Guidelines for Writing Requirements

OOAD

Chea Daly

Guidelines for Writing Requirements

- Feasible
- Keep it Short and Clear
- Use Complete Sentence
- Unambiguous
- Avoid Generalization
- Verifiable
- Prioritized
- Consistent
- …

Guidelines for Writing Requirements

Feasible (Realistic, Possible)

- Requirements are part of a contract between customer and developer.
- There is no room for "wish lists" – general terms about things that some body probably wants.

- Example:
  - The network shall handle all unexpected errors without crashing.
    - How to handle unexpected errors?

Guidelines for Writing Requirements

Keep it Short and Clear

- Requirements should be as short as possible.
- They should be stated clearly and simply.
- Long paragraphs risk ambiguity and confusion.

- Example:
  - Sometimes the user will enter Airport Code, which the system will understand, but sometimes the closest city may replace it, so the user does not need to know what the airport code is, and it will still be understood by the system.

This sentence may be replaced by a simpler one:

- The system shall identify the airport based on either an Airport Code or a City Name.

Guidelines for Writing Requirements

Use Complete Sentence

- Every requirement should contain a **subject** and a **predicate.**
  - **Subject**: an actor (who/that performs the action).
  - **Imperatives**: Example: "**shall**" / "**will**" / "**must**" to show **mandatory nature**; "**may**" / "**should**" to show **optionality**.
  - **Predicate**: a condition, action, or intended result.

- Example: The **pilot shall** be able to **view the airspeed**.

Guidelines for Writing Requirements

Unambiguous

- Reader should be able to draw only one interpretation.
- Avoid using subjective words (easy, simple, quick, efficient, etc.)

- Example: The system shall be able to **quickly** generate a visible warning signal for the co-pilot **or** navigator.
  - How quick? (in less than 5 seconds?)
  - attention of co-pilot or navigator or both?

Guidelines for Writing Requirements

Avoid Generalization

- Generalization words: usually, generally, often, normally, typically, probably, etc.

- Example: Users **normally** require early indication of intrusion into the system.
  - 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 not.

Guidelines for Writing Requirements

Verifiable (Testable)

- Contains a success criterion/other measurable indication of quality.
- Unambiguous

- Example1: The system shall be able to **quickly** generate a visible warning signal for the co-pilot. → **Unverifiable**

- Example2: The system shall be able to generate a visible warning signal for the co-pilot **in least than 5 seconds**. → **Verifiable**

Guidelines for Writing Requirements

Prioritized

- Assigns implementation priority for Requirements.
- Requirement priority levels example
  - **High** priority – must be in system release
  - **Medium** priority – necessary, but can be deferred to later release
  - **Low** priority – nice to have, but may be dropped if insufficient time or resources

- Example:

ID	Requirement	Priority
Req01	The pilot shall be able to view the airspeed.	High
Req02	The system should be able to generate a visible warning signal for the co-pilot.	Low

Guidelines for Writing Requirements

Consistent

- Use consistent terminology
  - If you're writing requirements for Admin users, don't flip back and forth between "Admin User" and "Administrator". Inconsistency leaves room for confusion.
- Be consistent with Imperatives (**shall**, **must**, **will**, **should**, etc.)
  - Once you have agreement on how each imperative term will be used within your organization, use the same terms consistently across all requirements.
- …

References

1. **Dennis A., Haley W. B., Tegarden D.** "System Analysis & Design - An Object-Oriented Approach with UML", 5e – 2015
2. **Sarnath R., Brahma D.** "Object-Oriented Analysis, De sign and Implementation", 2e – 2015
3. **Naoufel B.** 'Guidelines for Good Requirements Writing with Examples' – 2014
4. **Peter Z.** 'Requirements Management Using IBM® Rational® RequisitePro®' - 2008
5. https://reqexperts.com