AIRBORNE SOFTWARE DEVELOPMENT PROCESSES CERTIFICATION REVIEW STRATEGY BASED ON RTCA/DO-178C

Jinghua Sun[1], Samuel Edwards[2], Nic Connelly[3],
Andrew Bridge[4] and Lei Zhang[1]

[1]COMAC Shanghai Aircraft Design and Research Institute, Shanghai, China
[2]Defence Aviation Safety Authority, 661 Bourke St, Melbourne, VIC, Australia
[3]School of Engineering, RMIT University, Melbourne, VIC, Australia
[4]European Union Aviation Safety Agency, Cologne, Germany

ABSTRACT

Airborne software is invisible and intangible, and is frequently used to provide safety critical functionality for aircraft. Highly complex software, however, cannot be exhaustively tested and only assured through a structured, process, activity, and objective-based approach. This paper studied the development processes and objectives applicable to different software levels based on RTCA/DO-178C, and identified 82 technical focus points based on each airborne software development sub-process, then created a Process Technology Coverage matrix to demonstrate the technical focuses of each process. This paper proposes an objective-oriented top-down and bottom-up sampling strategy for the four software Stage of Involvement reviews by considering the frequency and depth of involvement. Finally, this paper provides a Technology Objective Coverage matrix, which can support the reviewers to perform the efficient risk-based SOI reviews by considering the identified technical points, thus efficiently achieving confidence in the level of safety of the aircraft from the software assurance perspective.

KEYWORDS

Airborne Software, Stage of Involvement, DO-178C, Safety Critical Software Oversight.

1. INTRODUCTION

Modern transport aircraft are developed and certified with numerous, complex systems that rely on embedded software to control and optimise the flight of the aircraft. With the development of computer technology, more and more aircraft system functions are implemented by airborne software, however, software is an intangible asset, having no physical presence, which is stored on various media (CASA, 2014). The software will fail only when there is a latent defect, virus, design error, or single event exception. Software design errors may exist for many years without manifesting or causing malfunctions. Thus quality should be built into the software and be reviewed by assuring the development and verification processes (CASA, 2014) (Rierson, 2013). Airborne software is always one of the critical concerns in the aircraft certification process (EASA, 2012).

Software safety is an increasingly prominent issue in today’s aviation industry (Mendis, 2008). The aircraft systems can directly affect the safety of aircraft, however, the software is fundamentally different from the physical components installed on the aircraft. The structural

David C. Wyld et al. (Eds): NLP, MLTEC, CLBD, SEAPP, NeTIoT, VLSIE, ITCS, ARIA, DMS - 2021
pp. 269-282, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.112321

components of the aircraft can be tested to ensure that there are no design and manufacturing defects, whereas the Mean Time between Failures (MTBF) and programmed replacements do not apply to software components (CASA, 2014). Continuous testing cannot demonstrate that software has a reliability level similar to that of physical components, as the software does not degrade with use, rather, defects are experienced in exact states of operation. The software embedded in physical systems directly impacts the safety of the aircraft and its occupants (Hilderman & Baghai, 2007). Employing software review technology can ensure that rigour has been applied during the applicant's design commensurate with the worst-case failure condition associated with airborne software (RTCA, 2011a).

A level of assurance is required to have confidence in software to ensure aircraft safety. In October 2018 and March 2019, two Boeing 737MAX planes belonging to Indonesian Lion Air and Ethiopian Airlines crashed, respectively, causing a total of 346 deaths, which was directly related to the design of the Manoeuvring Characteristics Augmentation System (MCAS) and its flight control law software (COMMITTEE, 2020). This tragedy is a stark reminder of the criticality of software, and has been a significant loss for Boeing, and the operators of aircraft that were grounded. At the same time, the FAA as the supervisor also triggered a crisis of public trust. Wayne Rash stated that "As is the case where software controls hardware, there are ways things can go wrong either because something happened that was not anticipated, or because the response was wrong" (Rash, 2019). So what can be done to ensure that the software to be maintained at an acceptable level of safety?

Due to the particularity of airborne software and the professionalism of software-related technologies, significant pressure is placed on airborne software reviewers. However, the complexity and scale of software keeps increasing as modern civil aircraft are getting more and more integrated and complex. Therefore, formulating a set of airborne software review strategies with related technical focuses is an important issue.

For more than three decades, airborne software has been developed and assured through a structured approach based on objectives and activities (Rierson, 2013). The most commonly used method to measure software goodness is DO-178[], which is recognised as Means of Compliance (MOC) by NAAs(National Airworthiness Authority) via their respective Advisory Circular (AC) (Hilderman & Baghai, 2007). This study was conducted based on DO-178C to establish the airborne software review strategy to support certification of safety critical software.

2. ANALYSIS OF SOFTWARE REVIEW TECHNICAL FOCUSES

2.1. Quantitative Analysis of DO-178C Software Life Cycle Process and Objectives

DO-178C is a process-based, activity-driven, objective-oriented standard. It is not a software development standard, but a method to measure the goodness of software, and provide a safety benchmark that is commensurate with the safety criticality. It contains six processes (represented in Figure 5), which are the planning process, development process, and four integral processes (verification process, configuration management process, quality assurance process, and certification liaison process). The integral processes are supported throughout the whole software lifecycle (RTCA, 2011a). Notably, not all the projects follow a Waterfall lifecycle model, but a variation in the representation of the waterfall model instead (Santos and Ferreira 2019). The DO-178C proposed process, output, and input are represented in Figure 1, and can be adapted to the project lifecycle model as required.

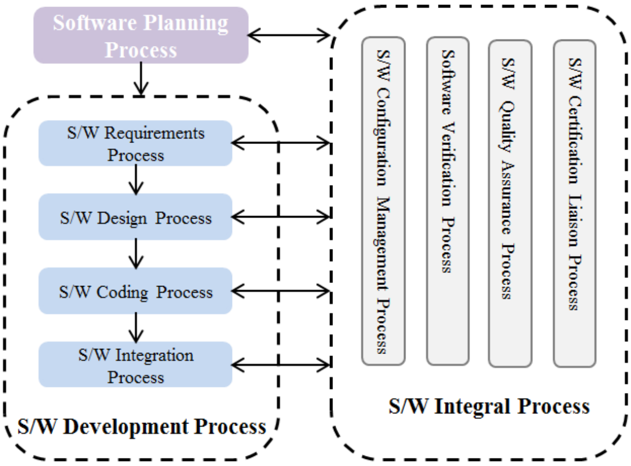
Software Planning Process
S/W Requirements Process
S/W Design Process
S/W Coding Process
S/W Integration Process
S/W Development Process
S/W Configuration Management Process
Software Verification Process
S/W Quality Assurance Process
S/W Certification Liaison Process
S/W Integral Process

Figure 1. DO-178C software life cycle processes

A latent software error in data or the final product can cause a fault of the software, then the abnormal behaviours of software can lead to a system failure condition, which can finally affect the aircraft operations. The rigour of software development is determined by the software level. DO-178C defined five software levels as listed in Table 1. DAL A is the severest, while DAL E has no safety impact. The software DAL is determined by the system safety assessment process. The different level has different objectives requirements. Table 2 *and Figure 2* are the comparison of DO-178C's Objectives in Annex A from Table A-1 to Table A-10 for different DALs of software.

Table 1. DO-178C Software DAL, related failure conditions and objectives.

Source: (Marques & Yelisetty , 2019) (Jimenez el. 2020)

System Failure Condition	Required Software Level	Number of Associated Objectives	Number of Associated Objectives with Independence
Catastrophic	A	71	31
Hazardous	B	69	19
Major	C	62	5
Minor	D	26	2
No Safety Effect	E	0	0

Table 2. Comparison of DO-178C objectives for different software levels.

Annex A	A	B	C	D
Table A-1 Software Planning Process	7	7	7	2
Table A-2 Software Development Process	7	7	7	4
Table A-3 Verification of Outputs of Software Requirements Process	7	7	6	3
Table A-4 Verification of Outputs of the Software Design Process	13	13	9	1
Table A-5 Verification of Outputs of Software Coding & Integration Processes	9	9	8	1
Table A-6 Testing of Outputs of Software Integration Process	5	5	5	3

Table A-7 Verification of Verification Process Results	9	7	6	1
Table A-8 Software Configuration Management Process	6	6	6	6
Table A-9 Software Quality Assurance Process	5	5	5	2
Table A-10 Certification Liaison Process	3	3	3	3

The experience accumulation of reviewers can start from Level D software review, and gradually master the review methods and techniques of higher-level software, to finally be competent for the review of Level A software:

a) Level D can be treated as a black box, focusing on high-level requirements development and verification. If updating a level D software to level C, there will be a leap of workload by 36 objectives.

b) The objectives differences between level C and level B include 1 Objective in Table A-3 "High-level requirements are compatible with target computer", 4 objectives in Table A-4 about the compatibility and verifiability of low-level requirements and architecture, 1 objective in Table A-5 "Source code is verifiable", and 1 objective about decision coverage in Table A-7.

c) The main differences between A and B are 2 objectives in Table A-7, which are requirements of MCDC Structural Coverage Analysis (SCA) and verification of additional code that cannot be traced to source code.

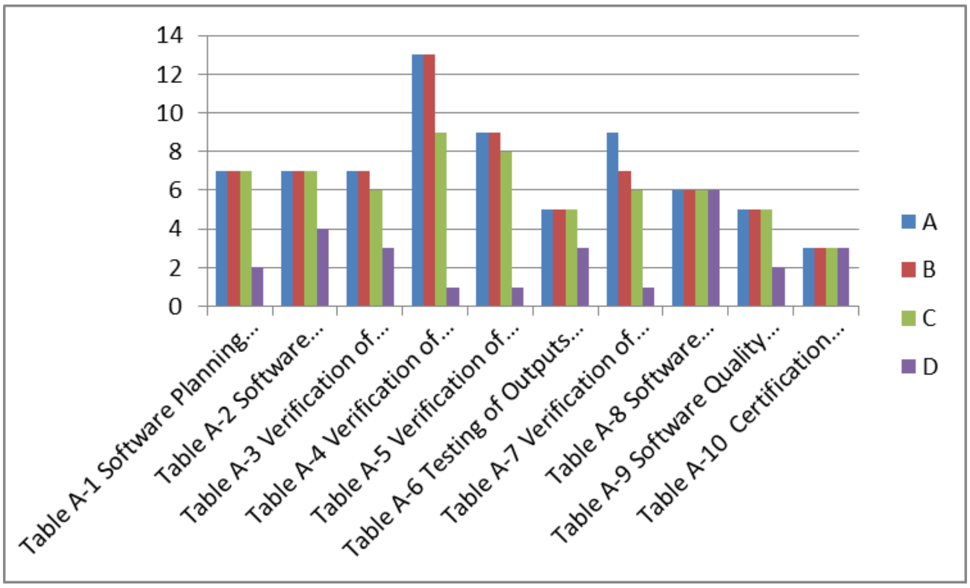

Figure 2. The comparison of applicable objectives in each Table of Annex A for different software levels

2.2. Analysis of Technical Focuses of DO-178C Process

The study on the DO-178C objectives and process can help software reviewers quickly locate the technical focuses and finding compliance. Table 3 is the analysis of the technologies based on DO-178C software life cycle processes. Each process of DO-178C may contain sub-process and components (RTCA, 2011a). The technical focus points are analysed based on each component covered and concerned during the software reviews. The technologies are from the research and analysis of the technical focus points, with most of them are described in DO-178C, and a few are from the industry practice, then they are compared with the CAST Paper research themes, finally re-analysed to ensure the completeness of the technology list.

Table 3. Qualitative analysis of technical focus points and related techniques of DO-178C

Source: (RTCA, 2011a) (FAA, 2003) (EASA, 2012) (FAA, 2004) (FAA, 2017) (CAST, 2002) (RTCA, 2011b)

Process	**Sub-process/Components**	**Technical Focus Points/Elements**	**Technologies (T_i, i=1…n)**
4.0 Software Planning	4.3 Software Plans	11.1 PSAC 11.2 SDP 11.3 SVP 11.4 SCMP 11.5 SQAP	1) Software DAL Determination 2) Partitioning 3) Multiple-Version Dissimilar Software 4) Safety Monitoring 5) PDI 6) User-Modifiable Software 7) COTS 8) Field-Loadable Software 9) Option-Selectable Software 10) Software Life Cycle Definition 11) Transition Criteria 12) Deactivated Code 13) PDS 14) Tool Qualification 15) Reuse of tool qualification data 16) Reuse of software life cycle data 17) Exhaustive Input Testing 18) Software Reliability Model 19) Product Service History 20) Database/PDI 21) Use of COTS Graphical Processor Unit (GPU) 22) Microprocessor 23) Multiple Core Processors 24) SEU (Single Event Upset) 25) Reverse engineering
	4.4 Software Life cycle Environment Planning	4.4.1 Software Development Environment 4.4.2 Language and Complier 4.4.3 Software Test Environment	
	4.5 Software Development Standards	11.6 Software Requirements Standards 11.7 Software Design Standards 11.8 Software Code standards	
5.0 Software Development	5.1 Software Requirements	11.9 Software Requirements Data	26) High-Level Requirements

Process	**Sub-process/Components**	**Technical Focus Points/Elements**	**Technologies (T_i, i=1…n)**
		11.22 Parameter Data Item File	27) Derived requirements 28) Merging high-level requirements and low-level requirements
	5.2 Software Design	11.10 Design Description	29) Control Flow Design 30) Data Flow Design 31) Low-Level Requirements 32) PDI Design
	5.3 Software Coding	11.11 Source Code 11.22 Parameter Data Item File	33) C, Ada, Assembly languages 34) Auto code generation 35) MBD 36) OOT 37) Cache 38) Stack
	5.4 Integration	11.12 Executable Object Code	39) Compiling 40) Complier library 41) Software Integrity Check (e.g. Cyclic redundancy check, Checksum)
	5.5 Traceability	11.21 Trace Data	42) Traceability Tools (e.g. DOORS)
6.0 Software Verification	6.3 Software review and analysis	Review and analysis of Software Plans and standards 6.3.1 Review and analysis of Software High-Level Requirements (HLRs) 6.3.2 Review and analysis of Software Low-Level Requirements (LLRs) 6.3.3 Review and analysis of Software Architecture 6.3.4 Review and analysis of Source Code 6.3.5 Review and analysis of the Outputs of the Integration Process 6.4.5 Review and analysis of Test Cases, procedures, and results 6.6 Review and analysis of PDI File	43) Plans and Standards Review 44) HLR Review and Analysis 45) LLR Review and Analysis 46) Architecture Review and Analysis 47) Source Code Review and Analysis 48) Outputs of the Integration Process Review and Analysis 49) Test Cases Review and Analysis 50) PDI file Review and Analysis 51) Worst-Case Execution Time 52) Verification of Stack Usage 53) Model Review and Analysis 54) Verification of independence
	6.4 Software Testing	6.4.1 Test Environment 6.4.2,6.2.3	55) Hardware/Software Integration Testing 56) Software Integration

Process	**Sub-process/Components**	**Technical Focus Points/Elements**	**Technologies (T_i, i=1…n)**
		Requirements-Based Test 6.4.4 Test coverage Analysis	Testing 57) Low-Level Testing 58) Normal Range Test Cases Selection 59) Robustness Test Cases Selection 60) MCDC 61) Decision Coverage Analysis 62) Statement Coverage Analysis 63) Data Coupling 64) Control Coupling 65) DAL A additional verification (Whether Object Code can directly traceable to source code) 66) Extraneous Code Resolution 67) Deactivated Code Handle
	6.5 Traceability	11.21 Trace Data	
Integral Process	7.0 Software Configuration Management	7.2.1 Configuration Identification	68) Software part numbering
		7.2.2 Baselines and Traceability	69) Baseline Definition
		7.2.3 Problem Reporting	70) OPR Category Definition
		7.2.4 Change Control	71) Software Change Control
		7.2.5 Change Review	
		7.2.6 Configuration Status Accounting	
		7.2.7 Archive, Retrieval, and Release	72) Media Selection, Refreshing, Duplication 73) Data Retention
		7.3 Data Control Category	
		7.4 Software Load Control	74) Software Conformity Inspection
		7.5 Software Life Cycle Environment Control	
	8.0 Software Quality Assurance	8.2 Software Quality Assurance Activities	
		8.3 Software Conformity Review (SCR)	75) SCR 76) First Article Inspection (FAI)
	9.0 Certification Liaison	9.1 Means of Compliance and Planning (LOI, Milestones, and Issue Papers, etc.)	77) LOI Criteria
		9.2 SOI Reviews	78) SOI Review Strategy 79) Sampling Strategy
		9.3 Software	80) Software maturity

Process	**Sub-process/Components**	**Technical Focus Points/Elements**	**Technologies (T_i, i=1…n)**
		Approval, including approval of Software Configuration Index (SCI) and Software Accomplishment Summary (SAS)	evaluation for Type Inspection Authorization (TIA) 81) Open Problem Report (OPR) Evaluation 82) Software Change Impact Analysis (CIA) to determine Major or Minor Changes

Note: In addition to the description of the items in the first three columns, the chapter number of the referenced DO-178C is also listed, such as 6.0 Software Verification, where 6.0 refers to DO-178C Chapter 6. The verification process is one of the four integral processes listed separately in the table because it is highly related to the software product. Each technology is identified as T_i. For instance, T_{81} refers to item 81) ORP technology in this table.

This paper identified a total of 82 technologies based on the DO-178C software assurance benchmark. The same technology may be used in different processes, but the focus will be on different perspectives. For example, MBD (Model Based Development) may be used in planning, design, coding, and verification processes. The technology distribution statistics in each process are shown in Table 4 and Figure 3.

Table 4. Software Process Technology Coverage (PTC) matrix

DO-178C Process	**Technology Coverage**	**Amount**
Planning Process	T_1 ~T_{25}	25
Development Process	T_{26} ~T_{42}	17
Verification Process	T_{43}~ T_{67}	25
Configuration Process	T_{68}~ T_{74}	7
Quality Assurance Process	T_{75}~ T_{76}	2
Certification Liaison Process	T_{77} ~T_{82}	6

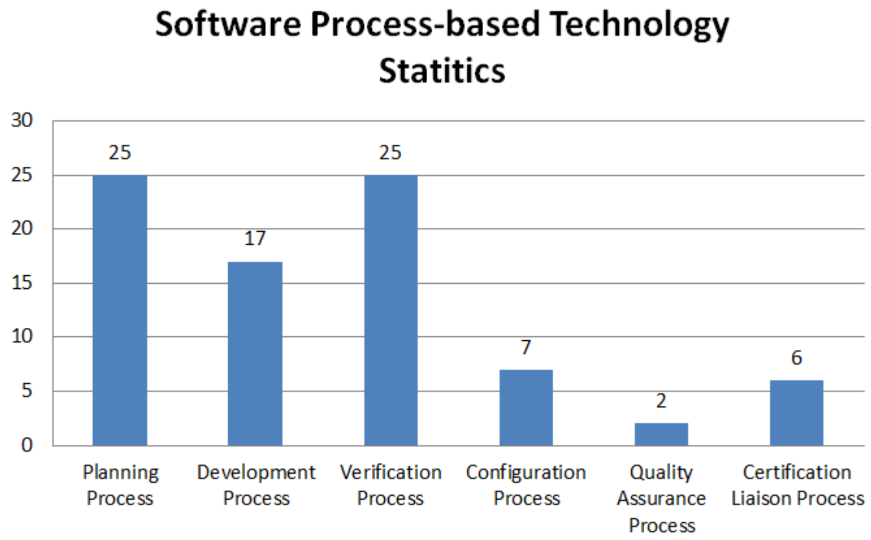

Figure 3. A quantitative analysis of the technology distribution of each process

3. Analysis of Software Reviews and Sampling Strategy

3.1. Analysis of SOI Reviews and LOI

SOI is a method of implementing process control on airborne software originally defined by the FAA in order to monitor the software life cycle process and assess compliance with the applicable objectives of DO-178B and related airworthiness requirements. According to FAA and EASA policy, four SOI reviews are defined, which are SOI#1 Software Planning Review, SOI#2 Software Development Review, SOI#3 Software Verification Review, and SOI#4 Final Certification Software Review (FAA, 2003) (EASA, 2012). The review aims to find systemic problems in the applicant's software developing processes and non-compliance issues with regulations and establish confidence in the software through the reviews (FAA, 2004). The purpose of software SOI review is to develop confidence in compliance with DO-178C objectives and other applicable software policy, guidance, and issue papers (FAA, 2018). The reviews can be conducted by a certification officer or delegated to a DER or ODA/DOA. The LOI depends on the project-specific conditions, which is executed through SOI. The main factors that can affect the SOI frequency are as follows:

a) the software category, which means PDS, COTS, new-developed software, TSOA software, libraries, RTOS, IMA hosted software, etc.,
b) the software DALs as determined by the system safety assessment process (EASA, 2012) (FAA, 2003),
c) the project characteristics, such as the tier of supplier-chain, the experience of the applicant, the complexity of the project, system functionality and novelty, software developing team human resources, and existence of issues associated with Section 12 of DO-178C (FAA, 2003),
d) the use of new technologies or unusual design features (EASA, 2012),
e) whether using alternative methods to show compliance,
f) the establishment and operation of the software assurance aspect of the applicant’s Design Assurance System (DAS), and
g) the amount of planning review activities of the delegation systems (e.g. DER or ODA) and the applicant’s self-monitoring status (EASA, 2012).

3.2. Analysis of SOI Review and Sampling Strategy

Studies indicate that developing a scientific and reasonable software review and sampling strategy, and mastering the technology related to each SOI review, especially the impact of this technology on software compliance verification, will facilitate the rapid identification of key clues during software reviews (Dodd & Habli, 2012). Each SOI review and sampling strategy and the applicable identified technologies for each SOI are analysed in the following sections.

3.2.1. SOI#1: Software Planning Review

The goal of SOI#1is to evaluate the compliance of the software planning with the applicable objectives of Table A-1 and A-8~A-10 of DO-178C Annex A (FAA, 2004). Review activities and review strategy of SOI#1 is suggested to firstly review the software interface with system development process, hardware design process, and system safety assessment process to assess the consistency among the plans and standards in compliance with the objectives of DO-178C Table A-1 (Chen, et al., 2015). Then review the verification results, the Software Quality Assurance (SQA) record, the Software Configuration Management (SCM) records, and the certification liaison process, and assess the compliance with the applicable objectives of DO-

178C Table A-8~A-10 (FAA, 2004). The important thing is to assess the consistency between software plans to determine that when the applicant follows their plans whether they will meet all applicable objectives of DO-178C and other applicable software policy or guidance (RTCA, 2011a). If tool qualification, MBD, OOT, or formal method is applied, the assessment could also cover additional aspects in RTCA/DO-330, DO-331,DO-332 and DO-333 (FAA, 2017).

3.2.2. SOI#2: Software Development Review.

The goal of SOI#2 is to assess whether the software plans and standards are effectively implemented and to evaluate the compliance of the software development process to the applicable objectives of DO-178C Table A-2~A-5, and A-8~A-10 (FAA, 2003). The review and sampling strategy is suggested to review the output of the software requirements process, design process, coding process, and integration process, and assess the compliance with applicable objectives of DO-178C Table A-2~A-5 through top-down and bottom-up thread review (illustrated in Figure 4) with the Risk-Based sampling strategy*(VanderLeest, 2013)*, by which the sampling covers each functional area until the reviewer has sufficient confidence in the software implementation of specific functional requirements set. It also need to assess the compliance of configuration management, quality assurance, and airworthiness liaison process with the applicable objectives of DO-178C A-8~A-10, and evaluate the closure status of review action items in SOI#1.

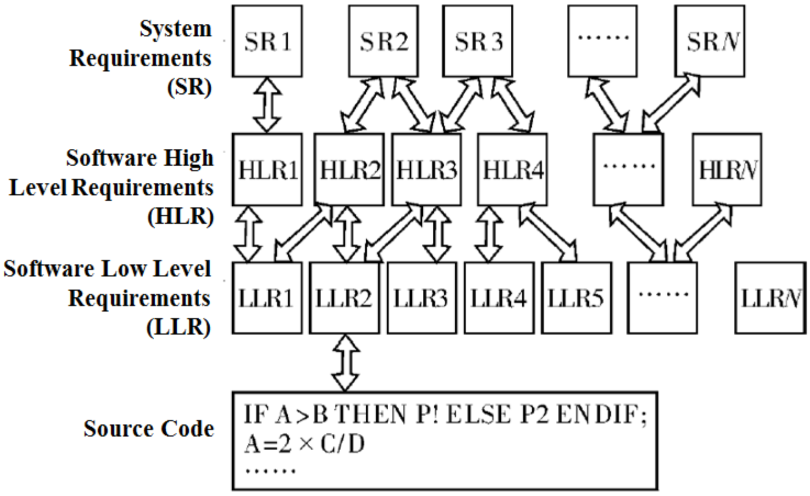

Figure 4. Illustration of top-down and bottom-up thread review

Source: (Xing & Mu, 2015)

3.2.3. SOI#3: Software Verification Review

The purpose of SOI#3 is to evaluate the compliance of the software verification process with the applicable objectives of DO-178C Table A-6, A-7, and A-8~A-10 to assess the effectiveness and implementation of verification plans and procedures (FAA, 2003). The SOI#3 software review and sampling strategy are suggested to be also risk-based to perform a delta review of the development data if there are major changes from the previous review, and to assess the test cases, test procedures, verification results, test coverage, and code structure coverage to the applicable objectives of DO-178C Table A-6 and A-7. The sampling strategy is the same as SOI#2.

Besides, it is recommended that reviewers pick some requirements that need additional considerations, for instance, if there are option selectable software or UMS, the verification of the protection mechanism should be reviewed. There are many cases, for example, the Fuel Quantity Indication Computer (FQIC) can use Litre or Kilogram as the unit of measurement, which is mainly configured through the pins of the computer. Some options are implemented by software design, for example, the B737Max MCAS system has two modes which are Auto Pilot (AP) and manual flight modes (Boeing, 2020), the mode selection is usually implemented by software logic with Case or If statement, according to such strategy, the reviewers is likely to pay attention to the protection mechanism to ensure there are no unintended behaviours and the system requirements and architecture has defined the mechanism (COMMITTEE, 2020).

3.2.4. SOI#4: Final Certification Software Review

The goal of SOI#4 is to determine compliance of the final software product with the appropriate objectives of RTCA/DO-178C and other applicable certification policies and guidance (FAA, 2003). The SOI#4 review strategy is suggested to evaluate the closure status of findings, observations, and action items of the previous reviews, to conduct a delta review of SOI#2 and SOI#3 when necessary if there are major changes or the reviewer does have sufficient confidence in the software product, to assess the OPRs(Open Problem Reports) to judge whether they can be deferred to post-TC, and to review the final SCI, SAS, tool qualification data, such as Tool Accomplishment Summary (TAS) if applicable, to judge whether the version of software product intended to be used in the certified system or equipment fully comply with all applicable DO-178C objectives, the policy, and guidance (FAA, 2004).

3.3. Quantitative Analysis of SOI Technology & Objective Coverage

Through the above analysis, it can be known that airborne software safety assurance can be achieved by a structured approach. Table 5 is the analysis result of the applicable technology and objectives of each SOI. The analysis approach and process are as follows:

a) Based on the analysis of the SOI review strategy in Section 4.3.2 of this paper, identify the appropriate technologies associated with each SOI by referring to the technology list in Table 3.
b) Based on DO-178C Annex A and the analysis of SOI review strategy in Section 4.3.2 of this paper, in conjunction with FAA Order 8110.49 Chapter 2 “Software Review Process” (FAA 2003), which was based on DO-178B, analyse the available data to identify applicable objectives for each SOI based on DO-178C.

Figure 5 is the quantitative analysis of the distribution of TOC of each SOI review, which demonstrated that 50% of the DO-178C objectives are assessed in SOI#2 review, with 35% of the technologies assessed. According to the number of objectives, SOI#3 is the second highest, with 32% of the objectives addressed, however, accounting for 26% of the technologies. SOI#1 accounts for 31% of the objectives, but covers 16%, of the technologies. Finally, SOI#4 objectives are at 2%, and technology accounts for 8%. SOI#4 is a review of the entire life cycle process. It is necessary to evaluate all previous SOI review opening items, non-conformance items, and observation items. Therefore, although the SOI#4 objectives are accounted for the least, it plays a very critical role in the entire software review process, as the reviewers will determine whether the software is in compliance with all the applicable objectives of DO-178C and whether it can obtain the final approval.

Table 5. TOC Matrix of each SOI.

Source: (FAA, 2004) (RTCA, 2011a)

SOI	Technology		Objectives	
	Identification	Amount	Identification	Amount
SOI#1	T_{1}~T_{25}, T_{43}, T_{68} ~ T_{69}, T_{77}~T_{78}	30	Table A-1: Objective1-7(All Objectives) Table A-8: Objective1-4 Table A-9: Objective 1 Table A-10: Objective1-2	14
SOI#2	T_{26} ~T_{42} T_{44}~ T_{54} T_{68}~T_{71} T_{78}~T_{79}	34	Table A-2: Objective 1-6 Table A-3: Objective1-7(All Objectives) Table A-4: Objective 1-13(All Objectives) Table A-5: Objective 1-6 Table A-8: Objective 1-4,6 Table A-9: Objective 1-4 Table A-10: Objective 1-2	43
SOI#3	T_{48}~T_{49}, T_{51}, T_{54} T_{55} ~ T_{67} T_{68}~ T_{71} T_{77}~T_{81}	26	Table A-5: Objective 7-9 Table A-6: Objective1-5(All Objectives) Table A-7: Objective 1-9(All Objectives) Table A-8: Objective1-6(All Objectives) Table A-9: Objective1-4 Table A-10: Objective 1-2	28
SOI#4	T_{75} ~ T_{82}	8	Table A-9: Objective 5 Table A-10: Objective 3	2

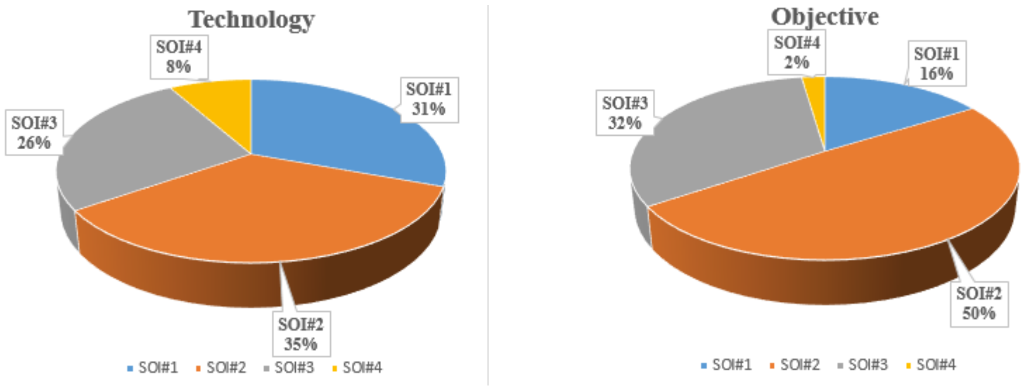

Figure 5. The TOC distribution of each SOI

4. Conclusions

Software reviews are always treated as a critical part of the system certification process, provided that it is conducted following each NAA's procedures and handbooks to finding compliance with the safety-related regulations § 25.1301 and § 25.1309. This research analysed regulation requirements and software review policies of the FAA, EASA, CASA, and CAAC using a comparative approach to establish the software certification basis and means of compliance. Given that the airborne software review is performed by people, the different working experiences, backgrounds, and technical capabilities of the reviewers may lead to different review conclusions.

An in-depth software review can discover the shortfalls existing in the design and potential risks to safety of aircraft design. This paper studied the technical focuses of airborne software review based on the DO-178C software life cycle process and identified 82 technology aspects through analysis of objectives and activities of each process. This paper also analysed the LOI impact factors of airborne software SOI review, and developed a set of Risk-based SOI reviews and sampling strategies, taking into account the applicable identified technologies and compliance objectives of DO-178C by developing the PTC and TOC matrixes. The study of this paper will help NAAs to maintain software expertise and formulate more effective software review procedures and guidance documents, and carry out corresponding technical research to ensure aircraft safety by conducting in-depth software reviews from a software certification perspective.

In the research process of this project, it was found that an Objective-oriented SOI review method based on DO-178C is meaningful. Software reviewers are required to apply their expertise and experience to judge compliance, while the software developer can provide effective assistance to demonstrate compliant evidence and perform software verification activities. This paper identified the necessity of future study to explore the applicable technical focuses and SOI review strategies for different DALs of airborne software based on each objective of DO-178C.

REFERENCES

[1] Boeing, 2020. *737Max Software Update.* [Online]
Available at: https://www.boeing.com/commercial/737max/737-max-software-updates.page
[Accessed 25 10 2020].

[2] CASA, 2014. *AC 21-50: Approval of software and electronic hardware parts,* Canberra: CASA.

[3] CAST, 2002. *CAST-10 "Literal" Interpretation of Decision Coverage Increases Rigor of Testing Requirements.* [Online]
Available at: https://www.rapitasystems.com/blog/cast-10-literal-interpretation-decision-coverage-increases-rigor-testing-requirements
[Accessed 6 July 2020].

[4] Chen, Y., Yan, L. & Sun, J., 2015. *Civil Aircraft Airborne Software Management.* Beijing: The Aviation Industry Press of China.

[5] COMMITTEE, T. H., 2020. *FINAL COMMITTEE REPORT: THE DESIGN, DEVELOPMENT & CERTIFICATION OF THE BOEING 737 MAX,* USA: THE House COMMITTEE on TRANSPORTATION AND INFRASTRUCTURE.

[6] Dodd, I. & Habli, I., 2012. Safety certification of airborne software: An empirical study. *Reliability Engineering & System Safety,* 98(1), pp. 7-23.

[7] EASA, 2012. *EASA CM – SWCEH – 002 Issue: 01 Revision: 01 Software Aspects of Certification.* [Online]
Available at: https://www.easa.europa.eu/sites/default/files/dfu/certification-docs-certification-memorandum-EASA-CM-SWCEH-002-Issue-01-Rev-01-Software-Aspects-of-Certification.pdf
[Accessed 19 Oct. 2020].

[8] FAA, 2003. *FAA Order8110.49: SOFTWARE APPROVAL GUIDELINES.* [Online]
Available at: https://www.faa.gov/documentLibrary/media/Order/FAA_Order_8110.49.pdf
[Accessed 19 Oct. 2020].

[9] FAA, 2004. *Job Aid: Conducting Software Reviews Prior to Certification.* [Online]
Available at: https://elsmar.com/elsmarqualityforum/attachments/jobaid-r1-1-pdf.14401/
[Accessed 19 Oct. 2020].

[10] FAA, 2011. *Order8110.49 Chg1: Software Approval Guidelines,* Washington: FAA.

[11] FAA, 2017. *AC 20-115D: Airborne Software Development Assurance Using EUROCAE ED-12( ) and RTCA DO-178( ),* Washington: FAA.

[12] FAA, 2018. *Order8110.49 A: Software Approval Guidelines,* Washington: FAA.

[13] Jimenez, J. A. et al., 2020. A Framework for Evaluating the Standards for the Production of Airborne and Ground Traffic Management Software. *IEEE Access,* 8(1), pp. 142-161.

[14] LU, Y. et al., 2011. Coverage analysis of airborne software testing based on DO178B standard. *Procedia Engineering,* I(17), pp. 480-488.

[15] Marques, J. & Yelisetty, S., 2019. AN ANALYSIS OF SOFTWARE REQUIREMENTS

SPECIFICATION CHARACTERISTICS IN REGULATED ENVIRONMENTS. *International Journal of Software Engineering & Applications (IJSEA),* 10(6), pp. 1-15.

[16] Mendis, K. S., 2008. Software Safety and Its Relation to Software Quality Assurance. In: G. G. Schulmeyer, ed. *Handbook of Software Quality Assurance.* Boston: ATECH HOUSE, p. 211.

[17] Rash, W., 2019. *eWEEK.* [Online]
Available at: https://www.eweek.com/mobile/how-software-can-make-an-airplane-crash
[Accessed 30 July 2020].

[18] Rierson, . L., 2013. *Developing Safety-Critical Software: A Practical Guide for Aviation Software and DO-178C Compliance.* 1 ed. Boca Raton: CRC Press.

[19] RTCA, 2011a. *DO-178C: Software Considerations in Airborne Systems and Equipment Certification,* Washington: RTCA, Inc.

[20] RTCA, 2011b. *DO-248C: Supporting Information for DO-178C and DO-278A,* Washington: RTCA, Inc.

[21] Xing, L. & Mu, M., 2015. Research On Airworthiness Standard DO-178B / C′s Object Analysis and Stage of Involvement Review in Airborne Software. *Aeronautical Computing Technique,* 45(5), pp. 97-101.

AUTHORS

Jinghua Sun : senior engineer, mainly study on airborne software, electronic hardware and system engineering areas. CAAC DER.

Samuel Edwards : DASA software specialist, mainly study on airborne software safety assurance and reviews.

Nic Connelly : RMIT Senior Lecturer, has more than 30 years’ experience in the aviation industry, having ever worked for Air services and Virgin Australia.

Andrew Bridge : EASA software, electronic hardware and safety expert.

Lei Zhang : senior engineer, mainly study on process control and quality management.