Requirements Verification Checklist

For the Assessment of:	
Project Name:	
Artifact Name:	
Artifact Author:	
Date:	

Criteria	Yes / No / NA
1. Cohesive—A set of requirements is cohesive if it relates to only one thing. All requirements in a set or group support the overall purpose and scope of the system or component under discussion, whether that is a business process definition, business rule, information flow, data flow or so forth.	
a. Are the requirements consolidated into groups such as products, components, use cases and user stories?	
b. Do the requirements align with their process definitions and models?	
c. Does each group contain 5 to 9 requirements? This is not a failing criterion if more than nine although, it may indicate the group can be divided.	
d. Do the grouped requirements flow logically from one to the next, from the more general to the more specific or from the least sensitive to the most sensitive?	
2. Complete—To be complete, all known relevant requirements are documented and all conditions under which a requirement applies are stated. A BRD is complete if, and only if, it includes the following elements: • All significant requirements, whether relating to functionality, performance, design constraints, attributes, or external interfaces. • Definitions of the responses of the system/software to all realizable classes of input data in all realizable classes of situations. • Descriptive labels for and references to all figures, tables, models and diagrams in the BRD and definition of all business terms, acronyms and units of measure.	
a. Does each requirement contain all the information necessary for the technical team to design, build and test that component of the solution?	
b. Are all the inputs to the system/software specified, including their source, accuracy, range of values and frequency?	
c. Are all the outputs from the system/software specified, including their destination, accuracy, and range of values, frequency and format?	
d. Are all the communication interfaces specified, including handshaking, error checking and communication protocols?	
e. Has analysis been performed to identify missing requirements?	
f. Are the areas of incompleteness specified when information is not available?	
g. Are the requirements complete such that if the product satisfied every	

requirement it would be acceptable?	
h. Are all figures, tables, models and diagrams labeled in a descriptive manner?	
i. Are all figures, tables, models and diagrams referenced within the document?	
j. Are all business terms, acronyms and units of measure defined appropriately?	
3. Consistent—Consistency demands that the requirement can be met without causing conflict or contradiction with any of the other requirements. Requirement should be stated in a way to allow the widest possible selection of implementation options.	
a. Do the requirements avoid prematurely determining a solution?	
b. Do the requirements avoid specifying a design?	
c. Are the requirements specified at a consistent level of detail?	
d. Should any requirements be specified in more detail?	
e. Should any requirements be specified in less detail?	
f. Are the requirements consistent with the content of other organizational and project documentation?	
4. Correct—Requirements must accurately describe the functionality to be built. Only the source of the requirements, the customers, users or stakeholders, can determine their correctness.	
a. Do the requirements fulfill the original business need?	
b. Has the scope of the system/software been bounded?	
c. Have the overall function and behavior of the system/software been defined?	
d. Has the required technology infrastructure for the system/software been adequately specified?	
e. Are all the tasks to be performed by the system/software specified?	
f. Can each requirement be allocated to an element of the solution design where it can be implemented?	
g. Does each task specify the data/information content used in the task and the data/information content resulting from the task?	
h. Is each requirement associated with a use case or process flow?	
i. Have requirements for communication among system/software components been specified?	
j. Have appropriate constraints, assumptions, and dependencies been explicitly and unambiguously stated?	
k. Are the hardware requirements specified?	
l. Are the physical security requirements specified?	
m. Are the operational security requirements specified?	
n. Is the maintainability of the system/software specified, including the ability to respond to changes in the operating environment, interfaces, accuracy, performance, and additional predicted capabilities?	
o. Is the reliability of the system/software specified, including the consequences of failure, vital information protected from failure, error detection, and recovery?	

p. Are internal interfaces such as software and hardware defined?	
q. Are external interfaces, such as users, software and hardware defined?	
r. Is each requirement relevant to the problem and its solution?	
s. Is the definition of the requirement's success included? Of failure?	
5. Feasible—Feasibility means each requirement is implementable within the existing infrastructure, budget, timeline and resources available to the team. The business analyst needs to work with the project team to make these determinations.	
a. Are the requirements technically feasible and do they fit within the project funding and timing constraints?	
b. If not, is the project able to develop the capability to implement the requirement?	
c. Even if a requirement is technically feasible, it may not be attainable due to constraints. Are there any constraints that prevent the requirement from being attained?	
d. Is it possible to implement each requirement within the capabilities and limitations of the technical and operational environment?	
e. Is it possible to implement each and every requirement?	
6. Necessary & Prioritized—Requirements must be ranked for importance and/or stability. A necessary requirement is one that is essential to meet business goals and objectives. A priority is assigned to each functional requirement or feature to indicate how essential it is to a particular solution release. If all requirements are considered equally important, it is difficult for the project team to respond to budget cuts, schedule overruns, staff turnover or new requirements added during development. Ranking requirements for stability is in terms of the number of expected changes to the requirement. Stable requirements are ready to be developed.	
a. Do requirements have an associated identifier to indicate either the importance or stability of that particular requirement?	
b. Do conflicts exist regarding the importance and/or stability ranking of the requirements?	
c. Are all requirements ranked the same?	
7. Measurable, Testable, & Verifiable—Verifiable means that the requirement states something that can be confirmed by examination, analysis, test or demonstration. A good requirement does not contain words that are not testable and measurable. If it is impossible to ensure that the requirement is met in the solution, it should be removed or revised. • Testable requirements are designed to demonstrate that the solution satisfies requirements. Tests may include functional, performance, regression, and stress tests. • The verification method and level (i.e., the location in the solution where the requirement is met) at which the requirement can be verified should be determined explicitly as part of the development of each requirement. Requirement statements that include words that have relative meaning are not verifiable. For example: o Adequate o Easy o More efficient o Better than o Maximum o Quality product o Comparison o Minimum o Substantial	

a. Are the requirements written in a language and vocabulary that anyone can understand? Do the stakeholders concur?	
b. Is the expected response time from the user's point of view specified for all necessary operations?	
c. Are other timing considerations specified such as processing time, data transfer and throughput?	
d. Are acceptable tradeoffs between competing attributes specified? For example, between robustness and correctness?	
e. Does each requirement capture a metric by which it can be measured?	
f. Is each requirement testable?	
g. Will it be possible for independent testing to determine whether each requirement has been satisfied?	
8. Traceable—Requirements are traceable if their origin is known and the requirement can be referenced or located throughout the solution. The requirement should be traceable to a goal stated in the project charter, vision document, business case or other initiating document. Requirements are traceable backwards and forwards. • Traceable backwards: each requirement can be traced back to specific customer, user or stakeholder input, such as a use case, a business rule, or some other origin. It can also be traced from any specific point in the life cycle back to an earlier phase, component or document. • Traceable forward: each requirement should have a unique identifier that assists in identification, maintaining change history and tracing the requirement through the solution components.	
a. Are requirements uniquely identified?	
b. Can each requirement be traced to its origin or source, such as a scope statement, change request, business objective or legislation?	
c. Is each requirement identified such that it facilitates referencing in future development and enhancement efforts?	
d. Has each requirement been cross-referenced to previous related project documents?	
9. Unambiguous—Requirements must be clear, concise, simple and free from ambiguity. They must be stated without technical jargon, acronyms (unless defined) or other obscure verbiage. Requirements express objective facts, not subjective opinions. Vague requirements are often misunderstood resulting in rework and corrective actions during the design, development and testing phases. If the requirement can be interpreted in more than one way, it should be removed or clarified. • All readers of a requirement should arrive at the same interpretation of its meaning. • All specialized terms and terms that might be subject to confusion should be well defined.	
a. Are the requirements written with simple, short sentences?	
b. Are the requirements specified clearly enough to be turned over to an independent group for implementation and still be understood?	
c. Are functional requirements separated from non-functional?	

d. Are requirements stated in a manner that avoids the likelihood of multiple interpretations?	
e. Do all the requirements avoid conflicts with other requirements?	
f. Do any of the requirements contain undefined acronyms?	
g. Are all requirements stated in one place only?	
h. Have redundant requirements been consolidated?	
i. Has each requirement been specified separately, avoiding compound requirements?	
j. Are the requirements written with proper grammar and correct spelling?	
k. Do any requirements contain vague subjects, adjectives, prepositions, verbs and subjective phrases?	
l. Do any of the requirements express negative statements?	