Certification Processes for Safety-Critical and Mission-Critical Aerospace Software

FOR

NASA Ames Research Center

Dated: June 30, 2003

Contributors: Stacy Nelson, ARC/Nelson Consulting

TABLE OF CONTENTS

RECORD OF REVISIONS

REVISION	DATE	SECTIONS INVOLVED	COMMENTS
Initial Delivery	1/31/03	Sections 1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12 and 13	This is a draft only and not intended to be the final deliverable.
Update	2/12/03	Section 8	Description of JPL mission critical processes
Final Version	6/30/03		Update per comments

1. TERMINOLOGY

The following terminology is used throughout this document:

- **Certification** - legal recognition by the certification authority that a software product complies with the requirements[1]
- **Mission critical**: loss of capability leading to possible reduction in mission effectiveness[2]
- **Safety-critical** means failure or design error could cause a risk to human life[2]

2. EXECUTIVE SUMMARY

This document is a quick reference guide with an overview of the processes required to certify safety-critical and mission-critical flight software at selected NASA centers and the FAA. Researchers and software developers can use this guide to jumpstart their understanding of how to get new or enhanced software onboard an aircraft or spacecraft.

The introduction contains aerospace industry definitions of safety and safety-critical software, as well as, the current rationale for certification of safety-critical software. The Standards for Safety-Critical Aerospace Software section lists and describes current standards including NASA standards and RTCA DO-178B.

The Mission-Critical versus Safety-Critical software section explains the difference between two important classes of software: safety-critical software involving the potential for loss of life due to software failure and mission-critical software involving the potential for aborting a mission due to software failure.

The DO-178B Safety-critical Certification Requirements section describes special processes and methods required to obtain a safety-critical certification for aerospace software flying on vehicles under auspices of the FAA

The final two sections give an overview of the certification process used at Dryden Flight Research Center and the approval process at the Jet Propulsion Lab (JPL).

- Results of all verification and validation activities
- Records of safety reviews
- Records of any incidents which occur throughout the life of the system
- Records of all changes to the system and justification of its continued safety[4]

The unique certification processes for each regulatory body have been published in standards. Section 4 provides an overview of NASA and RTCA standards for certification of safety-critical software.

Section 5 provides a definition of safety-critical versus mission critical software used at NASA and in DO-178B. Sections 6 through 8 contain examples of certification or approval processes for FAA, DFRC, JPL and ARC as follows:

- Section 6 describes the certification processes for RTCA DO-178B safety-critical certification
- Section 7 contains an overview of the Dryden Flight Research Center (DFRC) safety-critical certification
- Section 8 depicts the Jet Propulsion Lab (JPL) Approval Process

4. STANDARDS FOR SAFETY-CRITICAL AEROSPACE SOFTWARE

Standards are nothing more than the accumulation of lessons learned from previous projects so the software development process continually improves and developers don't make the same mistakes over and over again. In an effort to produce safe, quality, aerospace software faster and cheaper, standards have been written containing the lessons learned on aerospace software projects. They have guidelines that can be tailored for the specific characteristics of a project. Standards pertaining to safety-critical aerospace software can be divided into three categories:

- NASA Standards including center specific standards like the Dryden Flight Research Center Policies
- RTCA DO-178B – used by the FAA to regulate commercial aerospace software
- MIL-STD 498 – military standards

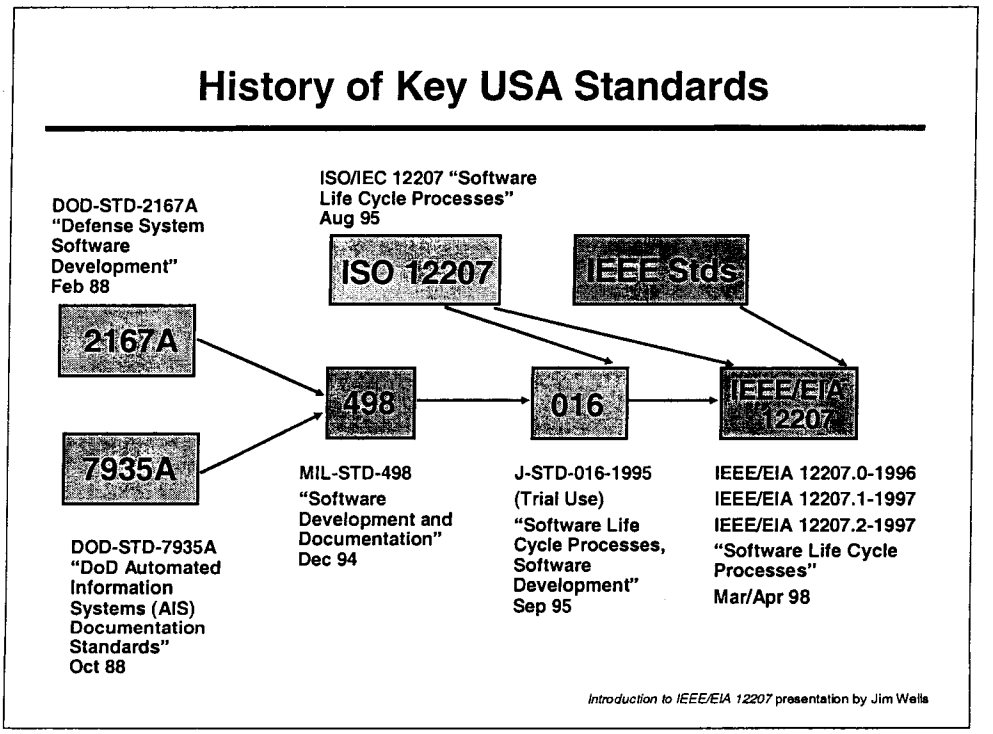


Figure 1: History of Key USA Standards[6]

Figure 1 depicts on overview of the history of key US standards. Reading from left to right, DOD-STD 2167A and DOD-STD-7935A were combined to form MIL-STD 498 which is currently used for military software development. Further discussion of military standards is outside the scope of this project.

Information from ISO/IEC 12207 in combination with J-STD-016-1995 and various IEEE standards was updated and clarified in IEEE/EIA 12207. IEEE/EIA 12207 contains concepts and guidelines to foster better understanding and application. It is divided into three volumes:

- 12207.0 – Software Life Cycle Processes
- 12207.1 – Software Life Cycle Processes Life Cycle Data
- 12207.2 – Software Life Cycle Processes Implementation Considerations

Each of these US Standards has at least one UK/European counterpart; however for purposes of this report, focus will be on the NASA and RTCA DO-178B standards.

Work is currently underway at NASA Ames Research Center to augment the IEEE/EIA 12207 standards with lessons learned from successful space missions and other pertinent research. NPG 2820.DRAFT is a preliminary draft of these standards.

NASA Dryden Flight Research Center defined special airworthiness and flight safety review standards to specify steps necessary to ensure safety and mission success for flight research missions. These are contained in Dryden Center Policies (DCP) and Handbooks (DHB) and can be found at http://www.dfrc.nasa.gov/DMS/dms/html.

4.1. NASA Standards Used at DFRC, ARC and JPL

The following list contains current standards used at Dryden, Ames and JPL for safety critical software:

- NASA Guidebook for Safety Critical Software, NASA-GB-1740.13-96
- Trial-Use Standard for Information Technology Software Life Cycle Processes - Software Development, J-STD-016-1995
- Dryden Flight Research Center policy: DCP-S-007
- Dryden Handbook Code X - Airworthiness and Flight Safety Review, Independent Review, Mission Success Review, Technical Brief and Mini-Tech Brief Guidelines DHB-X-001 Revision D
- Dryden Flight Research Center Policy: Airworthiness and Flight Safety Review Process, DCP-X-009 Revision B
- Dryden Flight Research Center Policy: Flight Operational Readiness Review (ORR) and Operational Readiness Review Panel (ORRP), DCP-X-020 Revision A
- IEEE Standard for Software Test Documentation, IEEE Std 829-1998 (Revision of IEEE Std 829-1983)
- NASA Software Safety Standard NASA STD 8719.13A
- NASA-GB-1740.13-96, NASA Guidebook for Safety Critical Software
- DRAFT NASA-GB-8719.13 NASA Software Safety Guidebook
- NASA Procedures and Guidelines (NPG) 2820.DRAFT, NASA Software Guidelines and Requirements[7]

NPG 2820.DRAFT references the following IEEE/EIA Standards[8]:

- *12207.0 - Standard for Information technology – Software Life Cycle Processes (March, 1998)12207.1 - Standard for Information technology – Software Life Cycle Data (April, 1998)*
- *12207.2 - Standard for Information technology – Software Implementation Considerations (April, 1998*

The IEEE documents reference the ISO and IEC standards published as ISO/IEC 12207 in 1995.

4.2. RTCA DO-178B[1]

In addition to the NASA standards, DO-178B, "Software Considerations in Airborne Systems and Equipment Certification" contains guidance for determining that software aspects of airborne systems and equipment comply with airworthiness certification requirements.

Written in 1980 by the Radio Technical Commission for Aeronautics (now RTCA, an association of aeronautical organizations of the United States from both government and industry), it was revised in

1985 and again in 1992. During the 1992 revision, it was compared with international standards: ISO 9000-3 (1991), "Guidelines for the Application of ISO 9001 to the Development, Supply and Maintenance of Software" and IEC 65A (Secretariat) 122 (Draft – 11-1991), "Software for Computers in the Application of Industrial Safety-Related Systems" and considered to generally satisfy the intent of those international standards.

RTCA also published the following documents to clarify DO-178B:

- DO-248B – explains best practices in applying DO-178B
- DO-278 – provides an extension to standards for ground-based facilities

5. MISSION-CRITICAL versus SAFETY-CRITICAL SOFTWARE

Both NASA and the FAA classify software into two broad categories based on the risk associated with software failure or defective software design:

- **Mission critical** meaning a loss of capability leading to possible reduction in mission effectiveness[2]
- **Safety-critical** meaning a failure or defective design could cause a risk to human life[2]

5.1. NASA Software Level Definitions

Software level definitions vary slightly across NASA centers but the broad categories are essentially the same. For purposes of this paper, the software level definitions at NASA Dryden are used as an example because Dryden is responsible for both safety and mission-critical software.

NASA Dryden denotes mission critical software as Class B and safety-critical software as Class A. For example, failure of Class B software might result in inability to collect data for a research project, but the pilot could safely fly and land the aircraft. While errors and/or design flaws in Class B software will not threaten life or aircraft, certification is still important due to the high cost of repeating missions should the software fail.[2]

Failure of Class A safety critical software would put the aircraft and pilot in danger. Therefore, testing involved in certification of Class A software is more stringent than for Class B.[2]

5.2. DO-178B Software Level Definitions

While DO-178B classifies software in more detail according to five levels described below, the overall idea is the same – more rigorous certification processes and methods are required for safety critical software.

- Level A – software whose anomalous behavior would cause or contribute to a catastrophic failure that would prevent safe flight and landing
- Level B - software whose anomalous behavior would cause or contribute to a hazardous/severe-major failure condition. Hazardous/Severe-Major is defined as failure conditions that reduce the capability of the aircraft or crew to cope with adverse operating conditions to the extent that safety is jeopardized, the physical demands on the crew are excessive to the point of being impossible and serious or fatal injuries may occur.
- Level C - software whose anomalous behavior would cause or contribute to a major failure with significant reduction in safety, increase in crew workload or conditions impairing crew efficiency or discomfort or injury to occupants
- Level D - software whose anomalous behavior would cause or contribute to a minor failure that would not significantly reduce aircraft safety and where crew actions would not be impaired but the crew might be inconvenienced
- Level E - software whose anomalous behavior would have no effect on operational capability of the aircraft and would not increase crew workload[1]

6. DO-178B SAFETY-CRITICAL CERTIFICATION REQUIREMENTS

DO-178B, "Software Considerations in Airborne Systems and Equipment Certification" contains specific guidance for certification of safety-critical software. It was developed by the RTCA in the United States while the European EUROCAE wrote the ED-12B, a similar standard.

Both RTCA and EUROCAE are independent industry-wide organizations comprised of a cross-section of members from the aerospace industry, as well as, the Federal Aviation Administration (FAA) and Joint Aviation Authorities (JAA).

The regulators in the United States and Europe decided that these two standards represent the best way to assure pilot and passenger safety so they published directives called Technical Standards Orders (TSO) that force aerospace companies to comply with these standards and to demonstrate compliance by certifying their software. Under Title 14 Code of Federal Regulations (CFR) Part 183, Representatives of the Administrator, the FAA is permitted to delegate some findings of compliance to Designated Engineering Representatives (DER). IN fact, DERs provide the majority of data approvals for airborne products in the US each year.

In order to comply, suppliers in the aerospace industry must understand that DO-178B considers software as part of the airborne system or equipment installed on the aircraft or engine and does not certify software as a unique, stand-alone product.

6.1. Safety-Critical Certification Process

The certification process includes the following steps where the applicant is a supplier of aerospace software and the certification authority is the organization or person responsible within the state or country concerned with the certification:

- Applicant meets with the certification authority to establish the certification basis or criteria for the aircraft or engine
- Applicant develops a Plan for Software Aspects of Certification (PSAC) to meet the certification basis. The PSAC includes:
    - System overview explaining the:
        - System functions and their allocation to the hardware and software
        - Architecture
        - Processor(s)
        - Hardware and software interfaces
        - Safety features
    - Software overview describing the software functions with emphasis on the proposed safety and partition concepts like resource sharing, redundancy, multiple-version dissimilar software, fault tolerance and timing/scheduling strategies
    - Certification considerations including:
        - Means of compliance
        - Software level (A-E)
        - Summary of the justification provided by the system safety assessment process including potential software contributions to failure conditions
    - Software Life Cycle section containing a description of the software with reference to respective detailed software plans and a summary explaining how the objectives of each software life cycle process will be satisfied and which organization is responsible. The

minimum software life cycle data that may be submitted to the Certification Authority is the:

- PSAC
- Software Configuration Index (SCI) – described in Section 6.5
- Software Accomplishment Summary (SAS) – described in Section 6.4
- Software Verification Cases and Procedures

- Software Life Cycle Data section including a description of any data to be produced and controlled by the software along with how the data relate to each other. It should also include information about how the data will be submitted to the certification authority (diskette, CD...) and the form of the data (ie text file, binary file...).
- Schedule
- Additional considerations like tool qualification, previously developed software, COTS software, et al.

- Certification authority assesses the PSAC for completeness and consistency by comparing it to the certification basis
- Certification authority satisfies itself that proposed software level is appropriate
- Certification authority apprises applicant of any issues that must be satisfied prior to certification
- Certification authority determines whether the aircraft or engine (including software) complies with the certification basis by reviewing the SAS and evidence of compliance. The Certification authority may also review at its discretion the software life cycle processes and their outputs.[9]

6.2. Additional Certification Considerations

Additional certification considerations include:

- Use of Previously Developed Software
- Tool Qualification
- Alternative Methods

6.2.1. Use of Previously Developed Software

Frequently, software will rely upon COTS or other previously developed software. Certification of these modifications takes into account the following:

- Change of software level
- Impact of modification on requirements, architecture, installation, development environment, target processor or other hardware and integration with other software

DO-178B lists specific methods for ensuring the safety of any modifications. These methods include:

- Reverse engineering to obtain software life cycle data that is inadequate or missing
- Comparison of failure conditions to previous application
- Upgrading development baseline if product history is necessary to satisfy certification objectives
- Repetition of hardware/software compatibility reviews
- Additional integration tests and reviews as necessary

6.2.2. Tool Qualification

Qualification of a tool is needed when processes described in DO-178B are automated. The objective is to ensure that the tool provides at least the same confidence as the manual process. The concept of tool qualification is unique to civilian aviation. Other industries do not typically require tool qualification prior to use.

For qualification purposes, tools are divided into two categories:

- Development Tools – tools whose output is part of airborne software and can introduce errors
- Verification Tools – tools that cannot introduce errors, but may fail to catch them

6.2.2.1. Qualification of Development Tools

Software development tools must be qualified to ensure they do not introduce errors into airborne software. Qualification criteria include the following:

- The software development process for the tool must satisfy the same objectives as the development process for airborne software
- The software level must be the same for the development tool and the airborne software. A different level may be applied if the tool provides a significant reduction in verification activities (like an auto-coder).
- The tool must be verified against Tool Operational Requirements. This may involve a trial period during which tool output is verified.

The certification authority qualifies a software development tool after considering the following:

- The tool must meet specific criteria outlined in the Tool Qualification Plan which contains:
  - Configuration identification of the tool
  - Details of the certification credit sought
  - Software level
  - Tool qualification activities to be performed
  - Tool qualification data to be produced
- The Tool Accomplishment Summary (similar to the Software Accomplishment Summary described in a Section 6.4) must be provided illustrating compliance with the Tool Qualification Plan.

6.2.2.2. Qualification of Verification Tools

Verification tools must be qualified to make sure that the tool catches the errors they were designed to find. Qualification criterion includes checking that the tool complies with its Tool Operational Requirements under normal operational conditions.

The certification authority qualifies a verification tool after inspecting the SAS and other materials necessary to prove that the tool complies with the PSAC.

6.2.3. Alternative Methods

Alternative methods may be used to support software qualification. DO-178B describes the following alternative methods:

- **Formal methods** involving the use of formal logic, discrete mathematics and computer-readable language to improve the specification and verification of software

- **Exhaustive Input Testing** for situations where the inputs and outputs of software can be bounded and exhaustively tested
- **Software reliability models** including methods for estimating the post-verification probabilities of software errors. At the time of publication, DO-178B did not consider these techniques mature enough for safety critical software.
- **Product service history** demonstrating that the software has a track record of safety

An alternative method cannot be considered in isolation from the software development processes. The applicant must show that the alternative method satisfies the objectives of DO-178B.

In order to use an alternative method, the applicant must specify the following in the PSAC:

- Impact of the proposed method on the software development process and life cycle data
- Rationale behind the alternative method clearly showing how it meets safety objectives

6.3. Modified Condition and Decision Coverage (MCDC)

Modified Condition/Decision Coverage is a structural coverage criterion required by DO-178B for Level A software. It addresses exercising of Boolean expressions throughout the software, presumably because Boolean logic is commonplace in flight-critical software, especially in control laws. Each decision (a top-level Boolean expression) must be exercised to check both True and False outcomes.

MCDC levies further coverage requirements if a decision is composed of multiple conditions (a condition is a Boolean subexpression of a decision) connected by Boolean operators, as in (A and (B or C)). The additional requirement is to demonstrate that each condition can independently influence the outcome of the decision. That is, there exists a set of value for all other conditions in the decision for which toggling the value of this one condition will toggle the outcome of the decision. For example, in the decision (A and (B or C)), a value of False for B and True for C will demonstrate that A can independently affect the outcome of the decision, as the following truth table illustrates:

A	B	C	(A and (B or C))
T	F	T	T
F	F	T	F

where T = True and F = False

Here, changing A from T to F while holding the values of B and C constant changes the value of the decision. There may be other combinations of values for B and C which will also demonstrate the independence of A, but one combination is all that is needed. Likewise, conditions B and C must be demonstrated to independently affect the outcome of the decision. It can be shown that a minimum of (N + 1) test cases will be needed to accomplish MCDC for a decision containing N distinct conditions.

There is one other, somewhat unrelated, requirement included in MCDC: each entry and exit point of a subprogram must be exercised. This requirement was most likely included for completeness to ensure explicit coverage of these entry and exit points for Level A systems.

Beyond the simple cases where decisions consist of familiar Boolean operators and all distinct conditions, there is controversy surrounding the meaning of MCDC and how to apply it. Investigation is under way at NASA Langley to study the variations that exist and to recommend ways of promoting a uniform interpretation of MCDC.[10]

Appendix E contains a simple Simulink code coverage example including MCDC.

6.4. Software Accomplishment Summary (SAS)[1]

The SAS is the primary document for showing compliance with the PSAC. It contains the following:

- System overview explaining the:
    - System functions and their allocation to the hardware and software
    - Architecture
    - Processor(s)
    - Hardware and software interfaces
    - Safety features

  Also describes any differences from the system overview in the PSAC.
- Software overview including software functions with emphasis on the proposed safety and partition concepts like resource sharing, redundancy, multiple-version dissimilar software, fault tolerance and timing and scheduling strategies. Also describes any differences from the system overview in the PSAC.
- Certification considerations including a restatement of PSAC certificate considerations and description of any differences.
- Software characteristics including the executable object code size, timing and memory margins, resource limitations and the means of measuring each characteristic
- Software Life Cycle section containing a description of the software with reference to respective detailed software plans and a summary explaining how the objectives of each software life cycle process will be satisfied, which organization is responsible and the certification liaison responsibilities. Also describes any differences from the system overview in the PSAC.
- Software Life Cycle Data section including a description of any data to be produced and controlled by the software along with how the data relate to each other. It should also include information about how the data will be submitted to the certification authority (diskette, CD...) and the form of the data (ie text file, binary file...). Also describes any differences from the system overview in the PSAC.
- Additional considerations section that summarizes certification issues that may warrant the attention of the certification authority
- Change history
- Software status section including a summary of problem reports unresolved at the time of certification
- Compliance statement section stating compliance with this document and summarizing the methods used to demonstrate compliance and any additional rulings or deviations for plans, standards or this document.

6.5. Software Configuration Index (SCI) [1]

The SCI identifies the configuration of the software and should identify the following:

- Software product
- Executable object code
- Source code components
- Previously developed software
- Software life cycle data

- Archive and release media
- Instructions for building the executable object code
- Reference to the Software Life Cycle Environment Configuration Index – identifies the environment where the software will run
- Data integrity checks, if used

6.6. Other Considerations for FAA Certification[11]

In addition to DO-178B, the FAA considers the documents shown in the following diagram:

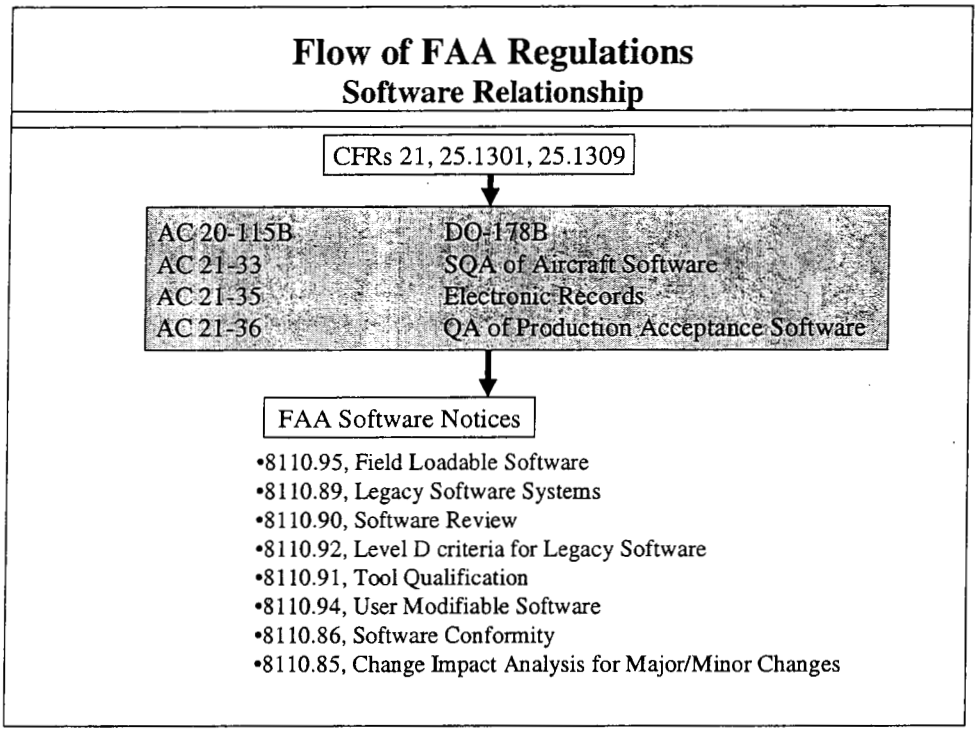


Figure 2: Flow of FAA Regulations

The following flow shows the FAA approval process.

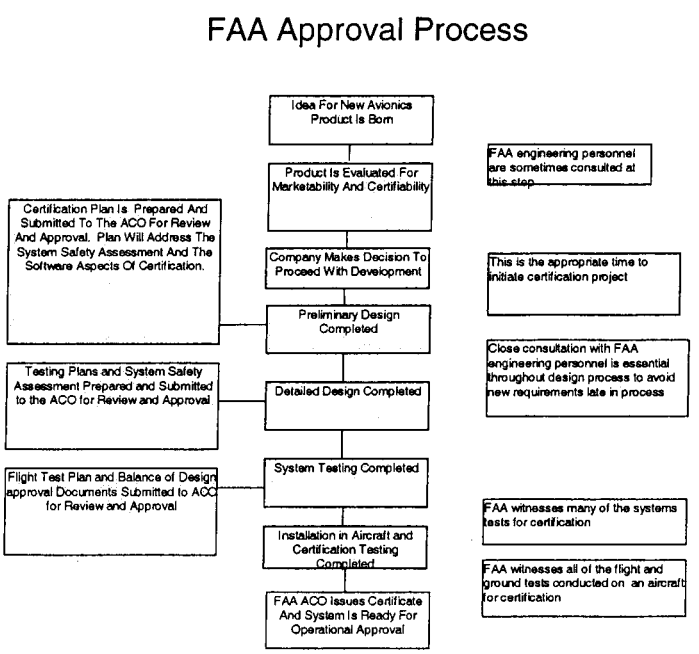


Figure 3: FAA Approval Process

A successful presentation to the FAA should include the following artifacts. The items highlighted in bold type are required for DO-178B and explained in the previous section. Other documents are standard life cycle material; however, the FAA requires that these artifacts be updated regularly so they contain the most recent information.

1. **Plan for Software Aspects of Certification (PSAC)**
2. Software Development Plan (SDP)
3. Software Verification Plan (SVP)
4. Software Configuration Management Plan (SCMP)
5. Software Quality Assurance Plan (SQAP)
6. Software Requirements Standards (SRS)
7. Software Design Standards (SDS)
8. Software Code Standards
9. Software Requirements Data
10. Design Description (SDD)
11. Source Code
12. Executable Object Code
13. Software Verification Cases and Procedures
14. Software Verification Results
15. Software Life Cycle Environment Configuration Index
16. **Software Configuration Index (SCI)**
17. Problem Reports
18. Software Configuration Management Records
19. Software Quality Assurance Records
20. **Software Accomplishment Summary (SAS)**

In order to obtain certification by the FAA, the applicant must prove that objectives have been met. For Level A there are 66 objectives, for Level B there are 65 objectives and for Level C there are 62 objectives.

7. DRYDEN FLIGHT RESEARCH CENTER CERTIFICATION PROCESS

The Dryden certification process is documented in:

- Dryden Handbook Code X - Airworthiness and Flight Safety Review, Independent Review, Mission Success Review, Technical Brief and Mini-Tech Brief Guidelines DHB-X-001 Revision D
- Dryden Flight Research Center Policy: Airworthiness and Flight Safety Review Process, DCP-X-009 Revision B
- Dryden Flight Research Center Policy: Flight Operational Readiness Review (ORR) and Operational Readiness Review Panel (ORRP), DCP-X-020 Revision A

These documents contain specific details about whom and how the process shall be implemented. Figure 2 below provides an overview of the process.

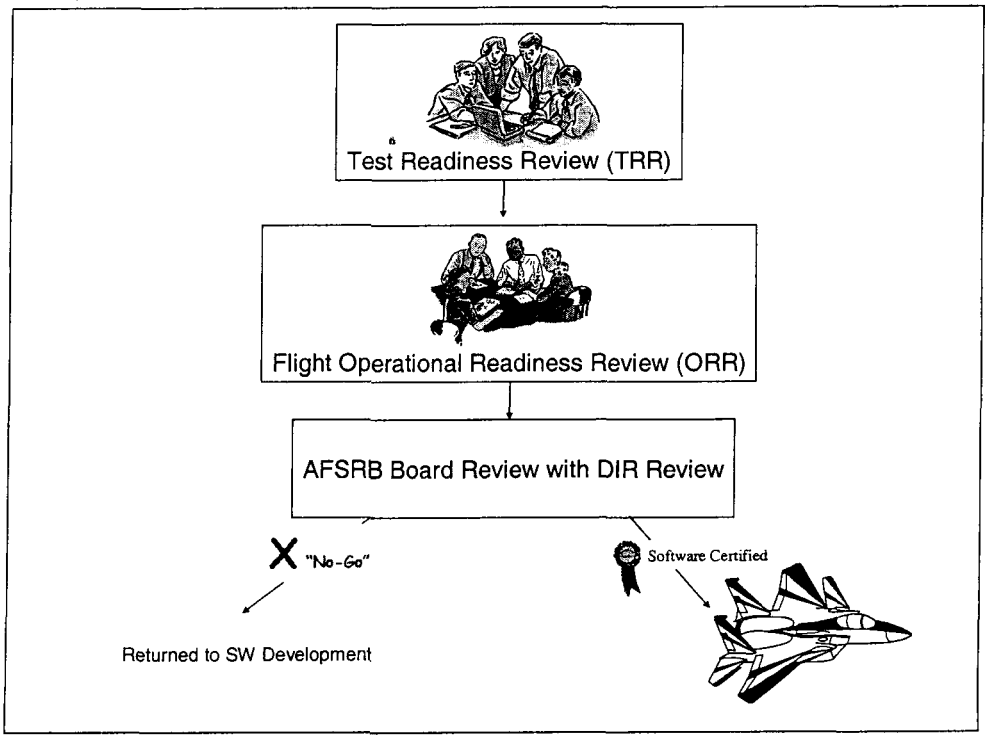


Figure 4: Overview of DFRC Certification Process for Class A Software

When software is ready for certification it is reviewed at the Test Readiness Review (TRR) by the internal project team. Once the software passes this internal review, it is reviewed by an independent team of engineers who have not worked on the project called the Operational Readiness Review Panel (ORRP).

The ORRP conducts a Flight Operational Readiness Review (ORR). When the software passes the ORR, the ORRP notifies the DFRC Chief Engineer.[12] For a detailed flow chart of the ORR process, see Appendix C.

Then, the Project or Mission Manager presents project plans and preparations to the Chair of the AFSRB, Air-worthiness Flight Safety Review Board. The Chair of the AFSRB is appointed by the DFRC Center Director. AFSRB board members include the line organizational Directors, ex Officio members, the Chief Pilot and the chief of the Safety Office. Other U.S. Government personnel may be appointed as necessary to provide a thorough review.

The Chair of the AFSRB may take one of four possible actions:

- After careful review, should the Chair deem the software flight worthy, he or she may certify it without further review by the Board
- The Chair may convene a small group of Dryden experts, independent to the project, to assist him/her in determining whether the proposed project is cleared for flight
- The Chair may request that plans and proposed conduct of the project be presented to the entire AFSR for its review. In this case, the Board shall pass judgment on whether a particular project has adequately considered and integrated flight safety into its proposed plans.
- The Chair may request that plans and proposed conduct of the project be presented to the ASFRB by the DFRC Independent Review (DIR). The Chair may establish a formal DIR based on the following criteria:
    - New program or operation with significant risk to personnel or property (Class A)
    - Phased program ready to enter a succeeding phase beyond that already approved
    - Program preparing to exceed some limit previously approved
    - Program requiring major modification of aircraft

After careful review and consideration, the AFSRB makes a "go" or "no-go" decision. If the software receives a "go" then it is certified and loaded onto the aircraft. If the software is lacking in some regard, and receives a "no-go" decision, then it returns to development for further work and the certification process starts over.[13]

For detailed flowchart of the AFSRB decision process, see Appendix D.

8. JET PROPULSION LAB APPROVAL PROCESS

The Jet Propulsion Lab is chartered to develop mission critical systems and software. At this time, JPL does not develop safety-critical software.

While JPL does not certify software per se, processes are in place to evaluate and approve systems and software based on four levels:

- Level A – applies to systems and software where failure could result in "loss of mission" defined as the inability to meet mission objectives
- Level B – applies to systems and software supporting science data processing for example flight software of a secondary nature that is isolated where failure produces no side effects to Level A systems
- Levels C and D – applies to other types of systems and software where failures are further isolated and do not produce side effects to Level A or B systems

Before any system can be implemented on a space craft, rover or other vehicle, it must be approved. Each mission team at JPL is responsible for developing a plan to ensure success of mission-critical systems. Teams adhere to NASA standards developed by the Software Working Group (http://swg.nasa.gov) and three documents specifically addressing system and software development at JPL including:

- Software Development Requirements – guide to development of Level A software including recommended reviews, stress testing, independent quality assessment, et al
- Set of Handbooks based on CMMI containing guidelines for costing, requirements etc
- Set of Design Principles describing recommended design methods and techniques for flight software

Generally, a stringent review process is followed to evaluate systems and software pending approval. A sample of this process is shown in the following diagram and described below:

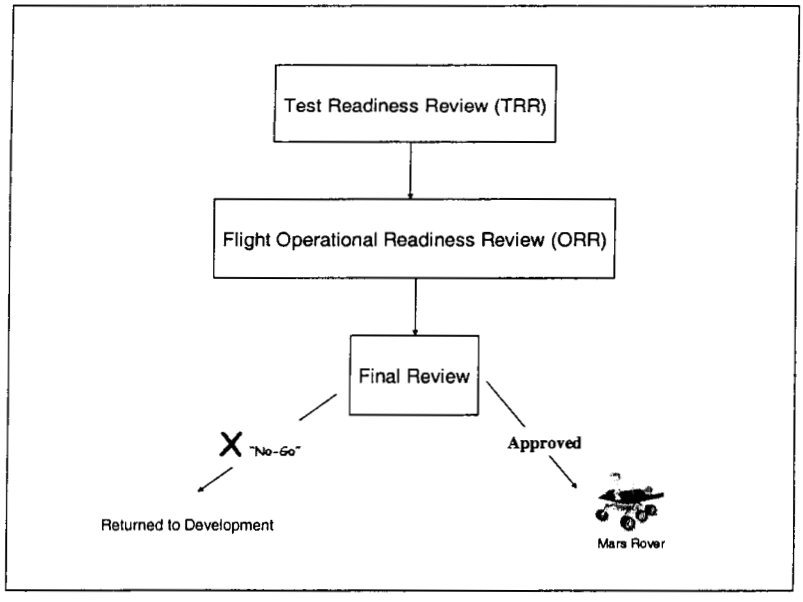


Figure 5: Overview of JPL Approval Process

When systems and software are ready for approval they are reviewed at the Test Readiness Review (TRR) by the internal project team. Once the software passes this internal review, it is reviewed by an independent team of engineers who have not worked on the project. The independent team conducts a Flight Operational Readiness Review (ORR). When the system and software pass the ORR, the Program Manager is notified and submits the project plans and preparations to the Final Reviewer(s). The Final Reviewer(s) determines whether the software is approved for implementation or must return to software development for further work.

9. APPENDIX A: ACRONYMS

Term	Definition
ANSI	American National Standards Institute
ARC	Ames Research Center
CM	Configuration Management
DFRC	Dryden Flight Research Center
FAA	Federal Aviation Administration
EIA	Electronic Industries Association
IEC	International Electro-technical Commission
IEEE	Institute of Electrical and Electronic Engineers
ISO	International Organization for Standardization
IV&V	(NASA) Independent Verification & Validation
JAA	Joint Aviation Authorities
JPL	Jet Propulsion Lab
MIL STD	Military Standard
NASA	National Aeronautical Space Administration
NPD	NASA Policy Directive
NPG	NASA Procedures and Guidelines
RTCA	Requirements and Technical Concepts for Aviation
USA	United Space Alliance
V&V	Verification & Validation

Note: *More Acronyms: http://www.ksc.nasa.gov/facts/acronyms.html*

10. APPENDIX B: GLOSSARY

Black Box testing: Requirements-driven testing where engineers select system input and observe system output/reactions

Certification: process for demonstrating that system safety is satisfactory fro flight operation

CSCI: Computer Software Configuration Item (a term used in NASA or Military standards to describe a product like a jet engine or a computer system

Fidelity: Integrity of testbed. For example: low fidelity testbed may have a simulator rather than actual spacecraft hardware. The highest fidelity testbed is the actual hardware being tested

Mission critical: loss of capability leading to possible reduction in mission effectiveness but cannot cause a risk to human life

Modified Condition And Decision Coverage (MCDC): defined as checking that *"every point of entry and exit in the program has been invoked at least once, every condition is a decision in the program has taken all possible outcomes at least once, every decision has been shown to independently affect that decision's outcome. A condition is shown to independently affect a decision's outcome by varying just that condition while holding fixed all other possible conditions."*[1]

Nominal: Expected behavior for no failure, for example: nominal behavior for a valve may be "open" or "shut"

Off-Nominal: Unexpected failure behavior, for example: off-nominal behavior for a valve may be "stuck open" or "stuck shut"

Safety-critical: failure or design error could cause a risk to human life

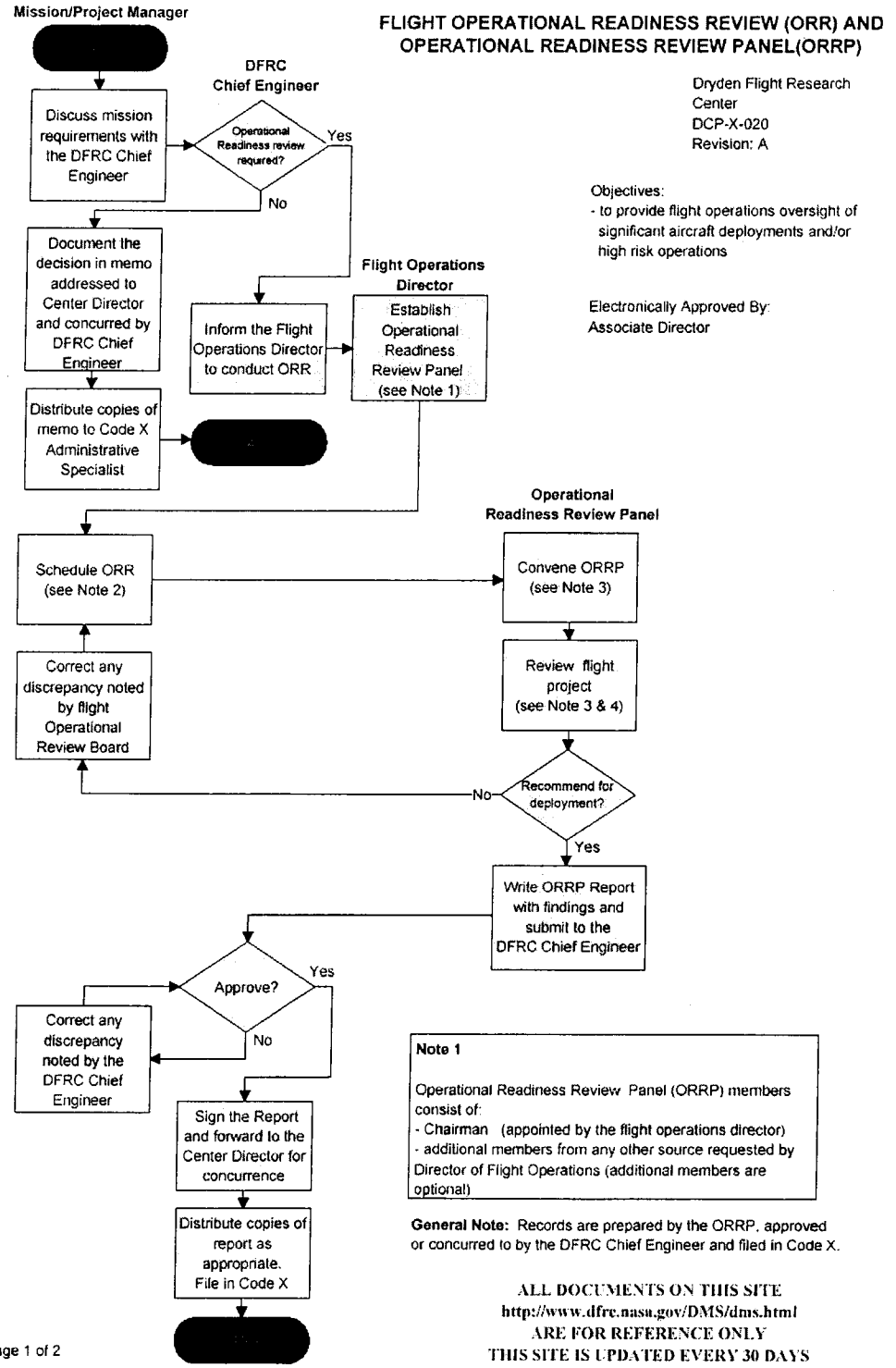
Validation: process of determining that the requirements are correct and complete

Verification: evaluation of results of a process to ensure correctness and consistency with respect to the input and standards provided to that process

White Box Testing: Design-driven testing where engineers examine internal workings of code

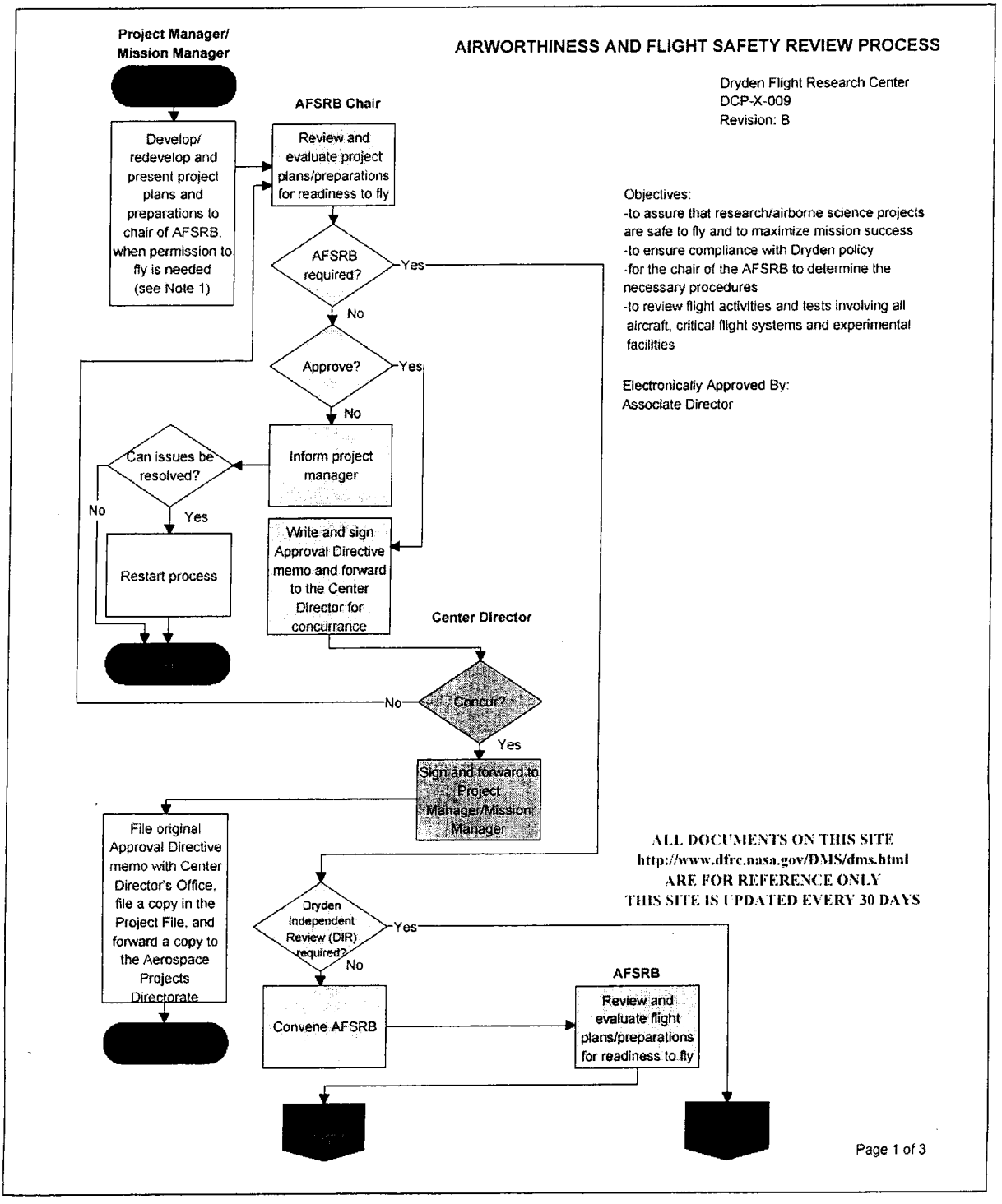
11.APPENDIX C: DFRC ORR and ORRP

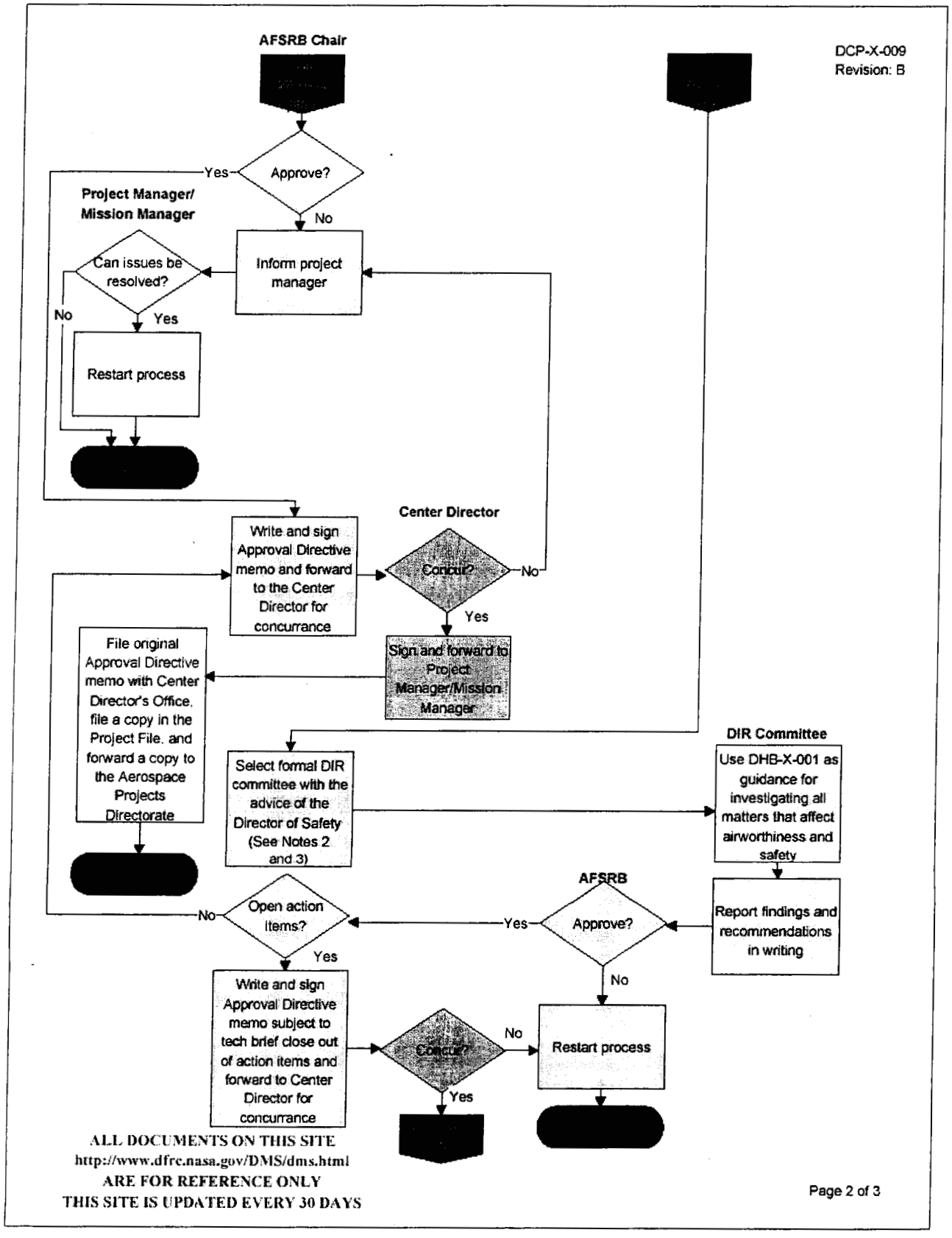
This section was copied from FLIGHT OPERATIONAL READINESSREVIEW (ORR) AND OPERATIONAL READINESS REVIEW PANEL (ORRP) Dryden Flight Research Center, DCP-X-020, Revision: A. For current revisions see http://www.dfrc.nasa.gov/DMS/dms.html

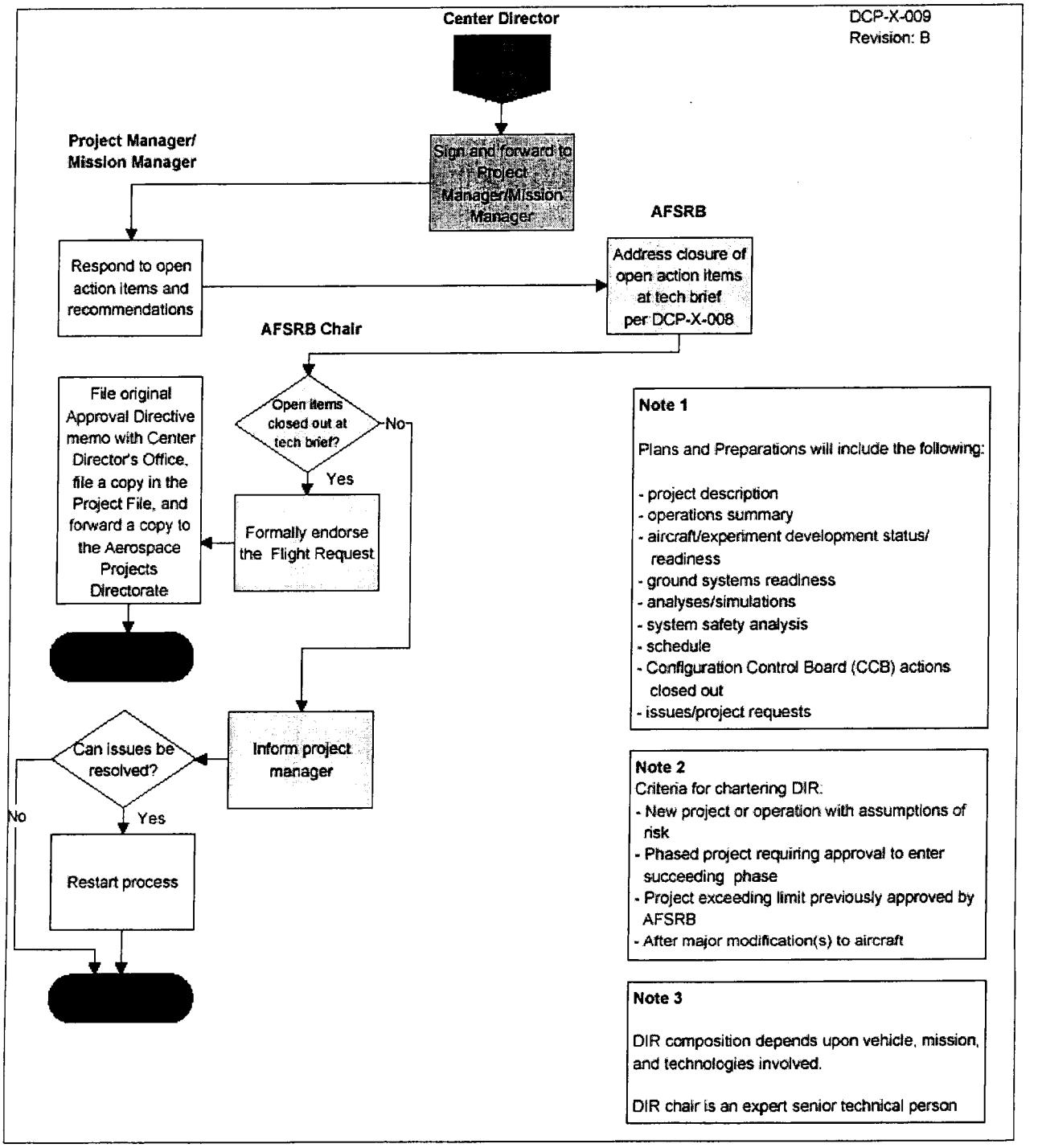


12. APPENDIX D: DFRC AIRWORTHINESS AND FLIGHT SAFETY REVIEW PROCESS

This section was copied from AIRWORTHINESS AND FLIGHT SAFETY REVIEW PROCESS Dryden Flight Research Center, DCP-X-009, Revision: B. For current revisions see http://www.dfrc.nasa.gov/DMS/dms.html

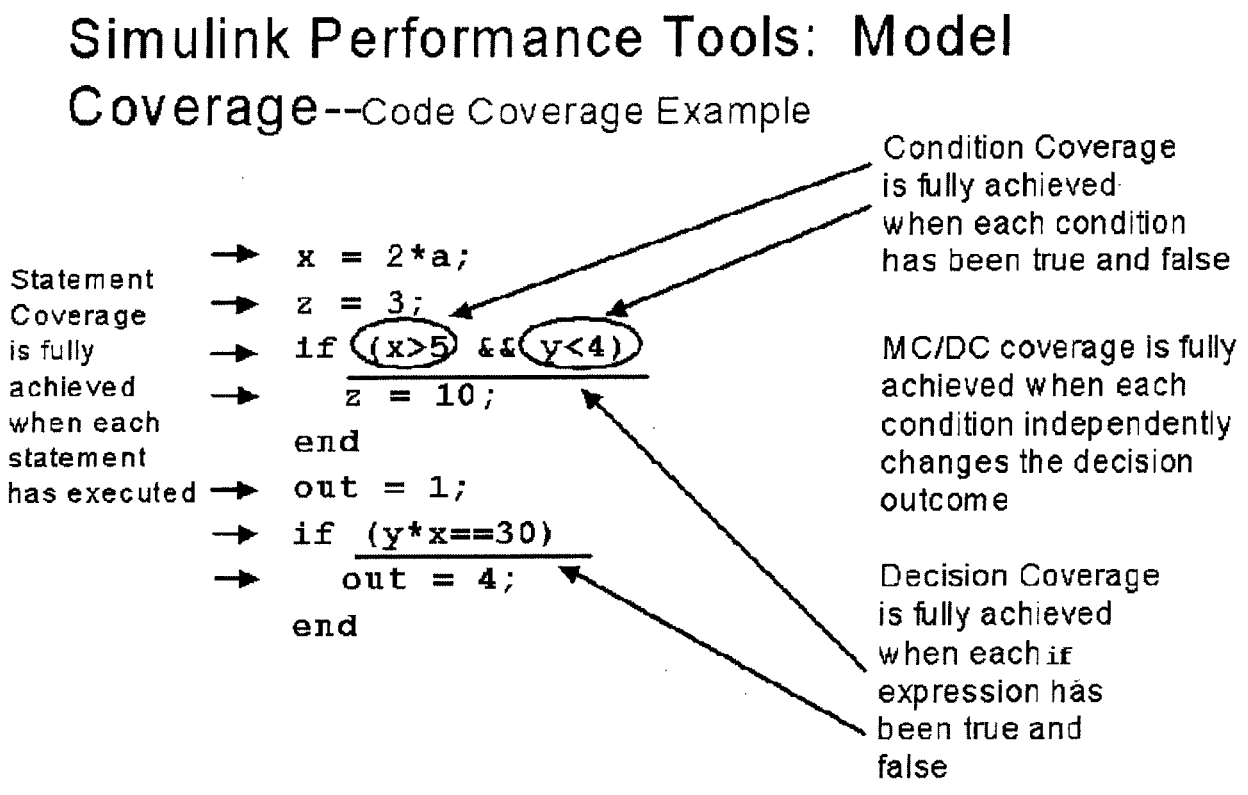



AFSRB Chair
DCP-X-009
Revision: B
Approve?
Yes
No
Project Manager/
Mission Manager
Can issues be resolved?
Inform project manager
No
Yes
Restart process
Center Director
Write and sign Approval Directive memo and forward to the Center Director for concurrance
Concur?
No
Yes
Sign and forward to Project Manager/Mission Manager
File original Approval Directive memo with Center Director's Office, file a copy in the Project File, and forward a copy to the Aerospace Projects Directorate
DIR Committee
Select formal DIR committee with the advice of the Director of Safety (See Notes 2 and 3)
Use DHB-X-001 as guidance for investigating all matters that affect airworthiness and safety
AFSRB
Open action items?
Approve?
Report findings and recommendations in writing
No
Yes
Yes
No
Write and sign Approval Directive memo subject to tech brief close out of action items and forward to Center Director for concurrance
Concur?
No
Yes
Restart process
ALL DOCUMENTS ON THIS SITE
http://www.dfrc.nasa.gov/DMS/dms.html
ARE FOR REFERENCE ONLY
THIS SITE IS UPDATED EVERY 30 DAYS
Page 2 of 3


Center Director
DCP-X-009
Revision: B
Project Manager/
Mission Manager
Sign and forward to Project Manager/Mission Manager
AFSRB
Respond to open action items and recommendations
Address closure of open action items at tech brief per DCP-X-008
AFSRB Chair
Open items closed out at tech brief?
No
Yes
File original Approval Directive memo with Center Director's Office, file a copy in the Project File, and forward a copy to the Aerospace Projects Directorate
Formally endorse the Flight Request
Inform project manager
Can issues be resolved?
No
Yes
Restart process
Note 1
Plans and Preparations will include the following:
- project description
- operations summary
- aircraft/experiment development status/ readiness
- ground systems readiness
- analyses/simulations
- system safety analysis
- schedule
- Configuration Control Board (CCB) actions closed out
- issues/project requests
Note 2
Criteria for chartering DIR:
- New project or operation with assumptions of risk
- Phased project requiring approval to enter succeeding phase
- Project exceeding limit previously approved by AFSRB
- After major modification(s) to aircraft
Note 3
DIR composition depends upon vehicle, mission, and technologies involved.
DIR chair is an expert senior technical person

13. APPENDIX E: SIMULINK MCDC EXAMPLE

The following example was copied from *Documenting, Testing and Verifying Your Designs in Simulink* by Valerie Lyons, The MathWorks, Inc.[14]



14. REFERENCES

[1] *Software Considerations in Airborne Systems and Equipment Certification,* Document No RTCA (Requirements and Technical Concepts for Aviation) /DO-178B, December 1, 1992. (Copies of this document may be obtained from RTCA, Inc., 1140 Connecticut Avenue, Northwest, Suite 1020, Washington, DC 20036-4001 USA. Phone: (202) 833-9339)

[2] Interview with Dale Mackall, Sr. Dryden Flight Research Center Verification and Validation engineer on January 16, 2003

[3] Interview with Dale Mackall, Sr. Dryden Flight Research Center Verification and Validation engineer on January 16, 2003

[4] Neil Storey, *Safety-Critical Computer Systems* Addison-Wesley Longman, 1996

[5] NASA/CR-2002- Verification & Validation of Neural Networks for Aerospace Applications, Dale Mackall, Johann Schumann and Stacy Nelson

[6] Introduction to IEEE/EIA 12207 presentation by Jim Wells - Software Engineering Process Office (SEPO - D12), Software Process Improvement Working Group (SPIWG), October 13, 1999

[7] *NASA Procedures and Guidelines NPG: 2820.DRAFT, NASA Software Guidelines and Requirements as of 3/19/01* (Responsible Office: Code AE/Office of the Chief Engineer), NASA Ames Research Center, Moffett Field, California, USA

[8] IEEE Standards 12207.0, 12207.1, 12207.2 located at the following web address (URL): http://ieeexplore.ieee.org/search97/s97is.vts?Action=FilterSearch&SearchPage=VSearch.htm&ResultTemplate=adv_crst.hts&Filter=adv_sch.hts&ViewTemplate=lpdocview.hts&query1=12207&scope1=&op1=and&query2=&scope2=&op2=and&query3=&scope3=&collection=jour&collection=conf&collection=stds&collection=pprint&py1=&py2=&SortField=pyr&SortOrder=desc&ResultCount=15

[9] RTCA DO-178B – Overview of Aircraft and Engine Certification p. 45

[10] Email dated 2/21/03 from Dr. Vdot Santhanam, Boeing

[11] Ty Startzman, Boeing Software and Languages Technology presentation titled *Certification - Its role in the Software Development Lifecycle, 2003.*

[12] Dryden Flight Research Center Policy: Flight Operational Readiness Review (ORR) and Operational Readiness Review Panel (ORRP), DCP-X-020 Revision A

[13] Dryden Handbook Code X - Airworthiness and Flight Safety Review, Independent Review, Mission Success Review, Technical Brief and Mini-Tech Brief Guidelines DHB-X-001 Revision D

[14] Valerie Lyons, *Documenting, Testing and Verifying Your Designs in Simulink*, The MathWorks, Inc located at: http://www.mathworks.com/company/events/archived_webinars.shtml