Requirements Analysis Checklist

from *Software Requirements Essentials*by Karl Wiegers and Candase Hokanson

*Analysis Considerations for Individual Requirements*

* Origin and rationale
* Decomposition and derivation
* Exceptions
* Acceptance criteria
* Assumptions
* Constraints
* Business rules
* Hazards and risks
* Reuse potential

*Quality Characteristics for Individual Requirements*

* **Complete**. No necessary information is missing.
* **Correct**. The requirement accurately states a stakeholder’s need or a necessary property of the solution.
* **Feasible**. The requirement can be implemented within known technical, business, and project constraints.
* **Necessary**. The statement documents something a stakeholder really needs.
* **Prioritized**. The requirement is ranked relative to others as to its importance and urgency of inclusion in the solution.
* **Unambiguous**. The statement conveys only one possible meaning to all readers.
* **Verifiable**. There’s some way to demonstrate that the requirement has been correctly implemented.

*Analysis Considerations for Requirement Sets*

* Gaps
* Conflicts and inconsistencies
* Overlaps and duplications
* Dependencies
* Representation in different forms
* Prioritization
* Assumed and implied requirements

*Quality Characteristics for Requirement Sets*

* **Complete**. No requirements are missing.
* **Consistent**. No requirement conflicts with another requirement.
* **Modifiable**. The collected requirements are organized and labeled in a way that makes it easy to change them and to keep records of the changes made.
* **Traceable**. It’s possible to document logical links from requirements back to their origins, to related requirements, and to other development products like designs, code, and tests.