TITLE

Control of Software

1. PURPOSE AND SCOPE

This document establishes a common framework for software life cycle processes. It contains requirements for the development, operation and maintenance of Manufacturing, Test and Support software (reference Section 4.1) and Deliverable Software (reference Section 4.2) and is used in the design, analysis, manufacture, inspection, test, or calibration which directly affects or verifies the configuration, conformity, or quality of a product.

Business Systems software is excluded from the requirements of this document.

2. APPLICATION

2.1 Supplier quality requirements defined in this document are agreed upon by and applicable to the following UTC Aerospace Business Entities (also known as Members):

UTC Aerospace Systems – Aircraft Systems	UTAS - AS
UTC Aerospace Systems– Power Control & Sensing Systems	UTAS - PCSS
Pratt & Whitney	PW
Pratt & Whitney Canada	PWC
Sikorsky Aircraft	SIK

Note: *The requirements of this document do not apply to Aftermarket Operations.*

2.2 This document applies to suppliers and their subcontractors who furnish product, material or services (as a manufacturer or maintenance provider) to any of the above Members as a contract requirement regardless of supplier's industry, regulatory accreditation, or certification status.

3. DEFINITIONS

3.1 Software Applications:

3.1.1 Airborne software: Software for airborne systems and equipment used on aircraft (aircraft Bill of Material).

3.1.2 Business Software: Office software not intended for delivery, and does not directly affect product configuration, conformity, or quality.

3.1.3 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.
Note: *Contract–negotiated software developed for a specific application is not COTS software.*

3.1.4 Deliverable Software: Software delivered to an external customer or supplier. This may be airborne, ground based, manufacturing, test and support software that may be embedded with hardware.

3.1.5 Executive Software: Machine system software. Non-part specific software used to run automated equipment, usually provided by the machine vendor.

3.1.6 Ground Based Software: Software used within ground based systems (non-flight software) such as engine monitoring systems.

3.1.7 Manufacturing and Test Software:

Software used in the design, analysis, manufacture, inspection, acceptance, test or calibration that has a direct effect on the configuration, conformity or quality of deliverable product.

Examples include:

- Part specific program (e.g. Computer Numerical Control (CNC) and Coordinate Measuring Machine (CMM)
- Gage Calibration
- Computer Aided Design Models (CAD)
- Programmable Logic Control (PLC)
- Executive software, (e.g., robot dipping, CMM, CNC, etc.)
- Special Process Software (e.g., heat treat, shot peen, sonic wall inspection, plasma spray, etc.)
- Dot Peen
- Performance Acceptance Test (PAT or Acceptance Test (AT)
- Burn-In
- Hardware/ Software Qualification
- Lab View scripts for PAT
- Control Model Scripts used in Production Validation, Compiler, Assembler

- Engineering test equipment software

3.2 Engineering Model: Un-shippable product (model) built with un-released documentation for engineering use.

3.3 External Customer: A customer who is external to their UTC business unit.

3.4 Independent Method of Inspection: Method of inspection using calibrated and traceable measuring & test equipment using inputs different from those of the software under test such that programming errors, specification interpretation errors, and decisions made by the programmer that affect the accuracy of the inspection are not duplicated. Independent methods of inspection may include hand gauging, bench layout, previously approved automated inspection programs, manual-mode machine inspection involving a different programmer/inspector or automated inspection programs involving a different programmer/inspector.

3.5 Non-Current Software: Software that is no longer required for production.

3.6 Purchased/Procured Software: Modified or customized software that is purchased from a third party vendor.

3.7 Recommended Practice: Items that are not required, however, are useful to an organization to enhance the quality, and maintainability of the software.

3.8 Software: Computer programs, associated documentation, and data pertaining to the operation of a computer system.

3.9 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.

3.10 Software Configuration Control: The change control and storage of software during the development process.

3.11 Software Maintenance: The modification of a software product after delivery to correct faults, to improve performance or other attributes, or to adapt the product to a modified environment.

3.12 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.).

3.13 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.

3.14 Traceability: Existence of materials or artifacts that show a direct link between requirements, program logic and test.

4. REQUIREMENTS

4.1 Manufacturing and Test and Support Software:

The supplier shall develop, release and update software in accordance with each phase and requirements defined below:

4.1.1 *Requirements Phase -*

The Requirements phase shall define the purpose or function of the software which includes the requirements and how those requirements are initiated, documented and approved.

The Requirements phase should include an Engineering drawing, specification and/or work instruction or equivalent that is specific to the requirements and it shall be revision controlled. The requirements (e.g., Engineering drawing, specification and/or work instructions or equivalent) shall be retained (e.g., electronically or manually) and placed under configuration control.

Note: *Performing a comparison from the current version and the previous version is not an acceptable form of documenting the changes to the software requirements.*

4.1.2 *Coding Phase -*

Coding standards shall be defined and include but not limited to:

- Software naming conventions (e.g. modules and executable software)
- Naming conventions including developmental and production file names
- Header information with unique identifier and revision as a minimum

4.1.2.1 Changes in code shall trace to requirements. The change history for modifications shall be documented in the program header or supporting version control system.

4.1.2.2 Provide the operator (Test, Manufacturing or Engineering) with the ability to verify that the correct software has been loaded. This can be done by verifying the following type of software identification software:

- Name or unique identifier
- Version
- Date
- Time

4.1.3 *Verification Phase -*

Prior to releasing the software to production the following applies:

- Define the verification phase
- All test procedures and results shall be documented and reviewed.
- The test documentation shall show that the software performs its required function and that the testing meets all the requirements.
- Each new or revised software program shall be tested prior to use and unapproved programs shall not be used for production purposes. The first run of a software program must be either:
  - Dry run
  - Tested on a suitable test piece and should be representative of the part
  - Verified using simulation software (e.g., vericut, etc.), or
  - A computerized comparison of the original software program prior to use

  Note: *An equivalent method to the above is acceptable.*
- All verification or test documentation, including results, shall be retained (e.g., electronically or manually) and placed under configuration control as objective evidence that "verification" or "test " has been performed prior to production use.

  Note: *The recommended practice for verification of the software/system (e.g., test bench) is that it be performed independent of the engineering or production unit to determine that the software/system functions as intended and to avoid any damage to the product prior to connection to the new/modified software.*
  - Software for automated inspection (e.g. CMM, etc.) shall be verified by correlation of the test results with the results from an independent method of inspection.
  - For P&W suppliers, 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.
  - Variable data shall be recorded and retained.
  - The independent method also requires a different person other than the person who created the inspection program to perform the verification.

4.1.4 *Version Control:*

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

  Note: *The suggested method is to release a drawing to describe the software, or a software configuration index to describe the version of the software which includes executable code*.

4.1.5 *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 of software to ensure that the modified software meets the changed requirements
  Note: *The recommended practice is to use a common software problem reporting/change management system to track changes to the software*.

4.1.6 *Access Control:*
Access control restrictions shall be defined and implemented. Examples of such controls include but not limited to:

- Read and write access of the master and copies
- Edit Key restrictions (e.g. NC, CNC Machine, etc.)
- Password protected

4.1.7 *Archiving, Backup and Recovery:*
Archiving, Backup and Recovery requirements shall be defined and implemented. Examples of such controls include but are not limited to:

- Master copies, duplicates, and user copies shall be restricted and/or removed from all areas except the archive.
- Non-current software in the archive shall have restricted access to prevent unauthorized use.
- All non-current versions and development software programs shall be removed from machine memory.
- Software programs shall be archived in a manner that allows retrieval of all released versions of software programs for traceability purposes.
- Software stored in the archive shall be identified, and segregated so that the appropriate version of software can be consistently and accurately located and retrieved
- The released software shall be able to be rebuilt from the archived source code, object code and components with only the addition of Purchased/Procured/COTS software (compilers, linkers, loaders). Components should include items such as required operating system, required software package and required hardware.
  Note: *Consideration should be given for obsolescence management for regeneration for teat equipment software (e.g., compilers, linkers).*
- Documentation (User Guides), text files (used for information or for test cases), batch files, executable files, and any similar files shall be archived with the source code, or the location of where they are archived should be referenced in the files that are archived/archive documentation.

4.1.8 *Identification, Storage, Handling and Release:*

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

4.1.9 Purchasing Information for Purchased/Procured Software: Purchase documentation (PO/Statement of Work) shall describe the product to be purchased, and be approved prior to communication to the supplier. Documentation shall include requirements for:

- Unique product identification
- Applicable specifications, drawings or technical data
- Inspection/acceptance requirements
- Testing and associated test requirements
- Problem reporting requirements
- Change notification requirements
- Export classification requirements
- Software licensing, and redistribution requirements

4.1.10 Export Compliance and Classification requirements shall be met. The preferred method is for the software and associated test artifacts to automatically display the export classification. All software shall contain an additional End User License Agreement (EULA) file that should contain details on any company licensing, Open Source re-distribution licensing and/or the application export classification.
Note: *It is suggested that the EULA file be named the same as the primary application or binary file (e.g., application.exe>application-EULA.text).*

4.1.11 The supplier shall have a defined training program.

4.1.12 The supplier shall be responsible for supplier oversight when work is outsourced (e.g., audits, product acceptance etc.).

4.1.13 The supplier shall be responsible, at a minimum, to test the COTS and Executive software in the environment prior to initial use. Objective evidence that the software meets its intended use shall be retained.

4.1.14 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 PO.

4.2.1 Suppliers who receive a PO from a Member where the supplier's product offering to said purchase order includes deliverable software, shall meet the intent of AS9115 as a minimum.

4.2.2 Deliverable Airborne Software

4.2.2.1 RTCA/DO-178 shall be the preferred approach for deliverable airborne software development. The supplier shall complete and maintain a checklist that shows compliance to RTCA/DO-178 requirements when requested by the Member.

4.2.2.2 All software plans required in RTCA/DO-178 shall be submitted to the applicable Member for review and approval prior to the start of the software development process. All subsequent revisions/changes shall also be submitted for review and approval.
Note: *Additional regulatory orders and issue papers may apply.*

4.2.2.3 For software that meets the equivalent of RTCA/DO-178level E criticality, the supplier shall submit its planned software development process for review and approval.

4.2.3 RTCA/DO-178 Alternative: When RTCA/DO-178 is not used for the software development process, the supplier shall submit to the applicable Member the alternative software development process for review and approval prior to the start of the software development process. All subsequent revisions/changes shall also be submitted for review and approval. Examples of this type of deliverable software include:

- Manufacturing
- Test
- Ground Based Systems
- Deliverable airborne software that does not meet the requirements of paragraph 4.2.2

5. RECORDS/FORMS

Completed Quality records generated electronically or on paper, shall be retained per the requirements of ASQR-01.

6. REFERENCES

It is the responsibility of the supplier to obtain copies of non–UTC documents specified herein. These include, but may not be limited to the following:

Document	Title
AS 9115	Quality Management Systems – Requirements for Aviation, Space and Defense Organizations – Deliverable Software
RTCA/DO-178	Software Considerations in Airborne Systems & Equipment Certification

7. NATURE OF CHANGE

This document has been completely revised. Major changes include:

- Document put in new template
- Definitions added
- Associated phases of software development process defined
- Updated requirements for archiving
- Requirements added for procured software, export compliance, RTCA/DO-178 Alternative and equivalent RTCA/DO-178 level E criticality
- Deliverable software requirements updated and deleted reference to software plans

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