Software Requirements Specification – Sample Checklist

1. Correct and Complete:
    a. All functional capabilities are documented.
    b. Reuse of existing software or use of COTS software is fully described.
    c. All adaptation requirements (e.g., geographic parameters, platform variations are identified).
    d. Applicable timing, resource usage (e.g., CPU, memory, bandwidth), and associated system load requirements are identified.
    e. The effect of the operating system, executive, or COTS has been factored into the performance requirements and resource budgets.
    f. Applicable safety requirements are identified.
    g. Applicable security requirements are identified.
    h. Applicable design constraint requirements (e.g., object-oriented design, language, support environment) are identified.
    i. Applicable software quality attributes (e.g., reliability, maintainability, testability) are identified.
    j. Applicable human performance/human engineering requirements are identified.
    k. Applicable acceptance criteria (e.g., test, inspection, demonstration) are identified.
    l. Delivery requirements (e.g., media, marking) are documented.
    m. Requirements are traceable to requirements allocated to software.
    n. TBD requirements are scheduled for completion as documented action items.
    o. All functional data flows are specified, including sources and destinations.
    p. Inputs and outputs of each requirement are necessary and sufficient for the specified processing. Purpose of the inputs and outputs are stated.
    q. Mathematical equations required as constraints on processing are given or referenced.
    r. Accuracy and precision requirements are defined.
    s. All software functions are considered (e.g., startup, restart, modes of operation, shutdown, normal terminations, abnormal conditions, performance monitoring and tuning).
    t. All operator interactions are considered.
    u. All functional processing requirements are specified for recognized error conditions (e.g., hardware faults, I/O errors, computational errors, processing overload, buffer overflow, events failing to occur, out-of-sequence events).
    v. All test requirements are defined (e.g., test levels; provisions to inject test data, to adjust parameters, to control or trace the execution of test runs, to extract and reduce test results).

2. Consistent:
    a. All software requirements are derived from the system specification.
    b. Each object is referred to by a unique name.
    c. Each object is defined by one set of characteristics which are not in conflict with one another.
    d. All requirements are free of logical conflicts.
    e. All requirements are free of timing conflicts.
    f. Each requirement is specified only once.
    g. Requirements do not conflict with each other.
    h. SRS is consistent with interface requirements specification and higher-level specifications.
    i. All data and messages are specified only once.
    j. Acronyms and abbreviations are defined and used consistently.
    k. Mathematical equations are defined consistently.
    l. Data flows are consistent with the specified inputs and outputs of the relevant requirements.
    m. Data flow notations are used consistently.
    n. Order and frequency of messages are consistent with the specified processing sequences and response times.
    o. Message data attributes are consistent with the inputs and outputs of relevant requirements.
    p. Loads used to allocate resource budgets are consistently specified for all functions.
    q. Function names used in diagrams are consistent with the requirements text.
    r. Requirements are consistent with the operational context.

3. Feasible:
    a. Data expected from external sources exists at those sources.
    b. Data sent to external destinations is expected at those destinations.
    c. Requirements are achievable with available technology.
    d. Necessary implementation tools are available.
    e. Scope of requirements is realistic, considering software estimates, schedules, and support facility plans.
    f. Based upon available facts or modeling information, the performance requirements are realistic (e.g., response times, accuracies, processing capacities).
    g. Resource budgets are realistic (e.g., CPU time, I/O utilization, memory, worst-case loads, data storage).

4. Meets Standards:
    a. Major software functions are described in relation to system operation.
    b. Requirements are clearly stated and unambiguous.
    c. Terminology is understandable and consistent.
    d. All notation and naming conventions are defined.
    e. Glossary is adequate.
    f. SRS adheres to required format.
    g. Requirements are clearly numbered or otherwise marked.
    h. Text has been edited for spelling and grammatical errors.
    i. Requirements terminology (i.e., “shall”, “will”, “may”, etc.) is used correctly.

5. Testable:
    a. All requirements are specified against the software (i.e., not against the hardware or the operator).
    b. All requirements can be verified by some (implicit, explicit, analytical, empirical) means.
    c. Test procedures can be written against all requirements, using existing or planned resources.
    d. Test results can be evaluated against predetermined acceptance criteria.