DESIGN

This checklist captures common elements that should be present in all design artifacts including documents, diagrams, prototype models, etc. Some items refer to the design of a system as a whole, and others can be applied individually to separate parts of a design. See the more focused CxOne design checklists for more detail in specific areas, e.g., *CxCheck_Architecture*.

General

SD-1 Do individual elements of the design (e.g., a diagram) conform to appropriate checklists?

SD-2 Do appropriate design artifacts conform to *CxCheck_CxOneArtifact*?

SD-3 Does the design support both product and project goals?

SD-4 Is the design feasible from a technology, cost, and schedule standpoint?

SD-5 Have known design risks been identified, analyzed, and planned for or mitigated?

SD-6 Are the methodologies, notations, etc. used to create and capture the design appropriate?

SD-7 Does the level of formality match the project size, project goals, and engineer expertise?

SD-8 Has the design been refined based on prototyping or implementation feedback?

SD-9 If possible, were proven past designs reused?

SD-10 Does the design support proceeding to the next development step?

Design Considerations

SD-11 Does the design have conceptual integrity? (i.e., does the whole thing hang together?)

SD-12 Can the design be implemented within technology and environmental constraints?

SD-13 Does the design use standard techniques and avoid exotic, hard-to-understand elements?

SD-14 Does the design emphasizes simplicity over cleverness?

SD-15 Is the design "as simple as possible, but no simpler"?

SD-16 Is the design lean? (i.e., are all of its parts strictly necessary?)

SD-17 If part of an evolutionary lifecycle or if change is expected in maintenance, do expected volatile areas of the design support expected changes or refactoring of the design? (e.g., have *design for change* principles been followed?)

SD-18 Does the design create reusable components if appropriate?

SD-19 Will the design be easy to port to another environment if appropriate?

SD-20 Does the design have low complexity?

SD-21 Is the design intellectually manageable?

SD-22 Is the design robust?

Requirements Traceability

SD-23 Does the design address all issues from the requirements?

SD-24 Does the design add features or functionality, which was not specified by the requirements? (i.e., are all parts of the design traceable back to requirements?)

SD-25 If appropriate has requirements coverage been documented with a completed requirements traceability matrix?

Level of Detail

SD-26   Is the level of detail appropriate for this design artifact?

SD-27   Has sufficient detail been included to allow the downstream consumer of the design to produce their artifacts (design, databases, code, test plan, etc.)?

Completeness

SD-28   Are all of the assumptions, constraints, design decisions, and dependencies documented?

SD-29   Has a risk plan been made for the parts of the design that may not be feasible?

SD-30   Are assumptions made due to missing information been documented?

SD-31   Have all reasonable alternative designs been considered, including not automating some processes in software?

SD-32   Have all goals, tradeoffs, and decisions been described?

Consistency

SD-33   Is the design consistent with its upstream and downstream artifacts?

SD-34   Does the design adequately address issues that were identified and deferred at previous upstream levels?

SD-35   Have the impacts of all 'To-Be-Determined' (TBD) issues in the upstream artifacts been assessed and addressed?

SD-36   Does the design make sense both from the top down and the bottom up?

SD-37   Is the design consistent with related artifacts? (i.e. other modules, designs, etc.)

SD-38   Is the design consistent with the development and operating environments?

Performance

SD-39   Are all performance attributes, assumptions, and constraints clearly defined?

SD-40   If appropriate are there justifications for design performance? (e.g., prototyping critical areas or reusing an existing design proven in the same context)

Maintainability

SD-41   Does the design allow for ease of maintenance?

SD-42   If reusable parts of other designs are being used, has their effect on design and integration been stated?

SD-43   Does the design account for future extensions to the program?

SD-44   Does the design resists erosion in the correctness of its content over time?

Compliance

SD-45   Does the design follow all standards necessary for the system? (e.g., date standards)

Modeling and Design Views

SD-46   When appropriate, are there multiple, consistent, models and/or views that represent of the design (i.e. static vs. dynamic)?

SD-47   When there are multiple models of the software (e.g., static and dynamic) are those models consistent with each other?