

The Role and Structure of the Software Development Plan (SDP) in DO-178C Compliance

1. Introduction

DO-178C, *Software Considerations in Airborne Systems and Equipment Certification*, is the cornerstone guidance document for software development in civil aviation. Issued by RTCA (U.S.) and EUROCAE (Europe), it outlines objectives that must be met to demonstrate software reliability, safety, and traceability to regulatory authorities such as the FAA or EASA. Among the suite of planning and process documents required under DO-178C, the **Software Development Plan (SDP)** is one of the most foundational. It provides a framework for how software development will be conducted within the constraints of DO-178C, ensuring that each phase of the software lifecycle aligns with applicable design assurance levels (DALs) and that development activities are consistent, auditable, and traceable.

2. Purpose of the SDP

The SDP serves as a **roadmap for software engineering activities** within a DO-178C-compliant project. It identifies development methods, standards, tools, responsibilities, and interfaces, ensuring that the development effort is systematic and aligned with regulatory expectations. For certification purposes, the SDP becomes part of the documentation set reviewed by certification authorities and is referenced in audits throughout the software lifecycle.

The SDP directly supports **DO-178C objectives 1-4** for all DALs, and is specifically called out in Table A-1 of DO-178C. It must be created early, preferably before software development begins in earnest, and maintained as a **living document** throughout the project.

3. Design Assurance Levels and the SDP

The complexity and rigor of an SDP depends on the **Design Assurance Level (DAL)** assigned to the software. These levels range from **DAL A** (catastrophic impact if failed) to **DAL E** (no effect on safety). For DAL A and B projects, the SDP must provide more granular detail, define tighter control processes, and explicitly document the rationale behind every methodology, tool, and standard used. For DAL C and D, the SDP can be less exhaustive but still must show conformance to applicable objectives.

4. Software Lifecycle Processes

A key function of the SDP is to define the **software development lifecycle** and the phases that will be used, including:

- **Requirements Engineering**
- **High-Level Design**
- **Low-Level Design**
- **Source Code Development**

- **Integration**
- **Verification**
- **Configuration Management**
- **Quality Assurance**
- **Problem Reporting and Correction**

Each phase must be accompanied by:

- **Entry and exit criteria**
- **Responsible roles**
- **Standards and conventions used**
- **Verification methods**

The SDP must define whether a **traditional waterfall model, V-model, or custom lifecycle model** (e.g., incremental or hybrid agile) is used. While DO-178C allows flexibility, the SDP must **justify** the chosen lifecycle in terms of how it supports traceability and completeness.

5. Integration with Other Planning Documents

DO-178C recommends five core plans for software certification, of which the SDP is one. The other four are:

- **Plan for Software Aspects of Certification (PSAC)**
- **Software Verification Plan (SVP)**
- **Software Configuration Management Plan (SCMP)**
- **Software Quality Assurance Plan (SQAP)**

The SDP is **referenced in and supports the PSAC**, but does not duplicate its contents. Rather, it expands on specific development activities (whereas the PSAC is a high-level agreement with certification authorities).

The SDP must also be **cross-referenced** with the SVP to ensure that:

- Requirements, design, and code are verifiable
- Test cases and analysis align with development outputs

It should also align with the SCMP and SQAP, clearly defining how changes, audits, and software quality will be managed within the context of development.

6. Standards and Methods

The SDP documents the **development standards** used in the project. This includes:

- **Requirements standards:** Natural language, formal, or semi-formal notation (e.g., DOORS templates)
- **Design standards:** Use of UML, SysML, or structured methods

- **Coding standards:** Commonly MISRA C, JSF AV C++, or in-house rules
- **Naming conventions and commenting rules**
- **Error handling mechanisms**
- **Modularity and reusability rules**

Each standard must support consistency and facilitate traceability, two central tenets of DO-178C.

7. Tool Usage and Qualification

The SDP must identify:

- All **software development tools** used (e.g., IDEs, compilers, static analyzers)
- Any tools that **automate or replace manual DO-178C objectives**
- Whether a tool must be **qualified** under DO-330 (Software Tool Qualification Considerations)

For instance, if a tool generates source code from models or performs structural coverage analysis, it likely needs qualification. The SDP should describe the tool chain, rationale for tool use, and summarize any qualification activities (with details deferred to tool-specific documentation).

8. Traceability and Configuration Management

Traceability is a central focus of DO-178C. The SDP must describe:

- How traceability is maintained from **high-level requirements** → **low-level requirements** → **code** → **tests**
- How change impacts are tracked
- What tools or methods are used to manage traceability (e.g., trace matrices, automated requirements tools)

It must also define how configuration items are created, managed, and version-controlled. While the SCMP covers this in detail, the SDP should state how software artifacts (requirements, design, code) are managed within development activities.

9. Roles and Responsibilities

The SDP outlines the project's organizational structure with a focus on:

- **Software Engineers:** who design and code
- **Verification Engineers:** who test and analyze
- **Configuration Managers**
- **Software Quality Assurance (SQA)**
- **Designated Engineering Representatives (DERs) or Certification Authorities**

It should state that verification is **independent** of development and explain how peer reviews, inspections, or audits are structured to ensure independence (required for DAL A/B).

10. Change Management

The plan must describe how **process deviations** and **software changes** are handled. This includes:

- How change requests are submitted and reviewed
- How regression testing is triggered
- How traceability and baselines are maintained after changes

This helps satisfy DO-178C objectives related to **problem reporting, change control, and configuration integrity**.

11. Lifecycle Data and SDP Maintenance

The SDP must define:

- What **lifecycle data** will be generated (e.g., source code, test cases, review records)
- How that data is **reviewed, approved, and archived**
- Whether the SDP itself is a **controlled document**, and how updates to the plan are tracked

In most cases, the SDP is **baselined** at a milestone like Preliminary Software Review (PSR) or Software Requirements Review (SRR), and controlled under the SCMP thereafter.

12. Certification Authority Expectations

When reviewed by certification authorities, the SDP is evaluated for:

- Consistency with the PSAC and other plans
- Adequacy of development and verification activities
- Clear linkage to DO-178C objectives
- Reasonable clarity and completeness of process definition

An incomplete or vague SDP can trigger Requests for Action (RFAs) from FAA Designated Engineering Representatives (DERs) or EASA software specialists. Therefore, organizations typically adopt **template-based** or **organization-wide SDP frameworks** that are adapted per project.

13. Best Practices and Common Pitfalls

Best Practices:

- Begin the SDP early and update it continuously.

- Align with a standard company template approved on prior projects.
- Ensure tight cross-referencing with SVP, SCMP, and SQAP.
- Have the SDP peer-reviewed before presenting it in certification reviews.

Common Pitfalls:

- Copying SDP content from other DALs without tailoring.
- Omitting details on tool usage or justification of lifecycle model.
- Failing to maintain the SDP as a living document during changes.
- Treating it as a formality instead of a practical project guide.

14. Example Outline of a DO-178C SDP

1. Introduction and Scope
2. Reference Documents
3. Software Lifecycle Model
4. Development Environment and Tools
5. Development Standards and Methods
6. Software Development Activities
7. Traceability Approach
8. Integration with SVP, SCMP, SQAP
9. Roles and Responsibilities
10. Change and Problem Reporting
11. Lifecycle Data and Milestones
12. SDP Configuration Management
13. Glossary and Acronyms

15. Conclusion

The **Software Development Plan (SDP)** is not just a procedural document — it is a vital, certifiable component of any DO-178C-compliant software project. It defines how software is developed in a way that satisfies safety-critical objectives, supports verification, and enables regulatory authorities to have confidence in the process. A well-written SDP reflects a disciplined engineering organization and provides a strong foundation for certification success.

As aviation systems become more complex, and as more development tools and agile techniques are adopted in safety-critical domains, the importance of a precise, traceable, and living SDP will only increase. Ultimately, the SDP ensures that development is not only consistent with DO-178C but also aligned with the broader mission: **delivering safe and certifiable airborne software.**

DO-178C Software Development Plan (SDP) Checklist

1. General Information

- Document title, version, and date
- Project or program name and system overview
- Applicable Design Assurance Level (DAL)
- Scope and purpose clearly defined
- Reference to the Plan for Software Aspects of Certification (PSAC)

2. Referenced Documents

- All applicable DO-178C standards and supplements (e.g., DO-330)
- Software standards (requirements, design, code)
- Related plans: SVP, SCMP, SQAP
- Traceability matrix location(s)

3. Software Lifecycle Model

- Lifecycle model described (e.g., waterfall, V-model, incremental)
- Rationale for lifecycle model choice
- Mapping of lifecycle phases to DO-178C objectives
- Phases and review gates clearly defined

4. Software Development Activities

For each phase (requirements, design, code, integration):

- Objectives and tasks defined
- Entry and exit criteria specified
- Input and output artifacts listed
- Review and approval methods described
- Assigned roles and responsibilities

5. Standards and Conventions

- Requirements development standards

- Design methods and notation standards
- Coding standards (e.g., MISRA, in-house)
- Commenting, naming, and formatting rules
- Handling of derived requirements

6. Development Tools

- Development and support tools listed (IDEs, compilers, etc.)
- Description of tool usage per lifecycle phase
- Identification of any tools requiring DO-330 qualification
- Summary of tool qualification approach or reference to Tool Qualification Plan

7. Traceability

- Traceability approach explained (manual, automated, hybrid)
- Coverage from high-level requirements → low-level requirements → code → test
- Traceability tools identified
- Trace matrix structure/location referenced

8. Roles and Responsibilities

- Clear roles for:
 - Software engineers
 - Verification engineers
 - SQA personnel
 - Configuration managers
- Independence of verification (especially for DAL A/B)
- Organization chart or resource assignment summary

9. Configuration and Change Management

- Interface with Software Configuration Management Plan (SCMP)
- Definition of configuration items for each development phase
- How changes are proposed, tracked, and verified

- Regression testing strategy

10. Integration with Other Plans

- Verification activities consistent with the SVP
- Quality assurance actions referenced from SQAP
- Configuration activities aligned with SCMP
- References to the PSAC and any DER involvement

11. Lifecycle Data and Reviews

- List of software lifecycle data produced
- Identification of review types (peer, formal, audit)
- Review criteria and documentation requirements
- Status accounting and documentation strategy

12. SDP Control and Updates

- Identification of document owner or control authority
- Version control and update procedure
- SDP baseline milestone defined (e.g., PSR or SRR)
- Review and approval signatures or reference to approval records

13. Annexes or Appendices (Optional)

- Glossary of terms and acronyms
- Development schedule or milestone summary
- Deviations or tailoring from DO-178C (if any)

Certification Considerations

- Content is consistent with the PSAC submitted to certification authorities
- Plan supports satisfaction of DO-178C objectives A-1 through A-8 (as applicable)
- DER or certification authority review comments addressed (if applicable)