Student Name: _____
Course Number and Name: _____

Software Verification and Validation

Requirements Ambiguity

Exercise Module Number: RM09

Instruction: Identify the ambiguity (or ambiguities) in these statements; discuss and propose how they may be corrected.

1. For a web-based system, it is required that loading of all web pages must be completed within a reasonable time duration.

2. Access right to the data is limited to the individuals with managerial rank, but those with access rights may also grant access rights to others.

3. Even though the (stock) market is open only during business hours, access to stock prices should be available 24/7, supporting client access to the market at the clients' time and locale.

Next Page

4. A user should be able to customize the system behavior to cater to his/her own needs. Yet the system should provide a reasonable default case for everyone.

5. Every book is identified by its ISBN in the catalog. When a member of the library takes a book out on loan, the system must also identify which copy of the book was loaned out, so that upon return the member will be responsible for any damage to that specific copy of the book.

This material is based upon work supported by the National Science Foundation under Grant Number 1245036

Some Ambiguous Terms to Avoid

Source: *Adapted from Karl E. Wiegers, More about Software Requirements, Microsoft Press, 2006*

Ambiguous Term	Improvement
acceptable, adequate	Define what constitutes acceptability and how the system can judge this.
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 minimum, not more than, not to exceed	Specify the minimum and maximum acceptable values.
between	Define whether the end points are included in the range.
depends on	Describe the nature of dependency. Does another system 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 services?
efficient	Define how efficiently the system uses resources, how quickly it performs specific operations, or how easy it is for people to use.
flexible	Describe the ways in which the system must change in response to changing conditions or business needs.
improved, better, faster, superior	Quantify how much better or faster constitutes adequate improvement in a specific functional area.
including, including but not limited to, and so on, such as	The list of items should include all possibilities. Otherwise, it can't be used for design or testing.
maximize, minimize, optimize	State the maximum and minimum acceptable values of some parameter.
normally, ideally	Also describe the system's behavior under abnormal or non-ideal conditions.
optionally	Clarify whether this means a system, user or developer choice.
reasonable, when necessary, where appropriate	Explain how to make this judgment.
robust	Define how the system is to handle exceptions and respond to unexpected operating conditions.
seamless, transparent, graceful	Translate the user's expectations into specific observable product characteristics.
several	State how many or provide the minimum and maximum bounds of a range.
shouldn't	Try to state requirements as positives, describing what the system will do.
state-of-the-art	Define what this means.
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 achieve the customer's usage needs and usability expectations.

This material is based upon work supported by the National Science Foundation under Grant Number 1245036