Plan for Software Aspects of Certification

Document Number: H398-001-001

Version No: 2.7

|  |
| --- |
| Proprietary Notice  This document and the information disclosed herein are proprietary data of Howell Instruments, Inc. Neither this document nor the information contained herein will be reproduced, used, or disclosed to others without the written authorization of Howell Instruments, Inc. |

|  |  |
| --- | --- |
| Written By: | Nabirasul Herkal (Technical Manager, ALTEN Global TechnologiesPrivate Limited).  image |
| Reviewed By: | Prajwal R (Software Engineer, ALTEN Global TechnologiesPrivate Limited)  .Pavitra D (Quality Engineer – Quality Assurance, ALTEN Global TechnologiesPrivate Limited)  . |
| Approved By: | Shyamala B (Senior Director - Quality Assurance, ALTEN Global TechnologiesPrivate Limited)  cid:image001.jpg@01D61E3F.6C97D0F0 . |
| Released By: | Madhuchandra S (Associate Director - Software, ALTEN Global TechnologiesPrivate Limited)  .C:\Users\afreen.p.SWSYS\Pictures\SIGNATURE.PNG |



****ALTEN Global Technologies Private Limited****

# Amendment Record

|  |  |  |  |
| --- | --- | --- | --- |
| **Version No.** | **Description of Amendment** | **Change**  **Request No.** | **Release**  **Date** |
| 1.1 | Initial Release | NA | 20-Sept-2021 |
| 1.2 | Updated to address review comments   1. Updated acronyms in section 4 | 100001 | 20-Sep-2021 |
| 1.3 | Updated to address the following:   1. Changed company name to "Accord Global Technology Solutions Pvt. Ltd" 2. Updated "Table 2: Acronyms and Definitions" Updated "Section 6 Responsibilities" 3. Updated "Figure 2: Functional block diagram of EDAU and CMU+" 4. Updated "Table 4: Software binaries, DAL and Responsibilities" 5. Updated last paragraph in section "8 SOFTWARE OVERVIEW" 6. Updated sections "8.1.1.1 Boot Loader", "8.1.2.1 Boot Loader", "8.1.3.1 Boot Loader" and "8.2.1 Boot Loader" to mention boot loader as certified software 7. Spell check done 8. Updated "Table 6: Roles and Responsibilities and Official Communications Contacts" to change role of Project Leader 9. Updated "Table 8: Software Development Process Data" to change naming conventions and to add build process document 10. Updated section "12.5 Previously Developed Software" 11. Updated last paragraph section "12.6.1 Aircraft Configuration Data File" 12. Updated "Figure 3: Functional block diagram of EIU and CMU+" 13. Updated Table 3 to add description about Boot Up Configuration file | 100005 | 28-Sep-2021 |
| 1.4 | Updated to address self review comments   1. Updated Table 8 and SI No 5 to modify the part numbers of softwares 2. Updated Table 8 and SI No 6 to modify the part number of build process document 3. Updated Table 8 and SI No 7 to diffrentiate the boot config for analog and discrete modules | 100036 | 23-Feb-2022 |
| 1.5 | Updated to change the existing planning documents to product specific planning documents | 100103 | 08-Aug-2022 |
| 2.0 | Updated to address the following:  Updated the following sections to change EIU to EDAU   1. Section 7.2 Configuration Management Unit Plus NVM (CMU+) 2. Section 8.3 Scheduler 3. Updated”Figure 3: Functional block diagram of EDAU and CMU+” 4. Section 8.6 Language Used for Software Development 5. Section 10.1 Organization – Roles and Responsibilities 6. Section 10.1.1 Accord Global Technology Solutions Pvt. Ltd 7. Section 12.1 Parametric Configuration Data 8. Section 8.4 Functional block diagram of EIU and CMU+ Software 9. Section 9.2 Hardware Safety Features 10. Section 12.2 Tool Qualification 11. Section 12.7 User Modifiable Software   Updated the following sections as per the Deviations to SAS   1. Section 7 System Overview 2. Section 7.1 Engine Data Acquisition Unit (EDAU) 3. Table 4 Software binaries, DAL and Responsibilities 4. Section 8.1.1.3 Application Software 5. Section 8.1.2.3 Application Software 6. Section 8.2.3 Application Software 7. Table 8: Software Development Process Data 8. Section 12.4 Field Loadable Software 9. Section 12.6 Option Selectable Software/Deactivated code 10. Section 12.6.2 Application Software   Updated ”Figure 4 Organizational Responsibilities and Independence.”  to modify the names and designations.  Updated “Table 6: Roles and Responsibilities and Official Communications Contacts’’ to modify the names and contact numbers  Updated ”Figure 2 Functional block diagram of EDAU and CMU+”.  Updated “Table 11: Tool Qualification”   1. Changed National Instrument (NI) Test Equipment to Test Box 2. Deleted NI Instrument Max   Updated ”Section 8.1.3.3 Application Software” to change the word “Depends” to “Depending”.  Updated ”Section 4 Acronyms and Definitions” removed the NI Acronyms.  Updated ”Table 8: Software Development Process Data” to change the Part Numbers of Module Configuration and Aircraft Configuration. | 100165 | 24-July-2023 |
| 2.1 | Updated the following as per the QA Review comments   1. Updated ”Section 4 Acronyms and Definitions” to change Definition of EIU. 2. Updated ”Section 7 System Overview” to change Engine Instrument System to Engine Interface Unit. 3. Updated “Table 6: Roles and Responsibilities and Official Communications Contacts’’ to change Organization name. 4. Updated ”Section 13 PROJECT SCHEDULE" 5. Updated ”Section 12.2 Tool Qualification” 6. Updated ”Table 13 Option Selectable Software-Application Software" 7. Updated ”Figure 4 Organizational Responsibilities and Independence.”to modify the names and designations. 8. Updated “Section 8.4 Functional block diagram of EDAU and CMU+ Software “ to change EIU to EDAU in Amendment Record version 2.0.   Updated the following as per Self Review comments   1. Updated the version from 2.1 to 2.0. 2. Updated ”Section 4 Acronyms and Definitions” 3. Updated ” Section 7.1 Engine Data Acquisition Unit (EDAU)” 4. Updated ”Section 8.1.1.1 Boot Loader" 5. Updated ”Section 8.1.1.2 Boot-Up Config file" 6. Updated ”Section 8.1.1.3 Application Software" 7. Updated ”Section 8.1.1.5 Software Loader" 8. Updated ”Section 8.1.2.1 Boot Loader" 9. Updated ”Section 8.1.2.2 Boot-Up Config file" 10. Updated ”Section 8.1.2.5 Software Loader" 11. Updated ”Section 8.1.3.1 Boot Loader" 12. Updated ”Section 8.1.3.2 Boot-Up Config filace" 13. Updated ”Section 8.1.3.5 Software Loader" 14. Updated ”Section 8.2.1 Boot Loader" 15. Updated ”Section 8.2.2 Boot-Up Config file" 16. Updated ”Section 8.2.4 Software Loader" 17. Updated ”Figure 2 Functional block diagram of EDAU and CMU+” to Architecture Block Diagram. | 100165 | 26-July-2023 |
| 2.2 | Added section 12.10 DO178B justification on AC-115D  Updated software partnumbers of application software in Table 8: Software Development Process Data | 100232 | 26-June-2024 |
| 2.3 | Updated the following as per the Review comments:   1. Changed all occurrences of Accord Global Technology Solutions Pvt. Ltd.” To “ALTEN Global Technologies Private Limited” in the document 2. Updated front page 3. Water mark removed 4. Updated Section 10.1 Organization – Roles and Responsibilities   Table 6: Roles and Responsibilities and Official Communications Contacts | 100232 | 5th July 2024 |
| 2.4 | Updated the following as per the Review comments   1. Changed all occurrences of ‘ALTEN’ To “ALTEN GT” in the document 2. Updated Section 10.1 Organization – Roles and Responsibilities | 100232 | 9th July 2024 |
| 2.5 | Updated the following as per the QA Review comments:  1.Amendment record  2.Section 10.1 Organization – Roles and Responsibilities  3.Updated front page | 100232 | 11th July 2024 |
| 2.6 | Below updates are done to address the FAA/DER comments:   1. Updated *Table 1: References* to add AC-115D 2. Added Native and Mixed System architecture in section 7 3. Added previously developed software information in section 12.5 4. Updated section 13.1 Project Milestones to reflect the latest dates | 100241 | 7th August 2024 |
| 2.7 | Updated AC-115D to AC 20-115D in section 3 and 12.10 | 100300 | 13th September 2024 |

**TABLE OF CONTENTS**

[Amendment Record 2](#_Toc177112077)

[1 Objective 17](#_Toc177112078)

[2 Document Scope 17](#_Toc177112079)

[3 References 17](#_Toc177112080)

[4 Acronyms, Terms and Definitions 18](#_Toc177112081)

[5 Document Control 22](#_Toc177112082)

[6 Program Responsibilities 22](#_Toc177112083)

[7 System Overview 23](#_Toc177112084)

[7.1 Engine Data Acquisition Unit (EDAU) 25](#_Toc177112085)

[7.2 Configuration Management Unit Plus NVM (CMU+) 26](#_Toc177112086)

[7.3 Functional block diagram of Engine Data Acquisition System (EDAU) and CMU+ 27](#_Toc177112087)

[8 Software Overview 28](#_Toc177112088)

[8.1 Engine Data Acquisition Unit (EDAU) 29](#_Toc177112089)

[8.1.1 Gateway Module 29](#_Toc177112090)

[8.1.2 Discrete Module 31](#_Toc177112091)

[8.1.3 Analog Module 32](#_Toc177112092)

[8.2 Configuration Management Unit Plus NVM (CMU+) 33](#_Toc177112093)

[8.2.1 Boot Loader 33](#_Toc177112094)

[8.2.2 Boot-Up Config file 33](#_Toc177112095)

[8.2.3 Application Software 33](#_Toc177112096)

[8.2.4 Software Loader 33](#_Toc177112097)

[8.2.5 Maintenance Application 33](#_Toc177112098)

[8.2.6 DL Application Software 33](#_Toc177112099)

[8.2.7 DL Application Configuration 33](#_Toc177112100)

[8.3 Scheduler 34](#_Toc177112101)

[8.4 Functional block diagram of EDAU and CMU+ Software 34](#_Toc177112102)

[8.5 Configuration Data 35](#_Toc177112103)

[8.5.1 Module Configuration Data 35](#_Toc177112104)

[8.5.2 Aircraft Configuration Data 36](#_Toc177112105)

[8.6 Language Used for Software Development 36](#_Toc177112106)

[9 Certification Considerations 37](#_Toc177112107)

[9.1 List of Hazardous Events 37](#_Toc177112108)

[9.2 Hardware Safety Features 39](#_Toc177112109)

[10 Software Life Cycle 40](#_Toc177112110)

[10.1 Organization – Roles and Responsibilities 40](#_Toc177112111)

[10.1.1 ALTEN Global Technologies Private Limited 42](#_Toc177112112)

[10.1.2 Howell Instruments, Inc. 42](#_Toc177112113)

[10.1.3 KTronics Aero Services. 43](#_Toc177112114)

[10.2 Software Life Cycle Model 43](#_Toc177112115)

[10.3 Software Life Cycle Processes 43](#_Toc177112116)

[10.3.1 Software Planning Process 44](#_Toc177112117)

[10.3.1.1 Inputs 44](#_Toc177112118)

[10.3.1.2 Activities 44](#_Toc177112119)

[10.3.1.3 Outputs 44](#_Toc177112120)

[10.3.1.4 Transition Criteria 44](#_Toc177112121)

[10.3.2 Software Development Process 44](#_Toc177112122)

[10.3.2.1 Software Requirements Process 45](#_Toc177112123)

[10.3.2.1.1 Inputs 45](#_Toc177112124)

[10.3.2.1.2 Activities 45](#_Toc177112125)

[10.3.2.1.3 Outputs 45](#_Toc177112126)

[10.3.2.1.4 Transition Criteria 45](#_Toc177112127)

[10.3.2.2 Software Architectural Design Process 46](#_Toc177112128)

[10.3.2.2.1 Inputs 46](#_Toc177112129)

[10.3.2.2.2 Activities 46](#_Toc177112130)

[10.3.2.2.3 Outputs 46](#_Toc177112131)

[10.3.2.2.4 Transition Criteria 46](#_Toc177112132)

[10.3.2.3 Software Low Level Requirements Process 46](#_Toc177112133)

[10.3.2.3.1 Inputs 47](#_Toc177112134)

[10.3.2.3.2 Activities 47](#_Toc177112135)

[10.3.2.3.3 Outputs 47](#_Toc177112136)

[10.3.2.3.4 Transition Criteria 47](#_Toc177112137)

[10.3.2.4 Software Coding and Integration Process 47](#_Toc177112138)

[10.3.2.4.1 Inputs 47](#_Toc177112139)

[10.3.2.4.2 Activities 48](#_Toc177112140)

[10.3.2.4.3 Outputs 48](#_Toc177112141)

[10.3.2.4.4 Transition Criteria 48](#_Toc177112142)

[10.3.3 Software Verification Process 48](#_Toc177112143)

[10.3.3.1 Verification of the Software Planning Process 49](#_Toc177112144)

[10.3.3.1.1 Inputs 49](#_Toc177112145)

[10.3.3.1.2 Activities 49](#_Toc177112146)

[10.3.3.1.3 Outputs 50](#_Toc177112147)

[10.3.3.1.4 Transition Criteria 50](#_Toc177112148)

[10.3.3.2 Verification of the Software Requirements Process 50](#_Toc177112149)

[10.3.3.2.1 Inputs 50](#_Toc177112150)

[10.3.3.2.2 Activities 51](#_Toc177112151)

[10.3.3.2.3 Outputs 51](#_Toc177112152)

[10.3.3.2.4 Transition Criteria 51](#_Toc177112153)

[10.3.3.3 Verification of the Software Architectural Design Process 51](#_Toc177112154)

[10.3.3.3.1 Inputs 51](#_Toc177112155)

[10.3.3.3.2 Activities 51](#_Toc177112156)

[10.3.3.3.3 Outputs 51](#_Toc177112157)

[10.3.3.3.4 Transition Criteria 52](#_Toc177112158)

[10.3.3.4 Verification of the Software Low Level Requirements Process 52](#_Toc177112159)

[10.3.3.4.1 Inputs 52](#_Toc177112160)

[10.3.3.4.2 Activities 52](#_Toc177112161)

[10.3.3.4.3 Outputs 52](#_Toc177112162)

[10.3.3.4.4 Transition Criteria 52](#_Toc177112163)

[10.3.3.5 Verification of the Software Coding and Integration Process 53](#_Toc177112164)

[10.3.3.5.1 Inputs 53](#_Toc177112165)

[10.3.3.5.2 Activities 53](#_Toc177112166)

[10.3.3.5.3 Outputs 53](#_Toc177112167)

[10.3.3.5.4 Transition Criteria 53](#_Toc177112168)

[10.3.3.6 Testing of Outputs of the Software Integration Process 54](#_Toc177112169)

[10.3.3.6.1 Inputs 54](#_Toc177112170)

[10.3.3.6.2 Activities 54](#_Toc177112171)

[10.3.3.6.3 Outputs 55](#_Toc177112172)

[10.3.3.6.4 Transition Criteria 55](#_Toc177112173)

[10.3.3.7 Verification of the Verification Process 56](#_Toc177112174)

[10.3.3.7.1 Inputs 56](#_Toc177112175)

[10.3.3.7.2 Activities 56](#_Toc177112176)

[10.3.3.7.3 Outputs 56](#_Toc177112177)

[10.3.3.7.4 Transition Criteria 56](#_Toc177112178)

[10.3.4 Software Configuration Management Process 57](#_Toc177112179)

[10.3.5 Software Quality Assurance Process 57](#_Toc177112180)

[10.3.5.1 Quality Assurance of Software Planning Process 58](#_Toc177112181)

[10.3.5.1.1 Inputs 58](#_Toc177112182)

[10.3.5.1.2 Activities 58](#_Toc177112183)

[10.3.5.1.3 Outputs 58](#_Toc177112184)

[10.3.5.1.4 Transition Criteria 58](#_Toc177112185)

[10.3.5.2 Quality Assurance of Software Requirements Process 59](#_Toc177112186)

[10.3.5.2.1 Inputs 59](#_Toc177112187)

[10.3.5.2.2 Activities 59](#_Toc177112188)

[10.3.5.2.3 Outputs 59](#_Toc177112189)

[10.3.5.2.4 Transition Criteria 59](#_Toc177112190)

[10.3.5.3 Quality Assurance of Software Architectural Design Process 59](#_Toc177112191)

[10.3.5.3.1 Inputs 59](#_Toc177112192)

[10.3.5.3.2 Activities 60](#_Toc177112193)

[10.3.5.3.3 Outputs 60](#_Toc177112194)

[10.3.5.3.4 Transition Criteria 60](#_Toc177112195)

[10.3.5.4 Quality Assurance of Software Low Level Requirements Process 60](#_Toc177112196)

[10.3.5.4.1 Inputs 60](#_Toc177112197)

[10.3.5.4.2 Activities 61](#_Toc177112198)

[10.3.5.4.3 Outputs 61](#_Toc177112199)

[10.3.5.4.4 Transition Criteria 61](#_Toc177112200)

[10.3.5.5 Quality Assurance of Software Coding and Integration Process 61](#_Toc177112201)

[10.3.5.5.1 Inputs 61](#_Toc177112202)

[10.3.5.5.2 Activities 61](#_Toc177112203)

[10.3.5.5.3 Outputs 62](#_Toc177112204)

[10.3.5.5.4 Transition Criteria 62](#_Toc177112205)

[10.3.5.6 Quality Assurance of Testing of Outputs of the Software Integration Process 62](#_Toc177112206)

[10.3.5.6.1 Inputs 62](#_Toc177112207)

[10.3.5.6.2 Activities 62](#_Toc177112208)

[10.3.5.6.3 Outputs 63](#_Toc177112209)

[10.3.5.6.4 Transition Criteria 63](#_Toc177112210)

[10.3.5.7 Quality Assurance of Verification of the Verification Process 63](#_Toc177112211)

[10.3.5.7.1 Inputs 63](#_Toc177112212)

[10.3.5.7.2 Activities 63](#_Toc177112213)

[10.3.5.7.3 Outputs 63](#_Toc177112214)

[10.3.5.7.4 Transition Criteria 64](#_Toc177112215)

[10.3.6 Certification Liaison Process 64](#_Toc177112216)

[10.3.6.1 Inputs 64](#_Toc177112217)

[10.3.6.2 Activities 64](#_Toc177112218)

[10.3.6.3 Outputs 64](#_Toc177112219)

[10.3.6.4 Transition Criteria 64](#_Toc177112220)

[10.3.7 System Safety Assessment Process 65](#_Toc177112221)

[10.3.7.1 Inputs 65](#_Toc177112222)

[10.3.7.2 Activities 65](#_Toc177112223)

[10.3.7.3 Outputs 65](#_Toc177112224)

[10.3.7.4 Transition Criteria 65](#_Toc177112225)

[11 Software Life Cycle Data 66](#_Toc177112226)

[11.1 Software Planning Process Data 66](#_Toc177112227)

[11.2 Software Development Process Data 72](#_Toc177112228)

[11.3 Software Verification, Software Configuration Management and Software Quality Assurance Process Data 76](#_Toc177112229)

[11.4 Certification Liaison Process Data 85](#_Toc177112230)

[12 Additional Considerations 86](#_Toc177112231)

[12.1 Parametric Configuration Data 86](#_Toc177112232)

[12.2 Tool Qualification 86](#_Toc177112233)

[12.3 Use of Software COTS Components and Software 88](#_Toc177112234)

[12.4 Field Loadable Software 88](#_Toc177112235)

[12.5 Previously Developed Software 89](#_Toc177112236)

[12.6 Option Selectable Software/Deactivated code 90](#_Toc177112237)

[12.6.1 Aircraft Configuration Data File 91](#_Toc177112238)

[12.6.2 Application Software 92](#_Toc177112239)

[12.6.3 Module Configuration Data File 92](#_Toc177112240)

[12.7 User Modifiable Software 93](#_Toc177112241)

[12.8 Multiple Version Dissimilar Software 93](#_Toc177112242)

[12.9 Product Service History 93](#_Toc177112243)

[12.10 DO178B justification on AC 20-115D 93](#_Toc177112244)

[13 Project Schedule 94](#_Toc177112245)

[13.1 Project Milestones 94](#_Toc177112246)

[13.2 Meetings 94](#_Toc177112247)

[14 PSAC Compliance to DO-178B 96](#_Toc177112248)

**LIST OF TABLES**

[**Table 1: References** 17](#_Toc177112249)

[**Table 2: Acronyms and Definitions** 18](#_Toc177112250)

[**Table 3: Terms and Definitions** 20](#_Toc177112251)

[**Table 4: Software binaries, DAL and Responsibilities** 28](#_Toc177112252)

[**Table 5: Preliminary Failure Hazard Analysis** 37](#_Toc177112253)

[**Table 6: Roles and Responsibilities and Official Communications Contacts** 41](#_Toc177112254)

[**Table 7: Software Planning Process Data** 66](#_Toc177112255)

[**Table 8: Software Development Process Data** 72](#_Toc177112256)

[**Table 9: Software Verification, Software Configuration Management and Software Quality Assurance Process Data** 76](#_Toc177112257)

[**Table 10: Certification Liaison Process Data** 85](#_Toc177112258)

[**Table 11: Tool Qualification** 87](#_Toc177112259)

[**Table 12: List pf Previously Developed Software** 89](#_Toc177112260)

[**Table 13: Options Selectable SW - Aircraft Parameters** 91](#_Toc177112261)

[**Table 14: Option Selectable Software –Application Software** 92](#_Toc177112262)

[**Table 15 DO178B justification on AC 20-115D** 93](#_Toc177112263)

[**Table 16: Project Milestones** 94](#_Toc177112264)

[**Table 17: PSAC Compliance to DO-178B** 96](#_Toc177112265)

LIST OF FIGURES

[**Figure 1: Functional block diagram – Engine Instrument System** 24](#_Toc173969776)

[**Figure 2: Native System Block Diagram** 24](#_Toc173969777)

[**Figure 3: Mixed System Block Diagram** 25](#_Toc173969778)

[**Figure 4: EIS Architecture Block Diagram** 27](#_Toc173969779)

[**Figure 5: Functional block diagram of EDAU and CMU+** 35](#_Toc173969780)

[**Figure 6: Organizational Chart Depicting Responsibilities and Independence** 40](#_Toc173969781)

# Objective

The objective of this document is to propose the Software Life Cycle Processes to the Certification Authority for agreement and defines the means of complying with RTCA DO-178B DAL A guidelines for Engine Data Acquisition Unit (EDAU) and Configuration Management Unit plus NVM (CMU+).

# Document Scope

The scope of this document is intended to apply only to the software assurance aspects related to Engine Data Acquisition Unit (EDAU) and Configuration Management Unit plus NVM (CMU+).

# References

**Table *1*** identifies Document numbers of the documents that are referenced by Title or Document number within the text of this Plan for Software Aspects of Certification.

**Table 1: References**

| **Source** | Document No. | **Title** |
| --- | --- | --- |
| RTCA | DO-178B | Software Considerations in Airborne Systems and Equipment Certification |
| FAA | FAA ORDER 8110.49A | FAA Order – Software Approval Guidelines |
| FAA | AC 20-115D | Airborne Software Development Assurance Using EUROCAE ED-12( ) and RTCA DO-178( ) |
| ARINC | July 14, 2011 | GENERAL STANDARDIZATION OF CAN (CONTROLLER AREA NETWORK) BUS PROTOCOL FOR AIRBORNE USE (ARINC 825-2) |
| **ALTEN Global Technologies Private Limited** | H398-001-002 | Software Development Plan |
| **ALTEN Global Technologies Private Limited** | H398-001-003 | Software Verification Plan |
| **ALTEN Global Technologies Private Limited** | H398-001-004 | Software Configuration Management Plan |
| **ALTEN Global Technologies Private Limited** | H398-001-005 | Software Quality Assurance Plan |

# Acronyms, Terms and Definitions

**Table *2*** describes Acronyms and their Definitions.

**Table 2: Acronyms and Definitions**

|  |  |
| --- | --- |
| **Acronyms** | **Definitions** |
| ALTEN GT | **ALTEN Global Technologies Private Limited**. |
| AC | Alternating Current |
| ACD | Aircraft Configuration Data |
| ADC | Air Data Computer |
| ARINC | Aeronautical Radio Incorporated |
| ASM | Assembly Language |
| BATT | Battery |
| BIT | Built-In Test |
| C | C Software Language |
| CA | Coverage Analysis |
| CAN | Controller Area Network |
| CAS | Crew Alert System |
| CBIT | Continuous Built-In Test |
| CH | Channel |
| CM | Configuration Management |
| CMS | Configuration Management System |
| CMU+ | Configuration Management Unit plus NVM |
| CPU | Central Processing Unit |
| CodeTrax | Coding standards verification Tool |
| COTS | Commercial Off-the-Shelf |
| CRC | Cyclic Redundancy Check |
| CSC | Computer Software Component |
| CVR | Cockpit Voice Recorder |
| DAL | Design Assurance Level |
| DAI | Direct Analog Interface |
| DER | Designated Engineering Representative |
| DLU | Data Logging Unit |
| DU | Display Unit |
| ECU | Engine Control Unit |
| EDAU | Engine Data Acquisition Unit |
| EIS | Engine Instrument System |
| EIU | Engine Interface Unit |
| FAA | Federal Aviation Administration |
| GSW | Ground Support Software |
| Howell | Howell Instruments, Inc. |
| HSI | Hardware-Software Integration |
| HSIT | Hardware-Software Integration Testing / Software High Level Requirements Based Testing |
| I/O | Input/output |
| IDAL | Item Development Assurance Level |
| LED | Light Emitting Diode |
| LLR | Low Level Requirements |
| LLRT | Low Level Requirements based Testing / Software Low Level Requirements Based Testing |
| LRU | Line Replaceable Unit |
| MCD | Module Configuration Data |
| NVM | Non-Volatile Memory |
| PEND | Pending |
| POST | Posting |
| PSAC | Plan for Software Aspects of Certification |
| QA | Quality Assurance |
| RPM | Revolutions Per Minute |
| RTCA | Radio Technical Commission for Aeronautics |
| RTRT | Rational Test Real Time Unit Test and Code Coverage Tool |
| SOI | Stage of Involvement |
| SQA | Software Quality Assurance |
| SRS | Software Requirements Specification / Software High Level Requirements Specification |
| UMS | User Modifiable Software |
| USA | United States of America |
| USB | Universal Serial Bus |

**Table *3*** describes Terms, and their Definitions are explained.

**Table 3: Terms and Definitions**

|  |  |
| --- | --- |
| **Terms** | **Definition** |
| Module | A hardware device (the Engine Interface Unit Board, for example) that is controlled by a microprocessor. At a minimum, a Module contains Application Software which directs the operation of the module. A Module may also include a Boot Loader, Software Loader and Module Configuration Data. |
| Application Software (AS) | Software that runs on a Module that controls the primary functions of the Module. Because its functions can be enabled, disabled, or controlled by specified configuration data, AS may also be considered "Option Selectable Software" by DO-178B. AS is also referred to as Flight Software. |
| Software Loader (SL) | Software that runs on the Module that performs secondary functions, such as calibration, for the Module. SL does not run at the same time as AS. Although it is present in the Module during flight, it is inactive, and cannot be activated during flight. SL may also be considered "Option Selectable Software." |
| Boot Loader (BL) | Software that runs when power is applied to a Module and selects, based on some criteria (for example, USB Flash Drive is present), which software, either the AS or SL (but not both), should control the Module during the current power cycle. A Boot Loader is required in a Module when SL is included in the Module, otherwise, it is optional (for future use) and will only select the Application Software. |
| Boot Up Config File | The Boot Loader software will get the possible number of programs the bootloader config file which allows boot software to jump, with the corresponding conditions (e.g.: CAN message or switch to be active) and the sector number with the size of the applications for the CRC checks before jumping to the application.  The boot-up configuration also contains heartbeat LED pins and ports with the error log sector number and size to write runtime errors in case of an application CRC fail. |
| Module Configuration Data (MCD) | Parametric data used by AS or SL (or both) to initialize and control the operation of the software. For example, MCD may contain a lookup table data used to control scaling for a parameter or a Boolean value which enables or disables a feature of the software. MCD is stored at a known location in flash memory and contains only data (no executable code). MCD specifies the options in "Option Selectable Software" (from DO-178B). |
| Engine Data Acquisition Unit (EDAU) | This unit contains two Engine Interface Unit (EIU) Modules. |
| Configuration Management Unit plus NVM (CMU+) | This unit contains two Configuration Management units. |
| Aircraft Configuration Data (ACD) | Configuration data which is specific to an Aircraft, such as: Aircraft Serial Number, Number of Engines, Engine Type(s), Engine Serial Number(s), etc. This data is specified in the UAS specification. According to DO-178B, ACD is considered "User Modifiable Software." |
| Ground Support Software (GSW) | Software that runs on a Windows based computer which is used to create ACD and transfer ACD to / from the CMU+. |
| Electronic Medium | Information on the computer disk. |

# Document Control

This document will be controlled in accordance with Software Configuration Management Plan (H398-001-004).

# Program Responsibilities

This program is jointly being developed by Howell Instruments, Inc., USA and **ALTEN Global Technologies Private Limited**, Bengaluru, India.

Here after **ALTEN Global Technologies Private Limited** will be referred as ALTEN GT and Howell Instruments, Inc. will be referred as Howell.

ALTEN GT will supply the documents required for FAA Certification to Howell. ALTEN GT will ensure that all data complies with DO-178B. All project data will be available in Configuration Management System (CMS) of ALTEN GT and transferred to Howell. All data required for SOI audits will be available at Howell.

# System Overview

Howell’s Engine Instrument System (EIS) will provide an engine instrument system upgrade for various variants of helicopters. Engine Instrument System will consist of one of two architectural configurations, the Native System and the Mixed System.

The Native System architecture will consist of one Howell Instruments H398 Data Acquisition Unit (DAU), one Howell Instruments H698 Configuration Management Unit (CMU+), and two Howell Instruments H420 Display Units as shown in **Figure 2: Native System Block Diagram**.

Mixed System Architecture will consist of one Howell Instruments Data Acquisition Unit (DAU), one Howell Instruments Configuration Management Unit (CMU+), and one Howell Instruments Display Unit, and one additional Display Unit as shown in **Figure *3*: Mixed System Block Diagram**. The non-Howell DU will be defined relative to its interface.

It will be complemented by pilot input devices and interconnecting harness. The EIS will interface with the aircraft engine sensors and other applicable aircraft systems.

Software for Engine Interface Unit (EIU) and Configuration Management Unit plus NVM (CMU+) are required to be designed and developed by ALTEN GT. Data Logging Unit (DLU) is to be designed and developed by Howell. Hardware Safety features are identified in Section 9.2.

Hardware Part Number for EDAU will be a series where the series part number will be H398B-[XXX], where XXX is aircraft configuration.  Note: Hardware Part Number is H398B-001 for EDAU and H698A-002 for CMU+.

**Note:** Subsequent changes to Hardware part numbers of H398 series will be captured under Software Change Impact Analysis document (H398-004-104)

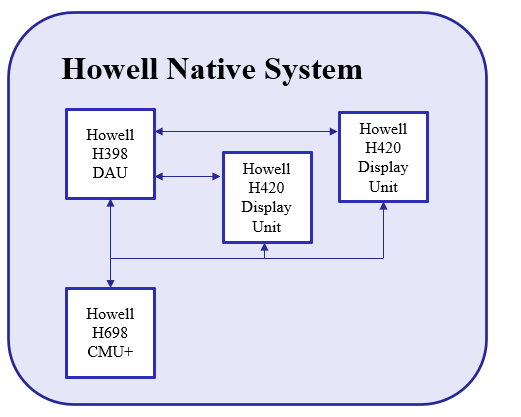
The processor used by all EDAU boards and CMU+ is STMicroelectronics STM32F407IGT6 32-bit embedded microcontroller.

**Figure *1*** describes the functional blocks of Engine Instrument System (EIS).

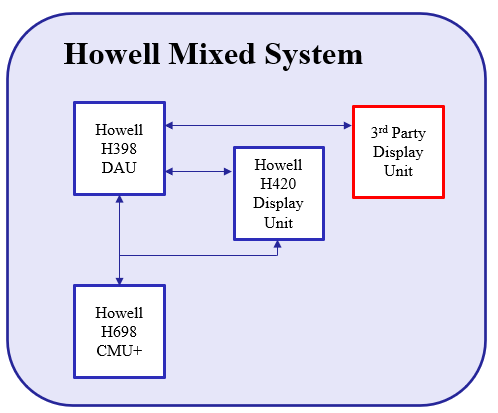
Diagram

Description automatically generated

**Figure 1: Functional block diagram – Engine Instrument System**



**Figure 2: Native System Block Diagram**



**Figure 3: Mixed System Block Diagram**

Overview of EDAU and CMU+ is described in section 7.1 and 7.2.

## Engine Data Acquisition Unit (EDAU)

The EDAU shall function to process airframe and engine data and transmit this information to be graphically presented on a cockpit display. The EDAU will process analog and discrete inputs, apply user defined logic to those inputs, and may generate aural and visual Caution and Warning alerts.

The EDAU is utilized to perform the following functions:

1. Provide analog to digital conversion of aircraft and engine systems
2. Provide discrete bit information via A429 for Crew Alerts (CAS)
3. Provide field loadable software update via maintenance port
4. Process data
5. Filter analog and digital signal data
6. Receive/Transmit data
7. Provide discrete strapping for multiple aircraft and engine configurations

To increase the reliability of the system, EDAU will have two Engine Interface Units (EIU) packed together. Both the Engine Interface Units (EIU) will run the data acquisition application software, process the same inputs and produce same outputs independently at the same time.

The primary function of the EIU will be acquiring analog engine parameters, discrete inputs, and other aircraft system information for processing and converting to a digital format for display via ARINC 429 to two Display Units (DU).

One EIU will act as a primary unit and the other EIU will act as a secondary unit. Each unit will operate independent from the other including the power source in an effort to mitigate single point failure within the system. The excitation of the analog sensors will be carried out by the excited EIU side, based on excitation condition.

Each EIU will contain three Boards/Modules, each with its own CPU: The Gateway Board (BH35112) for digital buses, the Analog Board (BH35113) for analog inputs and the Discrete Board (BH35114) for discrete I/O and chip detection. Inter-board communication and communication with CMU+ / DLU will be on CAN bus based on ARINC 825 protocol.

The EDAU will be designed to have the following features based on aircraft variant.

* RS-422 inputs/outputs
* Bi-directional RS-232 serial data bus
* ARINC 429 receive/transmit functionalities.
* Programmable discrete inputs.
* Configuration strapping discrete inputs for selection aircraft engines.
* Ground/open discrete inputs/outputs
* 28V/Open discrete inputs/ and outputs.
* Ground/open discrete outputs.
* Audio output
* Analog inputs and thermocouple inputs.
* Analog output.
* AC pressure transducer inputs.
* AC synchro inputs.
* DC differential voltage pressure transducer inputs.
* 2-wire differential millivolt inputs.
* Single-wire voltage inputs.
* Two-wire bulb inputs.
* 2-wire tachometer inputs.

This architecture will be designed such that it ensures a failure in the EDAU, an aircraft sensor, or an engine sensor cannot disable any other parameters for a particular system.

## Configuration Management Unit Plus NVM (CMU+)

The CMU+ stores Aircraft Configuration Data (ACD). The ACD includes options related to the engine parameters, sensor interface, exceedance parameters, and airframe configurations. The EDAU accesses CMU+ on system initialization to determine the current configuration and active interfaces.

The CMU+ is utilized to perform the following functions:

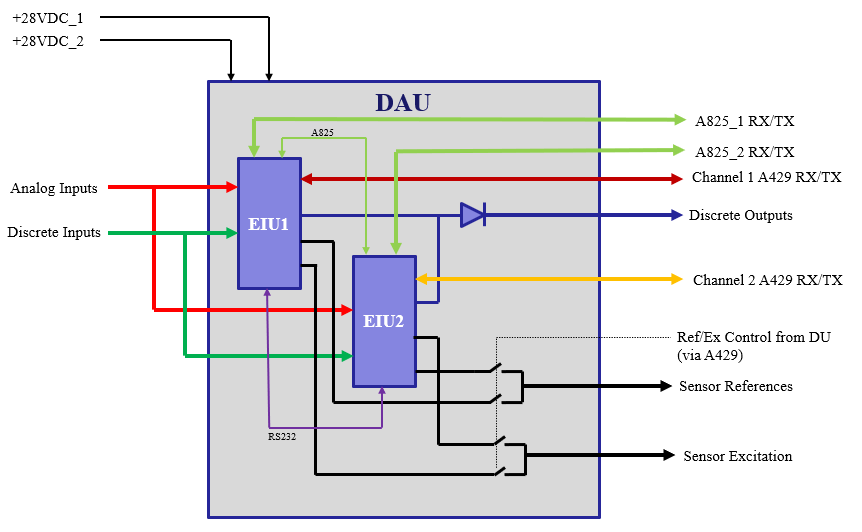
1. Provide configuration data to each EIU utilizing separate, isolated hardware for each EIU
2. Support the EDAU system configuration so that a replacement EIU will function the same as the EIU being replaced after installation and initialization
3. The EDAU shall provide the capability to support different engine and aircraft configurations
4. The CMU+ shall store item configuration data for entire EIS
5. The CMU+ shall contain system NVM

The CMU+ is internally redundant with two separate configuration sections packaged into one enclosure (separate power supplies). One section CMU+ #1 is connected to EIU#1 through ARINC825 (CAN #1), and the other section CMU+ #2 is connected to EIU#2 through ARINC825 (CAN #1). Internally CMU+ #1 and CMU+ #2 are connected through an independent ARINC825 (CAN #2).

The CMU+ installed in position #1 and #2 will have a USB device interface for programming/accessing configuration information in Loader Mode.

## Functional block diagram of Engine Data Acquisition System (EDAU) and CMU+

**Figure *4*** depicts the EIS Architecture Block Diagram and their I/O interfaces.

**

**Figure 4: EIS Architecture Block Diagram**

# Software Overview

Software Overview section describes the software aspects of Engine Data Acquisition Unit (EDAU) and Configuration Management Unit plus NVM (CMU+)

**Table *4*** lists the Software Binaries and their corresponding development responsibilities for Engine Interface Unit (EIU) of Engine Data Acquisition Unit (EDAU) and Configuration Management Unit plus NVM (CMU+)

| **Table 4: Software binaries, DAL and Responsibilities** | | |
| --- | --- | --- |
| **Software Binary** | **DO-178B DAL** | **Responsibility** |
| EDAU – Gateway Module Application | | |
| Application Software | A | ALTEN GT |
| Module Configuration Data | A | ALTEN GT |
| Boot Loader | A | Howell |
| Boot-Up Config file | A | Howell |
| Calibration Software | E | Howell |
| Software Loader | E | Howell |
| EDAU – Discrete Module Application | | |
| Application Software | A | ALTEN GT |
| Module Configuration Data | A | ALTEN GT |
| Boot Loader | A | Howell |
| Boot-Up Config file | A | Howell |
| Software Loader | E | Howell |
| EDAU – Analog Module Application | | |
| Application Software | A | ALTEN GT |
| Module Configuration Data | A | ALTEN GT |
| Boot Loader | A | Howell |
| Boot-Up Config file | A | Howell |
| Calibration Software | E | Howell |
| Software Loader | E | Howell |
| Configuration Management Unit Plus NVM (CMU+) | | |
| Application Software | A | ALTEN GT |
| Boot Loader | A | Howell |
| Boot-Up Config file | A | Howell |
| Software Loader | E | Howell |
| Maintenance Application | E | Howell |
| DL Application Software | E | Howell |
| DL Application Configuration | E | Howell |
| Aircraft Configuration Data (ACD) | A | ALTEN GT |

**Note:** Subsequent changes to Software binaries along with Aircraft configuration data, DAL, and Responsibilities of H398 series will be captured under Software Change Impact Analysis document (H398-004-104).

## Engine Data Acquisition Unit (EDAU)

The EDAU is data conversion unit capable of processing input data from engine and various aircraft sensors and providing data to the displays and discrete, analog, audio outputs. EDAU consists of two Engine Interface Units (EIU). Each Engine Interface Unit Module will have identical Software running on it.

The EIU shall be utilized to perform the following functions:

* Provide analog to digital conversion of aircraft and engine systems
* Provide discrete bit information via A429 for Crew Alerts (CAS)
* Provide field loadable software update via maintenance port
* Process data
* Filter analog and digital signal data
* Receive/Transmit data
* Provide discrete strapping for multiple aircraft and engine configurations

Each Engine Interface Unit (EIU) consists of Gateway, Analog and Discrete Modules.

### Gateway Module

Software Components of Gateway Module are explained in section 8.1.1.1 through 8.1.1.6.

#### Boot Loader

The Bootloader's responsibility is to jump to different applications based on the jump condition's specified in the boot configuration file. Before jumping to a application the bootloader will do a CRC check to verify the application's integrity. In the event of CRC failure the bootloader will jump to a default application mentioned on the boot-up config file(Refer section 12.5).

#### Boot-Up Config file

Boot-Up configuration file provides the possible number of applications for the bootloader to jump. The configuration file will also provide the memory size and sector number associated with the applications and their jump conditions. The boot-up configuration file also contains the error log memory details and a default jump application memory detail.

#### Application Software

The Gateway section is responsible for EIU coordination and communication with the offside EIU, communication with the external subsystems, performing calculations, scaling, and functionality specific to the aircraft configuration.

Gateway Module processes following input data:

1. A429 Input Data from DU’s
2. RS232 Serial Input Data
3. Channel-to-channel data link via ARINC 825

Gateway Module processes following output data:

1. ARINC 429 Output to the Display units.
2. ARINC 825 to the Analog, Discrete, Data Logger and Configuration Module
3. RS-232 Serial Data Output.
4. RS-422 Serial Data Output
5. Processing of Audio Outputs

#### Module Configuration Data

Refer Section 8.5.1.

#### Software Loader

Software loader application provides a facility to do software update via ARINC 825.This module consist of a windows application and a embedded application. The windows application acts as a server and the embedded acts as a client. Based on the binary file transferred from the windows application the embedded application will do the software update.

#### Calibration Software

Analog signals are prone to gain and offset errors. Calibration is the process of correcting these errors. In the EDAU this is done by the Calibration Software. This software consists of a Calibration Windows Application and a Calibration Application s19 binary. The Windows Application acts as a server and the s19 binary acts as a client. Together, these 2 software calculate the gain and offset values for each analog signal. The Calibration Windows Application then uses this set of gain and offset values to create a Calibration Data Table s19 binary. This Calibration Data Table binary is then loaded into the unit. The Flight Application Software then uses this binary to process analog signals.

### Discrete Module

Discrete section will be capable of processing discrete inputs from various engine and aircraft subsystems. The Alarm caution and warning system shall provide signals to the DU to display various cautions and warnings. Alarm functions shall clear when the alarm condition is no longer present.

Software Components of Discrete Module are explained in section 8.1.2.1 through 8.1.2.5.

#### Boot Loader

The Bootloader's responsibility is to jump to different applications based on the jump condition's specified in the boot configuration file. Before jumping to a application the bootloader will do a CRC check to verify the application's integrity. In the event of CRC failure the bootloader will jump to a default application mentioned on the boot-up config file(Refer section 12.5).

#### Boot-Up Config file

Boot-Up configuration file provides the possible number of applications for the bootloader to jump. The configuration file will also provide the memory size and sector number associated with the applications and their jump conditions. The boot-up configuration file also contains the error log memory details and a default jump application memory detail.

#### Application Software

Discrete Module processes different types of discrete input and output data. Depending on the aircraft the discrete inputs & outputs might vary.

#### Module Configuration Data

Refer Section 8.5.1.

#### Software Loader

Software loader application provides a facility to do software update via ARINC 825.This module consist of a windows application and a embedded application. The windows application acts as a server and the embedded acts as a client. Based on the binary file transferred from the windows application the embedded application will do the software update.

### Analog Module

Analog section will be capable of acquiring signals typically present in aircraft operation including but not limited to torque, engine power turbine speed, engine gas generator speed, turbine outlet temperature, oil temperatures, oil pressures, fuel flow, fuel quantity, and outside air temperature.

Software Components of Analog Module are explained in section 8.1.3.1 through 8.1.3.6.

#### Boot Loader

The Bootloader's responsibility is to jump to different applications based on the jump condition's specified in the boot configuration file. Before jumping to a application the bootloader will do a CRC check to verify the application's integrity. In the event of CRC failure the bootloader will jump to a default application mentioned on the boot-up config file (Refer section 12.5).

#### Boot-Up Config file

Boot-Up configuration file provides the possible number of applications for the bootloader to jump. The configuration file will also provide the memory size and sector number associated with the applications and their jump conditions. The boot-up configuration file also contains the error log memory details and a default jump application memory detail.

#### Application Software

Analog Module processes following different kind of analog input and output data. Depending on the aircraft the analog inputs & outputs might vary.

#### Module Configuration Data

Refer Section 8.5.1.

#### Software Loader

Software loader application provides a facility to do software update via ARINC 825.This module consist of a windows application and a embedded application. The windows application acts as a server and the embedded acts as a client. Based on the binary file transferred from the windows application the embedded application will do the software update.

#### Calibration Software

Analog signals are prone to gain and offset errors. Calibration is the process of correcting these errors. In the EDAU this is done by the Calibration Software. This software consists of a Calibration Windows Application and a Calibration Application s19 binary. The Windows Application acts as a server and the s19 binary acts as a client. Together, these 2 software calculate the gain and offset values for each analog signal. The Calibration Windows Application then uses this set of gain and offset values to create a Calibration Data Table s19 binary. This Calibration Data Table binary is then loaded into the unit. The Flight Application Software then uses this binary to process analog signals.

## Configuration Management Unit Plus NVM (CMU+)

Software Components of Configuration Management Unit are explained in section 8.2.1 through 8.2.7.

### Boot Loader

The Bootloader's responsibility is to jump to different applications based on the jump condition's specified in the boot configuration file. Before jumping to a application the bootloader will do a CRC check to verify the application's integrity. In the event of CRC failure the bootloader will jump to a default application mentioned on the boot-up config file (Refer section 12.5).

### Boot-Up Config file

Boot-Up configuration file provides the possible number of applications for the bootloader to jump. The configuration file will also provide the memory size and sector number associated with the applications and their jump conditions. The boot-up configuration file also contains the error log memory details and a default jump application memory detail.

### Application Software

The CMU+ is responsible for transferring the Aircraft Configuration Data to both EIUs.

The EDAU shall support an external configuration management unit that will store these parameters related to system/equipment variations as well as engine parameters. Upon power up, the EIU CPU shall access this configuration management unit to determine the initialization configuration and active interfaces. The configuration management unit shall be programmed by a means of an attached PC using a USB port.

### Software Loader

Software loader application provides a facility to do software update via USB. This module consist of a windows application and a embedded application. The windows application acts as a server and the embedded acts as a client. Based on the binary file transferred from the windows application the embedded application will do the software update.

### Maintenance Application

Maintenance application software is used to send Aircraft Configuration Data (ACD) to the CMU+. Howell will be developing this Software.

### DL Application Software

DL flight application software is used to log the data. Howell will be developing this Software.

### DL Application Configuration

DL application configuration file used by DL flight application software. Howell will be developing this Software.

## Scheduler

The EDAU Application Software and CMU+ Application Software use a kernel to create and schedule jobs. The kernel has a fixed priority pre-emptive scheduler. The scheduler ensures that at any given time, the processor executes the highest priority task of all those tasks that are currently ready to execute. The pre-emptive scheduler has a clock interrupt task that provides the scheduler with an option to switch after the task has had a given period to execute—the time slice.

The kernel uses semaphores to synchronize the tasks. Three operations can be performed on a semaphore: Creation, Pending, and Posting. Each task has to run periodically. A unique semaphore is created for every task. A task desiring to run will perform a PEND on its corresponding semaphore. The scheduler is responsible for releasing the semaphores at correct intervals so that the tasks execute at the required frequency. The scheduler releases the semaphore for a task by doing a POST on the semaphore. Once the semaphore is released, the task is made to run. The task then executes its set of instructions and makes the next PEND. It then waits for a POST from the scheduler. This process goes indefinitely.

## Functional block diagram of EDAU and CMU+ Software

**Figure *5*** depicts the functional blocks of EDAU and CMU+ and their I/O interfaces.



ARINC825

28VPower

**Figure 5: Functional block diagram of EDAU and CMU+**

## Configuration Data

Configuration data are explained in section 8.5.1 through 8.5.2.

### Module Configuration Data

The EIU Software will use a Module Configuration Data binary to make itself configurable. This is done so that EIU Software can support different input signals and be able to format output data as per the characteristic of the connected display unit without having to modify itself. Module Configuration Data allows defining the following:

1. Configure the interfaces
2. Enable/Disable specific input and output pins
3. Specify the characteristics of inputs/outputs like scaling, range, default value, range error default value, de-bounce time and persistence period
4. Threshold values
5. Fault confirmation times.

### Aircraft Configuration Data

Aircraft has multiple systems/equipment variations as well as engine parameter differences. The CMU+ Software will have an Aircraft Configuration Data Binary to support these different aircraft and engine configurations. The Aircraft Configuration Data will be defined by Aircraft Manufacturer and will configure the following items:

1. Aircraft Serial Number
2. Aircraft Registration
3. Characteristics of the Engines installed on the Aircraft
4. Installation status of Subsystems
5. Constants and correction factors required to calculate Engine related parameters

## Language Used for Software Development

The C language from ISO/IEC 9899:1999, version C99 will be used as the programming language to implement the software for EDAU and CMU+. The Cortex-M4 assembly Thumb2 instructions will be used for hardware intensive and processor related functions.

# Certification Considerations

The EDAU and CMU+ are part of Engine Instrument System (EIS). The EIS is a potential contributor to a hazardous failure condition. **Table *5*** provides Preliminary Failure Hazard Analysis. This project has been determined to require DAL A project development controls. EIS is a future product series, hence, to meet the need for future requirements Howell Instruments, Inc. has chosen to develop the software to a catastrophic failure condition.

Refer **Table *4*** for the Design Assurance Level of the Software Binaries.

## List of Hazardous Events

**Table *5*** lists the Preliminary Failure Hazard Analysis.

**Table 5: Preliminary Failure Hazard Analysis**

| **Hazard** | **Classification of Failure Condition (FDAL)** | **Hardware Quantitative Probability (IDAL)** | **Risk Mitigation** | **Failure Condition Effect** |
| --- | --- | --- | --- | --- |
| Loss of EDAU rotor RPM function (Nr) | Hazardous/ Severe Major | <10-7 | Failure condition detected will be recorded in the fault log and the appropriate display annunciation will be activated. | Review pilot action |
| Erroneous EDAU rotor RPM function w/o detection | Hazardous/ Severe Major | <10-7 | Failure condition detected will be recorded in the fault log and the appropriate display annunciation will be activated. | Review pilot action |
| Erroneous temperature functions of either channel of the EDAU | Hazardous/ Severe Major | <10-7 | Failure condition detected will be recorded in the fault log and the appropriate display annunciation will be activated. | Review pilot action |
| Configuration data mismatch causes Misleading Primary Engine parameters | Hazardous/ Severe Major | <10-7 | Failure condition detected will be recorded in the fault log and the appropriate display annunciation will be activated. | Review pilot action |
| EDAU channel failed to provide cross-side engine data | Hazardous/ Severe Major | <10-7 | Failure condition detected will be recorded in the fault log and the appropriate display annunciation will be activated. | Review pilot action |
| EDAU channel failed to provide engine data | Hazardous/ Severe Major | <10-7 | Failure condition detected will be recorded in the fault log and the appropriate display annunciation will be activated. | Review pilot action |
| EDAU channel provides erroneous high fuel quantity for all on-side tanks | Hazardous/ Severe Major | <10-7 | Failure condition detected will be recorded in the fault log and the appropriate display annunciation will be activated. | Review pilot action |
| EDAU channel provides erroneous low fuel quantity for all on-side tanks | Hazardous/ Severe Major | <10-7 | Failure condition detected will be recorded in the fault log and the appropriate display annunciation will be activated. | Review pilot action |
| Loss of EDAU detection of an unreasonable OAT data | Hazardous/ Severe Major | <10-7 | Failure condition detected will be recorded in the fault log and the appropriate display annunciation will be activated. | Review pilot action |
| The exceedance is generated by the EDAU | Hazardous/ Severe Major | <10-7 | Failure condition detected will be recorded in the fault log and the appropriate display annunciation will be activated. | Review pilot action |

**Note**: Subsequent changes to Preliminary Failure Hazard Analysis of H398 series will be captured under Software Change Impact Analysis document (H398-004-104)

## Hardware Safety Features

In Normal Mode, the EDAU and CMU+ software regularly restarts the watchdog timer to prevent the watchdog timer from timing out. Any failure to restart the watchdog timer indicates a hardware fault or a program error. The timing out of the watchdog timer causes the processor to reset. After reset, a Power-On BIT is carried out which will try to identify the fault and take necessary corrective action.

# Software Life Cycle

Organization roles and responsibilities are described in section 10.1. Software Life Cycle Processes are described in section 10.2 through 10.3.

## Organization – Roles and Responsibilities

ALTEN GT and Howell will work as an integrated project team to manage the development of the Software for EDAU and CMU+.

**Figure *6*** depicts the Organizational Responsibilities and Independence.



**Figure 6: Organizational Chart Depicting Responsibilities and Independence**

**Table *6*** describes the roles, responsibilities and contact details of all the relevant persons involved in this program.

**Table 6: Roles and Responsibilities and Official Communications Contacts**

| **Name** | **Contact** | **Role In this Program** | **Organization** |
| --- | --- | --- | --- |
| Vasanthi TP | [vasanthi.panduranga@alten.com](mailto:vasanthi.panduranga@alten.com)  +91 80 25350105/45463000  Extn:4011 | Director of Engineering – Responsible for the overall Program Management. | **ALTEN Global Technologies Private Limited** |
| Madhu Chandra S | [madhuchandra.srinivas@alten.com](mailto:mailtomadhuchandra.srinivas@alten.com) +91 80 25350105/45463000 Extn:4018 | Associate Director - Responsible for the Software Creation of DO-178B Data items and Responsible for Software Development and Verification Activities. | **ALTEN Global Technologies Private Limited** |
| Nabirasul Herkal | [nabirasul.herkal@alten.com](mailto:nabirasul.herkal@alten.com)  +9180 25350105/45463000 | Technical Manager - Responsible for Software Requirements, Design, Implementation and Software Verification | **ALTEN Global Technologies Private Limited** |
| Shyamala B | [shyamala.bappanadu@alten.com](mailto:shyamala.bappanadu@alten.com) +91 80 2535 0105/45463000  Ext: 1009 | Senior Director - Quality Assurance  Responsible for the Quality Assurance of DO-178B | **ALTEN Global Technologies Private Limited** |
| Kevin Crozier | [kevin@ktronicsaero.com](mailto:kevin@ktronicsaero.com)  (408) 910-1191 | Designated Engineering Representative (DER) | KTronics Aero Services |
| Michael James | [mjames@howellinst.com](mailto:mjames@howellinst.com)  (817) 568-5855 | Director of Programs – Responsible for the overall Program Management. | Howell Instruments, Inc. |
| Jackie Barnes | [JBarnes@howellinst.com](mailto:JBarnes@howellinst.com)  (817) 568 – 5818 | Quality Assurance Manager – Responsible for Quality Assurance and Certification Liaison with FAA | Howell Instruments, Inc. |
| Kamal Ramakrishnan | [KRamakrishnan@howellinst.com](mailto:KRamakrishnan@howellinst.com)  (817) 568-5848 | Director of Engineering – Responsible for all Software Related Activities. | Howell Instruments, Inc. |
| Sarath Chandar Ramamurthy | [SRamamurthy@howellinst.com](mailto:SRamamurthy@howellinst.com)  (817)568-5803 | Project Lead - Responsible for all Software Related Activities. | Howell Instruments, Inc. |
| Shayanne Guijosa | [SGuijosa@howellinst.com](mailto:SGuijosa@howellinst.com)  817-568-5846 | Electrical Engineer | Howell Instruments, Inc. |

### **ALTEN Global Technologies Private Limited**

ALTEN GT is responsible for the Software High Level Requirements, Software Design, Software Implementation and Software Verification of EDAU and CMU+ Modules.

ALTEN GT is responsible for the generation of Software Life Cycle Data items as mentioned in Section 11 for showing compliance to DO-178B objectives for all DAL A software.

ALTEN GT’s Quality Assurance is responsible to carry out audits as defined in Software Quality Assurance Plan (H398-001-005). Quality function is an independent function directly reporting to the Chief Executive Officer of ALTEN GT.

ALTEN GT’s Technical Manager tracks and oversees the project and ensures proper co-ordination across various functions. The Engineering Manager serves as the day-to-day contact point and is responsible for ensuring all technical and program milestones are met.

An Independent Verification and Validation team is responsible for carrying out the Verification and Validation activities.

### Howell Instruments, Inc.

Howell is responsible for overall program management, oversight of ALTEN GT, and Certification Liaison activities, as well as software design, development, and certification activities for software components other than DAL A.

### KTronics Aero Services.

Kevin Crozier has been contracted to perform Designated Engineering Representative duties for this project. With Certification Authority approval, he will conduct SOI audits 1 - 4 and will provide data documenting his review of, and recommendation of approval for, FAA required certification data.

## Software Life Cycle Model

The Software Life Cycle is based on Modified Waterfall Model. The major phases of the Software Life Cycle Model are Software Planning, Software Requirements, Software Architectural Design, Software Low Level Requirements, Software Coding and Integration and Software Verification. A new phase is entered after meeting the entry criteria for that phase, as defined in Section 10.3. Any phase can be re-entered if a Change Request or Problem Report warrants such a re-entry. Details of the Software Development Processes, Inputs, Outputs, and means of compliance to DO-178B objectives are described in the Software Development Plan (H398-001-002). Details of the Software Verification Processes, Inputs, Outputs, and means of compliance to DO-178B objectives are described in the Software Verification Plan (H398-001-003).

Software Life Cycle Process is entered once the Engine Instrument System Requirements Specification from Aircraft manufacturer is available.

## Software Life Cycle Processes

This section discusses the Software Life Cycle Processes and transition criteria between Software Life Cycle Processes.

The Software Life Cycle Processes are the Software Planning Process, the Software Development Process and the Integral Process. The Software Planning Process defines and coordinates the activities of the Software Development and Integral processes for this project. The Software Development Process produces the Software Product. The Integral Processes ensure the correctness, control, and confidence of the Software Life Cycle Processes and their outputs.

The Integral Process activities carried out for each of the Software Life Cycle Processes are Configuration Identification and Configuration Controlled as per Software Configuration Management Plan (H398-001-004), Verification of Outputs as per Software Verification Plan (H398-001-003), Quality Assurance activities as per Software Quality Assurance Plan (H398-001-005) and Certification Liaison activities as per Plan for Software Aspects of Certification (H398-001-001).

The Inputs, Activities, Outputs and Transition criteria for each of the processes are defined in Section 10.3.1 through Section 10.3.7.

### Software Planning Process

The purpose of the Software Planning Process is to develop the Software Planning Documents.

### Inputs

The input to this process is Engine Instrument System Requirements Specification from Aircraft manufacturer

### Activities

Following activities are performed during this process:

1. Develop Plan for Software Configuration Management
2. Develop Plan for Software Development
3. Develop Plan for Software Verification and Review Checklist Templates
4. Develop Plan for Software Quality Assurance and Review Checklist Templates
5. Develop Standards for Requirements, Design and Code
6. Develop Qualification Plans for the Tools selected

### Outputs

The Outputs of this process are described in **Table *7*** and under CM as per Software Configuration Management Plan (H398-001-004).

### Transition Criteria

The Software Planning Process is entered when the Engine Instrument System Requirements Specification from Aircraft manufacturer is under CM as per Software Configuration Management Plan (H398-001-004).

The Software Planning Process can be re-entered through a Problem Report after a CCB meeting.

### Software Development Process

The Software Development Plan (H398-001-002) describes the activities, standards and processes used in the Software Development Process of the Engine Instrument System program. The Software Development Plan (H398-001-002) describes the Software Development approach for the Software Requirements, Software Architectural Design, Software Low Level Requirements, Coding and Integration and the Tools used for the Software Development and the Software Life Cycle Processes.

The Software Development Process is as per the DO-178B guidelines and includes the following processes:

1. Software Requirements Process
2. Software Architectural Design Process
3. Software Low Level Requirements Process
4. Software Coding and Integration Process

### Software Requirements Process

The purpose of the Software Requirements Process is to develop the Software Requirements Specification, which contain Software High Level Requirements, and Software Requirements Traceability Matrix (which bi-directionally maps Software High Level Requirements to Engine Instrument System Requirements Specification from Aircraft manufacturer .

### Inputs

The inputs to this process are Engine Instrument System Requirements Specification from Aircraft manufacturer, Software Development Plan (H398-001-002) and Software Requirements Standards (H398-001-006).

### Activities

Following activities are performed during this process:

1. Define all interfaces as per Software Requirements Standards (H398-001-006)
2. Identify System Safety features
3. Identify the derived requirements. Decisions, taken while identifying the software components, are translated into derived requirements. These derived requirements are not directly traceable to Engine Instrument System Requirements Specification from Aircraft manufacturer. All the derived requirements are provided to the System Safety Assessment Process to analyze their effects on the system safety
4. Generate the Software High Level Requirements from the Engine Instrument System Requirements Specification from Aircraft manufacturer as per Software Requirements Standards (H398-001-006)
5. Create Software Requirements Traceability Matrix (which bi-directionally maps Software High Level Requirements Specification to Engine Instrument System Requirements Specification from Aircraft manufacturer.

### Outputs

The output of this process is described in **Table *8*** (SI No 1) and under CM as per Software Configuration Management Plan (H398-001-004).

### Transition Criteria

The Software Requirements Process is entered when Engine Instrument System Requirements Specification from Aircraft manufacturer, Software Development Plan (H398-001-002) and Software Requirements Standards (H398-001-006) are under CM as per Software Configuration Management Plan (H398-001-004).

The Software Requirement Process can be re-entered through a problem report after a CCB meeting.

### Software Architectural Design Process

The purpose of the Software Architectural Design Process is to develop the Software Architectural Design and Software Architectural Design Traceability Matrix (which bi-directionally maps Software Architectural Design to Software High Level Requirements).

### Inputs

The inputs to this process are Software Requirements Specification, Software Development Plan (H398-001-002) and Software Design Standards (H398-001-007).

### Activities

Following activities are performed during this process:

1. Identify CSCs from Software High Level Requirements as per Software Design Standards (H398-001-007)
2. Document CSC Interfaces as per Software Design Standards (H398-001-007)
3. Identify Global Data Structures as per Software Design Standards (H398-001-007)
4. Create Software Architectural Design as per Software Design Standards (H398-001-007) from the Software High Level Requirements Specification
5. Create Software Architectural Design Traceability Matrix (which bi-directionally maps Software Architectural Design to Software High Level Requirements)

### Outputs

The output of this process is described in **Table *8*** (SI No 2) and under CM as per Software Configuration Management Plan (H398-001-004).

### Transition Criteria

The Software Architectural Process is entered when Software Requirements Specification, Software Development Plan (H398-001-002) and Software Design Standards (H398-001-007) are baselined in CM as per Software Configuration Management Plan (H398-001-004).

The Software Architectural Design Process can be re-entered through a Problem Report after a CCB meeting.

### Software Low Level Requirements Process

The purpose of the Software Low Level Requirements Process is to develop the Software Low Level Requirements and Software Low Level Requirements Traceability Matrix (which bi-directionally maps Software Low Level Requirements to Software High Level Requirements and Software Architectural Design).

### Inputs

The inputs to this process are Software Requirements Specification, Software Architectural Design, Software Development Plan (H398-001-002) and Software Design Standards (H398-001-007).

### Activities

Following activities are performed during this process:

1. Identify all local data structures as per Software Design Standards (H398-001-007)
2. Identify the functions (CSUs) as per Software Architectural Design and Software Design Standards (H398-001-007)
3. Identify the derived requirements. Decisions, taken while identifying the software components, are translated into derived requirements. These derived requirements are not directly traceable to Software High Level Requirements Specification. All the derived requirements are provided to the System Safety Assessment Process to analyze their effects on the system safety
4. Generate Software Low Level Requirements as per Software Design Standards (H398-001-007) from the Software High Level Requirements
5. Create Software Low Level Requirements Traceability Matrix (which bi-directionally maps Software Low Level Requirements to Software High Level Requirements and Software Architectural Design)

### Outputs

The output of this process is described in **Table *8*** (SI No 3) and under CM as per Software Configuration Management Plan (H398-001-004).

### Transition Criteria

The Software Low Level Requirements Process is entered when Software Requirements Specification, Software Architectural Design, Software Development Plan (H398-001-002) and Software Design Standards (H398-001-007) are baselined in CM as per Software Configuration Management Plan (H398-001-004).

The Software Low Level Requirements Process can be re-entered through a Problem Report after a CCB meeting.

### Software Coding and Integration Process

The purpose of the Software Coding and Integration Process is to develop the Software code, Executable code and Software Code Traceability Matrix (which bi-directionally maps Software Code to Software Low Level Requirements).

### Inputs

The inputs to this process are Software Architectural Design, Software Low Level Requirements, Software Development Plan (H398-001-002), Software C Coding Standards (H398-001-008) and Software ASM Coding Standards (H398-001-009).

### Activities

Following activities are performed during this process:

1. Implement Source Code from Software Low Level Requirements in C and ASM as per Software C Coding Standards (H398-001-008), Software ASM Coding Standards (H398-001-009) and Software Development Plan (H398-001-002)
2. Create Build instructions in Software Configuration Index (H398-004-101)
3. Integrate the code to produce the Executable Code as per Build Instructions in Software Configuration Index (H398-004-101)
4. Create Load instructions in Software Configuration Index (H398-004-101)
5. The Executable Code is loaded on the target hardware as per Load Instructions in Software Configuration Index (H398-004-101)
6. Create Software Code Traceability Matrix (which bi-directionally maps Software Code to Software Low Level Requirements)

### Outputs

The outputs of this process are described in **Table *8*** (SI No 4, 5, 6) and under CM as per Software Configuration Management Plan (H398-001-004).

### Transition Criteria

The Software Coding and Integration Process is entered when Software Architectural Design, Software Low Level Requirements, Software Development Plan (H398-001-002), Software C Coding Standards (H398-001-008) and Software ASM Coding Standards (H398-001-009) are baselined in CM as per Software Configuration Management Plan (H398-001-004).

The Software Coding and Integration Process can be re-entered from a Problem Report after a CCB meeting.

### Software Verification Process

The Software Verification Plan (H398-001-003) describes all the Software verification activities for different phases of the Software development life cycle, and establishes the independence required. The Software Verification Plan (H398-001-003) also describes in detail the Software verification approach, the tools used for structural coverage, the Software verification activities at different phases and re-verification criteria. The Software Verification Plan (H398-001-003) explains the process of performing the Hardware Software Integration Testing and Software Low Level Requirements based Testing. The Software Verification Plan (H398-001-003) also describes obtaining the structural coverage during Software Low Level Requirements based Testing.

The Software Verification phase consists of a verification process for each of the Software Development Processes. The Software Verification processes satisfy the DO-178B verification guidelines. These processes are as follows:

1. Verification of the Software Planning Process
2. Verification of the Software Requirements Process
3. Verification of the Software Architectural Design Process
4. Verification of Software Low Level Requirements Process
5. Verification of the Software Coding and Integration Process
6. Testing of Outputs of the Software Coding and Integration Process
7. Verification of Verification Process

The Software Verification processes are carried out using three methods – reviews, analysis and testing. Software Test cases are captured in Software Requirements based Test Plan. The following section establishes the inputs, activities and outputs of each of the software verification processes and the criteria for transitioning from one process to another.

The Review Checklists templates developed in Software Planning Process used for review in each of the Software Verification Processes identify the applicable objectives from DO178 Appendix A.

The Software Verification Processes establishes the required Independence. This will be achieved by ensuring that the person who verifies the work-product is not involved with the development of that work-product.

### Verification of the Software Planning Process

The purpose of the Verification of the Software Planning Process is to verify the outputs of Software Planning Process (Section 10.3.1) satisfy the objectives of DO-178B Table A-1: items 1 to 7 and Table A-10: items 1 and 2. The outputs of Software Planning Process are Software Planning documents, Software Standards, Tool Qualification Plans and Review Checklist Templates that are described in **Table *7***.

### Inputs

The inputs to this process are Software Planning documents, Software Standards, Tool Qualification Plans and Review Checklist Templates that are described in **Table *7*** and Engine Instrument System Requirements Specification from Aircraft manufacturer.

### Activities

Following activities are performed during this process,

1. Document review of Plan for Software Aspects of Certification (H398-001-001)
2. Document review of Software Development Plan (H398-001-002)
3. Document review of Software Verification Plan (H398-001-003) and Review Checklist Templates
4. Document review of Software Configuration Management Plan (H398-001-004)
5. Document review of Software Quality Assurance Plan (H398-001-005) and Review Checklist Templates
6. Document review of Software Requirements Standards (H398-001-006)
7. Document review of Software Design Standards (H398-001-007)
8. Document review of Software C Coding standards (H398-001-008)
9. Document review of Software ASM Coding standards (H398-001-009)
10. Document review of Tool Qualification Plan for CodeTrax (H398-001-011)
11. Document review of Tool Qualification Plan for RTRT (H398-001-010)
12. Review of Outputs of Software Planning Process satisfy the objectives of DO-178B Table A-1: items 1 to 7 and Table A-10: items 1 and 2 by DER and Approval by FAA.

Note: Document review will be performed by ALTEN GT before the Software Planning documents are submitted to DER for review. Outputs generated from Document review by ALTEN GT are mentioned in following section 10.3.3.1.3.

### Outputs

The outputs of this process are described in **Table *9*** (SI No. 8, 10)

### Transition Criteria

The Verification of the Software Planning Process is entered when the Software Planning documents are under CM as per Software Configuration Management Plan (H398-001-004).

The Verification of the Software Planning Process can be re-entered when Software Planning documents are updated due to Problem Report(s).

### Verification of the Software Requirements Process

The purpose of the Verification of the Software Requirements Process is to verify the outputs of Software Requirements Process (Section 10.3.2.1) satisfy the objectives of DO-178B Table A-3: items 1 to 7. The outputs of Software Requirements Process are Software Requirements Specification and Software Requirements Traceability Matrix that are described in **Table *8*** (SI No 1).

### Inputs

The inputs to this process are

1. Engine Instrument System Requirements Specification from Aircraft manufacturer
2. Software Requirements Specification and Software Requirements Traceability Matrix that are described in Table 8 (SI No 1)
3. Software Requirements Standards described in Table 7 (SI No 6)
4. Software Verification Plan (H398-001-003)
5. Review Checklist Templates described in Table 7 (SI No 16, 17)

### Activities

Activity performed is checklist driven review of Software Requirements Specification and Software Requirements Traceability Matrix.

### Outputs

The outputs of this process are described in **Table *9*** (SI No 8, 11).

### Transition Criteria

The Verification of the Software Requirements Process is entered when the Software Requirements Specification, Software Requirements Traceability Matrix, Software Requirements Standards (H398-001-006), Review Checklist described in **Table *7*** (SI No 16, 17) are under CM as per Software Configuration Management Plan (H398-001-004).

The Verification of the Software Requirements Process can be re-entered when Software Requirements Specification and Software Requirements Traceability Matrix are updated due to Problem Report(s).

### Verification of the Software Architectural Design Process

The purpose of the Verification of the Software Architectural Design Process is to verify the outputs of Software Architectural Design Process (Section 10.3.2.2) satisfy the objectives of DO-178B Table A-4: items 8 to 13. The outputs of Software Architectural Design Process are Software Architectural Design and Software Architectural Design Traceability Matrix that are described in **Table *8*** (SI No 2).

### Inputs

The inputs to this process are

1. Software Requirements Specification that is described in Table 8 (SI No 1)
2. Software Architectural Design and Software Architectural Design Traceability Matrix that are described in Table 8 (SI No 2)
3. Software Verification Plan (H398-001-003)
4. Software Design Standards (H398-001-007) described in Table 7 (SI No 7)
5. Review Checklist Templates described in Table 7 (SI No 19, 20)

### Activities

Following activities are performed during this process:

1. Checklist driven review of Software Architectural Design
2. Checklist driven review of Software Architectural Design Traceability Matrix

### Outputs

The output of this process is filled review checklists described in **Table *9*** (SI No 8, 13).

### Transition Criteria

The Verification of the Software Architectural Design Process is entered when Software Architectural Design, Software Architectural Traceability Matrix, Software Design Standards (H398-001-007) and Review Checklists described in **Table *7*** (SI No 19, 20) are under CM as per Software Configuration Management Plan (H398-001-004).

The Verification of the Software Architectural Design Process can be re-entered when Software Architectural Design and Software Architectural Traceability Matrix are updated due to Problem Report(s).

### Verification of the Software Low Level Requirements Process

The purpose of the Verification of the Software Low Level Requirements Process is to verify the outputs of Software Low Level Requirements Process (Section 10.3.2.3) satisfy the objectives of DO-178B Table A-4: items 1 to 7. The outputs of Software Low Level Requirements Process are Software Low Level Requirements and Software Low Level Requirements Traceability Matrix that are described in **Table *8*** (SI No 3).

### Inputs

The inputs to this process are

1. Software Requirements Specification that is described in Table 8 (SI No 1)
2. Software Architectural Design that is described in Table 8 (SI No 2)
3. Software Low Level Requirements and Software Low Level Requirements Traceability Matrix that are described in Table 8 (SI No 3)
4. Software Verification Plan (H398-001-003)
5. Software Design Standards (H398-001-007) described in Table 7 (SI No 7)
6. Review Checklist Templates described in Table 7 (SI No 21, 22)

### Activities

Following activities are performed during this process:

1. Checklist driven review of Software Low Level Requirements
2. Checklist driven review of Software Low Level Requirements Traceability Matrix

### Outputs

The output of this process is filled review checklists described in **Table *9*** (SI No 8, 14).

### Transition Criteria

The Verification of the Software Low Level Requirements Process is entered when Software Low Level Requirements, Software Low Level Requirements Traceability Matrix, Software Design Standards (H398-001-007) and Review Checklists described in **Table *7*** (SI No 21, 22) are under CM as per Software Configuration Management Plan (H398-001-004).

The Verification of the Software Low Level Requirements Process can be re-entered when Software Low Level Requirements and Software Low Level Requirements Traceability Matrix are updated due to Problem Report(s).

### Verification of the Software Coding and Integration Process

The purpose of the Verification of the Software Coding and Integration Process is to verify the outputs of Software Coding and Integration Process (Section 10.3.2.4) satisfy the objectives of DO-178B Table A-5: items 1 to 7. Software Code, Software Code Traceability Matrix and Executable Code are the outputs of Software Coding and Integration Process.

### Inputs

The inputs to this process are

1. Software Low Level Requirements, Software Code and Executable Code described in Table 8 (SI No 3, 4 and 5) respectively
2. Software Verification Plan (H398-001-003)
3. Software Coding Standards described in Table 7 (SI No 8)
4. Review checklists described in Table 7 (SI No 23, 24 and 25)

### Activities

Following activities are performed during this process:

1. Software Coding Standards Conformance Reports are generated using CodeTrax Tool
2. Checklist driven review of Software Code
3. Checklist driven review of Software Code Traceability Matrix
4. Linker Definition File and Make file are reviewed to check incorrect hardware address, memory overlaps and missing Software components

Note:

1. Tool Qualification Index for CodeTrax will be generated and SQA verified before the checklist driven review of Source Code.

### Outputs

The outputs of this process are described in **Table *9*** (SI No 15, 16).

### Transition Criteria

The Verification of the Software Coding and Integration Process is entered when Software Code, Software Code Traceability Matrix, Software Coding Standards, Review Checklist templates described in **Table *7*** (SI No 23, 24 and 25) and Tool Qualification Index for CodeTrax are configured in CM as per Software Configuration Management Plan (H398-001-004).

The Verification of the Software Coding and Integration Process can be re-entered when Software Code and/or Software Code Traceability Matrix are updated due to Problem Report(s).

### Testing of Outputs of the Software Integration Process

The purpose of the Testing of Outputs of the Software Integration Process is to test the Software Code and Executable Code, which are the outputs of Software Coding and Integration Process (Section 10.3.2.4) satisfy the objectives of DO-178B Table A-6: items 1 to 5. Software High Level Software Requirements are tested through Hardware Software Integration Testing (HSIT) and Software Low Level Software Requirements are tested through Software Low Level Requirements based Testing using RTRT. Software Structural Code Coverage is achieved through Software Low Level Requirements based Testing.

### Inputs

The inputs to this process are

1. Software Requirements Specification Documents, Software Low Level Requirements Documents, Source Code and Executable code described in Table 8 (SI No 1, 3, 4 and 5) respectively
2. Software Configuration Index described in Table 8 (SI No 6)
3. Software Verification Plan (H398-001-003)
4. Review checklists described in Table 7 (SI No 27, 28, 29, 30, 31, 32, 33, 38, 39)

### Activities

Following activities are performed during this process:

1. Development of Test plan and Test Scripts for HSIT. Test Plan for HSIT describes the Software High Level Requirements based test cases, Test Setup, Procedure to Locate CRC, Methods of Testing and Analysis. Test scripts will have implementation of the Software High Level Requirements based Test cases.
2. Development of Test plan and Test Scripts for LLRT. Test Plan for LLRT describes Test Setup, Test Strategies and identification of the test scripts. Test Scripts will have implementation of Software Low Level Requirements based Test cases.
3. Create HSIT Traceability Matrix (bi-directionally mapping HSI tests to Software High Level Requirements)
4. Create LLRT Traceability Matrix (bi-directionally mapping LLR tests to Software Low Level Requirements)
5. Review of Test Readiness using Review Checklist for Test Readiness (H398-006-013)
6. Execution of HSIT, Compiler and Linker Analysis, Memory Margin Analysis, Stack Depth Analysis, Timing Throughput Analysis and generation of HSI Test Report
7. Execution of LLRT and generation of LLR Test Report, Coverage Analysis Report and Data and Control Coupling Analysis Report

Note:

1. Tool Qualification Index for RTRT will be generated and SQA verified before the execution of LLRT.
2. Test Readiness Review using Review Checklist for Test Readiness (H398-006-013) is performed before the execution of HSIT.
3. Stack Depth Analysis, Timing Throughput Analysis Report, Compiler and Linker Analysis and Memory Margin Analysis are captured as part of HSI Test Report.
4. Data and Control Coupling Analysis is captured as part of Coverage Analysis Report.
5. Structural Coverage Report for each source code file detailing test coverage of Software Structure (Statement Coverage, Decision Coverage and Modified Condition/Decision Coverage) is generated during Software Low Level Requirements based Testing (LLRT) by RTRT tool.
6. For Execution of HSIT and LLRT, Independence is established. That is, Person, who executes tests, is not the developer of Test Scripts.

### Outputs

The outputs of this process are described in **Table *9*** (SI No 1, 2, 3, 4, 5, 6, 7, 8 and 30)

### Transition Criteria

The Testing of Outputs of the Software Integration Process is entered when one of the following is satisfied:

1. For development of HSI Test Plan and Scripts, Software High Level Requirements Specification and Document Templates described in Table 7 (SI No 27, 28 and 38) are baselined in CM as per Software Configuration Management Plan (H398-001-004).
2. For development of LLR Test Plan and Scripts, Software Low Level Requirements and Document Templates described in Table 7 (SI No 29, 30, 31, 32, 33 and 39) are baselined in CM as per Software Configuration Management Plan (H398-001-004).
3. For execution of HSIT, Executable Code, Software Configuration Index, HSI Test Plan and Scripts are configured into CM as per Software Configuration Management Plan (H398-001-004)
4. For execution of LLRT, Source Code, LLR Test Plan and Scripts and Tool Qualification Index for RTRT are configured into CM as per Software Configuration Management Plan (H398-001-004).

The Testing of Outputs of the Software Integration Process can be re-entered when one of the following is satisfied:

1. For HSI Test Plan and Scripts, Software High Level Requirements Specification is updated due to Problem Report(s).
2. For LLR Test Plan and Scripts, Software Low Level Requirements is updated due to Problem Report(s).
3. For execution of HSIT, Executable Code, Software Configuration Index, HSI Test Plan and Scripts are updated due to Problem Report(s).
4. For execution of LLRT, Source Code, LLR Test Plan and Scripts are updated due to Problem Report(s).

### Verification of the Verification Process

The purpose of the Verification of the Verification Process is to verify the outputs of Testing of Outputs of Software Coding and Integration Process (Section 10.3.3.6) satisfy the objectives of DO-178B Table A-7: items 1 to 8. Independence is required for verification. This will be achieved by ensuring that the person who verifies the work-product (outputs of Verification of Software Integration Process) is not involved with the development of that work-product. The outputs of Verification of Software Integration Process are,

1. Software High Level Requirements based Test Plan, HSI Test Scripts, HSI Test Reports and HSIT Traceability Matrix
2. Software Low Level Requirements based Test Plan, LLR Test Scripts, LLR Test Reports, LLRT Traceability Matrix and Coverage Analysis Reports

### Inputs

The inputs to this process are,

1. Review Checklist templates described in Table 7 (SI No 34, 35, 36, 37, 40, 41, 48)
2. Software Life Cycle Data Items described in Table 9 (SI No 1, 2, 3, 4, 5, 6, 7, 8)
3. Software Requirements Specification Documents, Software Low Level Requirements Documents, and Source Code described in Table 8 (SI No 1, 3 and 4) respectively

### Activities

Following activities are performed during this process,

1. Checklist driven review of HSI Test Plan, Test Scripts and HSIT Traceability Matrix
2. Checklist driven review of HSI Test Reports
3. Checklist driven review of LLR Test Plan, Test Scripts, LLR Test Reports, LLRT Traceability Matrix and Coverage Analysis Reports
4. Compiler Validation is performed for Object Code traceability to ensure that no additional Code is introduced by the Compiler in the Object Code

### Outputs

The outputs of this process are described in **Table *9*** (SI No 8, 21, 22, 23, 24 and 33)

### Transition Criteria

The Verification of the Verification Process is entered when one of the following is satisfied:

1. For Activity (1), HSI Test Plan, Test Scripts, HSIT Traceability Matrix and Review Checklist Templates described in Table 7 (SI No 34, 40) are configured into CM as per Software Configuration Management Plan (H398-001-004).
2. For Activity (2), HSI Test Reports and Review Checklist Template described in Table 7 (SI No 35) is configured into CM as per Software Configuration Management Plan (H398-001-004).
3. For Activity (3), LLR Test Plan, Test Scripts, Test Reports, LLRT Traceability Matrix and Coverage Analysis Reports and Review Checklist Templates described in Table 7 (SI No 36, 37, 41) are configured into CM as per Software Configuration Management Plan (H398-001-004).
4. For Activity (4), Compiler tool is configured into CM as per Software Configuration Management Plan (H398-001-004).

The Verification of the Verification Process can be re-entered when one of the following is satisfied:

1. HSI Test Plan, Test Scripts and HSIT Traceability Matrix are updated due to Problem Report(s).
2. LLR Test Plan, Test Scripts and LLRT Traceability Matrix are updated due to Problem Report(s).
3. Compiler version and/or switch options are updated due to Problem Report(s).

### Software Configuration Management Process

Software Configuration Management (SCM) is the mechanism to identify, control, baseline, and archive all items required to define a Software Product. The transition criterion to enter this phase is the start of the Project. The SCM process is performed concurrently throughout the Software Life Cycle Process.

The Software Configuration Management Plan (H398-001-004) defines methods to identify, control, baseline, release and store / archive all the Software Life Cycle Data items. This document also describes the Problem Reporting and Change Control Process using the tool SmartWorks: SmartTracker and the activities performed during a Problem Reporting Process. This document also describes the configuration activity in detail and the Change Management Process.

The Project documents are maintained in Configuration Management System as per the Software Configuration Management Plan (H398-001-004).

### Software Quality Assurance Process

The Software Quality Assurance Plan defines all the quality assurance activities.

This Plan establishes the Quality Assurance activities performed throughout the Software Life Cycle Processes. This Plan applies to Software Life Cycle Process activities ranging from the Planning Phase of a project through the conformity review and subsequent release to customer.

Quality Assurance, documents (non-)compliance with software plans and standards, verifies transition criteria are satisfied, and records the results of all software conformity reviews.

Archival Report (H398-006-015), Configuration Status Accounting (H398-006-016) and Software Conformity Report (H398-006-014) are generated by SQA before SOI#4 Audit.

The following section establishes the inputs, activities and outputs of each of the Quality Assurance processes and the criteria for transitioning from one process to another.

### Quality Assurance of Software Planning Process

The purpose of the Quality Assurance of Software Planning Process is to ensure compliance of the outputs of Software Planning Process (Section 10.3.1) and Verification of Software Planning Process (Section 10.3.3.1) with the objectives of DO-178B Table A-9: items 1 to 2.

### Inputs

The inputs to this process are,

1. Engine Instrument System Requirements Specification from Aircraft manufacturer
2. Data described in **Table *7*** under CM as per Software Configuration Management Plan (H398-001-004)
3. Internal Audit Report for Software Planning Process (H398-006-006) (Template) described in **Table *7*** (SI No 15)
4. **Table *9*** (SI No 8 and 10)

### Activities

Following activities are performed during this process:

1. Review of Plan for Software Aspects of Certification
2. Review of Plan for Software Configuration Management
3. Review of Plan for Software Development
4. Review of Plan for Software Verification
5. Review of Plan for Software Quality Assurance
6. Generate Internal Audit Report for Software Planning Process (H398-006-006)
7. File Problem Reports for audit findings during the Verification of Software Planning Process
8. Verification and closure of SmartWorks: SmartTracker problem report (if any)

### Outputs

The output of this process is Internal Audit Report for Software Planning Process (H398-006-006) and Review checklists for Software Planning Documents described in **Table *9*** (SI No 9).

### Transition Criteria

Quality Assurance of Software Planning Process is entered when all the documents described in **Table *7*** are baselined under CM as per Software Configuration Management Plan (H398-001-004).

Quality Assurance of Software Planning Process can be re-entered when one of the documents described in **Table *7*** is updated due to Problem Report(s).

### Quality Assurance of Software Requirements Process

The purpose of the Quality Assurance of Software Requirements Process is to ensure compliance of the outputs of Software Requirements Process (Section 10.3.2.1) and Verification of Software Requirements Process (Section 10.3.3.2) with the objectives of DO-178B Table A-9: items 1 to 2.

### Inputs

The inputs to this process are,

1. **Table *8*** (SI No 1) and under CM as per Software Configuration Management Plan (H398-001-004)
2. Internal Audit Report for Software Requirement Process (H398-006-007) (Template) described in **Table *7*** (SI No 18)
3. **Table *9*** (SI No 8 and 11)

### Activities

Following activities are performed during this process:

1. Generate Internal Audit Report for Software Requirement Process (H398-006-007)
2. File Problem Reports for audit findings during Verification of Software Requirements Process
3. Verification and closure of SmartWorks: SmartTracker problem report (if any)

### Outputs

The output of this process is Internal Audit Report for Software Requirement Process (H398-006-007) described in **Table *9*** (SI No 12).

### Transition Criteria

Quality Assurance of Software Requirements Process is entered when all the documents described in **Table *8*** (SI No 1) and **Table *7*** (SI No 18) are baselined under CM as per Software Configuration Management Plan (H398-001-004).

Quality Assurance of Software Requirements Process can be re-entered when one of the documents described in **Table *8*** (SI No 1) is updated due to Problem Report(s).

### Quality Assurance of Software Architectural Design Process

The purpose of the Quality Assurance of Software Architectural Design Process is to ensure compliance of the outputs of Software Architectural Design Process (Section 10.3.2.2) and Verification of Software Architectural Design Process (Section 10.3.3.3) with the objectives of DO-178B Table A-9: items 1 to 2.

### Inputs

The inputs of this process are,

1. **Table *8*** (SI No 2) and under CM as per Software Configuration Management Plan (H398-001-004)
2. Internal Audit Report for Software Architectural Design Process (H398-006-008) (Template) described in **Table *7*** (SI No 26)
3. **Table *9*** (8 and 13)

### Activities

Following activities are performed during this process:

1. Generate Internal Audit Report for Software Architectural Design Process (H398-006-008)
2. File problem reports for audit findings (if any) during Verification of Software Architectural Design Process
3. Verification and closure of SmartWorks: SmartTracker problem report (if any)

### Outputs

The output of this process is Internal Audit Report for Software Architectural Design Process (H398-006-008) described in **Table *9*** (SI No 17).

### Transition Criteria

Quality Assurance of Software Architectural Design Process is entered when all the documents described in **Table *8*** (SI No 2) and Internal Audit Report for Software Architectural Design Process (H398-006-008) (Template) described in **Table *7*** (SI No 26) are baselined under CM as per Software Configuration Management Plan (H398-001-004).

Quality Assurance of Software Architectural Design Process can be re-entered when one of the documents described in **Table *8*** (SI No 2) is updated due to Problem Report(s).

### Quality Assurance of Software Low Level Requirements Process

The purpose of the Quality Assurance of Software Low Level Requirements Process is to ensure compliance of the outputs of Software Low Level Requirements Process (Section 10.3.2.3) and Verification of Software Low Level Requirements Process (Section 10.3.3.4) with the objectives of DO-178B Table A-9: items 1 to 2.

### Inputs

The inputs of this process are,

1. **Table *8*** (SI No 3) and under CM as per Software Configuration Management Plan (H398-001-004)
2. Internal Audit Report for Software Low Level Requirements Process (H398-006-009) (Template) described in **Table *7*** (SI No 26)
3. **Table *9*** (SI No 8, 14)

### Activities

Following activities are performed during this process:

1. Generate Internal Audit Report for Software Low Level Requirements Process (H398-006-009)
2. File problem reports for audit findings (if any) during Verification of Software Low Level Requirements Process
3. Verification and closure of SmartWorks: SmartTracker problem report (if any)

### Outputs

The output of this process is Internal Audit Report for Software Low Level Requirements Process (H398-006-009) described in **Table *9*** (SI No 17).

### Transition Criteria

Quality Assurance of Software Low Level Requirements Process is entered when all the documents described in **Table *8*** (SI No 3) and Internal Audit Report for Software Low Level Requirements Process (H398-006-009) (Template) described in **Table *7*** (SI No 26) are baselined under CM as per Software Configuration Management Plan (H398-001-004).

Quality Assurance of Software Low Level Requirements Process can be re-entered when one of the documents described in **Table *8*** (SI No 3) is updated due to Problem Report(s).

### Quality Assurance of Software Coding and Integration Process

The purpose of the Quality Assurance of Software Coding and Integration Process is to ensure compliance of the outputs of Software Coding and Integration Process (Section 10.3.2.4) and Verification of Software Coding and Integration Process (Section 10.3.3.5) with the objectives of DO-178B Table A-9: items 1 and 2.

### Inputs

The inputs of this process are,

1. Table 8 (SI No 4, 5, 6 and 7) and under CM as per Software Configuration Management Plan (H398-001-004),
2. Internal Audit Report for Software Coding and Integration Process (H398-006-010) (Template) described in Table 7 (SI No 47)
3. Table 9 (SI No 8).
4. Tool Qualification Index for CodeTrax (H398-005-052) described in Table 9 (SI No 32)
5. Table 9 (SI No 15 and 16)

### Activities

Following activities are performed during this process:

1. Generate Internal Audit Report for Coding and Integration Process (H398-006-010)
2. File problem reports for audit findings (if any) during Verification of Coding and Integration Process
3. Verification and closure of SmartWorks: SmartTracker problem report (if any)

### Outputs

The output of this process is Internal Audit Report for Software Coding and Integration Process (H398-006-010) described in **Table *9*** (SI No 28).

### Transition Criteria

Quality Assurance of Software Coding and Integration Process is entered when all the items described in **Table *8*** (SI No 4, 5, 6 and 7) and **Table *7*** (SI No 47) are baselined under CM as per Software Configuration Management Plan (H398-001-004).

Quality Assurance of Software Coding and Integration Process can be re-entered when one of the items described in **Table *8*** (SI No 4, 5, 6 and 7) is updated due to Problem Report(s).

### Quality Assurance of Testing of Outputs of the Software Integration Process

The purpose of the Quality Assurance of Testing of Outputs of the Software Integration Process is to ensure compliance of the outputs of Testing of Outputs of the Software Integration Process (10.3.3.6) with the objectives of DO-178B Table A-9: items 1 and 2.

### Inputs

The inputs of this process are

1. Table 9 (SI No 1, 2, 3, 4, 5, 6 and 7) and under CM as per Software Configuration Management Plan (H398-001-004),
2. Internal Audit Report for Testing of Outputs of the Software Integration Process (H398-006-017) (Template) described in Table 7 (SI No 50)
3. Table 9 (SI No 8)
4. Tool Qualification Index for RTRT (H398-005-051) described in Table 9 (SI No 32)
5. Review Checklist for Test Readiness (H398-006-013)) described in Table 9 (SI No 30)
6. Test Witnessing Checklist (H398-006-012) described in Table 7 (SI No 43)

### Activities

Following activities are performed during this process:

1. Review of Test Readiness Review (Review Checklist for Test Readiness (H398-006-013))
2. Generate Internal Audit Report for Testing of Outputs of the Software Integration Process (H398-006-017)
3. Test Witnessing using Test Witnessing Checklist (H398-006-012)
4. File problem reports for audit findings (if any) during Testing of Outputs of Software Integration Process
5. Verification and closure of SmartWorks: SmartTracker problem report (if any)

### Outputs

The output of this process are Test Witnessing Checklist (H398-006-012) and Internal Audit Report for Testing of Outputs of the Software Integration Process (H398-006-017) described in **Table *9*** (SI No 29, 37) respectively and Review results of Test Readiness Review.

### Transition Criteria

Quality Assurance of Testing of Outputs of Software Integration Process is entered when all the items described in **Table *8*** (SI No 4, 5, 6 and 7), **Table *9*** (SI No 1, 2, 6 and 7) and **Table *7*** (SI No 50) are baselined under CM as per Software Configuration Management Plan (H398-001-004).

Quality Assurance of Testing of Outputs of Software Integration Process can be re-entered when one of the items described in **Table *8*** (SI No 4, 5, 6 and 7) and **Table *9*** (SI No 1, 2, 6 and 7) is updated due to Problem Report(s)

### Quality Assurance of Verification of the Verification Process

The purpose of the Quality Assurance of Verification of the Verification Process is to ensure compliance of the outputs of Verification of the Verification Process with the objectives of DO-178B Table A-9: items 1 and 2.

### Inputs

The inputs of this process are,

1. Table 9 (SI No 20, 21, 22, 23, 24 and 33) and under CM as per Software Configuration Management Plan (H398-001-004)
2. Internal Audit Report for Verification of the Verification Process (H398-006-011) (Template) described in Table 7 (SI No 42)
3. Table 9 (SI No 8)

### Activities

Following activities are performed during this process:

1. Generate Internal Audit Report for Verification of the Verification Process (H398-006-011)
2. File problem reports for audit findings (if any) during Verification of the Verification Process
3. Verification and closure of SmartWorks: SmartTracker problem report (if any)

### Outputs

The output of this process is Internal Audit Report for Verification of Verification Process (H398-006-011) described in **Table *9*** (SI No 35).

### Transition Criteria

Quality Assurance of Verification of the Verification Process is entered when all the items described in **Table *9*** (SI No 1, 2, 3, 4, 5, 6, 7, 33, 20, 21, 22, 23, 24) and **Table *7*** (SI No 42) are baselined under CM as per Software Configuration Management Plan (H398-001-004).

Quality Assurance of Verification of the Verification Process can be re-entered when one of the items described in **Table *9*** (SI No 1, 2, 6 and 7) is updated due to Problem Report(s)

### Certification Liaison Process

The purpose of this process is to generate Software Accomplishment Summary, Software Configuration Index and Software Life Cycle Environment Configuration Index and verification of outputs of Certification Liaison Process satisfy the objectives of DO-178B Table A-10 item 3.

### Inputs

The inputs of this process are

1. For Software Accomplishment Summary, Software Life Cycle Data described in **Table *7***, **Table *8***, **Table *9*** and Software Conformity Review Report (H398-006-014)
2. For Software Configuration Index, Software Life Cycle Data described in **Table *8***, **Table *9*** and **Table *10*** (SI No 1)
3. For Software Life Cycle Environment Configuration Index, Software Life Cycle Data described in **Table *7***

### Activities

Following activities are performed during this process:

1. Generation of Software Accomplishment Summary
2. Generation of Software Configuration Index
3. Generation of Software Life Cycle Environment Configuration Index
4. Review of outputs of Certification Liaison Process satisfy the objectives of DO-178B Table A-10 item 3 by DER and Approval by FAA

### Outputs

The output of this process is described in **Table *10***.

### Transition Criteria

The Certification Liaison Process is entered when,

1. For Activity (1), Software Life Cycle Data described in **Table *7***, **Table *8***, **Table *9*** and Software Conformity Review Report (H398-006-014) are under CM as per Software Configuration Management Plan (H398-001-004).
2. For Activity (2), Software Life Cycle Data described in **Table *8***, **Table *9*** and **Table *10*** (SI No 1) are under CM as per Software Configuration Management Plan (H398-001-004).
3. For Activity (3), Software Life Cycle Data described in **Table *7*** are under CM as per Software Configuration Management Plan (H398-001-004).

The Certification Liaison Process is re-entered when,

1. For Activity (1), after the Software Life Cycle Data described in **Table *7***, **Table *8*** and **Table *9*** are updated due to Problem Report(s).
2. For Activity (2), after the Software Life Cycle Data described in **Table *8***, **Table *9*** and **Table *10*** (SI No 1) are updated due to Problem Report(s).
3. For Activity (3), after the Software Life Cycle Data described in **Table *7*** are updated due to Problem Report(s).

### System Safety Assessment Process

The purpose of System Safety Assessment Process is to assess the Derived Requirements generated during Software Requirement Process, Software Architectural Design Process and Software Low Level Requirements Process.

### Inputs

Inputs to this process are

1. Derived Requirements documented in Software Requirements Specification
2. Derived Requirements documented in Software Architectural Design
3. Derived Requirements documented in Software Low Level Requirements

### Activities

Following activity is performed during this process:

1. Howell assesses the Derived Requirements and identifies those requirements which affect the system safety.

### Outputs

Output of this process is the Howell produced Derived Requirements System Safety Assessment Report.

### Transition Criteria

System Safety Assessment Process is entered when Derived Requirements are identified in Software Life Cycle Data described in **Table *8*** (SI No 1, 2 and 3).

System Safety Assessment Process is re-entered when Derived Requirements identified in Software Life Cycle Data described in **Table *8*** (SI No 1, 2 and 3) are updated due to Problem Report(s).

# Software Life Cycle Data

**Table *7***, **Table *8***, **Table *9*** and **Table *10*** define the DO-178B Software Life Cycle Data generated during Software Life Cycle Processes.

## Software Planning Process Data

**Table *7*** describes the Software Planning Process Data.

**Table 7: Software Planning Process Data**

| Sl. No. | Planning Data | DO-178B Paragraph Reference | Delivery Medium |
| --- | --- | --- | --- |
|  | Plan for Software Aspects of Certification  H398-001-001 | 11.1 | Electronic Medium |
|  | Software Development Plan  H398-001-002 | 11.2 | Electronic Medium |
|  | Software Verification Plan  H398-001-003 | 11.3 | Electronic Medium |
|  | Software Configuration Management Plan  H398-001-004 | 11.4 | Electronic Medium |
|  | Software Quality Assurance Plan  H398-001-005 | 11.5 | Electronic Medium |
|  | Software Requirements Standards  H398-001-006 | 11.6 | Electronic Medium |
|  | Software Design Standards  H398-001-007 | 11.7 | Electronic Medium |
|  | Software C Coding standards  H398-001-008  Software ASM Coding standards  H398-001-009 | 11.8 | Electronic Medium |
|  | Tool Qualification Plan for RTRT  H398-001-010  Tool Qualification Plan for CodeTrax  H398-001-011 | 12.2 | Electronic Medium |
|  | Review Checklist for Plan for Software Aspects of Certification (Template)  H398-006-001 | 11.5 | Electronic Medium |
|  | Review Checklist for Software Development Plan (Template)  H398-006-002 | 11.5 | Electronic Medium |
|  | Review Checklist for Software Verification Plan (Template)  H398-006-003 | 11.5 | Electronic Medium |
|  | Review Checklist for Software Configuration Management Plan (Template)  H398-006-004 | 11.5 | Electronic Medium |
|  | Review Checklist for Software Quality Assurance Plan (Template)  H398-006-005 | 11.5 | Electronic Medium |
|  | Internal Audit Report for Software Planning Process (Template)  H398-006-006 | 11.5 | Electronic Medium |
|  | Review Checklist for Software Requirements Specification (Template)  H398-005-001 | 11.3 | Electronic Medium |
|  | Review Checklist for Software Requirements Traceability Matrix (Template)  H398-005-002 | 11.3 | Electronic Medium |
|  | Internal Audit Report for Software Requirements Process (Template)  H398-006-007 | 11.5 | Electronic Medium |
|  | Review Checklist for Software Architectural Design (Template)  H398-005-003 | 11.3 | Electronic Medium |
|  | Review Checklist for Software Architectural Design Traceability Matrix (Template)  H398-005-004 | 11.3 | Electronic Medium |
|  | Review Checklist for Software Low Level Requirements (Template)  H398-005-005 | 11.3 | Electronic Medium |
|  | Review Checklist for Software Low Level Requirements Traceability Matrix (Template)  H398-005-006 | 11.3 | Electronic Medium |
|  | Review Checklist for C Source Code (Template)  H398-005-007 | 11.3 | Electronic Medium |
|  | Review Checklist for ASM Source Code (Template)  H398-005-008 | 11.3 | Electronic Medium |
|  | Review Checklist for Source Code Traceability Matrix (Template)  H398-005-009 | 11.3 | Electronic Medium |
|  | Internal Audit Report for Software Architectural Design Process (Template)  H398-006-008  Internal Audit Report for Software Low Level Requirements Process (Template)  H398-006-009 | 11.5 | Electronic Medium |
|  | Software High Level Requirement based Test Plan (Template)  H398-005-010 | 11.3 | Electronic Medium |
|  | Software High Level Requirement based Test Report (Template)  H398-005-011 | 11.3 | Electronic Medium |
|  | Software Low Level Requirement based Test Plan (Template)  H398-005-013 | 11.3 | Electronic Medium |
|  | Software Low Level Requirement based Test Report (Template)  H398-005-014 | 11.3 | Electronic Medium |
|  | Software Low Level Requirement based C Test Script (Template)  H398-005-015 | 11.3 | Electronic Medium |
|  | Software Low Level Requirement based ASM Test Script (Template)  H398-005-016 | 11.3 | Electronic Medium |
|  | Coverage Analysis (Template)  H398-005-017 | 11.3 | Electronic Medium |
|  | Review Checklist for Software High Level Requirements based Test Plan and Scripts (Template)  H398-005-020 | 11.3 | Electronic Medium |
|  | Review Checklist for Software High Level Requirements based Test Report (Template)  H398-005-021 | 11.3 | Electronic Medium |
|  | Review Checklist for Software Low Level Requirements based C Test Script and Report (Template)  H398-005-022 | 11.3 | Electronic Medium |
|  | Review Checklist for Software Low Level Requirements based ASM Test Script and Report (Template)  H398-005-023 | 11.3 | Electronic Medium |
|  | Traceability Matrix of Software High Level Requirements based Test Procedure and Test Report to Software Requirements Specification (Template)  H398-005-025 | 11.3 | Electronic Medium |
|  | Traceability Matrix of Software Low Level Requirements based Test Script, Test Report and Structural Coverage Report to Software Low Level Requirements (Template)  H398-005-026 | 11.3 | Electronic Medium |
|  | Review Checklist for Traceability Matrix of Software High Level Requirements based Test Procedures and Test Report to Software Requirements Specification (Template)  H398-005-027 | 11.3 | Electronic Medium |
|  | Review Checklist for Traceability Matrix of Software Low Level Requirements based Test Script, Test Report and Structural Coverage Report to Software Low Level Requirements (Template)  H398-005-028 | 11.3 | Electronic Medium |
|  | Internal Audit Report for Verification of Verification Process (Template)  H398-006-011 | 11.5 | Electronic Medium |
|  | Test Witnessing Checklist (Template)  H398-006-012 | 11.5 | Electronic Medium |
|  | Review Checklist for Test Readiness (Template)  H398-006-013 | 11.5 | Electronic Medium |
|  | Software Conformity Report (Template)  H398-006-014 | 11.5 | Electronic Medium |
|  | Archival Report (Template)  H398-006-015 | 11.4 | Electronic Medium |
|  | Internal Audit Report for Software Coding and Integration Process (Template)  H398-006-010 | 11.5 | Electronic Medium |
|  | Review Checklist for Coverage Analysis (Template)  H398-005-012 | 11.3 | Electronic Medium |
|  | Configuration Status Accounting (Template)  H398-006-016 | 11.4 | Electronic Medium |
|  | Internal Audit Report for Testing of Outputs of the Software Integration Process (Template)  H398-006-017 | 11.5 | Electronic Medium |
|  | Test Bench Conformity Report  H398-006-018 | 11.5 | Electronic Medium |

**Note**: Subsequent changes to Software Planning Process Data of H398 series will be captured under Software Change Impact Analysis document (H398-004-104)

## Software Development Process Data

**Table *8*** describes the Software Development Process Data.

**Table 8: Software Development Process Data**

| Sl. No. | Development Data | DO-178B Paragraph Reference | Delivery Medium |
| --- | --- | --- | --- |
|  | Software Requirements Specification Document  Gateway Module (H398-002-001-GWY)  Analog Module (H398-002-001-ANA)  Discrete Module (H398-002-001-DSC)  Configuration Management Unit (H698-002-001-CMU)  Traceability Matrix  H398-002-002-GWY  H398-002-002-ANA  H398-002-002-DSC  H698-002-002-CMU | 11.9 | Electronic Medium |
|  | Software Architectural Design Document  Gateway Module (H398-003-001-GWY)  Analog Module (H398-003-001-ANA)  Discrete Module (H398-003-001-DSC)  Configuration Management Unit (H698-003-001-CMU)  Traceability Matrix  H398-003-002-GWY  H398-003-002-DSC  H398-003-002-ANA  H698-003-002-CMU | 11.10 | Electronic Medium |
|  | Software Low Level Requirements Document  Gateway Module (H398-003-011-GWY)  Analog Module (H398-003-011-ANA)  Discrete Module (H398-003-011-DSC)  Configuration Management Unit (H698-003-011-CMU)  Traceability Matrix  H398-003-012-GWY  H398-003-012-DSC  H398-003-012-ANA  H698-003-012-CMU | 11.10 | Electronic Medium |
|  | Software Derived Requirements Report  Gateway Module (H398-003-021-GWY)  Analog Module (H398-003-021-ANA)  Discrete Module (H398-003-021-DSC)  Configuration Management Unit (H698-003-021-CMU) | 11.10 | Electronic Medium |
|  | **Gateway Module:**  H398-004-001 (Gateway Boot-Up Config file)  H398-004-002 (Gateway Flight Application Software)  H398-004-003 (Gateway Module Configuration)  **Discrete Module:**  H398-004-004 (Discrete Boot-Up Config file)  H398-004-005 (Discrete Flight Application Software)  H398-004-006 (Discrete Module Configuration)  **Analog Module:**  H398-004-007 (Analog Boot-Up Config file)  H398-004-008 (Analog Flight Application Software)  H398-004-009 (Analog Module Configuration)  **Configuration Management Unit +:**  H698-004-001 (CMU+ Boot-Up Config file)  H698-004-002 (CMU+ Flight Application Software)  H698-004-003 (CMU+ Aircraft Configuration)  Traceability Matrix  H398-004-010-GWY  H398-004-010-DSC  H398-004-010-ANA  H698-004-010-CMU | 11.11 | Electronic Medium |
|  | **Build Process Document**  H398-004-011 | 11.12 | Electronic Medium |
|  | Executable Code  Boot-Up Software (Common for Gateway, Analog, Discrete and CMU+) H108E-217 (Howell)  Software Loader (Common for Gateway, Analog & Discrete) H108E-647 (Howell)  **Gateway Module:**  Gateway Flight Application H108E-808  Gateway Module Configuration Generic[[1]](#footnote-1)  Gateway Calibration Application H108E-654 (Howell)  Gateway Boot-Up Config file H108E-655  **Discrete Module**  Discrete Flight Application H108E-809  Discrete Module Configuration Generic1  Discrete Boot-Up Config file H108E-648  **Analog Module:**  Analog Flight Application H108E-810  Analog Module Configuration Generic1  Analog Calibration Application H108E-651 (Howell)  Analog Boot-Up Config file H108E-665  **Configuration Management Unit +:**  CM Flight Application H108E-658  DL Flight Application H108E-757 (Howell)  CMU + Software Loader H108E-660 (Howell)  CMU+ Boot-Up Configuration H108E-661  CMU+ Aircraft Configuration Generic1  DL App Configuration H108E-758 (Howell)  CMU+ Maintenance Application H108E-759 (Howell) | 11.12 | Electronic Medium |
|  | Software Configuration Index  H398-004-101 | 11.16 | Electronic Medium |
|  | Software Life Cycle Environment Configuration Index for EDAU and CMU+  H398-004-102 | 11.15 | Electronic Medium |

**Note**: Subsequent changes to Software Development Process Data of H398 series will be captured under Software Change Impact Analysis document (H398-004-104)

## Software Verification, Software Configuration Management and Software Quality Assurance Process Data

**Table *9*** describes the Software Verification, Software Configuration Management and Software Quality Assurance Process Data.

**Table 9: Software Verification, Software Configuration Management and Software Quality Assurance Process Data**

| Sl. No. | Verification, CM, SQA Data | DO-178B Paragraph Reference | Delivery Medium |
| --- | --- | --- | --- |
| 1 | Software High Level Requirement based Test Plan for EDAU and CMU+  H398-005-010-GWY (Gateway HRTP)  H398-005-010-ANA (Analog HRTP)  H398-005-010-DSC (Discrete HRTP)  H698-005-010-CMU (CMU HRTP)  HSI Test Scripts | 11.13 | Electronic Medium |
| 2 | Software Low Level Requirement based Test Plan for EDAU and CMU+  H398-005-013-GWY (Gateway LRTP)  H398-005-013-ANA (Analog LRTP)  H398-005-013-DSC (Discrete LRTP)  H698-005-013-CMU (CMU LRTP)  C Test Scripts based on template H398-005-015  ASM Test Scripts based on template H398-005-016 | 11.13 | Electronic Medium |
| 3 | Software High Level Requirement based Test Report for EDAU and CMU+  H398-005-011-GWY (Gateway HRTR)  H398-005-011-ANA (Analog HRTR)  H398-005-011-DSC (Discrete HRTR)  H698-005-011-CMU (CMU HRTR)  HSI Test Reports | 11.14 | Electronic Medium |
| 4 | Software Low Level Requirement based Test Report for EDAU and CMU+  H398-005-014-GWY (Gateway LRTR)  H398-005-014-ANA (Analog LRTR)  H398-005-014-DSC (Discrete LRTR)  H698-005-014-CMU (CMU LRTR)  C Test Reports  ASM Test Reports | 11.14 | Electronic Medium |
| 5 | Coverage Analysis  H398-005-017-GWY (Gateway CA)  H398-005-017-ANA (Analog CA)  H398-005-017-DSC (Discrete CA)  H698-005-017-CMU (CMU CA) | 11.14 | Electronic Medium |
| 6 | Traceability Matrix of HSI Test Procedure and Test Report to Software Requirements Specification  H398-005-025-GWY  H398-005-025-ANA  H398-005-025-DSC  H698-005-025-CMU | 11.13 | Electronic Medium |
| 7 | Traceability Matrix of LLR Test Script, Test Report and Structural Coverage Report to Software Low Level Requirements  H398-005-026-GWY  H398-005-026-ANA  H398-005-026-DSC  H698-005-026-CMU | 11.13 | Electronic Medium |
| 8 | Problems Reports  <Smartworks: SmartTracker Problem Report Number> | 11.17 | Electronic Medium |
| 9 | Filled Review Checklists for Software Planning Documents (QA)  H398-006-001  H398-006-002  H398-006-003  H398-006-004  H398-006-005  Filled Internal Audit Report for Software Planning Process  H398-006-006 | 11.19 | Electronic Medium |
| 10 | Filled Review Checklists for Software Planning Documents  H398-001-001-PeerReview  H398-001-002-PeerReview  H398-001-003-PeerReview  H398-001-004-PeerReview  H398-001-005-PeerReview  H398-001-006-PeerReview  H398-001-007-PeerReview  H398-001-008-PeerReview  H398-001-009-PeerReview  H398-001-010-PeerReview  H398-001-011-PeerReview | 11.14 | Electronic Medium |
| 11 | Filled Review Checklist for Software Requirements Specification  H398-005-001-GWY  H398-005-001-ANA  H398-005-001-DSC  H698-005-001-CMU  Filled Review Checklist for Software Requirements Specification Traceability Matrix  H398-005-002-GWY  H398-005-002-ANA  H398-005-002-DSC  H698-005-002-CMU | 11.14 | Electronic Medium |
| 12 | Filled Internal Audit Report for Software Requirements Process  H398-006-007 | 11.19 | Electronic Medium |
| 13 | Filled Review Checklist for Software Architectural Design  H398-005-003-GWY  H398-005-003-ANA  H398-005-003-DSC  H698-005-003-CMU  Filled Review Checklist for Software Architectural Design Traceability Matrix  H398-005-004-GWY  H398-005-004-ANA  H398-005-004-DSC  H698-005-004-CMU | 11.14 | Electronic Medium |
| 14 | Filled Review Checklist for Software Low Level Requirements  H398-005-005-GWY  H398-005-005-ANA  H398-005-005-DSC  H698-005-005-CMU  Filled Review Checklist for Software Low Level Requirements Traceability Matrix  H398-005-006-GWY  H398-005-006-ANA  H398-005-006-DSC  H698-005-006-CMU | 11.14 | Electronic Medium |
| 15 | Filled Review Checklist for Source Code  For EDAU  H398-005-007-<SRCFileName>  H398-005-008-<ASMFileName>  For CMU+  H698-005-007-<SRCFileName>  H698-005-008-<ASMFileName>  **CodeTrax Report**  For EDAU  H398-005-054-<SRCFileName>  For CMU+  H698-005-054-<SRCFileName> | 11.14 | Electronic Medium |
| 16 | Filled Review Checklist for Source Code Traceability Matrix  H398-005-009-GWY  H398-005-009-ANA  H398-005-009-DSC  H698-005-009-CMU | 11.14 | Electronic Medium |
| 17 | Filled Internal Audit Report for Software Architectural Design Process  H398-006-008  Filled Internal Audit Report for Software Low Level Requirements Process  H398-006-009 | 11.19 | Electronic Medium |
| 18 | Filled Review Checklist for RTRT Tool Qualification Index  H398-005-051-PeerReview | 11.14 | Electronic Medium |
| 19 | Filled Review Checklist for CodeTrax Tool Qualification Index  H398-005-052-PeerReview | 11.14 | Electronic Medium |
| 20 | Filled Review Checklist for Compiler Validation  H398-005-050-PeerReview | 11.14 | Electronic Medium |
| 21 | Filled Review Checklist for Software High Level Requirements based Test Plan and Scripts  H398-005-020-GWY  H398-005-020-ANA  H398-005-020-DSC  H698-005-020-CMU | 11.14 | Electronic Medium |
| 22 | Filled Review Checklist for Software High Level Requirements based Test Report  H398-005-021-GWY  H398-005-021-ANA  H398-005-021-DSC  H698-005-021-CMU  Filled Review Checklist for Traceability Matrix of HSI Test Procedure and Test Report to Software Requirements Specification  H398-005-027-GWY  H398-005-027-ANA  H398-005-027-DSC  H698-005-027-CMU | 11.14 | Electronic Medium |
| 23 | Filled Review Checklist for Software Low Level Requirements based Test Scripts and Report  For EDAU  H398-005-022-<C SRC File Name>  H398-005-023-<ASM SRC File Name>  For CMU+  H698-005-022-<C SRC File Name>  H698-005-023-<ASM SRC File Name>  Filled Review Checklist for Software Low Level Requirements based Test Plan  H398-005-013-GWY-PeerReview  H398-005-013-ANA-PeerReview  H398-005-013-DSC-PeerReview  H698-005-013-CMU-PeerReview  Filled Review Checklist for Software Low Level Requirement based Test Report  H398-005-014-GWY-PeerReview  H398-005-014-ANA-PeerReview  H398-005-014-DSC-PeerReview  H698-005-014-CMU-PeerReview  Filled Review Checklist for Traceability Matrix of LLR Test Script, Test Report and Structural Coverage Report to Software Low Level Requirements  H398-005-028-GWY  H398-005-028-ANA  H398-005-028-DSC  H698-005-028-CMU | 11.14 | Electronic Medium |
| 24 | Filled Review Checklist for Coverage Analysis  H398-005-012-GWY  H398-005-012-ANA  H398-005-012-DSC  H698-005-012-CMU | 11.14 | Electronic Medium |
| 25 | Filled Review Checklist for Software Configuration Index  H398-004-101-PeerReview | 11.14 | Electronic Medium |
| 26 | Filled Review Checklist for Software Life Cycle Environment Configuration Index  H398-004-102-PeerReview | 11.14 | Electronic Medium |
| 27 | Filled Review Checklist for Software Accomplishment Summary  H398-004-103-PeerReview | 11.14 | Electronic Medium |
| 28 | Filled Internal Audit Report for Software Coding and Integration Process  H398-006-010 | 11.19 | Electronic Medium |
| 29 | Filled Test Witnessing Checklist  H398-006-012 | 11.19 | Electronic Medium |
| 30 | Filled Review Checklist for Test Readiness  H398-006-013 | 11.19 | Electronic Medium |
| 31 | Filled Software Conformity Report  H398-006-014 | 11.19 | Electronic Medium |
| 32 | Tool Qualification Index for CodeTrax  H398-005-052  Tool Qualification Index for RTRT  H398-005-051 | 12.2 | Electronic Medium |
| 33 | Compiler Validation Report  H398-005-050 | 6.4.4.2 (b) | Electronic Medium |
| 34 | Archival Report  H398-006-015 | 11.18 | Electronic Medium |
| 35 | Filled Internal Audit Report for Verification of Verification Process  H398-006-011 | 11.19 | Electronic Medium |
| 36 | Configuration Status Accounting  H398-006-016 | 11.18 | Electronic Medium |
| 37 | Internal Audit Report for Testing of Outputs of the Software Integration Process  H398-006-017 | 11.19 | Electronic Medium |

**Note**: Subsequent changes to Software Verification, Software Configuration Management and Software Quality Assurance Process Data of H398 series will be captured under Software Change Impact Analysis document (H398-004-104)

## Certification Liaison Process Data

**Table *10*** describes the Certification Liaison Process Data.

**Table 10: Certification Liaison Process Data**

| Sl. No. | Certification Liaison Data | DO-178B Paragraph Reference | Delivery Medium |
| --- | --- | --- | --- |
| 1 | Software Life Cycle Environment Configuration Index for EDAU and CMU+  H398-004-102 | 11.15 | Electronic Medium |
| 2 | Software Configuration Index for EDAU and CMU+  H398-004-101 | 11.16 | Electronic Medium |
| 3 | Software Accomplishment Summary for EDAU and CMU+  H398-004-103 | 11.20 | Electronic Medium |

**Note**: Subsequent changes to Certification Liaison Process Data of H398 series will be captured under Software Change Impact Analysis document (H398-004-104)

# Additional Considerations

Additional Considerations are described in section 12.1 through 12.9.

## Parametric Configuration Data

The EDAU and CMU+ Software supports the usage of Parameter Configuration Data. There are 2 types of Configuration Data namely,

1. Module Configuration Data – This data defines the characteristics of the hardware on which the application binary is installed. Module Configuration Data is described in Section 8.5.1. Module Configuration Data is used by the Application Software to initialize and control the operation of the Software. Hence Module Configuration Data is considered to be Option Selectable Software. Section 12.6 describe the measures taken to address this consideration
2. Aircraft Configuration Data – This data defines the Aircraft Configuration on which the EDAU is installed. This data consists of Aircraft and Engine parameters. Aircraft Configuration Data is described in Section 8.5.2. The Aircraft Configuration Data is intended to be used by the maintenance personnel to create an appropriate file based on the Aircraft and the Engines that are installed. Hence Aircraft Configuration Data is considered to be Option Selectable Software, Field Loadable Software and User Modifiable Software at the same time. Sections 12.6, 12.4 and 12.7 describe the measures taken to address these considerations.

The above two Configuration Data are in the form of a file. This data file will be protected by a CRC and tested as part of the Verification of the Software Coding and Integration Process. The CRC is embedded at the end of the Configuration Data file. During Power-on, EDAU Module downloads the Aircraft Configuration Data file from the CMU+. It then calculates CRC over the copied data and compares it with the CRC embedded in the Configuration Data file. If the check fails, then a fault is raised. Similarly, during Power-on, Module Configuration Data is checked for its CRC by its Application Software.

## Tool Qualification

**Table *11*** identifies all the tools used in the Software Development and Verification of EDAU and CMU+ Modules and the rationale for their formal qualification requirement as per FAA Order 8110.49A.

If a tool is not qualified, the results obtained from that tool are formally verified. The tools that will be qualified are RTRT and CodeTrax.

RTRT is a unit testing (Software Low Level Requirements based Testing) tool for C source code that generates Structural Coverage information. In addition to the Structural Coverage Report, RTRT also generates a report of the Test Results. The qualification data that includes generation of Structural Coverage and Test Result Reports for basic C constructs and source code is described in Tool Qualification Index for RTRT (H398-005-051).

CodeTrax is a static analysis tool for C source code. It is used for automated verification of coding standards. The tool qualification of CodeTrax is described in Tool Qualification Index for CodeTrax (H398-005-052).

The Tool Qualification Plans and Indices contain Tool Qualification Data. Tool operation instructions are available in the user manuals of each of the tools.

RTRT tool will be qualified before the Software Low Level Requirements based testing and CodeTrax will be qualified before the Software Code Review activity.

| **Table 11: Tool Qualification** | | | | | |
| --- | --- | --- | --- | --- | --- |
| Tool, Version/Model | Development/ Verification Tool | Purpose of Use | Replaces a Process | To be qualified | Justification for not doing Tool Qualification |
| Test Box | Verification Tool | Test Box is used to verify the functionality of EIU and CMU+ | No | No | This tool is used to observe the data from the EDAU and CMU+. The output produced by the tool will be manually verified. |
| Eclipse  Integrated development environment Version 4.3 with Compiler (Compiler/Linker- arm-none-eabi-gcc (Sourcery G++ Lite 2011.03-42) 4.5.2) and higher versions | Development and  Verification Tool  (The compiler is used for Verification & Validation) | It provides flexible project management for the development of Embedded applications on STM32F407 and STM32F405 | No | No | The compiler is validated. |
| RTRT – Ver  8.3.1 and higher versions | Verification Tool | Statement Coverage  Decision / Condition coverage  Modified /Decision Coverage | Yes, satisfies objectives DO-178B Table A-7, excluding 3 | Yes |  |
| ReMa  5.2 and higher versions | Development and  Verification Tool | Traceability analysis | No | No | This is a requirement management tool and does not result in any process getting automated |
| CodeTrax  6.2 and higher versions | Verification Tool | ALTEN GT developed tool for checking coding standards conformance | Yes, satisfies objective DO-178B Table A-5-4  Note: Coding Standards that are not verified by the CodeTrax tool will be verified Manually. | Yes |  |
| CoPilot | Verification Tool | This tool will be used to simulate the ARINC 429 data | No | No | This tool is used to observe the data from the EDAU and CMU+. The output produced by the tool will be manually verified. |

## Use of Software COTS Components and Software

This program does not use COTS Components and Software.

## Field Loadable Software

This program uses Gateway Application Software, Discrete Application Software and Analog application software which can be Field Loadable.

This program also uses Aircraft Configuration Data file which can be Field Loadable. This data file defines the characteristics of the Aircraft and installed Engines as described in Section 8.5.2. A Ground Support Software will be developed to generate and load this file into the CMU+.

The following measures will be taken to ensure that there are no unwanted side-effects of using Field Loadable Software:

1. Protection from inadvertent loading – Inadvertent loading happens when the software unit allows the Software to be loaded during flight. Inadvertent loading is not possible in CMU+. This is because the only way to enter loader mode is to have it plugged into a computer upon startup. But after installation the USB port will not be accessible to anyone during the flight.
2. Detection of failed/partial/corrupted loads – A failed/partial/corrupted load can lead to an anomalous behavior of the Software. This will be prevented by having a Software mechanism. The mechanism will be to verify the CRC of the configuration data after the load has been completed. The Ground Support Software will then indicate the success/failure of the loading process.
3. Ensuring the validity of the file load – Application Software ensures that CRC of the ACD file is correct. ACD requirements are documented in Software Requirements Specification and these requirements are tested as part of HSIT. ACD Review has been done to make sure the ACD data is valid and within range.

## Previously Developed Software

The below list of software are developed and certified by Howell in the previous programs and all the software life cycle artifacts corresponding to these will be used in the current program.

**Table 12: List pf Previously Developed Software**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Module Name** | **Binary Name** | **Part Number** | **DO-178B DAL** | **Tag** |
| Gateway | Common Boot-Up Software | H108E-217 | A | TE/0082/r2 |
| Gateway | Software Loader | H108E-647 | E | N/A |
| Gateway | Boot-Up Config File | H108E-655 | A | TE/0082/r2 |
| Gateway | Calibration Application | H108E-654 | E | N/A |
|  |  |  |  |  |
| Analog | Common Boot-Up Software | H108E-217 | A | TE/0082/r2 |
| Analog | Software Loader | H108E-647 | E | N/A |
| Analog | Boot-Up Config File | H108E-665 | A | TE/0082/r2 |
| Analog | Calibration Application | H108E-651 | E | N/A |
|  |  |  |  |  |
| Discrete | Common Boot-Up Software | H108E-217 | A | TE/0082/r2 |
| Discrete | Software Loader | H108E-647 | E | N/A |
| Discrete | Boot-Up Config File | H108E-648 | A | TE/0082/r2 |
|  |  |  |  |  |
| CMU | DL Application Software | H108E-775 | A | ./svn/A21HOWBLDAU/ACCORD/SW/Tag/H398\_SOI4\_RELEASE\_02 |
| Configuration Management Unit +( CMU+) | Common Boot-Up Software | H108E-217 | A | TE/0082/r2 |
| CMU+ | DL Flight Application | H108E-659 | E | N/A |
| CMU+ | CMU+ Software Loader | H108E-660 | E | N/A |
| CMU+ | CMU+ Boot-Up Configuration | H108E-661 | A | TE/0082/r2 |
| CMU+ | DL App Configuration | H108E-663 | E | N/A |
| CMU+ | CMU+ Maintenance Application | H108E-664 | E | N/A |

The usage of this software in EDAU and CMU+ is explained under sections 8.1 and 8.2 respectively.

## Option Selectable Software/Deactivated code

This program will have following three software’s as Option Selectable Software:

1. Aircraft Configuration Data file
2. Application Software
3. Module Configuration Data file

The Deactivated Code in the above Software’s will be used in some configurations only. The “Option Selectable” part of the Application Software will be developed to DAL A. For the other two (1) and (3) Configuration Data files, Application Software in the EIU (DAL A) ensures that CRC of the ACD file is correct. ACD Review has been done to make sure the ACD data used are valid and within range.

Development consideration for the deactivated code is such that all the sections of code which will be deactivated will be tied to Software Requirements.

Verification consideration for the deactivated code is such that tests will be added as part of HSIT, which ensure that only those outputs for the activated code are observed and the outputs for the deactivated code are not observed.

The following measures will be taken to ensure that database errors and corruption do not have any adverse side-effects of using Option Selection Software:

1. Data Validity checks: On Power-on, the Discrete and Analog Application Software checks the values of each configured parameter in Module Configuration Data against its corresponding range of valid values. If any parameter is configured with an invalid value, then the Application Software halts execution and enter Error mode.

Data Validity Checks for Gateway MCD and ACD is not performed. ACD and MCD Review has been done to make sure the ACD and MCD data is valid and within range

2. CRC checks: Each Configuration Data file is protected by a CRC. Refer section 12.1 for a detailed explanation.

## Aircraft Configuration Data File

Table **13** provides the Options of Aircraft Configuration Data which can be selected by the Maintenance Personnel to define the Aircraft Configuration on which the Software is installed:

**Table 13: Options Selectable SW - Aircraft Parameters**

| **Attributes** | **Options** | **Applicable Aircraft Configuration** | **Description** |
| --- | --- | --- | --- |
| Engine Type Configuration | <Engine Type> | <Aircraft Type> | Application Software uses the configuration parameters for each engine in all calculations. |

DAL for Aircraft Configuration Data has not been specifically assigned as there is no software configuration item identified as Aircraft Configuration Data (ACD) to be delivered as part of the system. The ACD is generated for each aircraft/engine combination by maintenance personnel using ground station software and range checks on each configuration parameter will be taken care by ground station software itself. This will be tested to DAL A as part of the EIU application software testing. Sections 12.4 and 12.7 provide additional details on ACD.

## Application Software

The Application Software will have portions of code that will depend on the option selected by the Maintenance Personnel. The options that can be selected are:

1. Engine Type - This is an Engine Configuration Parameter. It is entered while loading the Aircraft Configuration Data binary onto the CMU+. Each type of Engine has its own parameters and constants which have to be used in the required Engine related calculations. This option will indicate which set of parameters to be used in the calculations.
2. EIU Select – There are two EIUs which are installed in the Aircraft. This has been done to provide redundancy. During normal operations the identical software executes on both the EIUs. Both the EIU drive the required outputs. But only one EIU drives the excitations at a time.

**Table *14*** describes the Options selectable through Application Software.

**Table 14: Option Selectable Software –Application Software**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Input Location | Attributes | Options | Applicable Aircraft Configuration | Description |
| Configuration Mirror Area | Engine Type configuration | <Engine Type> | <Aircraft Type> | Application Software uses the configuration parameters for each engine in all calculations. |

## Module Configuration Data File

Refer Section 8.5.1. DAL for Module Configuration Data is defined to be DAL A in **Table *4***.

## User Modifiable Software

The CMU+ contains "User Modifiable Software (UMS)". The Software Loader receives Aircraft Configuration Data from a Ground Support Software and then writes the data to a specific non-volatile memory location in the CMU+. When requested, this data is transmitted to both EIUs where it is used to configure the EDAU with Aircraft and Engine specific information.

The following measures will be taken to ensure that there are no unwanted side-effects of using User Modifiable Software:

1. UMS do not change the safety margins and operational capability of the Aircraft or Engine, this will be prevented by Software mechanism. The mechanism is to have checks implemented in the Application Software. These checks will ensure that the values entered by Maintenance Personnel are within the limits as specified in Engine Instrument System Requirements Specification from Aircraft manufacturer.
2. Protect non-modifiable Software: This will be ensured in two ways.
   1. Built-in Tests – The integrity of the Application Software will be checked during PBIT and CBIT. Any modification to the non-modifiable Software (Application Software) will be detected by these tests and the corresponding recovery action initiated.
   2. A test will be performed in HSIT to ensure that the loading of the Aircraft Configuration Data file does not affect other parts of the memory.
3. Identifying procedure constraints to users: Software Configuration Index will describe procedures, methods and tools for modifying User Modifiable Software. These along with constraints will be provided to the end user.

## Multiple Version Dissimilar Software

Not Applicable.

## Product Service History

Not Applicable.

## DO178B justification on AC 20-115D

Here is our brief answer on each of the items of 5.Using ED-12B/D0-178B Processes and Procedures for New Development.

**Table 15 DO178B justification on AC 20-115D**

|  |  |
| --- | --- |
| AC Section | Justification |
| 5.a.(1) | Yes, Meeting the Criteria. |
| 5.a.(2) | Yes, Meeting the Criteria |
| 5.a.(3) | Not applicable for Howell on this project |
| 5.a.(4) | Yes, Meeting the Criteria by either checking the PDI using checklist or by writing test procedures to confirm all the PDI items are within the range. |
| 5.a.(5) | Yes, Meeting the Criteria. No change to the process |
| 5.a.(6) | Yes, Meeting the Criteria as we are not declaring D0178C. |
| 5.b | Not needed as we are meeting all of 5.a |
| 5.c | Not establishing new software lifecycle process |

# Project Schedule

Project Milestones are described in section 13.1. Meetings are described in section 13.2.

## Project Milestones

**Table *16*** describes the major Milestones for the Software of EDAU and CMU+ certification schedule.

**Table 16: Project Milestones**

| **No** | **Milestones** | **Approximate End Dates** |
| --- | --- | --- |
| 1 | Project Kick-off | 10 July 2023 |
| 2 | SOI#1 Audit | 11 July 2024 |
| 3 | Requirements, Design and Implementation | 10 July 2024 |
| 4 | SOI#2 Audit | 11 July 2024 |
| 5 | TRR Audit, HSIT, LLRT, SOI#3 | 23 August 2024 |
| 6 | SOI#4 Audit | 31 August 2024 |

**Note:** Subsequent changes to Project Milestones of H398 series will be captured under Software Change Impact Analysis document (H398-004-104)

## Meetings

Meetings will be held with various stakeholders of the team as necessary to verify that the software is being developed and verified (including the witnessing of testing) that conforms to the Software Life Cycle Processes.

Status report depicting the project progress will be sent fortnightly to Howell.

Quality Assurance (QA) will review the documents as described in Software Quality Assurance Plan (H398-001-005) prior to the SOI audits for the process adherence. The four SOI audits will be carried out as follows:

1. SOI#1: To be carried out by Kevin Crozier after the Software Planning Process is complete.
2. SOI#2: To be carried out at Howell. ALTEN GT Personnel will travel to Howell facility to assist them after the Software Design and Coding are complete and ready for Software Verification.
3. SOI#3: To be carried out at Howell. ALTEN GT Personnel will travel to Howell facility to assist them after the Software Verification is complete.
4. SOI#4: To be carried out by Howell. ALTEN GT’s QA and Project Team will support Howell after the submission of Software Life Cycle Data items for final certification.

# PSAC Compliance to DO-178B

**Table *17*** describes PSAC compliance to DO-178B.

**Table 17: PSAC Compliance to DO-178B**

|  |  |
| --- | --- |
| **DO-178B Section** | **PSAC Section** |
| 11.1a System Overview | 7 System Overview |
| 11.1b Software Overview | 8 Software Overview |
| 11.1c Certification Considerations | 9 Certification Considerations |
| 11.1d Software Life Cycle | 10 Software Life Cycle |
| 11.1e Software Life Cycle Data | 11 Software Life Cycle Data |
| 11.1f Schedule | 13 Project Schedule |
| 11.1g Additional Considerations | 12 Additional Considerations |

**END OF DOCUMENT**

1. This is User Modifiable Software. Hence the part numbers for these files cannot always be determined by Howell or ALTEN GT and it varies with different aircraft variants. [↑](#footnote-ref-1)