
TITLE:**Control of Software**

WARNING

This document is the property of United Technologies Corporation (UTC). You may not possess, use, copy or disclose this document or any information in it, for any purpose, including without limitation to design, manufacture, or repair parts, or obtain FAA or other government approval to do so, without UTC's express written permission. Neither receipt nor possession of this document alone, from any source, constitutes such permission. Possession, use, copying or disclosure by anyone without UTC's express written permission is not authorized and may result in criminal and/or civil liability.

1. PURPOSE & SCOPE

This procedure establishes a common framework for software life cycle processes. It contains processes, activities, and tasks that are to be applied during the development, operation and maintenance of deliverable and non-deliverable software products.

2. APPLICATION

- 2.1 This procedure defines supplier quality requirements as agreed upon by the following business entities as UTC members of the ASQR Common Specification Team herein referred to as "UTC member".

Aftermarket Operations	AO
Hamilton Sundstrand	HS
Pratt & Whitney	PW
Pratt & Whitney Canada	PWC
Sikorsky Aircraft	SAC
UTC Fuel Cells	UTCFC

- 2.2 This document applies to suppliers and all members of their supply chain who furnish product, material or services to any of the above UTC members.

3. DEFINITIONS

- 3.1 **Commercial-Off-The-Shelf (COTS) Software:** Commercially available applications sold by vendors through public catalog listings. COTS software is not intended to be customized or enhanced. Contract-negotiated software developed for a specific application is not COTS software.
- 3.2 **Deliverable Software:** All software, including software embedded in deliverable hardware.

- 3.3 **Non-Deliverable Software:** Software used in the design, manufacture, inspection, test acceptance or calibration that has a direct effect on a deliverable product. Examples include:
- Computer Numerical Control (CNC)
 - Numerical Control (NC)
 - Data Numerical Control (DNC)
 - Gage Calibration
 - Computer Aided Design (CAD)
 - Statistical Process Control (SPC)
 - Coordinate Measuring Machine (CMM)
 - Programmable Logic Control (PLC)
 - Executive, Robot Dipping
 - Sonic Wall Inspection
 - Heat Treat
 - Dot Peen
 - Shot Peen
 - Plasma Spray
 - Performance Acceptance Test
 - Burn-In
 - Hardware/ Software Qualification
 - Lab View scripts for PAT
 - Control Model Scripts used in Production Validation, Compiler, Assembler
- 3.4 **Obsolete Software:** Software that is no longer required for production.
- 3.5 **Product Integrity role:** The planned actions necessary to provide adequate confidence and evidence that a product satisfies a given set of requirements.
- 3.6 **Purchased/Procured Software:** Software that is not modified or customized by the receiving party. This type includes Commercial-Off-The-Shelf (COTS) software used for product realization.
- 3.7 **Software:** Computer programs, associated documentation, and data pertaining to the operation of a computer system.
- 3.8 **Software Archive:** The long-term storage to assure that software, documents, and life cycle data associated with the software product are retrievable for all versions of software used in production.
- 3.9 **Software Configuration Control:** The change control and storage of software during the development process.

- 3.10 **Software Support Tools:** Used for the analysis, design or support of products but not used in the manufacture or acceptance of production product (e.g., software models, tools used to facilitate repetitive calculations, excel tracking spreadsheet, software tools used to eliminate manual steps, web tools used to track development effort, access databases used to track and tabulate data, etc.).
- 3.11 **Software Validation:** Process to confirm that the software (1) conforms to its development standards, (2) all requirements are fulfilled and, (3) works as intended in the target environment. This confirmation process determines that the fully integrated software functions correctly, completely, and consistently with system specifications and requirements.
- 3.12 **Software Verification:** Evaluation which may occur at various times during a software process to assure input requirements at the end of a development stage have been met. Verification includes review, analysis, inspection and test.

4. REQUIREMENTS

- 4.1 **Non-Deliverable Software** – The supplier shall have procedure(s) that address the following minimum requirements:

- 4.1.1 Organizational responsibility and authority including product and process integrity.
- 4.1.2 Identification of requirements:
- Define the purpose or function of the software
 - Define the requirements and how the software requirements are initiated, documented and approved

- 4.1.3 Define Coding standards:
- Naming conventions including developmental version production file names.
 - Software Version
 - Header information
 - Comments

Note: *The preferred method is to segregate the production software from the test and development programs.*

Note: *In cases where the library contains production, test and developmental software programs, there shall be a unique identifier assigned to distinguish the three types {e.g., CMM_V1_dev, CMM_V1_test, and CMM_V1_Prod etc.}.*

4.1.4 Verification and Validation:

- Define the Verification and Validation process.
- Test procedure or test description and results shall be documented, reviewed and retained.
- Provide objective evidence that the software performs its required function.
- Trace software to requirements.
- Inspection review and approval of software must be performed by someone acting in an acknowledged product integrity role. Software used to verify quantitative values (e.g., CMM, etc.) requires an independent method of validation (i.e., layout inspection, fixture check or comparison with another CMM program previously verified by an independent method) and correlation of the two sets of results.
 - Acceptable correlation requires the difference to be within 10% of the tolerance for each characteristic. Differences greater than 10% but not exceeding 25% may be acceptable with documented justification.
 - Differences greater than 25% are not acceptable.
 - Variable data shall be recorded and retained.

4.1.5 Target Environment:

- Identify interfaces to other software and to target computer hardware.
- Identify the target computer hardware and software environment.

4.1.6 Version Control:

- Uniquely identify each version of the software.
- Identify each item that makes up a software product.

4.1.7 Change Control:

Define the software change process. This includes, but is not limited to:

- Identifying problems.
 - Analysis for problem cause
 - Implementation and verification of corrective action
- Re-verification and re-validation of software shall be employed to ensure that the modified software meets the changed requirements.

4.1.8 Access Control:

- Limited access control shall be defined and implemented. Examples of such controls include:
 - (a) Read and write access of the master and copies.
 - (b) Edit Key restrictions (e.g. NC, CNC Machine, etc.).

4.1.9 Archiving, Backup and Recovery:

- Define the process used to prevent the use of obsolete software programs. Software that is no longer required for production shall be restricted and/or removed from all systems so it is no longer available for use.
- Master copies, duplicates, and user copies shall be restricted and/or removed from all areas except the archive.
- Obsolete software in the archive shall have restricted access to prevent unauthorized use.
- Master copies shall be stored in a secure location.
- Software programs shall be archived in a manner that allows retrieval of all released versions of software programs for traceability purposes.

4.1.10 Identification, Storage, Handling and Release:

Define the method for identification, storage, handling and release of software to the user. The end user shall only access the latest software program version.

Note: *Multiple software programs may be stored in machine memory (e.g., NC, CNC, etc.) however, it is not recommended since the wrong software production program may be used. It is strongly recommended that only the production software program, in use, be stored in machine memory.*

4.1.11 Define training and maintenance requirements.**4.1.12 Documentation:**

- Define required documentation for software development.
- Define approval requirements for software being released to production.

4.1.13 Define the process of supplier oversight (i.e., audit and product acceptance).**4.1.14 Define the process used to accept Purchased or Vendor Supplied software (COTS) prior to initial use.**

- 4.1.15 Analysis of Risks and Criticality as applicable.
- 4.1.16 Software Support Tool Development Process:
- Define the software requirements and document them in the program folder or equivalent.
 - Design and code the software and document the activity in the program folder or equivalent.
 - Execute a functional test of the software and document the activity in the program folder or equivalent.
 - Control the Software and documentation using internal configuration management procedures.
- 4.1.17 Define the internal audit or review processes for software to ensure compliance to established software development, procurement and control procedures.
- 4.2 **Deliverable Software** – Maintain a system that meets or exceeds the following requirements or as specified by the contract or Purchase Order.
- 4.2.1 All software plans discussed in this section shall be submitted to the applicable UTC member for review and approval prior to the start of the software development process via [ASQR-01 Form 3](#). All subsequent revisions/changes shall also be submitted for review and approval.
- 4.2.2 [RTCA/DO-178](#) shall be the preferred approach for all software development and meets the requirements as specified herein. The supplier shall complete and maintain a checklist that defines the [RTCA/DO-178](#) requirements when submitted by the appropriate UTC member.
- 4.2.3 The Software Quality Assurance (S/W QA) Plan shall include the following:
- A description of the S/W QA environment, including the scope, organizational responsibilities and interfaces, standards, procedures, tools and methods.
 - A statement of the S/W QA authority, responsibility and independence, including the approval authority for software products.
 - The S/W QA activities that are to be performed for each software life cycle process and throughout the product development including:
 - (1) S/W QA methods, (e.g., reviews, audits, reporting, inspections, and monitoring of the software life cycle processes, etc.)
 - (2) Activities related to the problem reporting, tracking and corrective action system

- (3) A description of the method used to ensure disposition and retention of any remaining S/W QA open action items, change requests, and completion of all software development tasks at the conclusion of the program
- A definition of the records to be produced by the S/W QA process.

4.2.4 The Software Development Plan (SDP) shall include the following:

- Identification of software being developed
- Resources (e.g., requirement, design code and verification environment etc.)
- Organizational structure and responsibilities
- Software Development process (e.g., including prototype and flight test software etc.)
- Software Development schedule and milestones
- Quality and project records
- Integrated Product Teams
- Formal reviews
- Computer resource utilization
- Corrective action process
- Risk management
- Control and development of software tools
- Software metrics
- Subcontractor management
- Security and safety requirements
- Data Management/Software Development libraries including documents to be produced
- Program language(s)
- Standards (e.g., requirements, Design code etc.)
- Development and formal configuration management (if included in the SDP)

4.2.5 The Software Configuration Management Plan shall include both the developmental and formal configuration management process and the following:

Note: *This plan can be included as part of the SDP.*

- Configuration Identification of all life cycle artifacts (e.g., Software unique identifier etc.)
- Configuration Control (e.g., Developmental and Formal, etc.)

- Subcontractor Configuration Management
- Organization and Resources
- Software Configuration Management Roles and Responsibilities
- Storage, Handling and Security
- Authorization including the release of project media and master versions of software
- Version Control
- Configuration Status Accounting
- Configuration Audits (e.g., Physical, Functional and Software Development Library Audits, etc.)
- Access Control

4.2.6 The Software Test or Verification Plan shall address Computer Software Unit or Module Test and Computer Software Configuration Item (CSCI) Test and include the following:

Note: *This plan can be included as part of the SDP.*

- Identification of the CSCI
- Software Test Environment including hardware and software elements
- How installation and Test activities are controlled
- Configuration and Change Control including test environment
- Regression Analysis
- Data Recording, Reduction, Analysis, and Retention including a plan for formal results

5. REFERENCES

5.1 Available from Radio Technical Commission for Aeronautics Publications RTCA Secretariat Suite 500, 1425 K Street N.W. Washington, D.C., 20005, USA

- [RTCA/DO-178](#) Software Considerations in Airborne Systems & Equipment Certification

5.2 UTC Member Company Publications/Forms:

- [ASQR-01 Form 3](#) Supplier Request for Information (SRI)

6. NATURE OF CHANGE

- Initial Issue – Software requirements formerly part of ASQR-01 are now contained in this document.

***** End of Document *****