Writing Good Requirements and RE in CSE

Colin Werner and Ze Shi (Zane) Li

thesegalgroup.org

lize@uvic.ca

Writing Excellent Requirements

Complete - requirement contains all the information to understand it

Correct - accurately describe a capability that will meet some stakeholder's need

Feasible - must be possible to implement each requirement within the known capabilities and limitations of the system

Writing Excellent Requirements

Necessary - describe a capability that provides business value, required for standard policy, or regulation, differentiates product

Prioritized - which are the most important to achieve value. Assign priority to each requirement

Unambiguous - natural language ambiguity

Verifiable - can QA/tester develop test to verify each requirement?

Two Key Factors

Each person reading the requirement should come to the same interpretation

Each reader's interpretation matches author's intended interpretation

Who’s Perspective

Traditionally, functional requirements were written from “the system’s” perspective

- fine to use “the system…” or “A user”

Or “An admin”, “A driver”, etc

Who’s Perspective

Traditionally, functional requirements were written from “the system’s” perspective

- fine to use “the system…” or “*A* user”

Or “An admin”, “A driver”, etc

Consistency

Be consistent with your terminology!

"Must" or "shall" or "needs to" or "should"

Pick one of these, and stick with the term

Active Voice

Use active voice to ensure clarity in *who* is taking the action

Not so great: of ITEM *will be modified*

Better: An admin *modifies* **ITEM**

Avoiding Ambiguity

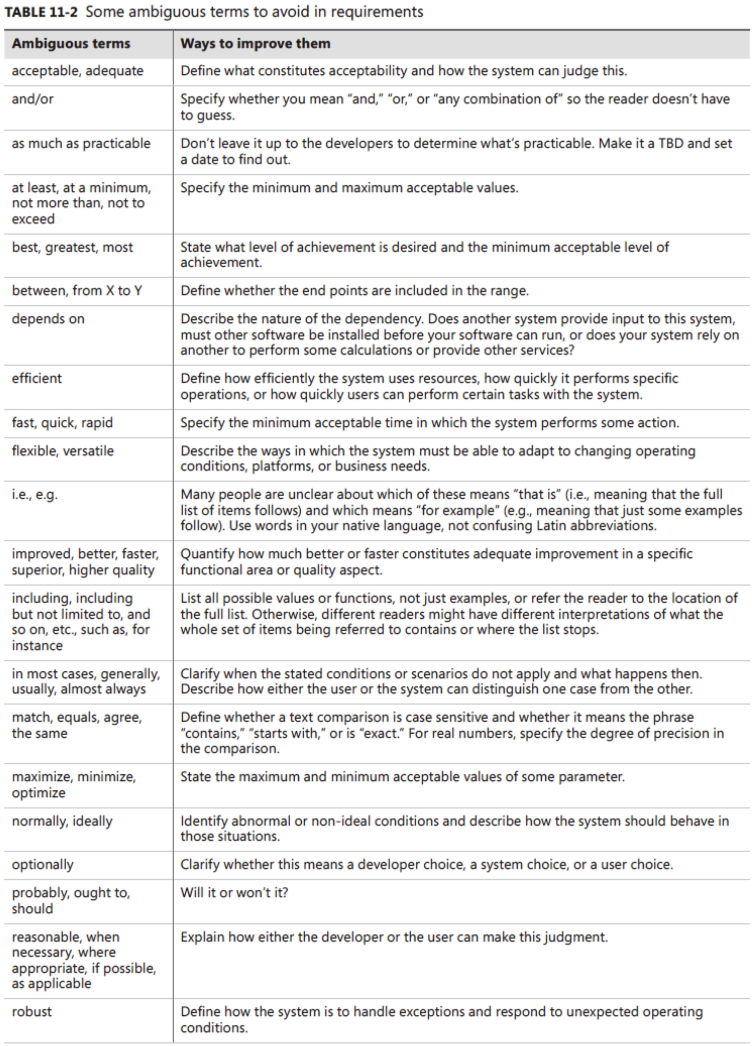
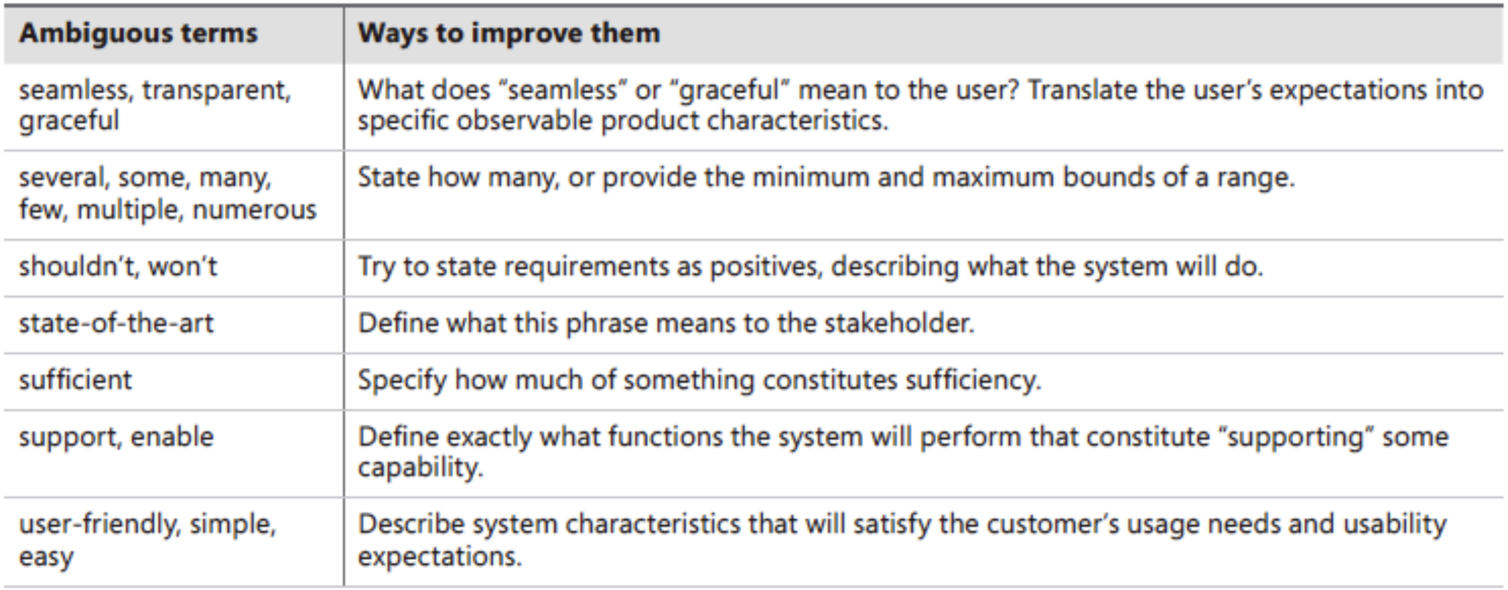


TABLE 11-2 Some ambiguous terms to avoid in requirements

Ambiguous terms	Ways to improve them
acceptable, adequate	Define what constitutes acceptability and how the system can judge this.
and/or	Specify whether you mean "and," "or," or "any combination of" so the reader doesn't have to guess.
as much as practicable	Don't leave it up to the developers to determine what's practicable. Make it a TBD and set a date to find out.
at least, at a minimum, not more than, not to exceed	Specify the minimum and maximum acceptable values.
best, greatest, most	State what level of achievement is desired and the minimum acceptable level of achievement.
between, from X to Y	Define whether the end points are included in the range.
depends on	Describe the nature of the dependency. Does another system provide input to this system, must other software be installed before your software can run, or does your system rely on another to perform some calculations or provide other services?
efficient	Define how efficiently the system uses resources, how quickly it performs specific operations, or how quickly users can perform certain tasks with the system.
fast, quick, rapid	Specify the minimum acceptable time in which the system performs some action.
flexible, versatile	Describe the ways in which the system must be able to adapt to changing operating conditions, platforms, or business needs.
i.e., e.g.	Many people are unclear about which of these means "that is" (i.e., meaning that the full list of items follows) and which means "for example" (e.g., meaning that just some examples follow). Use words in your native language, not confusing Latin abbreviations.
improved, better, faster, superior, higher quality	Quantify how much better or faster constitutes adequate improvement in a specific functional area or quality aspect.
including, including but not limited to, and so on, etc., such as, for instance	List all possible values or functions, not just examples, or refer the reader to the location of the full list. Otherwise, different readers might have different interpretations of what the whole set of items being referred to contains or where the list stops.
in most cases, generally, usually, almost always	Clarify when the stated conditions or scenarios do not apply and what happens then. Describe how either the user or the system can distinguish one case from the other.
match, equals, agree, the same	Define whether a text comparison is case sensitive and whether it means the phrase "contains," "starts with," or is "exact." For real numbers, specify the degree of precision in the comparison.
maximize, minimize, optimize	State the maximum and minimum acceptable values of some parameter.
normally, ideally	Identify abnormal or non-ideal conditions and describe how the system should behave in those situations.
optionally	Clarify whether this means a developer choice, a system choice, or a user choice.
probably, ought to, should	Will it or won't it?
reasonable, when necessary, where appropriate, if possible, as applicable	Explain how either the developer or the user can make this judgment.
robust	Define how the system is to handle exceptions and respond to unexpected operating conditions.

Avoiding Ambiguity



Ambiguous terms	Ways to improve them
seamless, transparent, graceful	What does "seamless" or "graceful" mean to the user? Translate the user's expectations into specific observable product characteristics.
several, some, many, few, multiple, numerous	State how many, or provide the minimum and maximum bounds of a range.
shouldn't, won't	Try to state requirements as positives, describing what the system will do.
state-of-the-art	Define what this phrase means to the stakeholder.
sufficient	Specify how much of something constitutes sufficiency.
support, enable	Define exactly what functions the system will perform that constitute "supporting" some capability.
user-friendly, simple, easy	Describe system characteristics that will satisfy the customer's usage needs and usability expectations.

Additional Ambiguity in Natural Language

Credit

These slides were adapted from Daniel Berry (https://cs.uwaterloo.ca/~dberry/)

A very useful and exhaustive resource is available: https://cs.uwaterloo.ca/~dberry/handbook/ambiguityHandbook.pdf

WRITING REQUIREMENTS IN NATURAL LANGUAGE

Because everyone knows at least one natural language, so we always have someone to do it!

And most people should speak the same language, so we should all understand it!

But natural languages are naturally ambiguous…

Writing Requirements in Natural Language

They are totally unambiguous!

But is there someone who can write in a formal language? You can probably find _someone_ …

But does everyone understand the formal language?

Why is ambiguity in natural language a problem

Read Dan's book!

Subconscious Disambiguation

Subconscious Ambiguation

Subconscious Disambiguation

In subconscious disambiguation (SD), the reader of an ambiguous phrase is not even aware that there is an interpretation other than the one that came first to his or her mind.

The reader understands an interpretation and thinks that it is the only one.

Subconscious Ambiguation

Subconscious ambiguation (SA) is the flip side of SD.

In SA, the writer of an ambiguous phrase is not even aware that he or she has written a phrase that has an interpretation other than what he or she thought to create the phrase.

Advice

Synonyms

Plural

Pronouns

‘Only’

‘This’

‘All’

Relative

Order

Synonyms

We were taught to use synonyms of words as opposed to repeating the same word over and over.

The user … the customer … the client … the consumer…

Vs

The user … the user … the user … the user …

Ambiguity of Plural

Students enroll in six courses per term.

Students enroll in hundreds of courses per term.

Ambiguity of Plural Fixed?

Each student enrolls in six courses per term.

The student body enrolls in hundreds of courses per term.

Ambiguity in Pronouns

Every student thinks she is a genius.

Ambiguity in Pronouns Fixed?

Every student thinks Ellen is a genius.

Each student thinks he or she is a genius.

Ambiguity of “Only”

I only nap after lunch.

Ambiguity of “Only” Fixed?

I nap only after lunch.

  Means the only time I nap is after lunch.

I only nap after lunch.

  Means the only thing I do after lunch is nap.

Ambiguity of “This”

This prevents security breaches.

Ambiguity of “This” Fixed

This encoding scheme prevents security breaches.

Ambiguity of “All”

All the lights in any room have a single on-off switch.

Ambiguity of “All” Fixed?

Each light in any room has a single on-off switch.

Ambiguity with Relative Terms

The response time must be fast.

Ambiguity with Relative Terms Fixed?

The response time must be less than one second.

Ambiguity with Order

The police shot the rioters with guns.

Ambiguity with Order - Fixed?

Using guns the police shot the rioters.

The police shot the armed rioters.

Other Words

almost, also, even, hardly, just, merely, mostly, nearly, really…

Final Advice

Avoid ambiguity

Be careful with names (pronouns)

Write for the reader

Careful organization

Avoid redundancy (not necessarily repetition)

Avoid superfluous text

READ WHAT YOU WRITE!!

The Good, The Bad, and the Ugly

The system allows a management user to create and edit the schedule for all other users.

The Good, The Bad, and the Ugly I

A management user must be able to create the schedule for a staff member.

A management user must be able to edit the schedule for a staff member.

The Good, The Bad, and the Ugly II

Academic advisors must be able to restrict access from the student which it is related to.

The Good, The Bad, and the Ugly II

An academic advisor must be able to restrict access for a student.

The Good, The Bad, and the Ugly III

Each student and teacher must be able to export records related to a student that he or she has access to.

The Good, The Bad, and the Ugly III

Each student must be able to export their respective record.

Each teacher must be able to export a student’s respective record of which the teacher has access to.

The Good, The Bad, and the Ugly IV, V, VI, VII & VIII

The Pipeline Manager should be able to record notes taken during meetings with debtors.

Each branch manager and debt counsellor should be required to sign in to use the Pipeline Manager.

Users must be able to view both real time and previously collected data.

The user has a basic knowledge of computer application use and purpose.

User inputs for cost estimations are correct regarding

CSE and RE

CSE Activities

Continuous integration

Continuous delivery

Continuous deployment

Continuous security

Continuous planning

…

Some Experiences from A Study of NFRs in Industry

Multiple case study to investigate the practices of NFRs in 3 organizations using CSE

- data collection analytics industry
- e-commerce platform
- content provider

What did we find? (Practices)

Put a number on the NFR

i.e., Establishing metrics to validate each NFR

- the number of customer actions to complete a transaction

- tracking website loading on a dashboard

NFRs that benefit from this: Usability, Performance, Security

What did we find? (Practices)

Let someone else manage the NFR

- offloading an NFR to someone else (cloud provider)

- scalability, reliability, security can be offloaded

- configurability for Docker, Terraform, and Kubernetes to recreate environments

- “Throw money at a third party” when needed

What did we find? (Practices)

Write your own tool to check the NFR

- custom, in-house tool
- usability tests via own tool for front-end code
- Sometimes custom tool is built on third-party tool
- Particularly useful highlighting security issues

What did we find? (Practices)

Put the NFR in source control

- NFRs are commented in code
- "Codification" -> helps with uncovering tacit knowledge

What did we find? (Challenges)

Not all NFRs are easy to automate

- usability is inherently difficult to verify (automate)
- usability testing can be a bottleneck (imagine 30 a day!)
- writing tests for NFRs
- more tests does not mean higher quality
- uncovering factors behind shifts in performance or scalability

What did we find? (Challenges)

Functional requirements get prioritized over NFRs

- NFRs are easy to put on the back burner

- Performance, efficiency are sometimes easy to overlook

- NFR trade-offs to achieve immediate functional behaviour

What did we find? (Challenges)

Lack of shared understanding of an NFR

- inconsistency in source control
- tacit knowledge
- bus factor
- role siloing
- documentation for knowledge transfer

"Being the one person specialized in this"

Some Trade-offs to Note

Losing control over offloaded NFR

- Organizations may be locked into vendor (cost of switching) and they lose control of NFR to the vendor (outages)

Hurt shared understanding of an NFR

- siloing between devops and developers, and other roles

The Growing Importance of Configurability

- to configure infrastructure and environments
- supports re-deployments, rapid deployment, rollbacks, etc
- configurability also depends on reliability, availability, etc
- code smells in configurability that has to be managed (technical debt)

In Practice