GUIDELINES TO VALID REQUIREMENTS

Extracted from: **Business Analysis: A Systems Approach to Solving Business Problems**

1. Single sentence (paragraph) per requirement
   - Explanations differentiated (different font/size, indented)

   Explanations can be included in a requirements document provided it is absolutely clear that it is an explanation and not a requirement. Make things easier for all readers of the requirements by setting explanations off from the requirements by placing the explanation in parentheses, in a different font/size, indented, in a different color, and so forth. It might also help to make it clear in the front of the document what is being used to distinguish the explanations from the requirements.

2. Numbered (identified)

   Each requirement must be identified with a unique number or label.

3. CompleteSelf-contained, stand-alone

   Every requirement must be self-contained and stand-alone. The understanding of a requirement must not depend on the content of another requirement. Understanding may depend on language defined in the glossary.

   > *Where completeness cannot be obtained (information not available, decisions not made, etc.) the requirement can be "defined later", but needs a date or a point in time (or process) by which it will be defined.*

   When dealing with an approved or baselined set of requirements, removing the "TBD" and replacing it with the real requirements requires a new version.

4. ConsistentLists only WHAT is required, not HOW
   - Does not conflict with any other requirements
   - Use same terminology throughout

   Lack of consistency in requirements is one of the major reasons for failures due to requirements errors. When inconsistency exists, assumptions will be made and chances are the assumptions could be wrong.

5. TestableUsing inspection, demonstration, execution or analysis
   - Single test per requirement with pass/fail results

6. Feasible
   - In scope
     States no requirements for anything or anyone not in the system
   - Technologies
     Does the technology exist to do what is needed?
     Can the technology be accessed/acquired (Buy. Build. Borrow)?
     Can the software team actually do it?
   - Economic (e.g., Cost / Benefits Analysis)
   - Legal and regulatory
     Do all the requirements meet or conform to all applicable regulations?
   - Usability
     Will the user actually be able to understand and use the feature to solve the business problem?

   Feasible means that it is possible to implement each requirement within the capabilities and limitations of the known technical and operational environment.
7. Traceable
   - Traceable to scope, concept of operations, etc.
   - Traceable to source of requirements
8. Written for the correct audience
   - User requirements UserFunctional requirements User / customerSystems requirements Architect / designerSoftware requirements Analyst / designer / developerProgram requirements Developer / testerHardware requirements Analyst / procurement / testers

9. Word Choice

   Precision

   Precision in professional language has two purposes: to make communication with fellow professionals specific and unambiguous, and to inform people on the outside meaningfully and clearly [Holmes, Neville, "In Praise of Professional Precision", *IEEE Computer*, 4/2006].

   Choice of words is important. There are a number of imprecise words that we think are precise. These words should be avoided. (See next page). Technical words or jargon tend to obfuscate, and should be avoided unless defined specifically in the glossary.

   Ambiguity

   Ambiguity exists when any word or phrase may be interpreted in more than one possible way. This can be verified by having another Business Analyst read the requirements aloud or have a Business Analyst paraphrase what she or he thinks the requirement is saying. If the meaning varies from the author's intended meaning, ambiguity exists and must be eliminated by rewording the requirement. Note that the requirement itself may not need to be changed to eliminate the ambiguity. A clearly defined and unambiguous entry in the glossary may eliminate the problem.

- Use rule-based verbs – “will”, “must”, active verbs
  Avoid statements of volition –

May	might
Should	could
If	can

- Use present rather than future tense, if possible

- No conjunctions: “and”, “but”, semicolon, “or”
  (“and” may be used as a connector only)

- Positive voice

- Avoid hyphenated phrases

Well-organized	Fully-automated
Trouble-free	State-of-the-art
Working-condition	User-friendly

Use nouns rather than pronouns or proper nouns

- Consistent (not contradictory – same use of word)
  Use Glossary for consistency

- Avoid vague or imprecise words

ability	about	
acceptable	adequate	all
appropriate	approximately	around
aspect		
case	characteristic*	current
communicate	comprehensive	convenient
detect	different*	
easy	efficient	employ
enable		
enough	every	everyone
factor	fast	function
feature*	high	immediate
instantaneously	kind	
large	logical	low
many	matter	measure(n)
modern	note (v)	numerous
operate		
perform	practical	present (adj)
present (v)*	quick	
realistic	reasonable	run*
satisfactory	set	simple
several	safe	small

sort	substantial	successfully
sufficient		
thing	type	
usable	useful	various

*without qualifying attribute

**unless dealing with mathematical specifics

- Avoid “automatic” words (without glossary definition)

accessible*	accessibility*	application
apply		artifact
architecture	architectural	availability*
automate(d)	automation	asynchronous
batch		build(n)
check (v)	client	communication*
component	concurrent	connection(s)
database*	decode*	design*
deployment	disk*	display*
documentation	drive(n)	
effective	efficient	element
encode*	entity	executable
execute		
feature	field	firmware
function	functional	functionality
generate	gui	
header	implementation	
input (n)	input (v)	interface
infrastructure	implement	
install	integrate*	integration
maintain	manual	
middleware	module	monitor (v)
network		
operation(s)	output (n)	output (v)
phase	port (v)	package
part	partition	populate
provide		
read*	real-time	record
relationship	robust	runtime
scale(v)	scan	screen
seamless	secure (v)	secure (n)
segment	server	serviceable
set	simultaneously	supply
support	survey(v)	switch (n)
synchronize	synchronous	
transaction	unit*	update
write*		

*without qualifying attribute(s)

- Phrases to avoid

more or less	network layer	physical design
present time	real-time	round-trip
the ability to	throw an error	user friendly
web-based		

- Avoid comparative adjectives or phrases

a little	as little as*	as much as*	bad
best	better	bigger	cheaper
considerable	considerably	definitely	
easier	enough	exceeding(ly)	excessive(ly)
extremely			
faster	good	highest	huge
large	larger	least	less
lowest	mega-**	meta-**	minimum*
maximum*	more	most	overly
really			
smaller	smallest	somewhat	too
very	worse		

*without a qualifying quantity (e.g., a minimum of twelve will be allowed)

**when referring to size rather than quantity