
بسم الله الرحمن الرحيم

Software Requirement Engineering

LECTURER: SYED HASNAIN ABBAS BUKHARI

Quality of Requirements

Quality of Requirements

- How can we distinguish between a high quality set of requirements and a low quality set?
- How can we measure the quality of a requirement?
- IEEE 830 standard identifies eight quality measures for evaluating an SRS
- RUP adds Understandability
- Karl Wiegers adds feasibility and necessary

Quality Measures

Statement Level

- ❑ Correct
- ❑ Unambiguous
- ❑ Verifiable
- ❑ Complete
- ❑ Understandable

Collection Level

- ❑ Complete
- ❑ Consistent
- ❑ Modifiable
- ❑ Traceable
- ❑ Understandable

Correct

- A requirement is unambiguous if and only if it can be subject to only one interpretation
- A typical problem whenever requirements are written in natural language
- There are really two types of ambiguity
  - Conceptual
  - Audience
- Conceptual ambiguity is independent of the audience. It is function of the logical structure of the statement.
- “The system shall be fast” is conceptually ambiguous, because speed is a matter of degree, and the statement contains no constraint on it.

Unambiguous

- A requirement is unambiguous if and only if it can be subject to only one interpretation
- A typical problem whenever requirements are written in natural language
- There are really two types of ambiguity
  - Conceptual
  - Audience
  - Conceptual ambiguity is independent of the audience. It is function of the logical structure of the statement.
  - “The system shall be fast” is conceptually ambiguous, because speed is a matter of degree, and the statement contains no constraint on it.

Unambiguous

- Audience ambiguity is present when different people read the requirement and come up with different interpretations of it.
- The requirement makes sense to each of them but means something different to each of them
- "Mary had a little lamb."
  - Does it mean she kept one as a pet?
  - Does it mean she gave birth to one?
- A formal peer review provides an opportunity for each participant to compare his understanding of each requirement to someone else's.

Checklist For Ambiguity

- Incomplete lists ending with "etc.," "and/or," and "TBD.“
- Vague words and phrases such as "generally," "normally," "to the greatest extent," and "where practicable.“
- Implied certainty, flagged by words such as "always," "never," "all,” or "every.“
- Passive voice, such as "the counter is set." (By whom or what?)
- Every pronoun, particularly "it" or "its." Each should have an explicit and unmistakable reference.
- Comparatives, such as "earliest," "latest," "highest." Words ending in “er" or "est" should be suspect.

Verifiable

- Can a tester devise tests or other verification approaches to determine whether each requirement is properly implemented?
- A requirement can be deemed verifiable if and only if there exists a finite, cost effective process with which a person or a machine can determine that the developed sotware system does indeed meet the requirement
- Requirements that are incomplete, inconsistent, infeasible, or ambiguous are also unverifiable
  - E.g. The car shall have power brakes **Not Testable**
  - The car should come to a full stop from 60 miles per hour within 5 sec**7 Testable**
  - The power brake shall fully engage with 4lbs of pressure applied to the
    - brake pedal **Testable**

Traceable

- A requirement is traceable if and only if the origin of each of its component requirements is clear, and if there is a mechanism that makes it feasible to refer to that requirement in future development efforts.

Understandable

- A requirement set is understandable if both the user and the developer communities are able to fully comprehend the individual requirements.

Complete

- **Statement level**
  - Each requirement must contain all the information necessary for the reader to understand it.
  - In the case of functional requirements, this means providing the information the developer needs to be able to implement it correctly.
  - If you know you're lacking certain information, use *TBD* (to be determined) as a standard flag to highlight these gaps, or log them in an issue8tracking system to follow up on later.
  - Resolve all TBDs in each portion of the requirements before the developers proceed with construction of that portion.

Complete

- **Collection level**
  - A set of requirements is complete if and only if it describes all significant requirements of concern to the user, including requirements associated with functionality, performance, design constraints, attributes, or external interfaces.
  - A complete set of requirements must define
    - The required response of the soTware to all realizable classes of inputs **(both valid and invalid)** in all realizable classes of situations.
    - It must provide complete **references** and labels for all of the **figures, tables, and diagrams** within the requirement set, as well as definitions of all terms and units of measure.

Consistency

- A requirement set is internally consistent if and only if no subset individual requirements described within it are in conflict with one another
- Consistent requirements don't conflict with other requirements of the same type or with higher-level business, user, or system requirements.
- One part of the requirements say
  - All employees who are 65 or older at the end of the calendar year shall receive a bonus of $1000
- Whereas another part of the requirements might say
  - All employees with 10 years or more of service at the end of the calendar year shall receive a bonus of $500
- Recording the originator of each requirement lets you know who to talk to if you discover conflicts.

Modifiable

- A requirement set is modifiable if and only if its structure and style are such that any changes to the requirements can be made easily, completely, and consistently, while retaining the existing structure and style of the set

- Modifiability dictates that each requirement be uniquely labeled and expressed separately from others so you can refer to it unambiguously.

- SRS be well organized, with a proper table of contents, index, and cross8referencing capabilities