

Criteria of a Good Requirement

Criterion	Description
Necessary	Can the system meet prioritized, real needs without it? If yes, the requirement isn't necessary.
Verifiable	Can one ensure that the requirement is met in the system? If not, the requirement should be removed or revised. Note: The verification method and level at which the requirement can be verified should be determined explicitly as part of the development for each of the requirements. (The verification level is the location in the system where the requirement is met (for example, the "system level," the "segment level," and the "subsystem level"). ¹
Attainable	Can the requirement be met in the system under development?
Unambiguous	Can the requirement be interpreted in more than one way? If yes, the requirement should be clarified or removed. Ambiguous or poorly worded writing can lead to serious misunderstandings and needless rework. Note: Specifications should include a list of acronyms and a glossary of terms to improve clarity.
Complete	Are all conditions under which the requirement applies stated? Also, does the specification document all known requirements? (Requirements are typically classified as functional, performance, interface, constraints, and environment.)
Consistent	Can the requirement be met without conflicting with all other requirements? If not, the requirement should be revised or removed.

¹See Grady, *System Validation and Verification*, pp. 101–102, for a discussion of verification levels.

Criterion	Description
Traceable	Is the origin (source) of the requirement known, and can the requirement be referenced (located) throughout the system? The automated requirements tool should enable finding the location in the system where each requirement is met.
Allocated	Can the requirement be allocated to an element of the system design where it can be implemented? If not, the requirement needs to be revised or eliminated. ²
Concise	Is the requirement stated simply and clearly?
Implementation free	The requirement should state what must be done without indicating how. The treatment of interface requirements is generally an exception.
Standard constructs	Requirements are stated as imperative needs using "shall." Statements indicating "goals" or using the word "will" are not imperatives.
Unique identifier	Each requirement should have a unique identifying number that assists in identification, maintaining change history, and providing traceability.

²The alternative is to risk a major costly change in the system or software architecture.