)
- **Better**: The airline shall be able to change the aircraft's seating from business to holidays charter use in less than 12 hours

naoufel.boulila@siemens.com

Guidelines for Good Requirements 3

Identify the type of user who needs the requirement

Every requirement should start by naming a class of user

Example

The **navigator** shall be able to …

naoufel.boulila@siemens.com

Guidelines for Good Requirements 4

Focus on stating results

Every requirement should name a single desired result. In a capability or affordance, this is something to be provided to the named class of user

Example

The navigator shall be able to **view storm clouds by radar**

naoufel.boulila@siemens.com

Guidelines for Good Requirements 5

Define verifiable criteria

- Every requirement must be verifiable
- You can indicate a possible test by adding a simple phrase to connect a specific criterion to the requirement.
- In a later step, the specific criterion can be extended to an acceptance criterion
- You do not necessarily have to write acceptance test criteria while preparing user requirements. However, for practical tests, verification criterion has to be defined. if it isn't verifiable, it isn't a requirement

Example

- **Simple Criterion**: The navigator shall be able to view storm clouds by radar **at least 100 km ahead**
- **Acceptance Criterion**: Aircraft flying at 800 km/h 10,000 meters towards a storm cloud identified by satellite; **storm cloud is detected at a range of at least 100km**

naoufel.boulila@siemens.com

Guidelines for Good Requirements 6

Avoid ambiguity

One of the most subtle and difficult issues in writing requirements

- Write clearly and explicitly
- Informal text, scribbled diagrams, conversations, phone calls can help removing ambiguity
- Dangerous ambiguities can be caused by the word “or” for instance

Example

The same subsystem shall also be able to generate a visible or audible caution/warning signal for the attention of the co-pilot or navigator

Issues

Which subsystem? Is the signal to be visible, audible or both? Is it both caution and warning, just caution, or just warning? …attention of co-pilot or navigator or both ?

naoufel.boulila@siemens.com

Guidelines for Good Requirements 7

Avoid making multiple requirements

Requirements which contains conjunctions such as “and”, “or”, “with”, “also” are dangerous and misleading

Example

The battery low warning lamp shall light up when the voltage drops below 3.6 volts, and the current workspace or input data shall be saved

Issues

How many requirements is that, what should be saved, and when?

naoufel.boulila@siemens.com

Guidelines for Good Requirements 8

Avoid building in let-out clauses

Requirements which contains let-out such as “if”, “when”, “but”, “except”, “unless”, “although”, are dangerous and misleading

Example

- The forward passenger doors shall open automatically when the aircraft has halted, except when the rear ramp is deployed
- The fire alarm shall always be sounded when smoke is detected, unless the alarm is being tested or the engineering has suppressed the alarm

Issues

There are two or more scenarios in both examples. The let-outs are trying to cover alternative cases. Handle as separate requirements

naoufel.boulila@siemens.com

Guidelines for Good Requirements 9

Avoid designing the system

- Requirements specify the design envelope, and too much detail will start to design the system, especially when it comes to our favorite areas.
- Danger signs: names of components, materials, software objects/procedures, database fields.

Example

The antenna shall be capable of receiving FM signals, using a cupper core with nylon armoring and water proof hardened rubber shield

Issues

Specifying design rather than actual need often increases the cost of systems by placing needless constraints on development and manufacture.
In this example there is indication that antenna should include a stronger reception and a water proof capabilities. Maybe the same goals can be achieved using a better material than cupper for less price..

naoufel.boulila@siemens.com

Guidelines for Good Requirements 10

Avoid mixing requirements and design

- The user requirements form a complete model of what users want
- System specifications form a complete model of the proposed system
- Confusion may happen when mixing up user requirements, system specifications, design elements, test cases, development guidelines, and installation instructions.

Example

The user shall be able to view the currently selected channel number which shall be displayed in 14pt Swiss type on an LCD panel tested to Federal regulation Standard 567-89 and mounted with shockproof rubber washers.

Issues

Mixes needs and a display constraints. The choice of an LCD Panel can be left until system design. LCD robustness might be considered as a constraint

naoufel.boulila@siemens.com

Guidelines for Good Requirements 11

Avoid mixing requirements and plan

- Mixing up requirements and plans and schedules is source of trouble.
- Plans are essential but do not belong in the requirements.
- Mixing both tends to increase the requirements workload through extra revisions and review meetings.
- Usually they are updated throughout the project whereas the requirements should stabilize.
- Danger signs: dates, project phases, and development activities

Example

The channel display type - LCD, LED, or TFT- shall be selected by 15 March and the first prototype panel shall be available for testing by the start of phase 3.

Issues

Although the statement is presented as a requirement, it is not!

naoufel.boulila@siemens.com

Guidelines for Good Requirements 12

Avoid speculation

- Requirements are part of a contract between customer and developer, with users as interested third parties.
- There is no room for “wish lists” – general terms about things that somebody probably wants.
- Danger signs include vagueness about which type of users is speaking and generalization words: usually, generally, often, normally, typically

Example

Users normally require early indication of intrusion into the system

Issues

This statement does not place any constraint or requirement on the system, though leaves the decision to the system developer to add that feature or suppress it, since he/she can think of a scenario where this is not the “normally” case.

naoufel.boulila@siemens.com

Guidelines for Good Requirements 13

Avoid playing on ambiguous requirements

- The use of “or” and “unless” in requirements, allow different groups of readers to understand different things from the same wording.
- A development manager could attempt to postpone, until too late any possibility of the customer’s asking for what was wanted
- A customer could hope to mislead the supplier into offering to complete a task at a lower price.

Example

Operators shall be able to back up any disk on to a high-speed removable disk drive or tape cartridge

Issues

This could be interpreted to mean “ a high-speed drive, or a high-speed tape cartridge” which is presumably what the users want; it could also mean “ a high-speed drive, or any sort of tape cartridge,” which might be cheaper for the developer to supply if users’ whishes can be ignored.

naoufel.boulila@siemens.com

Guidelines for Good Requirements 14

Avoid using vague, indefinable terms

- Many words used informally to indicate desired qualities are too vague for use in requirements.
- Vague terms include: user-friendly, versatile, flexible, approximately, as possible, efficient, improved, high performance, modern.
- Requirements using these terms are unverifiable because there is no definite test to show whether the system has the indicated properly.

Example

- The print dialog shall be versatile and user-friendly
- The OK status indicator lamp shall be illuminated as soon as possible after the system self-check is completed.

Issues

What is versatile? and user-friendly?

naoufel.boulila@siemens.com

Guidelines for Good Requirements 15

Avoid rambling

Long rambling sentences quickly lead to confusion and error.

Example

Provided that the designated input signals from the specified devices are received in the correct order where the system is able to differentiate the designators, the output signal shall comply with the required framework of section 3.1.5 to indicate the desired input state

Issues

A lot of unnecessary information in this example, that only causes confusion and eventually a misinterpretation of main goal.

naoufel.boulila@siemens.com

Guidelines for Good Requirements 16

Avoid expressing possibilities

- Possibilities are indicated with terms such as: may, might, should, ought, perhaps, probably.
- If developers do only what they have to, they will always ignore things that users might possibly require

Example

The reception subsystem probably ought to be sensitive enough to receive a signal inside a steel-framed building

Issues

A design decision based on the above statement could lead to the wrong system. As probably provide a reason for the developer to completely ignore it. That's less work and efforts, but leads to the wrong system.

naoufel.boulila@siemens.com

Guidelines for Good Requirements 17

Avoid wishful thinking

- Engineering is real-world activity.
- No system or component is perfect
- Wishful thinking means asking for the impossible
- Developers will rightly reject wishful requirements
- wishful terms include: 100% reliable, Safe, Handle all unexpected failures, please all users, run on all platforms, never fail, upgradeable to all future situations

Example

- The gearbox shall be 100% safe in normal operation
- The network shall handle all unexpected errors without crashing

Issues

What is 100% safety ? How to handle unexpected errors?

naoufel.boulila@siemens.com

Guidelines for Good Requirements 18

Avoid Negative Specification

- Use negative specification sparingly, for emphasis
- Don't use negative specification for requirements that could be stated in the positive

Example

- Users shall not be prevented from deleting data they have entered

Issues

Negative Specification.

- The system shall provide users with the ability to delete data they have entered

naoufel.boulila@siemens.com