

| **PROJECT TITLE:** | **X-Wings X1 Head Up Display (HUD) System** |
| --- | --- |
| **DOCUMENT NAME:** | **HUD Plan for Software Aspects Specification** |
| **Document ID:** | **000001-00** |
| **Revision:** | **Rev 00** |
| **Classification:** | **Unclassified** |

Prepared by:

Aircraft Electronic Systems

Lorenzini Systems, Ltd.  
Porto Alegre, Rio Grande do Sul, Brazil

| Classification: Unclassified This is for learning only and not intended to be anyhow commercially used. |
| --- |

**This page contains the instructions on how to create an PSAC and shall be removed on the final version.**

• This template is recommended for documents to be archived in the Project Life Cycle software (PLM).

• When converting to PDF format, PLM automatically includes a cover page with document status and confirmation of approvals.

• Documents for PLM should start on page 2 (already considered on this template) since PLM PDF includes its own cover page which is considered page number 1. Therefore, the table of contents should be consistent.

• The footer included automatically on the PLM PDF format contains the document status, ID, revision and page number.

• To include Notes in the footer, use the Foot Note option (on References Tab).

• List of Changes table: It’s recommended to include also the first revision (-- for Release) of the document with the responsible for the preparation, since sometimes it’s different from the owner in the PLM.

• Program name information may be omitted when this software is intended to be used in different programs.

**This document is a template for the Version Description Document.**

• The text in blue color is an explanation and should be removed from the final document.

• The text written in *italic blue* color is a parameter, which should be replaced by the actual one in the final document.

• The text written in *italic green* color is an example whose purpose is to clarify the explanation, and should be used as recommendation only.

• The text written in black color is considered as recommended text and may be used in order to provide clear statements as per the subject of the said section.

List of Changes

| Section | Change Description | REV |
| --- | --- | --- |
|  |  |  |
|  |  |  |

List of TBDs

| TBD Number | Reference |
| --- | --- |
|  |  |
|  |  |

### 

[LIST OF FIGURES 5](#_bsnhfcyt1zzt)

[LIST OF TABLES 5](#_ahi6nx8vpeyi)

[**1. Objectives or Scope 6**](#_g58yaypvz8b5)

[1.1. Identification 7](#_itv5uz9ddvgq)

[1.2. Document Overview 7](#_268o9nj30oeg)

[1.3. List of Abbreviations 8](#_2f64ldwd8agf)

[**2. Referenced Documents 9**](#_axcbir5j4puc)

[2.1. Project Documents 9](#_zbuzd3idm6bq)

[2.2. Vendor Documents 10](#_7vokybracd3z)

[**3. System Overview 11**](#_ii2ieojns7pj)

[3.1. Functional Description 11](#_4jhdjgui3v83)

[**4 Certification Considerations 12**](#_75h8so73yav3)

[**5 Software Life Cycle 12**](#_en5xfqmflpxs)

[5.2.1 Software Planning Process Objectives 13](#_pnlnd4uy1rq4)

[5.2.2 Software Planning Process Input 13](#_z8wwnhhld7m0)

[5.2.3 Software Planning Process Output 13](#_cigrwoh72jah)

[5.2.4 Software Planning Process Activities 13](#_lj5alxozd8dz)

[5.2.5 Software Planning Process Transition Criteria 14](#_fd76noj6r8p8)

[5.2.5.1 Software Planning Process Entry Criteria 14](#_hosd9ntg5ebz)

[5.2.5.2 Software Planning Process Exit Criteria 14](#_2m9qkadlcx4y)

[5.2.6 Integral Processes 14](#_u0370domymlv)

[5.2.6.1 Software Configuration Management Activities 14](#_t3vv4bz6x1wm)

[5.2.6.2 Software Quality Assurance Activities 14](#_li1c2h6feehp)

[5.3.1 Software Requirements Process Objectives 14](#_hfsr09vryi2r)

[**6 Additional Considerations 21**](#_t6ykl3cf7mxd)

[6.1.1 General 21](#_p37lm5cv8in)

[**7 Software Configuration Management Process 24**](#_rz62ftfywt1z)

[**8 Software Quality Assurance Process 24**](#_3kwq0rx4gpgx)

[**9 Certification Liaison Process 25**](#_n1amb4si2ig6)

[**10 Software Life Cycle Data 25**](#_xq1mdu6sw9b0)

[**11 Organization and Responsibilities 27**](#_yrne3s54b97v)

[**12 Milestones and Schedule 28**](#_hx7nmfaaq2v6)

### 

### LIST OF FIGURES

### LIST OF TABLES

## Objectives or Scope

#### <You can use this PSAC template as a resource when creating a PSAC. If you are updating an existing PSAC to support Model-Based Design (MBD), you can use this template as a reference document. Although representative of PSACs used in the industry, this PSAC template has not been reviewed, approved, or accepted by any certification authority. It is the user’s responsibility to gain approval and acceptance of their PSAC by the appropriate certification authority.>

The purpose of this document is to provide the Plan for Software Aspects of Certification (PSAC) for the project *<project>*. This document provides the planning data defined in DO-178C, Section 11.1. The certification authority uses the PSAC for a project as the primary means to determine whether an applicant is proposing a software life cycle that is commensurate with the rigor required for the level of software being developed.

Therefore, this document contains an overview of the project *<project>*, including:

* Overview of system and its software.
* Statement of certification considerations.
* Discussion of the software development plan, including the software life cycle processes and corresponding data.
* Project milestones and schedule.

The purpose of the PSAC is to describe the software life cycle for the project *<project>*, its development process, and all integral processes.

This document provides the planning data defined in [DO-178C] Section 11.1 and [DO-331] Section MB.11.1, respectively. It summarizes or references the software development, verification, configuration management, and quality assurance activities for the project *<project>*.

This document establishes the Software Aspects of Certification (PSAC) plan for the application of the *Head Up Display (HUD)* System on BERNARDOs *Aircrafts X-Wing X1.* The *HUD* software's development will align closely with EUROCAE ED12B / RTCA DO-178B guidelines and adhere to the Critical Items (CRIs) and Important Points (IPs) detailed in section ‎21. The primary objective of this development process is as follows:

1. To fully address the requirements stipulated by the PSAC, as elaborated in paragraph 11.1 of EUROCAE ED12B / RTCA DO-178B.
2. To meet the criteria outlined in documentXXT123456 - "Software, Airborne Electronic Hardware, and Databases Qualification Methodology," originating from the X-WING Certification Collection.
3. To effectively resolve the considerations highlighted in the Critical Items (CRIs) and Important Points (IPs) expounded upon in section 5.

### Identification

| System Name | – | HUD (Head Up Display) |
| --- | --- | --- |
| System Abbreviation | – | *HUD* |
| Document Id. | – | *000001-00* |

### Document Overview

| System Name | – | *HUD (Head Up Display)* |
| --- | --- | --- |
| System Abbreviation | – | *HUD* |
| Document Id. | – | *000001-00* |

### 

### List of Abbreviations

|  |  |
| --- | --- |
|  |  |
| *HUD* | *Head Up Display* |
|  |  |

## Referenced Documents

| Document Id. | Document Name |
| --- | --- |
| RTCA/DO-178C | SOFTWARE CONSIDERATIONS IN AIRBORNE SYSTEMS AND EQUIPMENT CERTIFICATION – DECEMBER 1, 1992 |
| EUROCAE ED-76 / RTCA DO-200A | Standards for Processing Aeronautical Data, September 28, 1998 |

* 1. Project Documents

| Document Id. | Document Name |
| --- | --- |
| 000002-00 | SDP - Software Development Plan |
| 000003-00 | SQAP - Software Quality Assurance Plan |
| 000004-00 | SVP - Software Verification Plan |
| 000005-00 | SCMP - Software Configuration Management Plan |
| 000006-00 | TQP - Tool Qualification Plan |

* 1. Vendor Documents

| Document Id. | Document Name |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## System Overview

* 1. Functional Description

*The (HUD) Head Up Display System is an intelligent display platform designed to furnish pilots with comprehensive flight information throughout all stages of flight. This includes critical data such as flight path, air data, altitude, attitude, flight director guidance, as well as other pertinent flight cues and alerts. This data is seamlessly projected onto a transparent combiner, ensuring pilots maintain visual contact with both the external surroundings and the seamlessly integrated image.*

*By merging flight and mission data with the EVS camera feed, the system projects an expansive display across a wide HUD Field of View. This approach enhances flight safety by delivering precisely timed and positioned information, thereby minimizing pilot errors and reducing cognitive load. The overarching objective of the EFVS is to heighten pilots' situational awareness during pivotal flight phases and in scenarios with terrain constraints. This technology significantly enhances approach stability, facilitates the transition from IMC to visual landing, refines touchdown precision, and acts as a safeguard against runway excursions.*

3.2 Safety and Partitioning

#### • <Describe your safety and partitioning strategy here, including resource sharing, redundancy, multiple-version dissimilar software, and fault tolerances aspects.>

3.3 Timing and Scheduling Strategies

#### • <Describe your software timing and scheduling strategies and techniques here. Consider using Model-Based Design for the implementation of the scheduling algorithms (e.g. using Stateflow).>

## 4 Certification Considerations

4.1 Certification Basis and Means of Compliance

#### • <Describe the means of compliance to be used for this project. Normally this section will reference DO-178C and supplements. This section must also include references to TC, STC, TSO, Advisory Circulars, Special Conditions, Certification Memoranda, etc.>

Means of compliance are based on the stringent application of [DO-178C] and [DO-331]. The development assurance strategy shall be based on a structured software development approach, as described in Chapter 5, “Software Life Cycle”.

For software development use the following development assurance methods:

• Traceability evidence: Traceability evidence will be generated during development to provide traceability between the applicable documents, the derived development artifacts (e.g. models, source code), through to the verification items.

• Reviews and analysis: Reviews and static analysis activities shall provide the evidence of the accuracy, completeness, and verifiability of the requirements, architecture, design, and implementation.

• Software testing: Dynamic testing will consist of systematic testing of all software functions of the project.

• Model simulation: Model simulation will be used to support model coverage analysis and to support the development of requirements-based functional test cases. It will not replace software integration testing,hardware software integration testing, or testing of the executable object code based on its requirements.

4.2 Software Level

The preliminary safety assessment process for the system resulted in the determination of the software level *<SoftwareLevel>* for the project *<project>*.

#### • <Insert a justification of the software level here, based on the preliminary system safety assessment.>

## 

## 5 Software Life Cycle

5.1 Software Life Cycle Description

#### • <Provide a description of the intended software life cycle here. Consider defining a unified life cycle for conventional and model-based developed components. Provide details for each workflow in the Software Development Plan and Software Verification Plan as described in DO-331 MB4.0. You can use MathWorks "Model-Based Design Workflow for DO-178C" document for support. See also DO-331 Section MB.11.1.d.>

Figure 51 Verification and Traceability Activities and Objectives (Model-Based Design Workflow for DO-178C)

5.2 Software Planning Process

### 5.2.1 Software Planning Process Objectives

#### • <The objectives of the software planning process are defined in DO-178C Section 4.1 and DO-331 SectionMB.4.1.>

### 5.2.2 Software Planning Process Input

The input to the software planning process is:

• DO-178C, DO-330, DO-331 and DO-333.

• The System Requirements for *<project>*.

• The preliminary System Safety Assessment for *<project>*.

#### • *<Add any other document references here (e.g. CRIs, Cert Memos, CAST papers, and others). These document references can also include company standards and process documents.>*

### 5.2.3 Software Planning Process Output

The output of the software planning process:

• Plan for Software Aspects of Certification for *<project>* (this document)

• Software Development Plan for *<project>*

• Software Verification Plan for *<project>*

• Software Configuration Management Plan for *<project>*

• Software Quality Assurance Plan for *<project>*

• Software Requirements Standards for *<project>*

• Software Design Standards for *<project>*

• Software Code Standards for *<project>*

• Software Model Standards for *<project>*

• Simulink Report Generator Tool Qualification Plan for *<project>*

• Simulink Verification and Validation Tool Qualification Plan for *<project>*

• Simulink Code Inspector Tool Qualification Plan for *<project>*

• Polyspace Code Prover Tool Qualification Plan for *<project>*

• Polyspace Bug Finder Tool Qualification Plan for *<project>*

• Simulink Test Tool Qualification Plan for *<project>*

#### <Add additional documents here.>

### 5.2.4 Software Planning Process Activities

#### • <The activities of the software planning process are defined in DO-178C Section 4.2 and DO-331 SectionMB.4.2.>

### 5.2.5 Software Planning Process Transition Criteria

##### 5.2.5.1 Software Planning Process Entry Criteria

#### • <Add the entry criteria for the planning process here. Normally, this criteria includes a reference to the complete planning process input.>

##### 5.2.5.2 Software Planning Process Exit Criteria

#### • <Add the exit criteria for the planning process here. Normally, this criteria includes a reference to the complete planning process outputs, the configured outputs, and the plans agreed to by the certification authority.>

### 5.2.6 Integral Processes

##### 5.2.6.1 Software Configuration Management Activities

#### • <Add a description here of how compliance to the software configuration management process described in Chapter 7 "Software Configuration Management Process" of this document will be achieved for the software planning process (as far as applicable).>

​Details are provided in the applicable Software Configuration Management Plan (SCMP).

##### 5.2.6.2 Software Quality Assurance Activities

#### • <Add a description here of how compliance to the software quality assurance process described in Chapter8 "Software Quality Assurance Process" of this document will be achieved for the software planning process.>

Details are provided in the applicable Software Quality Assurance Plan (SQAP).

•

5.3 Software Requirements Process

### 5.3.1 Software Requirements Process Objectives

#### • <The objectives of the software requirements process are defined in DO-178C Section 5.1.1 and DO-331 Section MB.5.1.1.>

### 5.3.2 Software Requirements Process Input

The input to the software requirements process is:

• The System Requirements for *<project>*.

• The Software Requirements Standards for *<project>*.

The Software Development Plan / or Plan of Software Aspects of Certification for *<project>*.

#### • <Add any other document references here, e.g., the Software Model Standards, if Specification Models are used. These references can include company standards and process documents.>

### 5.3.3 Software Requirements Process Output

The output of the software requirements process is:

• The Software Requirements Document

#### <Add any other document references here, e.g. Specification Models (if used).>

### 5.3.4 Software Requirements Process Activities

#### • <The activities of the software requirements process are defined in DO-178C Section 5.1.2 and DO-331 Section MB.5.1.2.>

### 5.3.5 Software Requirements Process Transition Criteria

##### 5.3.5.1 Software Requirements Process Entry Criteria

#### • <Add the entry criteria for the requirements process here.>

##### 5.3.5.2 Software Requirements Process Exit Criteria

#### • <Add the exit criteria for the requirements process here.>

### 5.3.6 Integral Processes

##### 5.3.6.1 Software Configuration Management Activities

#### • <Add a description here of how compliance to the software configuration management process described in Chapter 7 "Software Configuration Management Process" of this document will be achieved for the software requirements process.>

​Details are provided in the applicable Software Configuration Management Plan (SCMP).

##### 5.3.6.2 Software Quality Assurance Activities

#### • <Add a description here of how compliance to the software quality assurance process described in Chapter8 "Software Quality Assurance Process" of this document will be achieved for the software requirements process.>

​Details are provided in the applicable Software Quality Assurance Plan (SQAP).

5.4 Software Design Process

### 5.4.1 Software Design Process Objectives

#### • <The objectives of the software design process are defined in DO-178C Section 5.2.1 and DO-331 SectionMB.5.2.1.>

### 5.4.2 Software Design Process Input

The input to the software design process is:

• The Software Requirements Document

#### • The Software Specification Model <if used>

• The Software Design Standards

• The Software Model Standards

• The Software Development Plan or Plan of Software Aspects of Certification for *<project>*

#### • <Add any other document references here. These references can include company standards and process documents.>

### 5.4.3 Software Design Process Output

The output of the software design process is:

• The Software Design Description

• The Software Design Model

#### • <Add any other document references here.>

### 5.4.4 Software Design Process Activities

#### • <The activities of the software design process are defined in DO-178C Section 5.2.2 and DO-331 SectionMB.5.2.2.>

### 5.4.5 Software Design Process Transition Criteria

##### 5.4.5.1 Software Requirements Process Entry Criteria

#### • <Add the entry criteria for the requirements process here.>

##### 5.4.5.2 Software Requirements Process Exit Criteria

#### • <Add the exit criteria for the requirements process here.>

### 5.4.6 Integral Processes

##### 5.4.6.1 Software Configuration Management Activities

#### • <Add a description here of how compliance to the software configuration management process described in Chapter 7 "Software Configuration Management Process" of this document will be achieved for the software design process.>

​Details are provided in the applicable Software Configuration Management Plan (SCMP).

##### 5.4.6.2 Software Quality Assurance Activities

#### • <Add a description here of how compliance to the software quality assurance process described in Chapter8 "Software Quality Assurance Process" of this document will be achieved for the software design process.>

​Details are provided in the applicable Software Quality Assurance Plan (SQAP).

**5.5 Software Coding Process**

### 5.5.1 Software Coding Process Objectives

#### • <The objectives of the software coding process are defined in DO-178C Section 5.3.1 and DO-331 SectionMB.5.3.1.>

### 

### 5.5.2 Software Coding Process Input

The input to the software coding process is:

• Software Design Description

• Software Design Model

• Software Code Standards

• Software Development Plan or Plan of Software Aspects of Certification for *<project>*

#### • <Add any other document references here. This might include company standards and process documents.>

### 5.5.3 Software Coding Process Output

The output to the software coding process is:

• Source Code

#### • <Add any other document references here.>

### 5.5.4 Software Coding Process Activities

#### <The activities of the software coding process are defined in DO-178C Section 5.3.2.>

### 5.5.5 Software Coding Process Transition Criteria

##### 5.5.5.1 Software Requirements Process Entry Criteria

#### • <Add the entry criteria for the requirements process here.>

##### 5.5.5.2 Software Requirements Process Exit Criteria

#### • <Add the exit criteria for the requirements process here.>

### 5.5.6 Integral Processes

##### 5.5.6.1 Software Configuration Management Activities

#### • <Add a description here of how compliance to the software configuration management process described in Chapter 7 "Software Configuration Management Process" of this document will be achieved for the software coding process.>

​Details are provided in the applicable Software Configuration Management Plan (SCMP).

**5.5.6.2 Software Quality Assurance Activities**

#### • <Add a description here of how compliance to the software quality assurance process described in Chapter8 "Software Quality Assurance Process" of this document will be achieved for the software coding process.>

​Details are provided in the applicable Software Quality Assurance Plan (SQAP).

5.6 Integration Process

### 5.6.1 Integration Process Objectives

#### • <The objectives of the software integration process are defined in DO-178C Section 5.4.1 and DO-331 Section MB.5.4.1.>

### 5.6.2 Integration Process Input

The input to the software integration process is:

• Software Architecture

• Source Code

• Software Development Plan / or Plan of Software Aspects of Certification for *<project>*

#### • <Add any other document references here (e.g. compiler libraries or precompiled COTS software components). This might include company standards and process documents.>

### 5.6.3 Integration Process Output

The output of the software integration process is:

• Object Code

• Executable Object Code

• Parameter Data Item Files

• Compiling, Linking and Loading Data

#### • <Add any other document references here.>

### 5.6.4 Integration Process Activities

#### <The activities of the software coding process are defined in DO-178C Section 5.4.2.>

### 5.6.5 Integration Process Transition Criteria

##### 5.6.5.1 Software Requirements Process Entry Criteria

#### • <Add the entry criteria for the requirements process here.>

##### 5.6.5.2 Software Requirements Process Exit Criteria

#### • <Add the exit criteria for the requirements process here.>

### 5.6.6 Integral Processes

##### 5.6.6.1 Software Configuration Management Activities

#### • <Add a description here of how compliance to the software configuration management process described in Chapter 7 "Software Configuration Management Process" of this document will be achieved for the software integration process.>

​Details are provided in the applicable Software Configuration Management Plan (SCMP).

##### 5.6.6.2 Software Quality Assurance Activities

#### 

#### <Add a description here of how compliance to the software quality assurance process described in Chapter8 "Software Quality Assurance Process" of this document will be achieved for the software integration process.>

​Details are provided in the applicable Software Quality Assurance Plan (SQAP).

5.7 Software Verification Process

### 5.7.1 Software Verification Process Overview

#### • <The software verification process is an integral process (DO-178C Section 3.1 and DO-331 SectionMB.3.1) that runs in parallel with the development processes. The purpose and activities of the software verification process are defined in DO-178C Section 6.1 - 6.2 and DO-331 Section MB.6.1 – MB.6.2. Add a general description of your software verification process here.>

### 5.7.2 Reviews and Analyses

##### 5.7.2.1 Review and Analysis of Software Planning Process

#### • <The objective of this review and analysis is to demonstrate that the software plans and software development standards comply with the applicable regulations and purchaser specifications. The verification objectives of the software planning process are defined in DO-178C Section 4.1 (f and g) and DO-331 Section 4.1 (f and g). The reviews and assurances of the software planning process are defined in DO-178C Section 4.6 and DO-331 Section MB.4.6. Add a description here explaining how the objectives, reviews, and assurances will be achieved>

##### 5.7.2.2 Review and Analysis of Software High-Level Requirements

#### • <The objectives of the reviews and analysis of high-level requirements are defined in DO-178C Section6.3.1 and DO-331 Section MB.6.3.1. For details, see DO-178C Table A-3 and DO-331 Table MB.A-3.

#### • Add a description here explaining how the objectives, reviews, and assurances will be achieved.>

##### 

##### 5.7.2.3 Review and Analysis of Specification Model

#### • <If you use specification models, the objectives of the review and analysis of high-level requirements expressed by a specification model are defined in DO-331 Sections MB.6.3.1 and MB.6.8. For details, see also DO-331 Table MB.A-3.

#### • Add a description here explaining how the objectives, reviews, and assurances will be achieved.>

##### 

##### 5.7.2.4 Review and Analysis of Software Design Description

#### • <The objectives of the review and analysis of the software design are defined in DO-178C Section 6.3.2 and 6.3.3. For details, see DO-178C Table A-4 and DO-331 Table MB.A-4.

#### • Add a description here explaining how the objectives, reviews, and assurances will be achieved>​

##### 5.7.2.5 Review and Analysis of Design Model

#### • <The objectives of the review and analysis of the design model are defined in DO-331 Section MB.6.3.2 and MB.6.3.3. According to Table MB.A-4 objective 1, a model coverage analysis must be performed to demonstrate the compliance of the design model with its high-level requirements. Therefore, DO-331 Section MB.6.7 and MB.6.8 also apply. The Simulink Model Advisor and Simulink Report Generator can support the review of the design model. For details, see DO-331 Table MB.A-4.

#### • Add a description here explaining how the objectives, reviews, and assurances will be achieved.>

##### 5.7.2.6 Review and Analysis of Source Code

#### • <The objectives of the review and analysis of the source code are defined in DO-178C Section 6.3.4 and DO-331 Section MB.6.3.4. Polyspace Bug Finder can support the code review activities. Simulink Code Inspector can support the review of the auto-generated source code. For details, see DO-178C Table A-5.

#### • Add a description here explaining how the objectives, reviews, and assurances will be achieved.>

##### 5.7.2.7 Test Coverage Analysis

#### • <Add a description here explaining how the objectives of test coverage of high-level requirements, of low-level requirements, and of software structure according to DO-178C Section 6.4.4 and DO-331 SectionMB.6.4.4 will be achieved. For details, see DO-178C Table A-7 and DO-331 Table MB.A-7.

#### • Also, consider performing the analysis of data and control coupling (DO-178c Section 6.4.4.d and DO-331 Section MB.6.4.4.d) in parallel with the structural coverage analysis. The use of Polyspace Code Proverand Simulink Code Inspector can assist this analysis.>

### 5.7.3 Model Simulation

#### < As defined in DO-331 Section MB.6.8, model simulation is different from testing. Typical benefits of model simulation are:

#### • Earlier and easier verification of the algorithm than during testing of the executable object code.

#### • Development of test cases for the testing the executable object code and to support model coverage analysis.

#### Add a description here explaining how model simulation will be used to support the objectives listed in DO-331 Table MB.A-3, MB.A-4, MB.A-6 and MB.A-7.>

### 

### 5.7.4 Testing

#### < Testing the executable object code is typically performed on the target computer hardware or in an equivalent environment. It consists of software integration testing where the integration of handwritten parts and auto-generated parts of the software is verified and hardware software integration testing where the correctness of the executable object code and the CPU is verified.>

#### Processor-in-the-loop testing is a state-of-the-art method to demonstrate the correctness of the modeled functionality in its target environment using requirements based test cases. Consider reusing the test cases from model simulation.

#### Add a description here explaining how the objectives of DO-178C Section 6.4 and DO-331 Section MB.6.4 will be achieved. For details, see DO-178C Table A-6 and DO-331 Table MB.A-6.>

**5.7.5 Additional Verification Requirements**

##### 5.7.5.1 Traceability

The traceability between artifacts is a fundamental objective of DO-178C and the corresponding supplements. According to DO-178C Section 11.21 and DO-331 Section MB.11.21, traceability shall be ensured between:

• System requirements and high-level requirements (Section (MB.)6.3.1.f)

• High-level requirements and low-level requirements (Section (MB.)6.3.2.f)

• Source code and low-level requirements (Section (MB.)6.3.4.e)

• Software requirements and test cases (Section 6.5 a)

• Test cases and test procedures (Section 6.5 b)

• Test procedures and test results (Section 6.5 c)

#### < Add a description here explaining how the objectives above will be achieved. MathWorks Model Advisor and Code Inspector can support the traceability review.>

### 5.7.6 Derived Requirements

#### <The software development processes might produce derived high-level or low-level requirements. In either case, these requirements shall be provided to the system safety process. Additional verification effort will be required to ensure the correctness of the resulting functionality (DO-331 Table MB.6-1).>

### 5.7.7 Independence of Verification

#### <Depending on the software design assurance level, the independence between software development processes and the software verification process might be mandatory. Add a description here explaining how this independence will be achieved. Consider the use of verification tools like the Simulink Code Inspector to claim credit for satisfied independence.>

### 5.7.8 Integral Processes

##### 5.7.8.1 Software Configuration Management Activities

#### • <Add a description here of how compliance to the software configuration management process described in Chapter 7 "Software Configuration Management Process" of this document will be achieved for the software verification process.>

#### ​Details are provided in the applicable Software Configuration Management Plan (SCMP).

##### 5.7.8.2 Software Quality Assurance Activities

#### • <Add a description here of how compliance to the software quality assurance process described in Chapter8 "Software Quality Assurance Process" of this document will be achieved for the software verification process.>

#### ​Details are provided in the applicable Software Quality Assurance Plan (SQAP).

## 6 Additional Considerations

6.1 Tool Use and Qualification

### 6.1.1 General

According to DO-178C Section 12.2, "Qualification of a tool is needed when processes of this document are eliminated, reduced, or automated by the use of a software tool without its output being verified as specified in Section 6. The purpose of the tool qualification process is to ensure that the tool provides confidence at least equivalent to that of the process(es) eliminated, reduced, or automated."

Therefore, DO-178C defines three categories of tools in table 12-1. The tool qualification process will be conducted in accordance with DO-330.

Sections 6.1.2 to 6.1.5 of this document discuss the use of tools, the determination of the tool qualification level (TQL), and qualification of tools throughout this project.

**6.1.2 Tools Used in Software Planning Process**

| **Software Tool** | **Version  (Release)** | **Tool Vendor** | **Tool Criteria** | **Usage Description and Tool Qualification** |
| --- | --- | --- | --- | --- |
| *<Insert tool>* | *<…>* | *<…>* | *<…>* | *<…>* |

### 6.1.3 Tools Used in Software Development Process

| **Software Tool** | **Version  (Release)** | **Tool Vendor** | **Tool Criteria** | **Usage Description and Tool Qualification** |
| --- | --- | --- | --- | --- |
| *<Insert tool>* | *<…>* | *<…>* | *<…>* | *<Requirements Engineering and Management Tool>* |
| MATLAB | See SCI | MathWorks, Inc. | 1 | Modeling and simulation environment. The output of the tool will be verified as specified in Section 6 of DO-331.No credit is sought for the use of the tool. Therefore, no tool qualification is required. |
| Simulink | See SCI | MathWorks, Inc. | 1 | Modeling and simulation environment. The output of the tool will be verified as specified in Section 6 of DO-331. No credit is sought for the use of the tool. Therefore, no tool qualification is required. |
| Stateflow | See SCI | MathWorks, Inc. | 1 | Modeling and simulation environment. The output of the tool will be verified as specified in Section 6 of DO-331. No credit is sought for the use of the tool. Therefore, no tool qualification is required. |
| Embedded Coder | See SCI | MathWorks, Inc. | 1 | Embedded Coder is used to automatically generate C code from a Simulink model. The output of the tool will be verified as specified in Section 6 of DO-331. No credit is sought for the use of the tool. Therefore, no tool qualification is required. |
| *<Insert tool>* | *<…>* | *<…>* | *<…>* | *<Software development environment (e.g. Eclipse IDE)>* |
| *<Insert tool>* | *<…>* | *<…>* | *<…>* | Compiler. According to DO-178C Section 4.4.2, for a compiler, no specific tool qualification is required. For more information on the compiler, see Section 6.4"Compiler Considerations" of this document. |
| *<Insert tool>* | *<…>* | *<…>* | *<…>* | *<Debugger>* |
| *<Insert additional tools>* | *<…>* | *<…>* | *<…>* | *<…>* |

### 6.1.4 Tools Used in Software Verification Process

| **Software Tool** | **Version  (Release)** | **Tool Vendor** | **Tool Criteria** | **Usage Description and Tool Qualification** |
| --- | --- | --- | --- | --- |
| Simulink®Verification and Validation™  Model Advisor | See SCI | MathWorks, Inc. | 3 | Simulink Verification and Validation Model Advisor is used for static analysis of the Specification Model. See Simulink Verification and Validation TQP for determination of TQL and for credits sought. |
| Simulink®Verification and Validation™  Model Coverage | See SCI | MathWorks, Inc. | 3 | Simulink Verification and Validation Model Coverage is used for model coverage analysis according to DO-331 MB.6.7. See Simulink Verification and Validation TQP for determination of TQL and for credits sought. |
| Simulink®Verification and Validation™  Requirements Management Interface | See SCI | MathWorks, Inc. | 3 | Simulink Verification and Validation comprises a Requirements Management Interface that is being used to link requirements to Simulink model elements. The output of the tool will be verified by review according to DO-331 MB.6.3.2. No credit is sought for the use of the tool. Therefore, no tool qualification is required. |
| Simulink Code Inspector | See SCI | MathWorks, Inc. | 2 | Simulink Code Inspector supports the verification of the source code according to DO-331 MB.6.3.4. See Simulink Code Inspector TQP for determination of TQL and for credits sought. |
| Simulink Report Generator | See SCI | MathWorks, Inc. | 3 | Simulink Report Generator comprises capabilities to generate System Design Description (SDD) and Model XML Comparison reports. See Simulink Report Generator TQP for detailed information. |
| Simulink®Test™ | See SCI | MathWorks, Inc. | 3 | Simulink Test supports the verification of the models according to DO-331 MB.6.8.3.2 and executable object code according to DO-331 MB.6.4.2, 6.4.3 and 6.4.4. See Simulink Test TQP for determination of TQL and for credits sought. |
| Polyspace®Bug Finder | See SCI | MathWorks, Inc. | 2 | Polyspace Bug Finder supports the verification of the source code according to DO-331 MB.6.3.4 and DO-333 FM.6.3.4. See Polyspace Bug Finder TQP for determination of TQL and for credits sought. |
| Polyspace®Code Prover | See SCI | MathWorks, Inc. | 2 | Polyspace Code Prover supports the verification of the source code and executable object code according to DO-331 MB6.3.4 and DO-333 FM.6.2.1, FM.6.3 and FM.6.7. See Polyspace Coder Prover TQP for determination of TQL and for credits sought. |
| *<Insert tool>* | *<…>* | *<…>* | *<…>* | *<Structural coverage analysis tool>* |
| *<Insert additional tools>* | *<…>* | *<…>* | *<…>* | *<…>* |

### 6.1.5 Development Environment and Documentation Tools

| **Software Tool** | **Version  (Release)** | **Tool Vendor** | **Tool Criteria** | **Usage Description and Tool Qualification** |
| --- | --- | --- | --- | --- |
| *<Insert tool>* | *<…>* | *<…>* | *<…>* | *<Desktop Operating System (e.g. Windows)>* |
| *<Insert tool>* | *<…>* | *<…>* | *<…>* | *<Documentation Tools (e.g. Microsoft Office)>* |
| *<Insert tool>* | *<…>* | *<…>* | *<…>* | *<Configuration management tool>* |
| *<Insert tool>* | *<…>* | *<…>* | *<…>* | *<Problem reporting tool>* |
| *<Insert additional tools>* | *<…>* | *<…>* | *<…>* | *<…>* |

6.2 Use of Floating Point Numbers

#### • <If floating-point arithmetic and number representation is used, reference the standard to be applied (e.g. IEEE 754), define the format to be used (e.g. simple accuracy or double accuracy), and provide the restrictions and constraints and guidelines in the Software Code Standards and Software Model Standards(e.g. comparison of floating-point numbers for in-/ equality).>

6.3 Use of Software Libraries

#### • <Add a description here explaining how calls to software libraries will be handled. If your software development environment does not support calls to standard compiler libraries, configure your model to avoid calls to the standard libraries in the generated code. In your model, to avoid calls, consider using:

#### - Code Replacement Libraries (CRLs) to remap standard library calls.

#### - Blocks and model constructs that do not generate calls to standard compiler libraries.>

6.4 Compiler Considerations

#### • <Provide a brief description of the cross-compiler to be used including information on "what compiler will be used", "product service history", and "compiler settings".>

6.5 Programming Languages

The programming language to be used is ANSI C. To assure the proper use of the programming language by defining guidelines and restrictions, the Software Code Standards shall be applied.

The modeling languages to be used are MATLAB, Simulink, and Stateflow, as defined and restricted in the Software Model Standards.

Assembly language shall be used for time-critical and hardware-related functions only.

## 

## 

## 7 Software Configuration Management Process

#### • <The objectives of the software configuration management process are defined in DO-178C Section 7.1 and DO-331 Section MB.7.1.

#### • Add a description here explaining how the software configuration management process will be established.>

## 8 Software Quality Assurance Process

#### • <The objectives of the software quality assurance process are defined in DO-178C Section 8.1 and DO-331 Section MB.8.1.

#### • Add a description here explaining how the software quality assurance process will be established.>

## 9 Certification Liaison Process

#### • <The objectives of the certification liaison process are defined in DO-178C Section 9 and DO-331 SectionMB.9.

#### • Add a description here explaining how the certification liaison process will be established.>

## 

## 

## 10 Software Life Cycle Data

The software life cycle data that will be produced in the project *<project>* is:

| **Document Title** | **Notes** |
| --- | --- |
| **Software Plans** |  |
| Plan for Software Aspects of Certification | (this document) |
| Software Development Plan |  |
| Software Verification Plan |  |
| Software Configuration Management Plan |  |
| Software Quality Assurance Plan |  |
| Software Requirements Standards |  |
| Software Design Standards |  |
| Software Code Standards |  |
| Software Model Standards |  |
| Simulink Report Generator Tool Qualification Plan |  |
| Simulink Verification and Validation Tool Qualification Plan |  |
| Simulink Code Inspector Tool Qualification Plan |  |
| Polyspace Code Prover Tool Qualification Plan |  |
| Polyspace Bug Finder Tool Qualification Plan |  |
| Simulink Test Tool Qualification Plan |  |
|  |  |
| **Software Documents** |  |
| Software Requirements Document |  |
| Software Design Description |  |
|  |  |
| **Software Artifacts** |  |
| Specification Model |  |
| Design Model |  |
| Source Code |  |
| Executable Object Code |  |
|  |  |
| **Verification Documents** |  |
| Software Verification Test Cases and Procedures |  |
| Software Accomplishment Summary |  |
| Software Configuration Index |  |
| <Insert additional artifacts here.> |  |

## 

## 11 Organization and Responsibilities

#### < Add a description here explaining the management structure within the project. Clearly define the required roles and responsibilities to ease the demonstration of independence (if required)

#### 

#### **12 Milestones and Schedule**

#### < Add an overview about the project schedule and the defined milestones here to ease the communication and planning of audits with the authorities.>

## 

#### 