Requirement Quality Checklist

1) **Does the requirement have a level of detail appropriate to the current lifecycle stage and purpose of the document?**

   Requirements may be less detailed early in the development lifecycle but need more detail before coding or testing. The level of detail should be appropriate to the level of document and its purpose in the development process at the current. This is particularly important when using incremental or evolutionary lifecycles such as those using Agile Methods.

2) **Is the requirement risk related?**

   Indicate if the requirement is risk related and if so what the safety significance is if it doesn't work properly or refer to more information is separate risk analysis documentation.

3) **Is the requirement correct?**

   Some requirements have a significance that warrants documenting their source. For example what industry standard, text book, or regulation they come from; what simulation, calculation, or usability study they were derived from.

4) **Are requirements specified in all relevant ways/contexts?**

   It can be important to specify requirements in more than one way to capture all aspects of requirements or organize requirements information for different types of testing. For instance, one might specify use cases or user scenarios to capture the sequence of user operations and yet include detailed requirement by requirement specification elsewhere in the same or a separate document. Another example is that some requirements might vary by machine state or model and the details might be best captured in tables separate from the base requirements.

5) **Is the requirement vague that is, not clearly or explicitly stated?**

   Even if requirement is a "high level" requirement, it still must not be vague. Example: "The user interface must be cutting edge."

6) **Is the requirement ambiguous that is, open to having several possible meanings or interpretations?**

   Any requirement with the words:

   all, any, and, or, and/or, but, unless, if, only, also, it, they, or plural nouns

   is potentially ambiguous. Other potentially ambiguous words: *simultaneously*

   Example: "Component must be able to receive RAPHA data from multiple modalities simultaneously." What is meant by simultaneously? What is meant be multiple, i.e. could be two up to infinity?

7) Is this requirement complete? Are there pieces missing?

Example: "The device shall have a standby and a ready state". Are there other states? What is the difference between the states and

8) Does this requirement conflict with other requirements, both stated and implied?

Conflicting with stated requirements is easier to detect with relational database approach to requirements. Consider creating multiple groupings and categorizations instead of just one hierarchical structure.

Implied requirements may be things like performance (e.g. user expects screen to update within x seconds) or safety (i.e. user expects data to be correct, DST to have adjusted, etc.).

9) Is the requirement at the proper level? Is it a parent (system level; product feature level) or a child requirement (software level)?

Develop a consistent model (i.e. create WI or procedure) for what levels will exist, and how they will be represented in tools (i.e. Test Track). Generally, high level requirements should "flow down" to lower level requirements.

10) Does the requirement specify time periods or temporal aspects needed for complete understanding?

Example: "Upon entering required case details, if all fields (both required and optional) pass validation the Submit button will indicate all fields are valid and the allow the case to be submitted." This requirement does not specify how long after all conditions are met the Submit button becomes valid. Temporal aspects such as this can be handled by a general UI timing requirement or by including timing more specifically in the requirement. Requirements of this type are needed to know when the tester "determines it has failed".

11) Does the requirement provide tolerances by which pass/fail will be determined?

Always provide tolerances so that test cases can be written to indicate fail condition.

12) Finally, is the requirement testable?

Are test conditions and pass/fail criteria derivable from the requirement or more detailed specifications related to the requirement?