**The Role and Structure of the Software Verification Plan (SVP) in DO-178C Compliance**

## 1. Introduction

The certification of airborne software requires not only high-quality development practices but also rigorous verification procedures that ensure safety, reliability, and compliance with regulatory objectives. Within the framework of DO-178C—formally known as *Software Considerations in Airborne Systems and Equipment Certification*—verification is one of the most emphasized aspects of the software lifecycle. The **Software Verification Plan (SVP)** plays a central role in documenting and guiding these activities. As one of the five core planning documents required by DO-178C, the SVP outlines how verification will be conducted, who is responsible for it, what tools and environments will be used, and how compliance with DO-178C's verification objectives will be demonstrated.

The SVP is more than a checklist or a compliance formality—it is a strategic engineering plan that directly supports the integrity of the software product and the credibility of the certification effort. Without a well-defined and consistently executed SVP, it is nearly impossible to demonstrate that the software meets its requirements, behaves reliably under all conditions, and is free from unintended functionality. This paper explores the structure, purpose, and practical execution of the SVP, providing a comprehensive view for developers, certification engineers, and project managers working under DO-178C.

## 2. Role of the SVP in DO-178C

DO-178C is structured around a series of objectives tailored to different Design Assurance Levels (DALs), ranging from **DAL A** (catastrophic) to **DAL E** (no safety impact). The SVP supports the satisfaction of many of these objectives, particularly those found in Tables A-2 through A-7, which define what must be verified and to what degree.

The SVP provides a **systematic approach** to planning verification activities that include reviews, analyses, and testing. It ensures that verification is **independent** where required, that all software requirements are tested, and that structural coverage meets the thresholds defined for each DAL (e.g., 100% MC/DC for DAL A).

Moreover, the SVP ensures traceability between the verification activities and the artifacts under test—requirements, design, and code—thus supporting traceability objectives found throughout the DO-178C standard.

## 3. Structure of the SVP

A well-structured SVP typically includes the following sections:

1. **Introduction and Scope**

2. **Applicable Documents and Standards**
3. **Verification Lifecycle Model**
4. **Verification Objectives and Activities**
5. **Verification Environment and Tools**
6. **Test Case Development and Execution**
7. **Structural Coverage Analysis**
8. **Verification Roles and Responsibilities**
9. **Traceability and Coverage Strategy**
10. **Configuration Control for Verification Artifacts**
11. **Milestones, Reviews, and Schedule**
12. **Integration with Other Certification Plans**

Each section defines a part of the overall verification process, offering a complete view of how verification aligns with project scope, regulatory needs, and internal quality expectations.

**4. Verification Processes Defined**

The SVP must describe how **verification processes** will be conducted throughout the software lifecycle. These include:

- **Reviews** of requirements, design, and source code
- **Static and dynamic analyses**
- **Test planning and execution**
- **Structural coverage analysis**

Each process must define its **entry criteria**, **exit criteria**, and **success metrics**. For example, requirements reviews may require that all high-level requirements be reviewed by independent engineers and approved before test development begins.

The SVP should also define how anomalies discovered during verification are reported, tracked, and resolved. This ties into configuration management and quality assurance activities, typically referenced in the SCMP and SQAP respectively.

**5. Types of Verification Activities**

DO-178C distinguishes among different **verification techniques**, and the SVP must detail how each is applied:

**5.1 Reviews and Analyses**

- **High-Level Requirements (HLR)** must be reviewed for correctness, consistency, and verifiability.
- **Low-Level Requirements (LLR)** must be shown to be consistent with HLR and implementable.
- **Source Code** must be reviewed against coding standards, design consistency, and proper use of data/control structures.

**5.2 Requirements-Based Testing**

Every software requirement—especially safety-critical ones—must be tested using clearly defined input/output conditions. The SVP describes how test cases are derived, how test coverage is measured, and how test results are documented.

**5.3 Structural Coverage Testing**

The SVP must define how code coverage is measured and what level is required:

- **Statement Coverage**: All DALs
- **Decision/Condition Coverage**: DAL B and above
- **Modified Condition/Decision Coverage (MC/DC)**: DAL A

Each coverage type must be justified and demonstrated with tooling.

**6. Verification Environment**

The SVP defines the tools and environments used in verification. This includes:

- **Test environments** (e.g., host-based, target hardware)
- **Simulators and emulators**
- **Structural coverage tools** (e.g., LDRA, VectorCAST, GCov)
- **Requirements and traceability tools** (e.g., IBM DOORS, Polarion, Jama)

If any of these tools **automate or replace a verification objective**, they may require qualification under **DO-330**, the companion document to DO-178C that covers tool qualification.

The SVP must state which tools are **development tools** and which are **verification tools**, as this distinction affects qualification requirements.

**7. Traceability and Coverage**

Traceability is a central objective of DO-178C. The SVP must describe how traceability is maintained:

- From **requirements to test cases**
- From **code to requirements and tests**
- From **test cases to test procedures and results**

This traceability is often enforced through trace matrices or traceability tools. The SVP must also describe how structural coverage data is mapped back to the code under test, and how any gaps are resolved.

**8. Verification Team Responsibilities**

The SVP should identify all personnel involved in verification, along with their responsibilities. This includes:

- **Verification engineers** who design and run tests
- **Reviewers** for requirements, design, and code
- **Test leads and coordinators**
- **Tool qualification engineers** (if applicable)

A crucial point emphasized in DO-178C is **verification independence**—especially for DAL A and B. The SVP must show how verification is independent from the development team in terms of reporting structure, review authority, and approval roles.

**9. Integration with Other Plans**

The SVP is tightly coupled with:

- The **Software Development Plan (SDP)**: Describes what will be developed and when.

- The **Software Configuration Management Plan (SCMP)**: Ensures test artifacts are controlled.
- The **Software Quality Assurance Plan (SQAP)**: Defines how verification activities are audited and monitored.

The SVP should reference these plans rather than duplicate content. It must also describe how verification results will be captured in the **Software Verification Results (SVR)** document, which is a required DO-178C output reviewed by certification authorities.

**10. Common Pitfalls and Best Practices**

**10.1 Pitfalls**

- **Late development of the SVP**, resulting in misaligned testing.
- **Insufficient independence**, especially in DAL A/B programs.
- **Tool use without qualification**, risking certification rework.
- **Poor traceability**, making objective evidence hard to demonstrate.

**10.2 Best Practices**

- Begin SVP planning **in parallel** with the SDP.
- Use **version-controlled test cases and results**.
- Automate traceability and coverage tracking where possible.
- Conduct **internal reviews of the SVP** prior to submitting for DER or EASA review.

**11. Certification Authority Expectations**

Certification authorities such as the FAA or EASA will review the SVP (either directly or indirectly via the PSAC and audits). They expect:

- A clear, DAL-appropriate plan
- Proper test coverage definition and justification
- Evidence of independence
- Tool qualification references if applicable
- Logical flow and traceability between the SVP and SVR

During audits, the SVP may be examined alongside actual test results to ensure what was planned was actually executed.

## 12. Conclusion

The **Software Verification Plan (SVP)** is one of the most critical documents in a DO-178C-compliant project. It defines how the software will be verified, how verification aligns with development and certification objectives, and how the integrity of the verification process will be preserved.

A well-written and properly executed SVP ensures that airborne software is tested thoroughly, objectively, and with traceable, auditable results. It provides confidence to the certification authorities and strengthens the engineering discipline of the development team. As systems grow in complexity and safety expectations increase, the SVP will remain a cornerstone of aviation software compliance, protecting both safety and certification integrity.

## DO-178C Software Verification Plan (SVP) Checklist

### 1. General Information

- SVP title, document ID, revision number, and date
- Project/system name and software context
- Stated Design Assurance Level (DAL)
- SVP scope and applicability clearly defined
- References to the PSAC and other certification plans (SDP, SCMP, SQAP)

### 2. Referenced Documents

- DO-178C and applicable supplements (e.g., DO-330)
- Internal development and verification standards
- Software requirements, design, and coding standards
- Verification tool manuals and test environments

### 3. Verification Approach

- Overview of software lifecycle phases and where verification applies
- Description of the verification lifecycle model (V-model, etc.)
- Definition of each verification activity:

    - Reviews
    - Analyses
    - Testing

**4. Verification Objectives**

- Mapping of verification objectives to DO-178C Table A-x items
- DAL-specific objectives clearly identified
- Strategy for satisfying each objective defined

**5. Verification Tools and Environment**

- List of all tools used for verification (e.g., code coverage, test harnesses)
- Indication of any tools requiring qualification under DO-330
- Description of test environment(s) – host-based, target, HIL, etc.
- Tool version control and configuration process defined

**6. Verification Artifacts**

- List of verification inputs and outputs (e.g., test cases, procedures, logs)
- Method for review and approval of artifacts defined
- Storage and retention policy for verification records

**7. Test Planning**

- Methodology for requirements-based testing defined
- Procedure for test case development and validation
- Definition of expected outputs and pass/fail criteria
- Use of automated vs. manual testing documented

**8. Structural Coverage**

- Required coverage level per DAL stated:
    - Statement Coverage
    - Decision/Condition Coverage
    - MC/DC (if DAL A)

- Coverage analysis tool identified
- Description of gap analysis and rework strategy

**9. Traceability**

- Approach to requirement-to-test traceability documented
- Code-to-requirement-to-test linkage strategy explained
- Structural coverage linked to code segments
- Use of traceability tools or matrices described

**10. Roles and Responsibilities**

- Identification of verification team members and roles
- Clear independence of verification from development (DAL A/B)
- Description of peer reviews and independent audits

**11. Verification Milestones**

- Key verification activities aligned with development schedule
- Entry/exit criteria for each activity defined
- Review and approval gates identified (e.g., TRR, CDR, SVR reviews)

**12. Configuration Management of Verification Data**

- Controlled storage of test cases, results, coverage reports
- Integration with SCMP-defined tools and procedures
- Revision tracking for test artifacts

**13. Integration with Other Plans**

- Clear reference to:
    - Software Development Plan (SDP)
    - Software Configuration Management Plan (SCMP)
    - Software Quality Assurance Plan (SQAP)
- Description of how verification results will be captured in the SVR

## 14. Certification Considerations

- SVP supports demonstration of compliance with applicable DO-178C objectives
- References to planned DER/FAA/EASA reviews (if applicable)
- Evidence strategy for audits and compliance findings