Dissertation presented to the Instituto Tecnológico de Aeronáutica, in partial fulfillment of the requirements for the degree of Master of Science in the Graduate Program of Computer Engineering, Field of Software Engineering.

Felipe Rodrigo Evangelista Matilde

A RTCA DO-178C ORIENTED METHOD TO DEVELOP ELECTRONIC FLIGHT BAG SOFTWARE

Dissertation approved in its final version by signatories below:

Prof. Dr. Johnny Cardoso Marques

Advisor

Campo Montenegro
São José dos Campos, SP - Brazil
2024

Cataloging-in Publication Data
Documentation and Information Division

Matilde, Felipe Rodrigo Evangelista
A RTCA DO-178C oriented method to develop Electronic Flight Bag Software / Felipe Rodrigo Evangelista Matilde.
São José dos Campos, 2024.
72f.

Dissertation of Master of Science – Course of Computer Engineering. Area of Software Engineering – Instituto Tecnológico de Aeronáutica, 2024. Advisor: Prof. Dr. Johnny Cardoso Marques.

1. EFB. 2. Electronic Flight Bag. I. Instituto Tecnológico de Aeronáutica. II. Title.

BIBLIOGRAPHIC REFERENCE

MATILDE, Felipe Rodrigo Evangelista. **A RTCA DO-178C oriented method to develop Electronic Flight Bag Software**. 2024. 72f. Dissertation of Master of Science – Instituto Tecnológico de Aeronáutica, São José dos Campos.

CESSION OF RIGHTS

AUTHOR'S NAME: Felipe Rodrigo Evangelista Matilde
PUBLICATION TITLE: A RTCA DO-178C oriented method to develop Electronic Flight Bag Software.
PUBLICATION KIND/YEAR: Dissertation / 2024

Felipe Rodrigo Evangelista Matilde
Rua Rui Sérgio Rodrigues de Moura, 842
12.244-465 – São José dos Campos–SP

A RTCA DO-178C ORIENTED METHOD TO DEVELOP ELECTRONIC FLIGHT BAG SOFTWARE

Felipe Rodrigo Evangelista Matilde

Thesis Committee Composition:

Prof. Dr.	Luiz Alberto Vieira Dias	President	-	ITA
Prof. Dr.	Johnny Cardoso Marques	Advisor	-	ITA
Prof. Dr.	Inaldo Capistrano Costa	Internal Member	-	ITA
Prof. Dr.	Breno Bernard Nicolau de França	External Member	-	UNICAMP

ITA

I dedicate this work my wife, to my children and to my parents. They were the ones who motivated me with their smiles and hugs; who supported me by offering their time and energy; and who inspired me to keep doing my best to deliver this dissertation.

Acknowledgments

There is a huge list of people who I have to be thankfull to because, directly or not, they supported me to make this research.

First of all, I would like to give a special thank my beloved wife Camila, who since the beggining motivated me to go for a master degree. She covered me when I needed to study and always encouraged me when I was discouraged.

I am also thank to my beloved children, Mathias and Samuel. They were my true motivation.

I must be grateful to my parents. I know it was not easy for them to raise my sister and I and provided us the best school we could have. Definetly, their dedication impacted us a lot.

Another special thanks go to my professor and friend Johnny Cardoso Marques. He kindly helped me since the first day to build a master research proposition, provided me good feedbacks, instructed me and guided me in the whole jorney.

I also need to thank my colleagues at Embraer. When I first joined the team I was an aeronautical engineer who did not know anything about software development. All I know about software and EFB I learned with them. A special thanks goes to my collegues Maíra, Luigi and Fábio Corrêa who tutored me from the beginning. Another special thanks goes to my leaders Nakata and Bernardo, who sponsored my academic studies.

Glory be to the Father, and to the Son, and to the Holy Spirit. As it was in the beginning, is now, and ever shall be, world without end. Amen.

"Set your hearts on his kingdom first, and on God's saving justice, and all these other things will be given you as well."

— MATTHEW 6, 33

Resumo

O desenvolvimento de EFBs - aplicações utilizadas por pilotos dentro da cabine de aeronaves- tem crescido nos últimos vinte anos. Essas aplicações, em sua grande maioria feitas para tablets iPad ®, ajudam significativamente a operação de cabine ao reduzir a carga de trabalho, otimizar o desempenho da aeronave e remover das aeronaves manuais em papéis. Em 2021 a EUROCAE publicou a norma EUROCAE ED-273 para o desenvolvimento de EFB's trazendo, entre outras coisas, requisitos relacionados à garantia de desenvolvimento. A norma, considerada inédita no mundo dos EFBs, gerou naturalmente a preocupação sobre como se aderir à nova norma. Esse trabalho se propôs a criar um método de desenvolvimento de EFBs que seja aderente às diretrizes da EUROCAE ED-273. O caminho adotado foi o de apontar para a RTCA DO-178C - norma reconhecida no contexto aeronáutico para o desenvolvimento de software embarcado- e tentar identificar semelhanças e diferenças entre as duas normas. Assim sendo, um mapeamento entre as duas normas foi realizado. Além disso, quanto às aplicações legadas (isto é, aplicações que já possuíam histórico em serviço anterior à publicação da norma), esse trabalho se propôs a criar um método para que novos incrementos de software legados sejam aderentes à norma.

Abstract

The development of EFBs - applications used by pilots inside the airplane cabin - has been growing in the last twenty years. These applications, most of them developed for iPad ®, help the pilots considerably to operate the airplane by reducing the workload, optimizing the airplane performance, and removing paper manuals from the airplane. In 2021 EUROCAE published the standard EUROCAE ED-273 for the development of EFBs and introduced, between some other relevant points, some requirements related to software development assurance. The standard, as expected, made EFB manufacturers concerned about how to follow the new standard. This work proposes to create an EFB development method that is compliant with EUROCAE ED-273 guidelines. The adopted method was to point to RTCA DO-178C - a widely recognized standard in aviation for embedded software development - and try to identify similarities and differences between the two standards. Therefore, a mapping between the two standards was performed. Furthermore, in what concerns legacy applications (i.e., applications that already have service history before the standard publication), this work proposes to create a method for new increments in legacy software to be compliant with the standard.

List of Figures

List of Tables

List of Abbreviations and Acronyms

AC	Advisory Circular
ACO	Aircraft Certification Office
AMC	Acceptable Means of Compliance
AMMD	Airport Moving Map Display
ANAC	Agência Nacional de Aviação Civil
ARP	Aerospace Recommended Practice
ATC	Air Traffic Control
CAT.SPA.EFB	European Operational Requirement for Electronic Flight Bag
DAL	Development Assurance Level
DO	Document
EASA	European Union Aviation Safety Agency
ED	EUROCAE Document
EFB	Electronic Flight Bag
EUROCAE	European Organisation for Civil Aviation Equipment
FAA	Federal Aviation Administration
FQL	Function Qualification Level
HMI	Human Machine Interface
ITA	Instituto Tecnológico de Aeronáutica
MOPS	Minimum Operational Performance Standard
QA	Quality Assurance
QRH	Quick Reference Handbook
RTCA	Radio Technical Commission for Aeronautics
SDP	Software Development Plan
TQL	Tool Qualification Level
WG	Working Group

Contents

1 Introduction

1.1 Context

Operating an airplane is not easy. The pilots have to check for weight and balance, communicate with towers, flight crew, and passengers, fly the airplane, navigate from origin to destination, sometimes deal with system failures, and perform many procedures. All of this is due to time constraints and workload.

Electronic Flight Bags (EFBs) were developed to simplify pilots' tasks. Most of these applications are developed for iPad ®and adopted almost universally as an EFB device.

Some EFB application domains include takeoff and landing performance calculation, weight and balance, airport navigation, moving maps applications, briefing, flight plan-

FIGURE 1.1 – A pilot interacting with an EFB software in an iPad®

ning, electronic QRH procedures, and replacement of paper manuals. Figure 1.1 presents a pilot interacting with an EFB.

It is important to highlight that when an airline decides to adopt an EFB as part of its operation, it has to apply for authorization from its civil aviation authority. This type of authorization normally evaluates flight crew training, hardware safety, and computed results correctness.

Due to its benefits, EFBs are widely adopted worldwide. The growing popularity of EFBs stimulated the emergence of many EFB manufacturers and the increasing number of EFB application domains. Consequently, this scenario makes the EFB operational evaluation by civil aviation authorities even harder.

Therefore, in 2021, EUROCAE published the EUROCAE ED-273 document entitled "Minimum Operational Performance Standards (MOPS) for Electronic Flight Bag (EFB) Application." The EUROCAE ED-273 addresses concerns like function eligibility, risk analysis, interface, databases, and security. One of these concerns, in particular, has not been addressed by any other document before, the EFB software development assurance.

1.2 Motivation

EASA adopted EUROCAE ED-273 to comply with the operational requirement CAT.SPA. EFB. 100 (b). Since a new EFB regulation is in place, EFB manufacturers are expected to be concerned about how to comply with it.

The previous concern could be from two different EFB manufacturers: one with many years of service history and many customers worldwide, and one new player about to launch its first EFB release. Naturally, both are expected to face different challenges in showing EFB compliance with EUROCAE ED-273. The first already has an EFB development process and a working EFB product. Is it supposed to re-develop its application? Can it take advantage of its service history to show compliance? And how about the second one? It may be easier for a new player to create a brand-new process that shows compliance with EUROCAE ED-273 since the beginning. These two scenarios will be addressed further in this dissertation.

One important software standard is RTCA DO-178C (RTCA, 2011a), which applies to embedded software. Comparing both documents (i.e., RTCA DO-178C and EUROCAE ED-273) reveals many common aspects of software development assurance. The risk of embedded software is higher than EFB's, and of course, the level of demand is proportionally higher, too. However, it seems that both standards share some similar philosophies.

In that regard, it would be very beneficial for organizations that already have an RTCA DO-178C compliant process to take advantage of it as a means of compliance for the new EUROCAE ED-273. After all, having only one development process instead of two brings efficiency, reduces waste, and keeps all organization employees informed about the process. By the time this dissertation is written, the scenario of an EFB manufacturer that is also an embedded software manufacturer compliant with RTCA DO-178C is the most frequent.

Even EFB manufacturers that do not have an RTCA DO-178C process can take advantage of the similarities between RTCA DO-178C and EUROCAE ED-273. In the year this dissertation is being presented, no literature about EUROCAE ED-273 was found that could help these manufacturers. On the other hand, the amount of literature about RTCA DO-178C available is huge, and the content is instructive.

Additionally, it is convenient to state that EUROCAE ED-273 is a brand-new regulation even for aviation authorities. It may be difficult for aviation regulators to evaluate an EFB application in light of EUROCAE ED-273. In contrast, RTCA DO-178C is a 30-year-old regulation widely known by regulators.

As previously stated, RTCA DO-178C has a risk level proportionally higher than EUROCAE ED-273, and therefore, the number of objectives is equally higher. If compliance with EUROCAE ED-273 is proposed via a RTCA DO-178C oriented method, the number of objectives must be trimmed. Otherwise, it would be too onerous.

1.3 Research question

Both RTCA DO-178C and EUROCAE ED-273 are regulations applied to software development. The first has more objectives than the second, but they seem to share the same philosophies regarding development assurance: both are for aeronautical domain and both are object-oriented.

With that said, the research question is: **Is the development assurance part of EUROCAE ED-273 compatible with RTCA DO-178C?** Based on the research question, our research hypothesis is **There is a subset of RTCA DO-178C that can be used to show compliance with EUROCAE ED-273 but not full compliance.**

1.4 Objective

The objective of this research is to **Create a software development method to be used both by a legacy EFB manufacturer and a new EFB manufacturer to show compliance**

with the development assurance part of EUROCAE ED-273.

1.5 Scope

This development method aims to fulfill most EUROCAE ED-273 development assurance requirements through RTCA DO-178C objectives. To avoid being too onerous, it must select only applicable RTCA DO-178C objectives. Therefore, it is expected that, to follow the method proposed by this dissertation, the applicant already has a RTCA DO-178C compliant process.

In this section, it is convenient to explain why the scope of the research is limited to the EUROCAE ED-273 development assurance part to the detriment of the other parts. As it will be further explained in section 2.2, it contains other parts like function eligibility and human-machine interface. It is the understanding of this author that the requirements from the other parts, for instance, are harder to show compliance by an alternative method as it will be proposed for the development assurance part. The difficulty comes especially from these requirements, which are particular to EFBs.

It is important to emphasize that following the proposed method does not show full compliance with EUROCAE ED-273. The other parts have to be understood and followed. Additionally, as a disclaimer, this dissertation does not constitute any operational approval, and civil aviation authority evaluation is still required.

As stated in the objective, both scenarios of a **new player (Scenario 1)** and a **legacy EFB manufacturer (Scenario 2)** will be considered. That means both scenarios can benefit from the proposed development method. However, the legacy EFB scenario is the most relevant in the year this dissertation is being presented. This is explained by the fact that there were more EFBs released before the publication than EFBs released after 2021.

1.6 Research method

To achieve the objective, a set of five steps were identified. Figure 1.2 represents the five steps.

Step 1 was focused on reading and studying aviation software development standards. Three main documents were selected to achieve this step: RTCA DO-178C, RTCA DO-330 (Software Tool Qualification Considerations), and EUROCAE ED-273. Each document was studied carefully to comprehend the context they are applied to and to identify commonalities and differences between them. Additional information is provided in Sec-

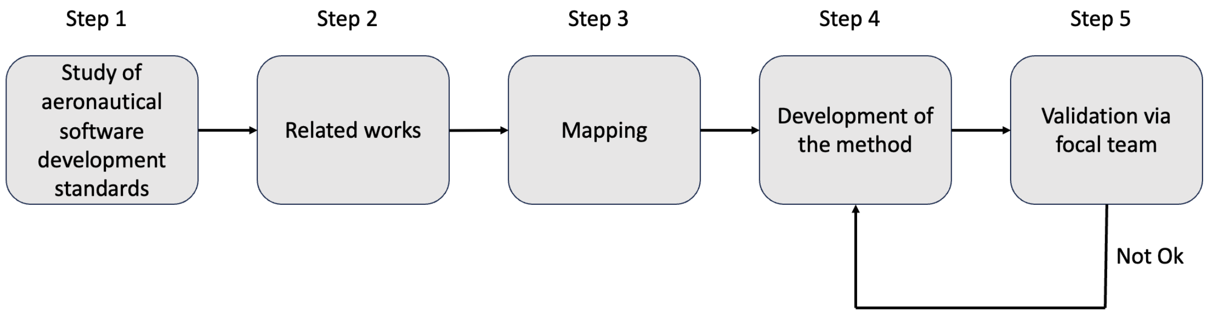

FIGURE 1.2 – Research method flowchart

tions 2.1 and 2.2. Step 2 was to read and study related works. The related works are summarized in Section 2.4. They were selected because they also performed mapping between two software standards. The related works were useful in identifying the adopted methodologies before performing the mapping to RTCA DO-178C.

Step 3 was the mapping itself. After a deep reading of both EUROCAE ED-273 and RTCA DO-178C, it was time to identify objectives and activities from RTCA DO-178C that could be used to show compliance with some of its development assurance requirements. Step 4 focused on building the development method, which is the main contribution of the present dissertation. Step 5 was the evaluation via the focal team. In this step, qualified experts in the domain were invited to evaluate the proposed method. After their evaluation, if the proposed development method is not approved, then it shall be revised.

1.7 Dissertation organization

In addition to Chapter 1, which covers the introduction, the other 4 chapters are included in this document. Chapter 2 covers the theoretical background, providing all the content to support the Development Method proposed in this research. Chapter 3 presents the main contribution of this dissertation, which is a Development Method to be used by EFB manufacturers to comply with EUROCAE ED-273. Chapter 4 presents the evaluation results of the proposed EFB Development method. Finally, Chapter 5 presents the conclusion, which summarizes this research, discusses the research question and hypothesis, presents the threads to validity and proposes future work.

2 Theoretical Background

This chapter provides the basic knowledge of the two main software development standards for civil aviation and some related works or documents.

2.1 RTCA DO-178C

2.1.1 Contextualization

Over the past four decades, the pervasive influence of software has profoundly reshaped various facets of our daily lives, particularly evident in the evolution of household technologies. A prime example lies in the transformation of television sets from mere receivers of analog signals to sophisticated multimedia hubs. Modern TVs now integrate advanced software components that enable diverse functionalities such as internet connectivity, voice recognition, and personalized content delivery.

This integration underscores the profound convergence of software and hardware, a phenomenon that has long been reshaping consumer electronics (ANDERSON; SMITH, 2019). As software continues to evolve, it plays an increasingly pivotal role in enhancing the functionality and user experience of devices (BROWN; JOHNSON, 2021). This trend reflects broader shifts towards interconnected, smart technologies that anticipate and adapt to user needs (JONES; PATEL, 2020).

Software has gained significant traction within the aviation industry. Yet, there exist notable distinctions between the development of aviation software and software for other applications, such as television. In this highly regulated realm, ensuring the safety of flight operations takes precedence, necessitating meticulous attention during software development to achieve the requisite level of airworthiness.

Aviation software encompasses a diverse array of areas, each subject to its own set of stringent development standards. For example, aeronautical databases integrated into aircraft systems must adhere to the guidelines outlined in RTCA DO-200B (RTCA, 2015). Similarly, compliance with RTCA DO-330 (RTCA, 2011b) is imperative for tools intended for use in the development process of aviation software. Moreover, when software is to be embedded within an aircraft's systems, adherence to the rigorous standards delineated in RTCA DO-178 is mandated.

2.1.2 History

In 1982, RTCA and EUROCAE collaboratively developed and published RTCA DO-178 (EUROCAE ED-12) to establish a comprehensive framework for embedded aviation software development. As noted by Rierson (2013), the standard underwent an update in 1985, resulting in RTCA DO-178A, which incorporated more robust software engineering principles.

It wasn't until 1992 that RTCA DO-178B was introduced. This version notably omitted explicit mention of system requirement validation, now assumed to be conducted within the broader system development context. RTCA DO-178B marked a significant expansion from its predecessor, incorporating key enhancements such as the introduction of development assurance levels and an emphasis on objective-oriented practices. It was under the enforcement of RTCA DO-178B that the standard garnered widespread adoption among manufacturers and gained recognition as a compliance mechanism by civil aviation authorities (AC 20-115D).

The fourth edition of the standard, RTCA DO-178C, was released in 2011. While not introducing a revolutionary approach to embedded software development, RTCA DO-178C clarified concepts that had previously been ambiguous within the aviation community, addressing lingering uncertainties from RTCA DO-178B.

The RTCA DO-178C is a well-established and mature standard, offering invaluable guidance to the aviation industry in developing dependable and secure software solutions. Although primarily focused on embedded software, RTCA DO-178C is often embraced in various aviation software domains as a valuable reference or as an alternative means of compliance.

2.1.3 Fundamental concepts

2.1.3.1 Software in the context of the system

One fundamental concept lies in the inherent interdependence of embedded software within a system. Put simply, the software and hardware components jointly define the system's functionality and purpose. It is imperative to recognize that for a system to operate effectively, both the software and hardware must execute their respective roles and interact in accordance with the intended design (SMITH, 2018)

As highlighted by Rierson (2013), RTCA DO-178 was published before SAE ARP-4754 (SAE INTERNATIONAL, 2010), which is nowadays the accepted standard for aircraft development. This is why RTCA DO-178 and DO-178A included in their scope the validation of the system requirement. However, RTCA DO-178B delimited the scope of what software development is and removed, therefore, the validation of the system requirement from it.

The concept underpinning SAE ARP-4754 revolves around the V-Model methodology, wherein high-level requirements for aircraft are decomposed into system-level requirements. Following this methodology, in order to effectively construct a system that meets the aircraft's overarching requirements, these system requirements must be allocated to its constituent components, whether software or hardware.

It is crucial to maintain clarity regarding the distinction between system, software, and hardware elements. While a system comprises both software and hardware components, inherent differences exist between them. It is precisely at this juncture that the scope of RTCA DO-178C comes into play.

The objective of RTCA DO-178C is to ensure the correct development of software as expected by the system. Hence, it is not within the purview of the software layer to validate the higher-level requirements of the system.

2.1.3.2 Development Assurance Level

As previously emphasized, it is crucial to comprehend the scope of software within the context of the overall system. Consequently, it is imperative to assess the potential impact that a failure in the software could have on the integrity of the entire system.

The concept of reliability holds significant importance in developing safe aircraft or systems. Reliability entails ensuring a sufficiently low probability of system failure and, by extension, failure of its components. Regarding "adequate" reliability, it implies recognizing the varying degrees of consequences associated with system failures. Put, the probability of a system failure that could result in severe consequences must be proportionately lower than that of a failure leading to mere inconvenience (DINGLE; TOOLEY, 2005).

In the development of an airplane and its associated systems, the anticipated probability of component failures is typically distributed through failure trees. However, assigning reliability in terms of probability becomes impractical when considering software as a component. This is due to unique properties inherent to software that hardware does not share; the software does not degrade over time or experience fatigue, stress, or physical consumption.

The concept of Development Assurance Level (DAL) is introduced to address this challenge. Under RTCA DO-178C, five levels ranging from A to E are established, each characterizing the potential impact of software failure on the system. Level A represents the most critical scenario, while level E indicates no impact on safety whatsoever.

The DAL designation dictates how much a development process must adhere to RTCA DO-178C's development assurance objectives. Consequently, the higher the DAL, the more stringent the compliance requirements outlined in this standard.

2.1.3.3 Objective oriented

As explained in Section 2.1.2, RTCA DO-178B introduced the concept of objective-oriented. This concept is strongly linked to the DAL concept previously explained. RTCA DO-178B and RTCA DO-178C do not prescribe a specific process for achieving compliance in embedded software development. Instead, they outline a set of criteria that the development process must satisfy to ensure compliance. These criteria are enumerated in ten tables presented in RTCA DO-178C's Appendix A. Figure 2.1 from Marques *et al.* (2023) illustrates the organization of these tables. Each criterion listed in these tables is referred to as an objective, reflecting the standard's 'objective-oriented" approach rather than being process-centric.

The relationship between Development Assurance Levels (DAL) and Objectives is integral to RTCA DO-178C. The standard specifies tables delineating a certain number of objectives for each DAL. As previously elucidated, the severity of consequences associated with a software error increases with higher DALs. Consequently, software assigned a higher DAL must demonstrate compliance with more objectives than software assigned a lower DAL.

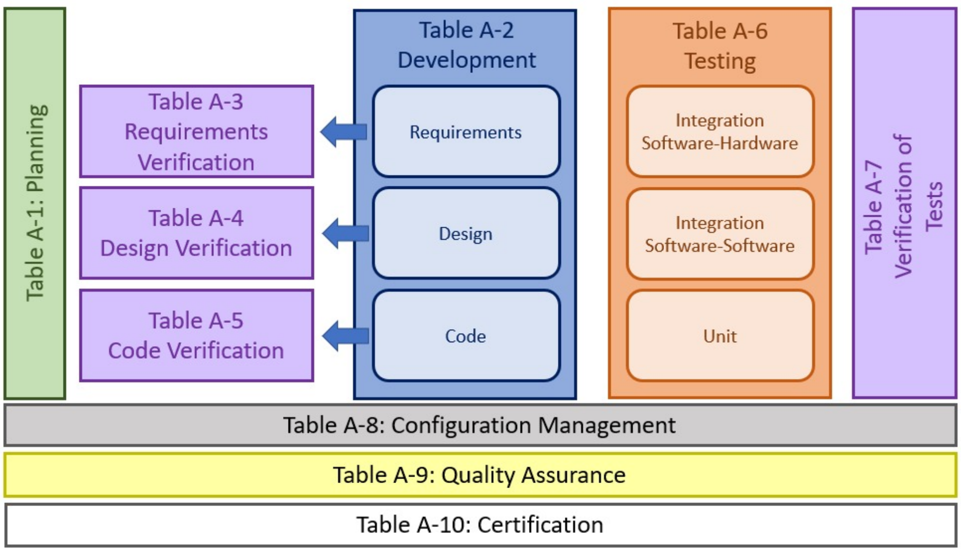

FIGURE 2.1 – Organization of Tables in RTCA DO-178C from Marques *et al.* (2023)

Furthermore, it is important to note that each objective comprises multiple activities. This implies that to achieve compliance with a particular objective, all activities associated with that objective must also be successfully demonstrated.

2.2 EUROCAE ED-273

2.2.1 Background

Integrating an Electronic Flight Bag (EFB) into real operations necessitates approval from the relevant civil aviation authority. The requesting operator must demonstrate various aspects, including adequate training for their crew on the specific EFB, the resilience of the EFB hardware under extreme flight conditions (e.g., depressurization), and the absence of interference between EFB wave signals and avionics systems. Moreover, they must ensure that the content of the EFB, particularly the software, is accurate and will not inadvertently impact flight operations.

The verification of software correctness is of particular concern, presenting a formidable challenge for civil aviation authorities due to the proliferation of applications available in the market. Over the past two decades, EFBs have gained popularity among operators, leading to a vast array of available applications encompassing functions such as aircraft performance calculation, moving maps, flight planning, briefings, weight and balance computations, digital logbooks, and paper manuals' replacement.

The absence of a standardized development framework for EFB applications compounds this challenge. Recognizing this need, EUROCAE established the Working Group WG-106 "EFB Applications" to examine and propose an industry-consensus standard for EFB development.

2.2.2 Structure and content

EUROCAE ED-273 is a sixty-nine-page standard composed of four chapters. Chapter 2 is the most relevant for this dissertation, and a deeper explanation will be provided below. In any event, a brief description of the other chapters is also provided next.

Chapter 1 is titled "Introduction" and presents the document's purpose and scope, the document structure, wording rules of the standard, definitions, and acronyms. In this first chapter, it is explained that the EUROCAE ED-273's purpose is to provide "MOPS for the design, development, evaluation, and validation of EFB applications and their functions". Additionally, it adds that "the MOPS include requirements, recommended practices and guidelines".

Chapter 2's title is "General Minimum Operational Performance Standard" and, as explained by the title, applies to all types of EFBs. The second chapter is divided into the following sections:

- Function Eligibility
  This section aims to define what functionalities can be implemented by an EFB and which ones require a certified airworthiness device or process. To do that, the standard provides a methodology to make a functional breakdown and identify the functions in an EFB.

  The section also presents a table with no authorized intended uses. For example, it is forbidden to use the following categories: "Aviate or Fly" the airplane and "Navigate." According to that table, it is also forbidden to communicate with ATC through an EFB.

- Safety Risk Assessment
  This section presents a methodology for identifying hazards that an EFB can promote, building a reasonable plan to mitigate and prevent damages, and validating the plan.

- Human Machine Interfaces
  This section presents some requirements for data validation and colors to be used or avoided in the airplane cockpit, among other concerns. The goal of this section is to guarantee that the flight crew will not be distracted from their duties and that all relevant information presented and asked by EFB is consistently planned in terms of HMI.

- Development Assurance
  In this section, if an EFB function presents a residual risk, its FQL shall be High. On the other hand, if there is no residual risk, the FQL of an EFB function shall be low.

  From this FQL definition, the standard proposes a table correlating each FQL to a set of development assurance objectives. All EFB functions have to comply with FQL Low-related objectives. However, if an EFB function's FQL is High, then it has to comply with a few more objectives.

  The table contains a set of 17 objectives, and each one of them is presented in a row. The table also classifies the requirements according to six categories. The six categories are Development Plan, Operational Requirements, Software Development, Configuration Management, Application Release, and Quality Assurance Process. Figure 2.2 illustrates the relationship between the process areas.

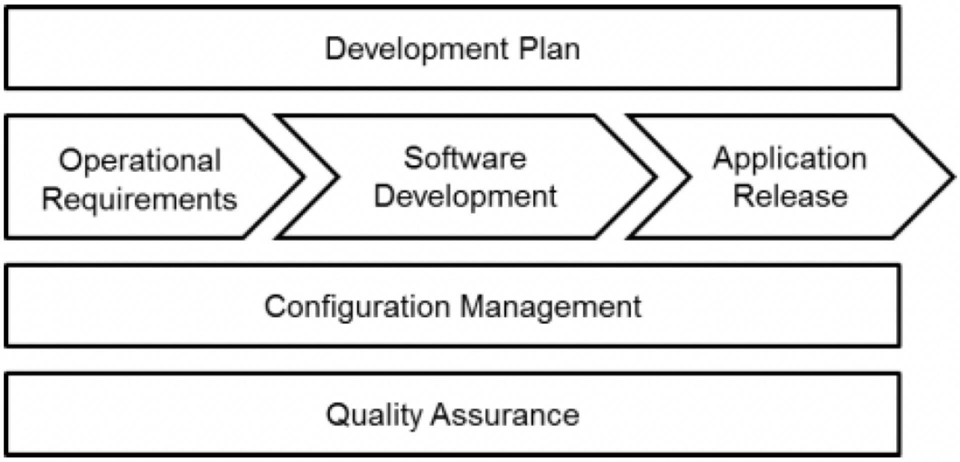

FIGURE 2.2 – Processes of an EFB application development (EUROCAE, 2021)

This table contains 5 columns: Requirements Category, EUROCAE ED-273 Section ID, Description of the objective, FQL Allocation High, and FQL Allocation Low.

From this table, it is possible to identify the sets of objectives applicable to each FQL. Unlike RTCA DO-178C, no activity is performed for any EUROCAE ED-273 objective.

- Databases

This section is related to a non-real-time acquisition database and does not cover parameter data items.

From an EFB point of view, EUROCAE ED-273 understands that two questions must be answered to identify the appropriate requirements for the database. The first question is: Does the database contribute to residual risk? The second question is: Is the database part of the EFB application?

With these two yes-no questions answered, the database can be situated among four possible combinations. For each of these four possible combinations, EUROCAE ED-273 defines requirements such as establishing a protocol or a document to specify the database structure precisely or to assure an appropriate quality level of the content.

- Security

This section defines a set of requirements related to EFB security. For instance, check the format and range of inputs, monitor for any third party, publish security alerts, correct any vulnerability found, or justify the lack of risk in case of security findings. If a residual risk is found in the Safety Risk Assessment, the EFB must comply with a few more security requirements.

Chapter 3' title is "Specific Minimum Operational Performance Standard". It defines a special set of requirements applicable to six EFB application domains. There is no additional requirement for development assurance in Chapter 3. The six domains are:

- Aircraft performance and mass and balance calculation functions;
- Function displaying own-ship position;
- Airport Moving Map Display(AMMD) function;
- Weather function;
- Electronic checklist function; and
- Electronic signature function.

Chapter 4's title is "Operational and Installation Instructions". It covers installation and administration instructions that are to be provided to users.

2.3 DOT-FAA-AR-01-116 Software Service History Handbook

In 2002, the Federal Aviation Administration (FAA) and the Department of Transportation (DOT) in the US released their final research report on assessing service history data for onboard applications. The executive summary explicitly states that this report aids industry stakeholders and the FAA in formulating and evaluating product service history data for certification credit.

According to the report, RTCA DO-178B proposes that service history could serve as an alternative means of compliance with certain requirements. However, the committee involved in the DOT/FAA research highlights the considerable challenge of qualitatively and quantitatively assessing the collected service history data.

The report delineates two essential criteria the applicant must meet: relevancy and sufficiency. Additionally, it identifies four key components of service history: problem reporting, operations, environment, and time. Each of these components must demonstrate the presence of relevant and sufficient data.

Furthermore, the report includes appendices containing spreadsheets with questions designed to guide applicants in systematically evaluating their service history. However, it emphasizes that this document is not an official FAA publication that can be used to demonstrate compliance with RTCA DO-178B. Therefore, it advises applicants to consult with their local civil aviation authority for further guidance and clarification.

2.4 Related work

As previously mentioned, the software development for embedded software is already standardized in aviation through the use of RTCA DO-178C (RTCA, 2011a) and its supplements. Recent works have discussed advances and new software development methods in this field. The authors of this work identified and grouped related works into 6 (six) themes:

- Impacts on the transition from RTCA DO-178B (RTCA, 1992) to RTCA DO-178C (RTCA, 2011a), as explored in the works of Marcil (2012) and Youn *et al.* (2015);
- Model-Based Development, as explored in the works Paz e Bousaidi (2016) and Sarkis *et al.* (2020);
- Use of Agile Methods in Software Development, as explored in the works Vander-Leest e Buter (2009) and Marsden *et al.* (2019);
- Formal Verification, as seen in the works of Moy *et al.* (2013) and Marques e Cunha (2017);
- Aircraft Embedded Software Loading, as reported in the works of Marques *et al.* (2019) and Marques *et al.* (2021);
- Mapping between standards, models, and norms with a focus on safety, as reported in the work of Bhansali (2005) and Ferreirós e Dias (2015).

The work of Marcil (2012) explains the benefits of formal methods and object-oriented technology that RTCA DO-178C offers in conjunction with RTCA DO-332 (RTCA, 2011d) and RTCA DO-333 (RTCA, 2011e). It also focuses specifically on modeling in software development and the qualification of tools that automate or facilitate the verification and validation of avionics applications built from models to ensure there are no unintended functions.

The work of Youn *et al.* (2015) presents an overview of the guidelines for aeronautical software contained in RTCA DO-178C and supplementary documents. It also addresses the similarity between RTCA DO-178B and DO-178C, reviewing the fundamentals of verification philosophy and an overview of crucial guidance included in RTCA DO-178C.

The work of Paz e Bousaidi (2016) presents a framework for using models for compliance with RTCA DO-178C. They also analyzed other approaches compared to the proposed framework, highlighting similarities, differences, strengths, and weaknesses.

The work of Sarkis *et al.* (2020) presented a set of guidelines for development based on Aeronautical Embedded Software models, ensuring compliance with RTCA DO-178C and RTCA DO-331 RTCA (2011c). In addition to the drivers, a case study is presented.

The work of VanderLeest e Buter (2009) provides a detailed analysis of the main agile practices, with a preliminary assessment of their ease of implementation. The authors highlighted that the transition to agile development does not require sudden and radical changes but can be accomplished by incorporating agile methods into an existing process.

The work of Marsden *et al.* (2019) shows how apparent contradictions between agile practices and aeronautical software certification objectives were resolved in several Airbus projects and quantifies the resulting financial gains.

The work of Moy *et al.* (2013) describes some of the goals and activities in the area of formal methods, explaining how these methods can be used instead of testing in an RTCA DO-178C context. The work summarizes the practical experience of Dassault-Aviation and Airbus in successfully applying formal methods for developing aeronautical embedded software.

The work of Marques e Cunha (2017) provides some scenarios for database verification using the RTCA DO-178C and RTCA DO-200B (RTCA, 2015) standards, including the use of tool qualification when processes are eliminated, reduced, or automated by the use of software tools without reviewing the output produced by such tools.

The work of Marques *et al.* (2019) characterizes the scenarios of software loading on aircraft and treatments for possible threats involving information security in this process. This work was later improved to a framework in the work of Marques *et al.* (2021) that presents a set of reusable requirements and general testing procedures for software loading involving manual and automatic checks. The authors believe that the framework can help smaller companies, especially those entering the market, to incorporate software loading capabilities into systems development.

In their study, Ferreirós e Dias (2015) proposed a method to estimate the distance a CMMI-DEV compliant team would have to overcome to be an embedded software provider compliant with RTCA DO-178C.

The method was based on ten tables - one for each RTCA DO-178C Appendix A table - whose columns represented each RTCA DO-178C objective and activity and whose rows represented each CMMI-DEV practice. Figure 2.3 presents an illustration of these tables.

The authors assessed the intersection of both standards in each of the ten tables. By doing so, they could estimate the level of accomplishment of RTCA DO-178C by a CMMI-DEV-compliant team.

The authors concluded that, despite similar philosophies and concerns in both stan-

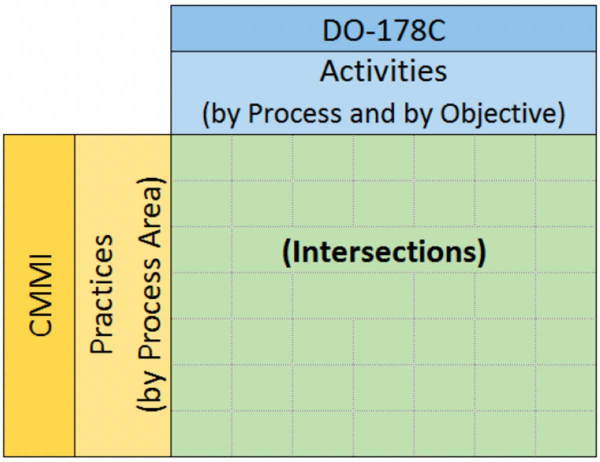

FIGURE 2.3 – Intersection table scheme proposed by (FERREIRóS; DIAS, 2015)

dards, it is impossible to adopt CMMI-DEV to comply with RTCA DO-178C. This statement of non-compliance is presented by the authors, especially when they point to the Verification of Verification Process Result and Certification Liaison Process,

A comparison of four compliance means for embedded software was performed by Yan (2009). The author evaluated RTCA DO-178B, Safety Engineering Approach for Software Assurance, Capability Maturity Model (e.g. CMMI), and Alternative Methods (e.g. formal methods and service history).

The author recommends checking first with local authorities to determine whether the alternative means of compliance are acceptable. However, the author performed a survey and concluded that it would be hard to convince aviation players to deviate from RTCA DO-178C. According to the author, the industry and aviation authorities widely adopted the RTCA DO-178C.

In his study, Bhansali (2005) wanted to find a subset of common attributes or objectives among 16 safety-related development standards. He concluded that there is a subset of common attributes or objectives (e.g. System Safety Assessment, Software Requirement Validation, and Traceability Analysis). However, it can be deduced from his work that different safety-related areas emphasize the software development process differently.

2.4.1 Contribution to the research

According to Ferreirós e Dias (2015), the alignment between standards does not necessitate a perfect match to derive valuable insights. Despite CMMI-DEV not aligning perfectly with RTCA DO-178C, Ferreiros identified key areas where a CMMI-DEV compliant team can focus to achieve RTCA DO-178C compliance. Furthermore, the method of comparing standards using tables was noted for its objectivity and visual efficiency.

Research by Yan (2009) underscores the significant role of RTCA DO-178C in aviation despite the existence of alternative compliance methods. Therefore, it would be beneficial for an Electronic Flight Bag (EFB) development method to adhere to RTCA DO-178 standards closely.

Bhansali (2005) emphasized that when attempting to identify common attributes or objectives among a set of standards, different areas of expertise may prioritize different aspects of the software development process. Consequently, there may be specific EFB development objectives that do not align with counterparts in RTCA DO-178C.

3 EFB Development Method

This chapter unveils the primary contribution of this research: an Electronic Flight Bag (EFB) development method designed to secure approval from civil aviation authorities as an acceptable means of compliance (AMC) for the software development assurance component outlined in EUROCAE ED-273. Figure 3.1 presents the proposed development method for EFBs.

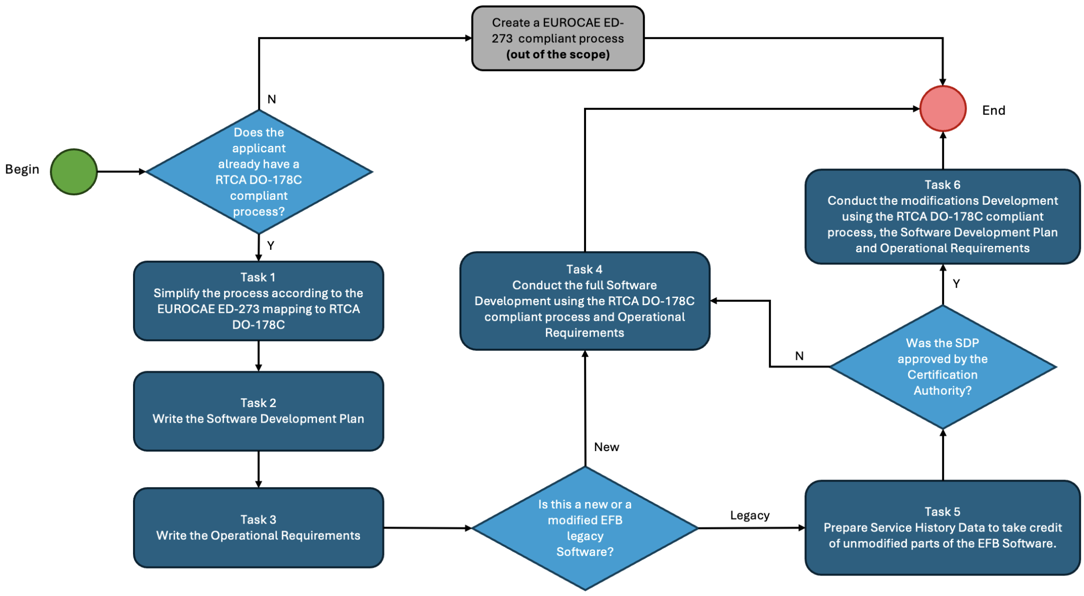

FIGURE 3.1 – Proposed development flowchart

Before executing any Task from Figure 3.1 a question about having or not an RTCA DO-178C compliant process must be answered. All the Tasks from 1 to 6 presented in Figure 3.1 and in this Section are to be executed only when there is an RTCA DO-178C compliant process. When the EFB applicant does not have an RTCA DO-178C compliant process then he/she must submit to his/her regulator an EUROCAE ED-273 compliant process. In this case, creating such a process is out of the scope of this research, even though the applicant could be inspired by the content here presented.

3.1 Task 1

Task 1 consists of simplifying the RTCA DO-178C compliant process according to EUROCAE ED-273 mapping to RTCA DO-178C. Assuming the applicant already has an RTCA DO-178C compliant process, it certainly addresses all applicable objectives and activities from the standard. Therefore, this dissertation proposes to perform a tailoring of the RTCA DO-178C compliant process so that only objectives and activities from Figure 3.3 are effectively executed.

Figure 3.3 illustrates the proposed mapping from EUROCAE ED-273 objectives to RTCA DO-178C objectives and activities. For the EUROCAE ED-273 portion of Figure 3.3, five columns are presented. The first column denotes the process area from which the objectives originate. The second column specifies the particular section within the EUROCAE ED-273 standard where each objective is located. The third column provides a concise description of the objective as outlined in the standard. The fourth and fifth columns indicate the applicability of these EUROCAE ED-273 objectives to FQL High and Low, respectively.

As delineated in Section 2.1, RTCA DO-178C tables comprise objectives, each composed of multiple activities. Accordingly, Figure 3.3 features three columns of RTCA DO-178C: Table, Objective, and Activity.

Figure 3.2 depicts the rationale behind the mapping. An objective from EUROCAE ED-273 is mapped, as needed, to one or more objectives from RTCA DO-178C, although not necessarily to all activities of those RTCA DO-178C objectives. For example, according to Figure 3.3, EUROCAE ED-273 objective 2.4.2.5.1 (EFB Application conformity) is mapped to objectives 1, 2, and 5 from RTCA DO-178C Table A-9. Furthermore, it is noted that objective 1 from RTCA DO-178C Table A-9 encompasses activities 8.2.b, 8.2.h, and 8.2.i, yet for the purpose of the mapping, only activity 8.2.b is included. Similarly, some objectives from EUROCAE ED-273 do not have equivalents in RTCA DO-178C, such as EUROCAE ED-273 item 2.4.2.2.1.

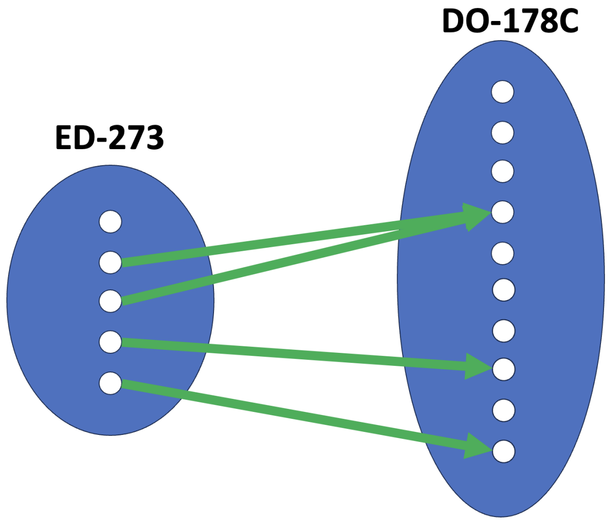

FIGURE 3.2 – Mapping diagram from EUROCAE ED-273 and RTCA DO-178C

EUROCAE ED-273					DO-178C		
	Development process objective		FQL allocation		Table	Objective	Activity
	Section	Description	High	Low			
Development plan	2.4.2.1.1	Minimum considerations	x	x	Table A-1 Software Planning Process	1 - The activities of the software life cycle processes are defined.	4.2.a, 4.2.c, 4.2.d, 4.2.e, 4.2.g, 4.2.i, 4.2.l, 4.3.c
						3 - Software life cycle environment is selected and defined.	4.4.1, 4.4.2.a, 4.4.2.b, 4.4.2.c, 4.4.3
	2.4.2.1.2	Additional considerations	x			4 - Additional considerations are addressed.	4.2.f, 4.2.h, 4.2.i, 4.2.j, 4.2.k
					Table A-2 Software Development Process	5 - Derived low-level requirements are defined and provided to the system processes, including the system safety assessment process	5.2.2.a, 5.2.2.b, 5.2.2.c
					Table A-8 Software Configuration Management Process	1 - Configuration items are identified.	7.2.1.b
					Table A-1 Software Planning Process	4 - Additional considerations are addressed.	4.2.k
Operational requirements	2.4.2.2.1	EFB Function operational requirements definition	x	x	NOT APPLICABLE	NOT APPLICABLE	NOT APPLICABLE
	2.4.2.2.2	EFB application architecture definition	x	x	Table A-2 Software Development Process	3 - Software architecture is developed	5.2.2.a, 5.2.2.d
	2.4.2.2.3	EFB Function Operational Requirements validation	x	x	NOT APPLICABLE	NOT APPLICABLE	NOT APPLICABLE
	2.4.2.2.4	EFB Function compliance with operational requirements	x	x	NOT APPLICABLE	NOT APPLICABLE	NOT APPLICABLE
Software development	2.4.2.3.1	EFB Function software requirements definition	x		Table A-2 Software Development Process	1- High-level requirements are developed	5.1.2.a, 5.1.2.b, 5.1.2.c, 5.1.2.d, 5.1.2.e, 5.1.2.f, 5.1.2.g, 5.1.2.j 5.5.a
						2-Derived high-level requirements are defined	5.1.2.h, 5.1.2.i
	2.4.2.3.2	EFB function software requirements validation	x		Table A-3 Verification of Outputs of Software Requirements Process	1- Software high-level requirements comply with system requirement	6.3.1
						2-High-level requirements are accurate and consistent	6.3.1
						3-High-level requirements are compatible with target computer	6.3.1
						4-High-level requirements are verifiable	6.3.1
						6-High-level requirements are traceable to system requirements	6.3.1
	2.4.2.3.3	EFB function compliance with software requirements	x		Table A-6 Testing of Outputs of Integration Process	1- Executable Object Code complies with high-level requirements	6.4.2, 6.4.2.1, 6.4.3, 6.5
						2-Executable Object Code is robust with high-level requirements	6.4.2, 6.4.2.2, 6.4.3, 6.5
					Table A-7 Verification of Verification Process Results	1-Test procedures are correct.	6.4.5
						2-Test results are correct and discrepancies explained.	6.4.5
						3-Test coverage of high-level requirements is achieved.	6.4.4.1
Configuration Management	2.4.2.4.1	Configuration identification	x	x	Table A-8 Software Configuration Management Process	1 - Configuration items are identified	7.2.1
	2.4.2.4.2	Baselines establishment	x			2 - Baselines and traceability are established.	7.2.2
	2.4.2.4.3	Problem reporting	x	x		3 - Problem reporting, change control, change review, and configuration status accounting are established	7.2.3
	2.4.2.4.4	Change control	x			3 - Problem reporting, change control, change review, and configuration status accounting are established	7.2.4
	2.4.2.4.5	Archive	x	x		4 - Archive, retrieval, and release are established	7.2.7
Application Release	2.4.2.5.1	EFB Application conformity	x	x	Table A-9 Software Quality Assurance Process	1- Assurance is obtained that software plans and standards are developed and reviewed for compliance with this document and for consistency	8.2.b
						2 - Assurance is obtained that software life cycle processes comply with approved software plans.	8.2.f, 8.2.d
						5 - Assurance is obtained that software conformity review is conducted.	8.3
	2.4.2.5.2	Impact analysis of known issues	x	x	Table A-9 Software Quality Assurance Process	2 - Assurance is obtained that software life cycle processes comply with approved software plans.	8.2.d
						5 - Assurance is obtained that software conformity review is conducted.	8.3
					Table A-8 Software Configuration Management Process	3 - Problem reporting, change control, change review, and configuration status accounting are established.	7.2.3
Quality Assurance Process	2.4.2.6	Quality assurance	x	x	Table A-9 Software Quality Assurance Process	1- Assurance is obtained that software plans and standards are developed and reviewed for compliance with this document and for consistency	8.2.b
						5 - Assurance is obtained that software conformity review is conducted.	8.3

FIGURE 3.3 – Mapping between EUROCAE ED-273 and RTCA DO-178

Observations from Figure 3.3 reveal two key insights. Firstly, there exists an extensive mapping from EUROCAE ED-273 to RTCA DO-178C, indicating a substantial alignment between the two standards. However, on the other hand, several requirements from

EUROCAE ED-273 lack a direct equivalent objective in RTCA DO-178C.

As depicted in Figure 3.3, the requirements from EUROCAE ED-273 that remain unmapped to RTCA DO-178C include 2.4.2.2.1 (EFB Function operational requirements definition), 2.4.2.2.3 (EFB Function Operational Requirements validation), and 2.4.2.2.4 (EFB Function compliance with operational requirements). The absence of equivalence in RTCA DO-178C for these requirements is primarily due to their focus on operational requirements.

EUROCAE ED-273 defines operational requirements, outlined in item 2.4.2.2, as functionalities aimed at assisting pilots in performing their duties. This aspect is not addressed within the scope of RTCA DO-178C.

It is worth emphasizing that RTCA DO-178C pertains specifically to embedded software. Embedded software operates within a system; therefore, it is nonsensical for the software component to introduce functionalities beyond what the system was originally designed for.

In essence, requirements are formulated and validated during system development at the system level, rather than at the software level. This is one of the fundamental reasons why RTCA DO-178C does not validate system requirements. In RTCA DO-178C, all development activities originate from and align with system requirements.

An EFB application cannot be viewed as a system component; it exists within an environment where pilots utilize the EFB to carry out their tasks. Therefore, the concept most closely resembling EUROCAE ED-273 operational requirements would be system requirements, which fall outside the scope of RTCA DO-178C.

The EUROCAE ED-273 standard distinguishes between operational requirements and software requirements. According to the standard, all Function Qualification Level (FQL) applications must address operational requirements, while only FQL High applications must address software requirements.

In contrast, RTCA DO-178C introduces two categories of software requirements: High-level Requirements (HLR) and Low-level Requirements (LLR). HLRs provide a high-level overview of the software's functionality, while LLRs offer detailed specifications on how the software will implement the HLRs.

The author posits that EUROCAE ED-273 operational requirements and software requirements for an Electronic Flight Bag (EFB) equate to system requirements and HLR requirements for embedded software, respectively. Since system requirements fall outside the scope of RTCA DO-178C, EUROCAE ED-273 objectives 2.4.2.2.1, 2.4.2.2.3, and 2.4.2.2.4 do not have equivalent objectives in RTCA DO-178C.

Additionally, there is no EUROCAE ED-273 objective specifically addressing LLR requirements. This absence aligns with the understanding that EFBs, being of lower criticality compared to RTCA DO-178C Design Assurance Level (DAL) D applications, do not necessitate LLR-related objectives. Consequently, the lack of LLR-related objectives in EUROCAE ED-273 is consistent with the absence of LLR-related objectives for RTCA DO-178C DAL D applications.

A further explanation about how to comply with these operational requirement-related objectives will be provided in Section 3.3.

3.2 Task 2

Task 2 consists of writing the Software Development Plan (SDP). This is a plan from RTCA DO-178C that contains a description of the software development procedures and software life cycle(s) (RTCA, 2011a). RTCA DO-178C requires more plans than SDP. However, EUROCAE ED-273 objective 2.4.2.1.1 requires only one, called Application Development Plan.

According to Figure 3.1, equivalence between SDP and the EFB's Application Development Plan is being proposed. The applicant does not need to write the other plans RTCA DO-178C prescribes.

SDP shall contain two important components:

- It shall present this dissertation's proposed method, so the Certification Authority can evaluate it
- It shall contain the service history credit if the applicant intends to take credit for it (explained in Task 5.2).

3.3 Task 3

As explained in Section 3.1, the only EUROCAE ED-273 objectives that are not mapped to RTCA DO-178C objectives are 2.4.2.2.1 (EFB Function operational requirements definition), 2.4.2.2.3 (EFB Function Operational Requirements validation) and 2.4.2.2.4 (EFB Function compliance with operational requirements). It was also stated that this lack of mapping is because system requirements are out of RTCA DO-178C's scope.

In this case, the EFB applicant has to create a method to define, validate, and verify the operational requirement to fulfill EUROCAE ED-273 objectives 2.4.2.2.1 (EFB

Function operational requirements definition), 2.4.2.2.3 (EFB Function Operational Requirements validation) and 2.4.2.2.4 (EFB Function compliance with operational requirements).

To avoid losing generality, the author understands that it shall not be proposed as a method for writing an operational requirement in the development method. It remains for the operator to choose his/her favorite methodology. For that matter, it can be mentioned the following from (VALENTE, 2024): user story, use case, diagrams, and textual requirement. However, considering that most EFB manufacturers are also RTCA DO-178C applicants, the recommendation would be to write EFB operational requirements similarly to the system requirements for embedded software. In this way, uniformity and consistency would be favored.

3.4 Task 4

According to Figure 3.1, when the applicant has a new EFB application (i.e., without service history), then Task 4 shall be executed. Task 4 shall also be executed by a legacy EFB application when its SDP is not approved by the certification authority for service history credit.

Task 4 consists of conducting the full Software Development using the RTCA DO-178C compliant process and Operational Requirements. In other words, to develop his/her EFB application, the applicant has to execute the simplified process from Task 1 for the whole EFB, write the Software Development Plan from Task 2, and write the operational requirements from Task 3 for the whole EFB.

3.5 Task 5

Figure 3.1 shows that when the application is considered a legacy EFB (i.e., with service history), then Task 5 shall be executed. It consists of preparing service history data to take credit for unmodified parts of the EFB Software.

Given the relatively recent introduction of EUROCAE ED-273 and the prevalence of EFB applications developed before its publication, this dissertation proposes a methodology for developing new software increments within the framework outlined in Sections 3.1 and 3.3, specifically tailored for EFBs with approved service history.

EUROCAE ED-273 allows EFB applicants to leverage service history as evidence of compliance with **certain** requirements, as stated in item 2.4.2.1.2.3. However, the standard does not clearly delineate which requirements can be demonstrated through service history and which cannot (EUROCAE, 2021).

As elucidated in Section 2.3, the FAA conducted research and published a handbook to assist both the Administration and the industry in qualitatively and quantitatively evaluating service history. However, the document acknowledges the difficulty in assessing the relevance and sufficiency of the collected data. Despite offering some spreadsheet tools, the document is not officially endorsed by the FAA and is not recognized as an alternative compliance method. Ultimately, the responsibility for evaluating and determining credit for service history rests with the Aircraft Certification Office (ACO) (DOT-FAA, 2002).

Given the aforementioned challenges, it is beyond the scope of this research to propose an objective methodology for evaluating the relevance and sufficiency of legacy EFB service history. Hence, the flowchart depicted in Figure 3.4 assumes that the ACO retains the authority to accept or reject the collected service history data.

Figure 3.4 presents a flowchart to assist legacy EFB applicants in maintaining and developing new increments while still adhering to EUROCAE ED-273 standards. It details what was presented as Task 5 in Figure 3.1.

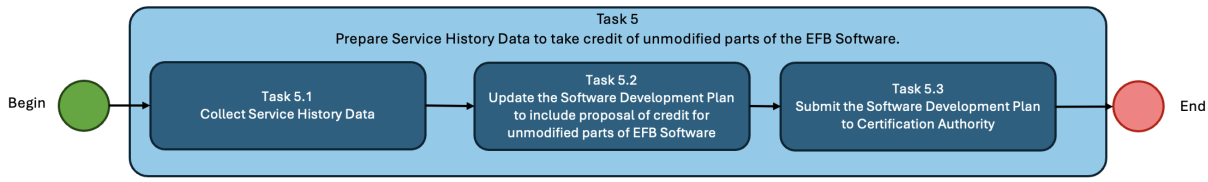

FIGURE 3.4 – Legacy EFB flowchart to request approval for service history credit

The Task 5.1 from Figure 3.4 entails collecting service history data. According to EUROCAE ED-273 item 2.4.2.1.2.3, the collected data must demonstrate relevance and include information on the problem report history. Additionally, there should be a demonstrated similarity between the operational environment from which the service history data is gathered, and the operational environment proposed by the EFB applicant. Guidance for this process can be derived from DOT/FAA/AR-01/116 content.

According to Task 5.2 from Figure 3.4, the applicant must elucidate their intention to seek credit for service history within the Development Plan. This entails attaching the collected content from Task 5.1 to the development plan, along with a clear rationale for seeking credit

According to Task 5.3, the applicant shall submit the Software Development Plan to the Certification Authority. The collected service history may either be accepted or rejected by the ACO. If not accepted, the only recourse for the EFB applicant is to undertake a complete redevelopment following Section 3.4 (Task 4). Conversely, if the ACO accepts the collected service history, only new increments to the legacy EFB are required to adhere to the proposed EFB development method (Task 6).

3.6 Task 6

As explained in Section 3.5, it is beyond the scope of this research to propose an objective methodology for evaluating the relevance and sufficiency of legacy EFB service history. Therefore, the flowchart from Figure 3.1 contains a decision point from where the applicant shall perform Task 4 or Task 6, depending on the outcome of service history credit from the Certification authority.

As explained in Section 3.1, in case the legacy EFB's service history is accepted by the certification authority, then Task 6 shall be performed. It consists of developing the modification using the RTCA DO-178C compliant process, the Software Development Plan, and Operational Requirements.

In other words, to develop a new increment of his/her EFB application, the applicant has to execute only for this new increment the simplified process from Task 1, write the Software Development Plan from Task 2, and write the operational requirements from Task 3 for new EFB's increment.

4 Method Evaluation

4.1 Focal team evaluation

After the development method proposed in Chapter 3 was created, it was evaluated by a focal team. Figure 4.1 presents the steps followed to evaluate the method by a focal team.

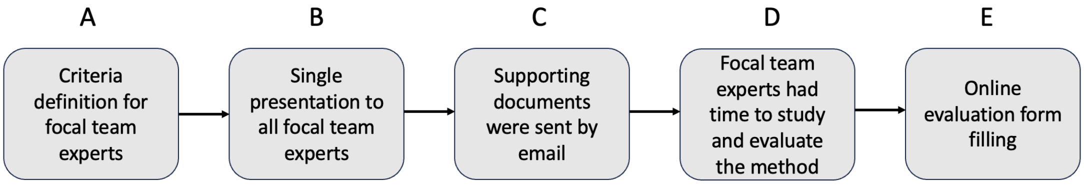

FIGURE 4.1 – Presentation to the focal team and evaluation steps

The focal team comprised five experts in some software domain areas like RTCA DO-178C, software certification, and EFB development. Before the method was evaluated, Step A pre-defined criteria for selecting the focal team experts. The criteria are presented in Section 4.1.1.

Step B was to make a single presentation to all focal team experts. It took about 45 minutes of research problem contextualization and development method presentation and 15 minutes for questions. The goal of this 60-minute total time was to present the same content for all focal team experts without giving one more information than the others.

After the presentation, the supporting documents were sent in Step C. These documents comprised the presentation slides and an extract from the development assurance part of EUROCAE ED-273. These documents were important to let focal team experts analyze the proposed development method in the slides with the requirements from EUROCAE ED-273 that the development method is supposed to comply with.

During the presentation, the author and the focal team experts agreed on a one-week deadline for study and evaluation. Step D was important because it gave evaluators enough time to study and make their comments. If more than one week had been proposed, then it might have compromised the understanding of the presentation.

Finally, Step E was filling out an evaluation form. An online version was adopted due to the geographic distance between evaluators and the agility of the answers. It was designed to take about 5 minutes to be answered. The evaluation was composed of two main parts. The first, composed of four questions, aims to qualify the evaluators and confirm that they comply with the criteria pre-defined in Step 1. The second, composed of five questions, aims to evaluate the method. For each question of the second part, the evaluators were supposed to select one and only one of the following options: Strongly agree, partially agree, neither agree, neither disagree, partially disagree, and strongly disagree. If an answer for the second part is not Strongly Agree, the evaluator was asked to provide a comment.

4.1.1 Criteria

4.1.1.1 Focal team selection

The first part of the online evaluation form was to qualify and confirm that evaluators comply with pre-defined criteria to be part of the focal team. The criteria contain four items:

- Criterion 1: The participant must have at least 10 years of professional experience in aviation;
- Criterion 2: The participant must hold an education background in engineering or computer science;
- Criterion 3: The participant must hold an Education degree in level of college, master's or Ph.D.
- Criterion 4: The participant must have professional experience in at least one of the following competencies: (I) RTCA DO-178C; (ii) aeronautical development software; (iii) development of software standards in general; or (iv) EFBs or application development

Additionally, it is convenient to highlight that the five evaluators were selected from three different aviation manufacturers. This diversity of employers reduces bias errors.

4.1.1.2 Development method acceptance

Each of the five focal team experts answered five questions related to the proposed development method. Therefore, a total number of 25 answers were provided by the focal team.

It was pre-defined that if all answers were Strongly Agree or Partially Agree, then the focal team could consider the development method valid.

On the other hand, if at least one of the 25 answers was Neither Agree nor Disagree, Partially Disagree, or Strongly Disagree, then a modification in the proposed development method would be required. If that is the case, a new evaluation by a focal team would be necessary.

If the proposed development method is accepted by the focal team but at least one answer is Partially Agreed, then the author evaluates the comment provided by the respondent. In this case, the author could change the method according to the feedback received.

4.1.2 Evaluation form

Questions from Q1 to Q4 were proposed to confirm the respondents comply with the pre-defined criteria to be part of the focal team. Questions Q5 to Q9 were proposed to evaluate the method itself. For these questions, the respondents must select one and only one answer between the following: Strongly Agree, Partially Agree, Neither Agree nor Disagree, Partially Disagree, Strongly Disagree. If an answer was not Strongly Agree, the evaluator was supposed to provide feedback. Table 4.1 presents the questions and the possible answers.

TABLE 4.1 – Proposed evaluation form's questions

ID	Evaluation type	Question	Possible Answers
Q1	Self-evaluation	How many years of professional experience do you have?	0-2 years 3-5 years 6-8 years 9-10 years 11-15 years 16-25 years more than 26 years
Q2		What is your education background area?	Engineering Computer Science Administration Accouting Others
Q3		What is your education degree?	High School College Master PhD Pos Doctoral Other(specify)
Q4		Select the technologies or areas of knowledge that you use to work daily or that you have experience with	RTCA DO-178C Aeronautical Software Development EUROCAE ED-273 Software Development Standard in general EFB Database Application Development
Q5	Method evaluation	Did you understand the research context?	Strongly Agree Partially Agree Neither Agree nor Disagree Partially Disagree Strongly Disagree
Q6		Do you consider that you received enough information to evaluate the method with strict criteria?	
Q7		Is the method correct? In other words, is the method defectless?	
Q8		Is the method complete? In other words, does it need nothing more to fulfill the method reason for existing?	
Q9		Is the method reasonable? In other words, is it not too much onerous compared to any other alternative means to fulfill the method reason for existing?	

4.1.3 Evaluation feedback

Questions from Q1 to Q4 are intended to evaluate the respondents and verify they comply with criteria defined in Section 4.1.1.1. According to the above-mentioned criteria, the results shown in Figures 4.2, 4.3, 4.4 and 4.5 proves the **respondents fulfilled the pre-defined criteria to be part of the focal-team**.

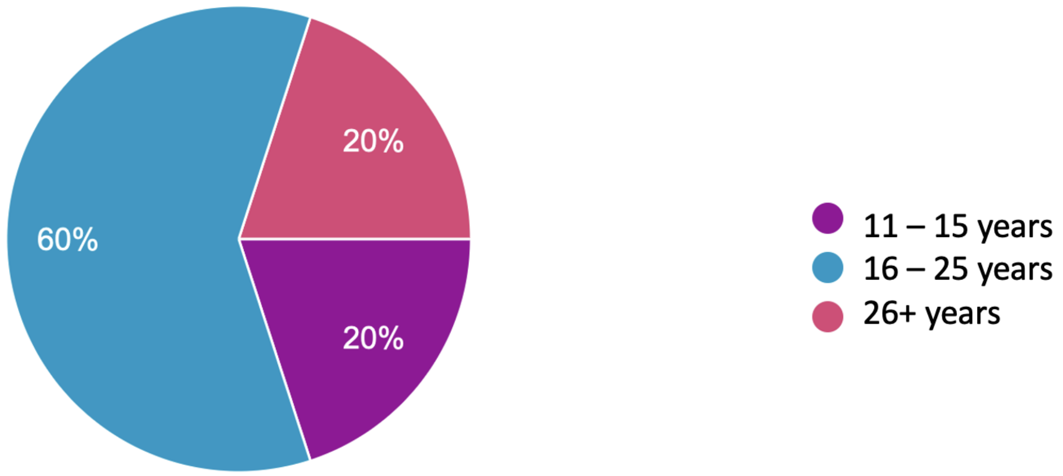

FIGURE 4.2 – Q1: How many years of professional experience do you have?

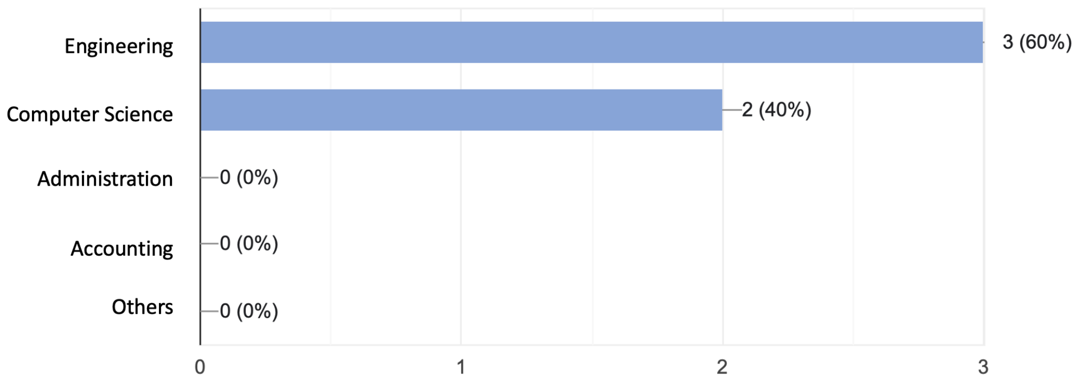

FIGURE 4.3 – Q2: What is your education background area?

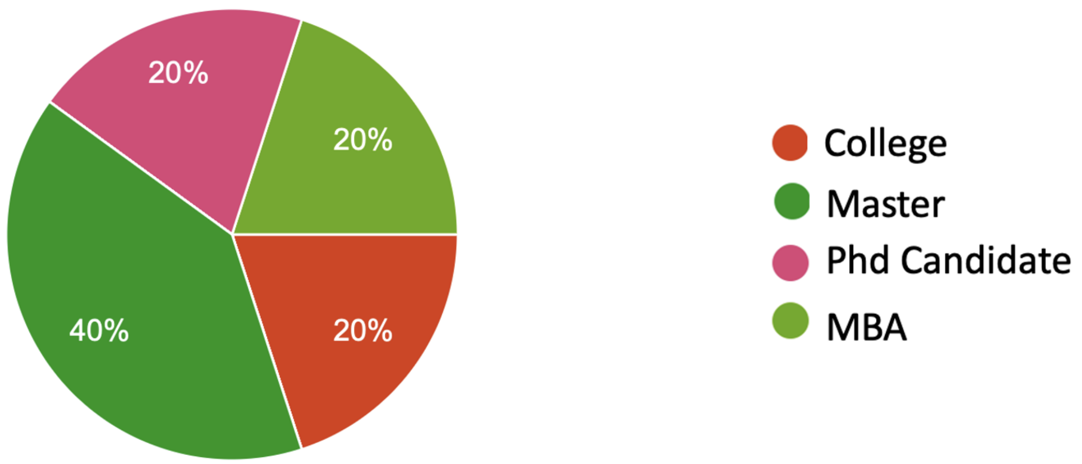

FIGURE 4.4 – Q3: What is your education degree?

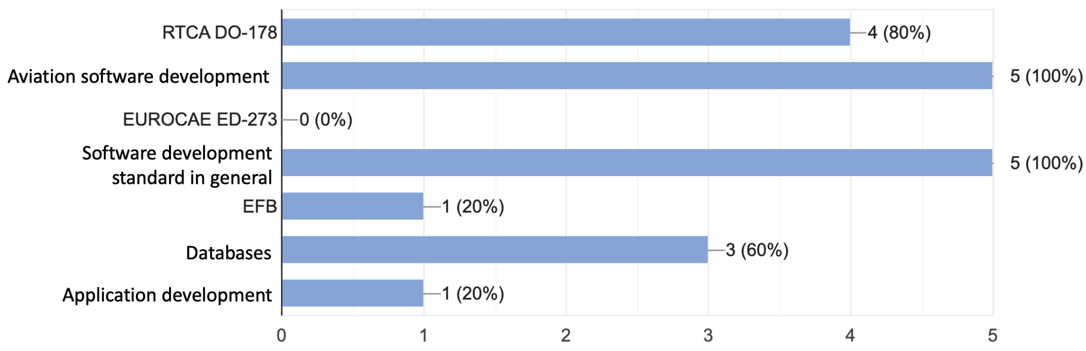

FIGURE 4.5 – Q4: Select the technologies or areas of knowledge that you use to work daily or that you have experience with

The questions from Q5 to Q9 evaluated the method itself. As it can been seen in Figures 4.6, 4.7, 4.8, 4.9 and 4.10, **all respondents Strongly or Partially agree with the statements**. As it was explained in Section 4.1.1.2, the results obtained by the focal team experiment prove the **method was accepted as valid by the experts**.

In this part of the evaluation, the respondents provided feedback and suggestions. All the comments were evaluated by the author. After the evaluation, some of the comments were **accepted by the author and incorporated into the method**, and some other comments were evaluated as **not applicable** or **unfounded**. The accepted comments/suggestions and the unfounded comments are presented below, followed by the author's justification. The non applicable comments will not be presented because they are non-method related or do not contribute to the discussion.

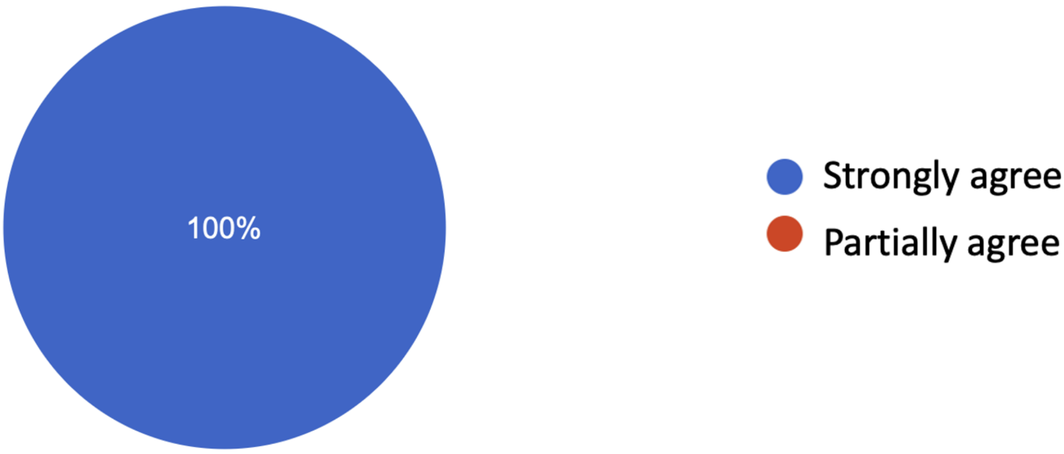

FIGURE 4.6 – Q5: Did you understand the research context?

For Q5, it can be seen that all respondents strongly agree with the statement. Therefore, no comments were provided.

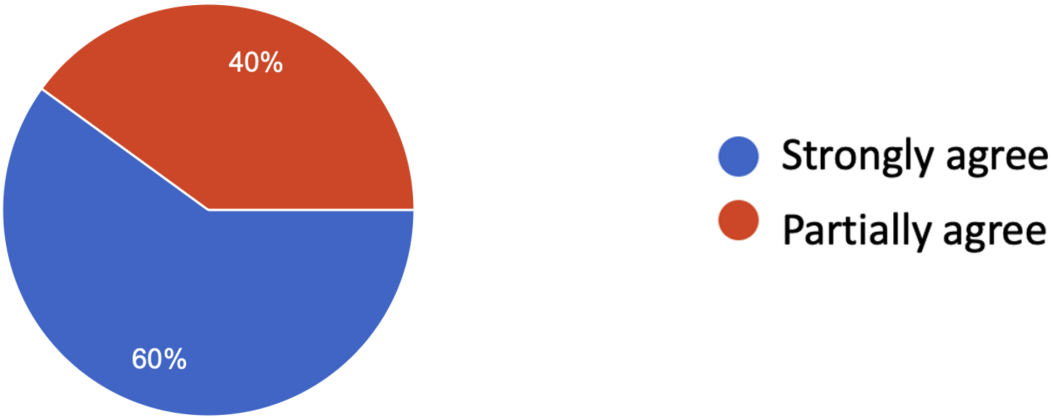

FIGURE 4.7 – Q6: Do you consider that you received enough information to evaluate the method with strict criteria?

For Q6, one respondent inquired about the required independence and configuration control level.

- Q6 - Comment 1: *"I did not find information regarding the independence required for the objectives of DO-178C in mapping. If independence is not applicable for mapping with ED-273, it may be worth mentioning that it is not necessary. The same comment applies to the configuration control levels of artifacts present in the DO-178C Tables and not mentioned in the method (mapping with ED-273)."*

The independence and configuration control level topics asked in Q6 Comment 1 do not apply to EUROCAE ED-273. However, the author evaluated it would be interesting to state that in this dissertation explicitly. Therefore, this comment was **accepted by the author and incorporated into the method**, as exposed in Section 3.1.

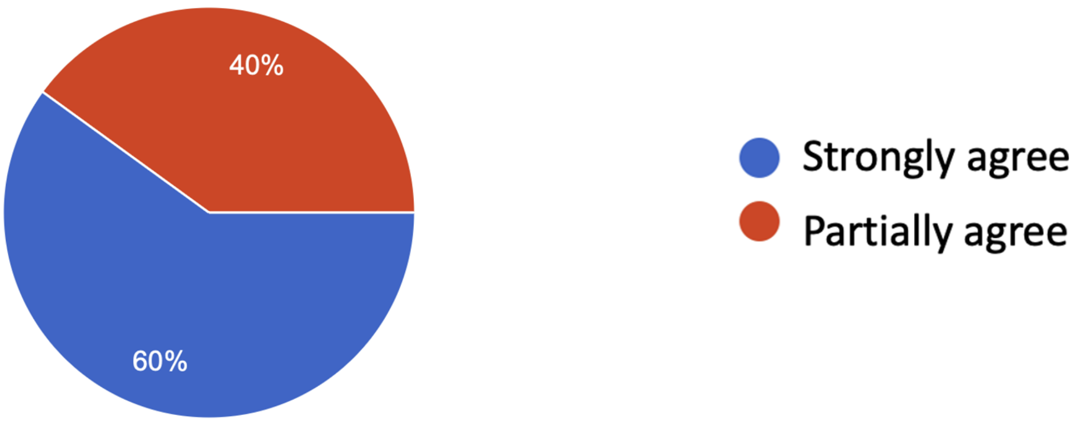

FIGURE 4.8 – Q7: is the method correct? In other words, is the method defectless?

For Q7, two respondents provided the following comments:

- Q7 - Comment 1: *"For ED-273 section 2.4.2.3.3, I understand that the objectives of Table A-7 of DO-178C should also be considered in the analysis. Furthermore, regarding ED-273 section 2.4.2.5.2, I believe that there are no objectives of DO-178C that fully comply with the concerns listed."*
- Q7 - Comment 2: *"I consider it hasty to say that the method is correct. Some requirements presented in the material do not seem to be fully aligned with the objectives indicated (e.g. REQ 40, 41, 42, 45, 46, 55, 58, 59, 60, 61, 69, 70, 71, 72)"*

The respondent from Q7 Comment 1 proposed to include Table A-7 from RTCA DO-178C to fully comply with EUROCAE ED-273 section 2.4.2.3.3 objective. This suggestion was **partially accepted by the author**. In this case, the author agrees that Table A-7 objectives 1, 2 and 3 are applicable to comply with EUROCAE ED-273 section 2.4.2.3.3 objective. After all, EUROCAE ED-273 section 2.4.2.3.3 is about compliance with software requirements. However, the remaining objectives (i.e., 4, 5, 6, 7, 8, and 9) from Table A-7 should not be pointed out by the mapping because they deal with RTCA DO-178C low-level requirements and software structures.

The respondent from Q7 Comment 1 also stated that there is no RTCA DO-178C objective that fully complies with EUROCAE ED-273 Section 2.4.2.5.2. However, the author does not agree with such a statement because EUROCAE ED-273 Section 2.4.2.5.2 concerns Known Issues in EFBs and matches with RTCA DO-178C Table A-8 objective 3, which concerns problem reporting.

The respondent from Q7 Comment 2 raised concerns about some EUROCAE ED-273 requirements. In the opinion of this respondent, some requirements from EUROCAE ED-273 were not addressed by the proposed mapping. The author evaluated the requirements the respondent mentioned, and the author **partially agrees with the respondent**. In other words, some of the requirements the respondents mentioned were indeed not addressed by the proposed mapping but others requirements were already addressed.

Requirements REQ 40, 41 and 42 are presented in EUROCAE ED-273 Section 2.4.2.1.2 *Development Plan - Additional Considerations*. REQ 40 aims to ensure that in an EFB with multiple FQL the FQL Low functions do not adversely affect the FQL High functions. This intent is similar to RTCA DO-178C Table A-2 Objective 5 and Activities 5.2.2.c. REQ 41 aims to identify all third-party pieces of software used in the EFB. This intent is similar to RTCA DO-178C Table A-8 Objective 1 and Activity 7.2.1.b. REQ 42 aims to get a justification and to collect data when an EFB service history is used to show compliance with the standard. This intent is similar to RTCA DO-178C Table A-1 Objective 4 and Activity 4.2.k. Therefore, the mapping from REQ 40, 41 and 42 to above mentioned RTCA DO-178C Tables, Objectives and Activities **was incorporated into the method** presented in this dissertation Section 3.

Requirement REQ 45 is presented in EUROCAE ED-273 Section 2.4.2.2.2 *Operational requirement - EFB application architecture definition* and it asks for EFB architecture definition. It is already addressed in the mapping by RTCA DO-178C Table A-2, Objective 3 and Activities 5.2.2.a and 5.2.2.d.

Requirement REQ 46 is presented in EUROCAE ED-273 Section 2.4.2.2.2 *Operational requirement - EFB application architecture definition* and it asks that a FQL shall be assigned to each EFB function. It is already addressed by EUROCAE ED-273 since it is the outcome of the safety risk assessment process.

Requirement REQ 55 is presented in EUROCAE ED-273 Section 2.4.2.3.1 *Software development - EFB function software requirement definition* and it asks for EFB software requirement definition. It is already addressed in the mapping by RTCA DO-178C Table A-3 Objectives 1 and 2 and Activities 6.3.1. These objectives from RTCA DO-178C address the HLR requirements, which were understood as being equivalent to EUROCAE ED-273 software requirements as explained in this dissertation Section 3.

Requirements REQ 58, 59 and 60 are presented in EUROCAE ED-273 Section 2.4.2.3.3 *Software development - EFB compliance with software requirements*. REQ 58 states that software requirements shall be fully covered by tests. It is already addressed in the mapping by RTCA DO-178C Table A-6 Objective 1 and 2 and Activities 6.4.2, 6.4.2.1, 6.4.3, 6.5. REQ 59 states that the results of the test shall be reviewed and that acceptable differences shall be explained. It is already addressed in the mapping by RTCA DO-178C Table A-7 Objectives 1, 2 and 3. REQ 60 asks for robustness tests. It is already addressed in the mapping by RTCA DO-178C Table A-6 Objective 2.

Requirements REQ 61 and 69 are presented in EUROCAE ED-273 Section 2.4.2.4 *Configuration Management* and Section 2.4.2.4.5 *Configuration Management - Archive*, respectively. REQ 61 states that a configuration management process shall be defined. It is already addressed in the mapping by RTCA DO-178C Table A-8 Objective 4 and Activity 7.2.7. REQ 69 asks for archiving artifacts. It is already addressed by the mapping by RTCA DO-178C Objective 4 and Activity 7.2.7.

Requirement REQ 70 is presented in EUROCAE ED-273 Section 2.4.2.5.1 *Application release - EFB application conformity*. REQ 70 ask for conducting a conformity review. It is already addressed in the mapping by RTCA DO-178C Table A-9 Objective 5 and Activity 8.3.

Requirement REQ 71 is presented in EUROCAE ED-273 Section 2.4.2.5.2 *Application release - Impact analysis of known issue*. REQ 71 asks for recording and assessing known issues. It is already addressed in the mapping by RTCA DO-178C Table A-8 Objective 3 and Activity 7.2.3. This topic was already addressed in Q7 Comment 1.

Requirement REQ 72 is presented in EUROCAE ED-273 Section 2.4.2.6 *Quality assurance process-quality assurance*. REQ 72 states that compliance with the development processes plan shall be obtained. It is already addressed in the mapping by RTCA DO-178C Table A-9 Objective 1 and 5 and Activities 8.2.b and 8.3.

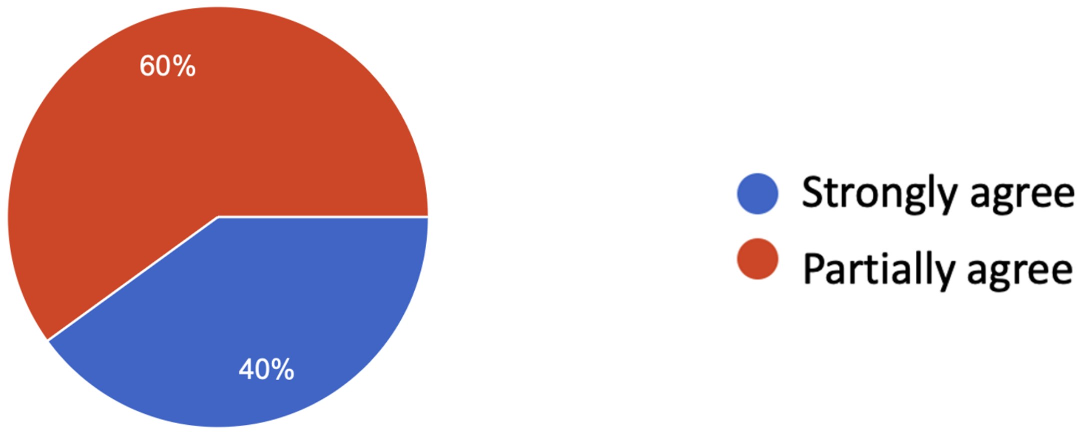

FIGURE 4.9 – Q8: is the method complete? In other words, does it need nothing more to fulfill the method's reason for existing?

For Q8, two comments were formulated:

- Q8 - Comment 1: *"For some of the requirements, it seems to me that some other objectives of DO-178 are missing to be completely met (Ex: ED-273 2.4.2.3.2 x DO-178C A-3: 7; ED-273 2.4.2.3.3 x DO-178C A-7: 2 and 3)"*
- Q8 - Comment 2: *"I understood that ED-273 section 2.4.2.2.1 was addressed as it was impossible to map it in DO-178C. However, I missed dealing with ED-273 sections 2.4.2.2.3 and 2.4.2.2.4.*

The respondent from Q8 Comment 1 proposes to include RTCA DO-178C Table A-3 objective 7 to fully comply with EUROCAE ED-273 Section 2.4.2.3.2. The author understands this comment is **unfound** because RTCA DO-178C Table A-3 objective 7 (Algorithms are accurate) is already contained in RTCA DO-178C Table A-3 objective 2 (High-level requirements are accurate). Once RTCA DO-178C Table A-3 objective 2 is already mapped by the method, then it would not be necessary to include RTCA DO-178C Table A-3 objective 7 in the method.

Additionally, the same respondent proposes including RTCA DO-178C Table A-7 objectives 2 and 3 to fully comply with ED-273 Section 2.4.2.3.3. This comment was **accepted by the author** and it was also proposed in Q7 Comment 1.

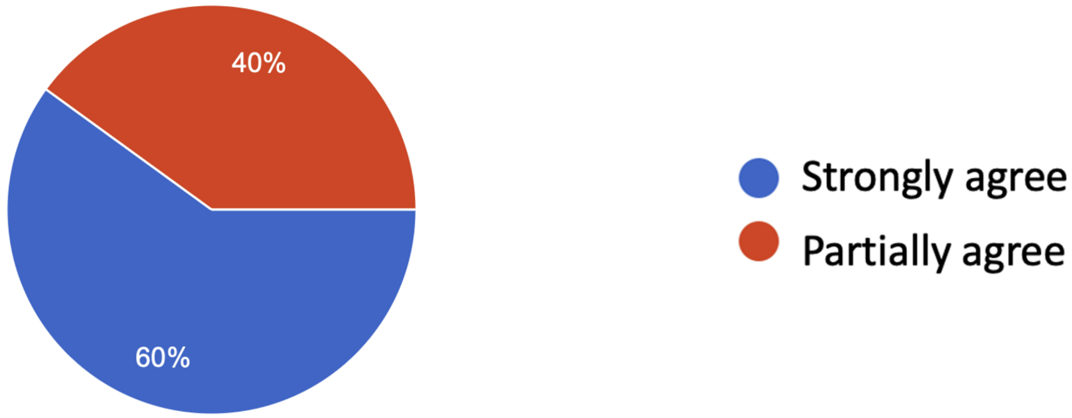

FIGURE 4.10 – Q9: Is the method reasonable? In other words, is it not too much onerous compared to any other alternative means to fulfill the method's reason for existing?

For Q9, no comment was incorporated.

5 Conclusion

According to Section 1.4, the research's objective was to **Create a software development method to be used both by a legacy EFB manufacturer and a new EFB manufacturer to show compliance with the development assurance part of EUROCAE ED-273**. After concluding the research, is the author's understanding that the objective was fulfilled.

This dissertation addressed a relevant topic for EFB manufacturers: compliance with EUROCAE ED-273. It is a relatively new standard that brought some unprecedented development assurance objectives for EFB development. It is a standard that brings quality to the software through the quality in the software process.

This dissertation followed the methodology presented in Section 1.6. A development method that could be used either by a new applicant or a legacy application manufacturer was proposed. The core of the method is the mapping between EUROCAE ED-273 objectives and RTCA DO-178C objectives and activities.

The proposed development method brings benefits to applicants. It was shown that some parts of EFB manufacturers already have a RTCA DO-178C-compliant process. Therefore, it would be much easier to adjust this process and follow only one process for the entire company.

In Section 1.3 it was presented the research question **Is the development assurance part of EUROCAE ED-273 compatible with RTCA DO-178C?** To answer this question, Section 3.1 was developed to find the correlation between both standard's objectives. According to Section 3.1, the number of EUROCAE ED-273's objectives mapped to RTCA DO-178C's objectives or activities is 14. This represents a 82% compatibility between the standards. In other terms, the research answered the research question and found a significant degree of compatibility between EUROCAE ED-273 and RTCA DO-178C.

In Section 1.3 it was also proposed that the research hypothesis is **There is a subset of RTCA DO-178C that can be used to show compliance with EUROCAE ED-273 but not full compliance**. To validate the research hypothesis two steps were followed: the first was to propose a mapping between the standards and the second was to validate the mapping proposal via the focal team.

In the mapping proposed in Section 3.1, it was shown that the only objectives that do not have a similar RTCA DO-178C objective or activity are those related to operational requirements. This lack of correspondence is because operational requirements are similar to system requirements, which are not part of RTCA DO-178C's scope. Therefore, to be compliant with EUROCAE ED-273's development assurance part it is mandatory that the applicant defines the operational requirements separately.

In Section 4.1, the mapping between EUROCAE ED-273 objectives and RTCA DO-178C proposed in Section 3.1 was submitted for the approval of a focal team composed of aviation software and EFB specialists. According to their evaluation in Section 4.1, the method was accepted as correct, complete and reasonable. Therefore, their evaluation validated the research hypothesis.

As explained in Section 1.4, this dissertation's scope was the development part of EUROCAE ED-273. The standard, however, is composed by other parts that must be addressed by the applicant to receive the compliance certificate.

5.1 Threads to validity

Both the method presented in Chapter 3 and its evaluation in Chapter 4 through a focus group may have biases that reflect the opinion of a small group of experts. EUROCAE ED-273 is a recent standard that, being issued in 2021, does not yet have a strong critical mass of industry users, which, despite reinforcing the importance of this work, carries a natural bias that the involved parties may have understandings that may not be fully convergent with the understanding of the community when this critical mass is fully established. To mitigate this risk, the five participants were selected with criteria that reinforce their know-how in the aeronautical sector, their professional experience in the industry, and their degree of education.

5.2 Future work

It was explained in Section 4.1 that the comments not related directly to the development method would not be evaluated by the author. However, in the online filling, some ideas were brought to light by one of the specialists.

According to the specialist, an important amount of time is spent in the EFB's design phase, especially because most error reports are related to operational errors, like providing the wrong weight to calculate the landing speeds. This type of evaluation was out of the research's scope and is also out of EUROCAE ED-273's scope. Therefore, a study could be conducted to identify how these operational errors could be avoided.

In chapter 3, it was presented a method to be used by legacy EFB applicants. In that method, the author considered it out of the research's scope to propose a criterion to accept or not service history data for EFBs. It would be interesting for future work to evaluate if the same guidance provided by (DOT-FAA, 2002) is applicable for EFBs.

5.3 Published article

The following article was published during the research:

1. **MATILDE, Felipe Rodrigo Evangelista**; MARQUES, Johnny. **Development Model for a Legacy Software Supporting Cabin Operation**. In: 2023 Workshop de Teses e Dissertações em Qualidade de Software - Simpósio Brasileiro de Qualidade de Software (SBQS), 2023, Brasília

The following article was accepted to be published and will be presented in a conference:

1. **MATILDE, Felipe Rodrigo Evangelista**; MARQUES, Johnny. **A Development Model for Electronic Flight Bag Software**. In: 43rd AIAA/IEE Digital Avionics Systems Conference (DASC), 2024, San Diego (CA)

Bibliography

ANDERSON, R.; SMITH, T. The convergence of software and hardware in consumer electronics: A review. **Journal of Consumer Technology**, v. 25, n. 3, p. 112–129, 2019.

BHANSALI, P. V. Universal software safety standard. **Software Engineering Notes**, ACM SIGSOFT, v. 30, n. 2, p. 1–4, 2005.

BROWN, L.; JOHNSON, M. Software integration and user experience: Exploring the impact on consumer electronics. **International Journal of Human-Computer Interaction**, v. 37, n. 2, p. 78–92, 2021.

DINGLE, L.; TOOLEY, M. **Aircraft Reliability Engineering: Principles and Practices**. [*S.l.*]: Taylor Francis, 2005.

DOT-FAA. **Software Service History Handbook**. 2002.

EUROCAE. **ED-273 Minimum Operational Performance Standards for Electronic Flight Bag (EFB) Application**. 2021.

FERREIRóS, A.; DIAS, L. A. V. Evaluation of accomplishment of do-178c objectives by cmmi-dev 1.3. *In*: **2015 12th International Conference on Information Technology - New Generations**. **Proceedings** [...]. [*S.l.: s.n.*], 2015. p. 759–760.

JONES, A.; PATEL, R. The impact of software on consumer electronics. **Journal of Consumer Technology**, v. 23, n. 2, p. 45–62, 2020.

MARCIL, L. Realizing do-178c's value by using new technology: Oot, mbdv, tqc fm. *In*: **2012 IEEE/AIAA 31st Digital Avionics Systems Conference (DASC)**. **Proceedings** [...]. [*S.l.: s.n.*], 2012.

MARQUES, J.; CUNHA, A. Verification scenarios of onboard databases under the rtca do-178c and the rtca do-200b. **36th AIAA/IEEE Digital Avionics System Conference (DASC 2017)**, 2017.

MARQUES, J.; HAYASHI, S.; BARROS, L. Caracterização de cenários e garantias de segurança em carregamentos de software aeronáutico. *In*: **Anais do V Workshop de Regulação, Avaliação da Conformidade e Certificação de Segurança**. **Proceedings** [...]. Porto Alegre, RS, Brasil: SBC, 2019. p. 1–10. ISSN 0000-0000. Available at: https://sol.sbc.org.br/index.php/wrac/article/view/14031.

MARQUES, J.; YELISETTY, S.; BARROS, L. A framework for loadable airborne systems. **IEEE Aerospace and Electronic Systems Magazine**, v. 36, n. 5, p. 38–47, 2021.

MARQUES, J.; YELISETTY, S.; SLAVOV, T.; BARROS, L. Enhancing aviation software development: An experience report on conducting audits. *In*: **Anais do XXII Simpósio Brasileiro de Qualidade de Software**. **Proceedings** [...]. Porto Alegre, RS, Brasil: SBC, 2023. p. 198–207. ISSN 0000-0000. Available at: https://sol.sbc.org.br/index.php/sbqs/article/view/27106.

MARSDEN, J.; WINDISCH, A.; MAYO, R.; GROSSI, J.; VILLERMIN, J.; FABRE, L.; AVENTINI, C. Ed-12c/do-178c vs. agile manifesto – a solution to agile development of certifiable avionics systems. *In*: **9th European Congress of Embedded Real-Time Software and Systems**. **Proceedings** [...]. [*S.l.: s.n.*], 2019.

MOY, Y.; LEDINOT, E.; DELSENY, H.; WIELS, V.; MONATE, B. Testing, or formal verification: Do-178c alternatives and industrial experience. **IEEE Software**, v. 30, n. 3, p. 50–57, 2013.

PAZ, A.; BOUSAIDI, G. On the exploration of model-based support for do-178c-compliant avionics software development and certification. *In*: **IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)**. **Proceedings** [...]. [*S.l.: s.n.*], 2016.

RIERSON, L. **Developing Safety-Critical Software**. 1st. ed. [*S.l.*]: CRC Press, 2013. 52 p.

RTCA. **DO-178B Software Considerations in Airborne Systems and Equipment Certification**. Washington, 1992.

RTCA. **DO-178C Software Considerations in Airborne Systems and Equipment Certification**. Washington, 2011.

RTCA. **DO-330 Software Tool Qualification Considerations**. Washington, 2011.

RTCA. **DO-331 Model-Based Development and Verification Supplement to DO-178C and DO-278A**. Washington, 2011.

RTCA. **DO-332 Object-Oriented Technology and Related Techniques Supplement to DO-178C and DO-278A**. Washington, 2011.

RTCA. **DO-333 Formal Methods Supplement to DO-178C and DO-278A**. Washington, 2011.

RTCA. **DO-200B Standards for Processing Aeronautical Data**. Washington, 2015.

SAE INTERNATIONAL. **SAE ARP-4754A: Guidelines for Development of Civil Aircraft and Systems**. 2010.

SARKIS, A.; MARQUES, J.; DIAS, L. A. V. Direcionadores para o desenvolvimento baseado emmodelos de software embarcado aeronáutico. **Cadernos do IME : Série Informática**, v. 44, n. 1, 2020.

SMITH, J. **Embedded Systems Design: Principles and Practices**. [*S.l.*]: Wiley, 2018.

VALENTE, M. T. **Software Engineering: A Modern Approach**. [*S.l.: s.n.*], 2024.

VANDERLEEST, S.; BUTER, A. Escape the waterfall: Agile for aerospace. *In*: **2009 IEEE/AIAA 28th Digital Avionics Systems Conference**. **Proceedings** [...]. [*S.l.: s.n.*], 2009.

YAN, F. Comparison of means of compliance for onboard software certification. **4th International Conference on Computer Science & Education**, IEEE, p. 1–4, 2009.

YOUN, W.; HONG, S.; OH, K.; SUNG, O. Software certification of safety-critical avionic systems: Do-178c and its impacts. **IEEE Aerospace and Electronic Systems Magazine**, v. 30, n. 4, p. 4–13, 2015.

Annex A - Artigo WTDQS 2023

Artigo aprovado e apresentado no Workshop de Teses e Dissertações de Qualidade de Software (WTDQS 2023) do XXII Simpósio Brasileiro de Qualidade de Software, evento realizado de 07 a 10 de novembro de 2023, na cidade de Brasília/DF.

Development Model for a Legacy Software Supporting Cabin Operation

Felipe Rodrigo Evangelista Matilde[1], **Johnny Marques**[1]

[1]Postgraduate Program in Electronic and Computer Engineering –
Instituto Tecnológico de Aeronáutica (ITA)
São José dos Campos – SP – Brazil

felipematildeita@gmail.com, johnny@ita.br

Master degree; Qualification date: August/2022; Planned end date: August/2024

Abstract. *Pilots have to follow procedures, perform checklists, receive and register some pieces of information, and perform many calculations. To facilitate some of these tasks, Electronic Flight Bags (EFBs) are developed to be used by pilots inside the cabin. Software Quality Assurance (SQA) is a vital discipline in software development, ensuring that software meets quality requirements through various activities such as testing, reviews, and adherence to established standards. This paper presents the current status of research that will propose a new software development model for legacy EFBs.*

1. Introduction

The primary objective of aviation is to transport people from one place to another safely. The significance of safety in aviation arises from the inherent risks associated with operating an airplane, which can result in numerous fatalities and substantial economic losses. To mitigate these undesirable consequences, aviation authorities enforce a set of stringent regulations [Marques and Yelisetty 2019].

Software Quality Assurance (SQA) is a vital discipline in software development, ensuring that software meets quality requirements through various activities such as testing, reviews, and adherence to established standards [Fernandes and França 2015].

Given that safety is of paramount concern in aviation, every aspect of an airplane must adhere to a rigorous set of regulations. These regulations encompass hardware components such as hydraulic systems, as well as software components like avionics [Marques and Yelisetty 2019]. Regarding software, the primary aeronautical standard adopted over the past 30 years to ensure the development of high-quality and safe embedded software is RTCA DO-178C [RTCA 2011].

2. Problem Characterization and Proposed Solution

Operating an airplane is a complex task. Pilots have to follow procedures, perform checklists, receive and register some pieces of information, and perform several calculations. To facilitate some of these tasks, Electronic Flight Bags (EFBs) are developed to be used by pilots inside the cabin, as showed in Figure 1.

EFBs are essentially a piece of hardware not embedded in the airplane - usually an iPad®- that runs an application related to an airplane operation task. Pilots are the final user the EFB is developed for.

For decades, EFB applications have been developed to support airplane operations. Consequently, many regulations came into place to define EFB's scope and hardware specifications. However, there were no software development criteria for EFBs until 2021, when EUROCAE released its ED-273 entitled "Minimum Operational Performance Standards for Electronic Flight Bag (EFB) Application" [EUROCAE 2021]. Regarding software development, ED-273 specifies a set of objectives such as a Software Development Plan, Requirement validation, Software architecture, and Test. Despite being less critical to flight safety than Airplane Embedded Software, EFBs now have their standard to ensure product quality.

In this scenario, there are two active standards for developing aviation software: the well-established embedded software, RTCA DO-178C, which has been through four versions since 1982, and the other is specific to EFB, EUROCAE ED-273, introduced in 2021.

The aviation industry may be interested in aligning software development processes with only one of these standards, and, in this case, the legacy of RTCA DO-178C holds significant importance within the industry. Therefore, this master's research aims to address the following research question: **Is there a subset of RTCA DO-178 objectives that could be used to show full compliance with EUROCAE ED-273?** Based on the research question, our research hypothesis is **There is a subset of RTCA DO-178 that can be used to show compliance with EUROCAE ED-273 but not full compliance.**.

The objective of this research is to propose an EFB development model that aviation software manufacturers can use for legacy EFBs to comply with EUROCAE ED-273, adapting an established RTCA DO-178C process. The importance this research gives to legacy EFB software comes from the fact that most of EFBs has been developed before EUROCAE ED-273.

The software development model is the outcome of this research and will be composed by three components: the first is the subset of RTCA DO-178C objectives that can be used to show partial compliance with EUROCAE ED-273, the second is a method of showing compliance with EUROCAE ED-273 remaining objectives and finally the third is an approach to deal with a legacy software development history.

Figura 1. A pilot interacting with an EFB software in an iPad®

3. Methodology

Five steps were identified in the methodology. Figure 2 presents the methodology in a flowchart. Steps 1 and 2 were already performed, and Step 3 was in progress when this paper was submitted.

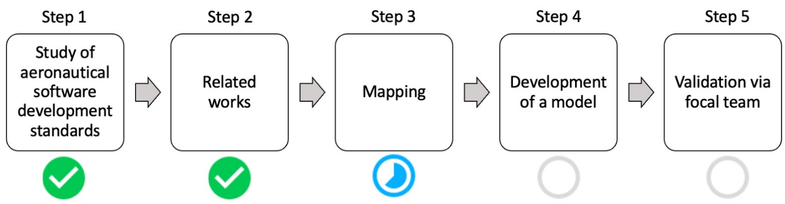

Figura 2. Research methodology flowchart

In Step 1, the aviation software development standards (RTCA DO-178C and EUROCAE ED-273) were studied. In this step, it was collected all information on such standards and tried to find common aspects between them.

Inside Step 2, the related work was identified. Because a mapping process between EUROCAE ED-273 and other aeronautical software standards will be performed in Step 3, the goal of Step 2 is to understand how to perform the mapping task. Consequently, some related works that also performed different mapping were selected. Step 3 is an important part of the research, and it identifies which aspects of EUROCAE ED-273 are already contemplated in RTCA DO-178C.

Step 4 focuses on the development of our main contribution to the research. The model will be composed of three elements: the subset of RTCA DO-178 that could be used to show partial compliance with EUROCAE ED-273 objectives, a method to address directly EUROCAE ED-273 remaining objectives, and a proposal to address the future modifications of legacy EFB software. Finally, inside Step 5, the model is evaluated by a focal team composed of experts in aviation software development with good knowledge of software standards for this field.

4. Background

4.1. RTCA DO-178C

According to [Rierson 2013], since the first version (1982), the goals of RTCA DO-178C were to promote the safe implementation of aviation software and to provide clear and consistent ties with the system and safety processes.

One important concept regarding RTCA DO-178C is the five Development Assurance Levels (DALs). According to the standard, a software risk assessment has to be done to evaluate the contribution a failure in the software will cause to the system. The DAL A is the most critical, while the remaining DALs (B, C, D, and E) progressively eliminate objectives based on the safety impact that software malfunctions can cause to the aircraft.[Marques and Yelisetty 2019]. The 71 objectives of the RTCA DO-178C are organized by DAL. There are objectives for development, verification, configuration control, and quality assurance [Marques and Cunha 2017].

4.2. EUROCAE ED-273

EUROCAE is an international organization in charge of developing standards for airborne equipment. In 2019, Working Group 106 (WG-106) studied and proposed the first standard for EFBs, EUROCAE ED-273, released in 2021.

The necessity for creating a standard came from the fact that the increasing number of EFBs applications became a challenge for aviation authorities such as the Federal Aviation Administration (FAA) in the USA, the European Aviation Safety Agency (EASA) in Europe, and *Agência Nacional de Aviação Civil* (ANAC) in Brazil to evaluate and approve them [EUROCAE 2021]. Despite many regulations created before to evaluate hardware components, EUROCAE ED-273 was the first to evaluate software aspects such as function suitability to EFBs, human-machine interface, development assurance, and security.

Similar to RTCA DO-178C, EUROCAE ED-273 also requires a risk assessment to evaluate the application. When this assessment is performed, the application receives one of the two possible Function Qualification Level (FQL): Low or High. Each FQL contains a set of objectives to be accomplished and registered by the development team according to the application FQL.

4.3. Related Works

In his study, Ferreirós & Dias (2015) evaluated the distance between CMMI-DEV 1.3 software maturity model and RTCA DO-178C. The purpose of such evaluation was to identify the gaps a software company compliant with CMMI-DEV 1.3 has to dedicate its effort to becoming an embedded software provider. The authors concluded that, despite having some points in common, RTCA DO-178C requires specific technical aspects that are out of CMMI-DEV 1.3 scope, like Verification of Verification Process Results.

A study to find a universal software safety standard was performed by Bhansali (2005) . This author read about 16 critical software standards and identified the 23 fundamental concepts behind them. The work presents and justifies each one of these concepts.

A comparison of means of compliance for Onboard Software Certification was performed by Yan (2009) . In the article, the author explained that despite being adopted as a means of compliance with certification regulation, RTCA DO-178C is not theoretically the only means.

5. Preliminary Results

The mapping between what EUROCAE ED-273 requires and what other standards require is still in progress. However, some relationships between these standards have already been found. In this section, two of them are presented.

5.1. Development Plan

EUROCAE ED-273's requirement 039 states that a development plan shall be defined. According to EUROCAE ED-273's recommendation 032, a development plan should describe software development methodology and processes, configuration management processes, quality assurance processes, and development environment, including framework and tools. Objectives from 1 to 4 found in RTCA DO-178C are equivalent to what EUROCAE ED-273 requires by "Development plan - minimum considerations".

5.2. EFB Function Operational Requirements Definition

EUROCAE ED-273's requirement 044 demands that all applicants have to define function operational requirements. There is not a RTCA DO-178 objective that matches such a requirement. After all, operational requirements define what the EFB is expected to perform, and according to Rierson (2013) , this would be similar to system requirements outside the scope of RTCA DO-178C. Figure 3 summarizes what was presented in this section.

EUROCAE ED-273				MAPPING		
Requirement ID	Requirement Title	Recommendation ID	Recommendation content	STANDARD	ITEM	COMMENT
REQ 039	Software Development Plan – minimum consideration	REC 032	Software development methodology and processes	DO-178	Table A-1 Objectives 1-4	Software Development Plan
			Development environment including framework and tools			
			Configuration management processes			Software Configuration Management Plan
			Quality assurance processes			Software Quality Assurance Plan
REQ 044	EFB Function Operational Requirements Definition	REC 037	The operational context	NA	NA	Not applicable
			The supported operational environment			
			Behaviour in normal and degraded operating conditions			
			The inputs entered by the user or acquired from other sources			
			Non-functional requirements			

Figura 3. Relation between two EUROCAE ED-273's requirements and other aeronautical standard

6. The Development Model

As stated previously, the development model will be composed by three elements: a subset of RTCA DO-178 objectives that could be used to show partial compliance with EUROCAE ED-273, a method to directly address EUROCAE ED-273's remaining objectives, and an approach to deal with legacy history.

Once the research is still in progress, the development model is not finally concluded. However, a small exemplification of the model can be presented. The model will assume that an existing RTCA DO-178 compliant process is running in the applicant organization. This process will be adapted so that a new DAL will be created to represent an EFB compliant with EUROCAE ED-273. Therefore, the subset of RTCA DO-178 objectives that could be used to show partial compliance with EUROCAE ED-273 will be applicable to this EFB's DAL. For instance, according to Figure 3, there will be an indication in the adapted RTCA DO-178 process stating that the new DAL has to comply with RTCA DO-178 Table A-1 Objectives 1-4.

We have already found some EUROCAE ED-273 objectives that do not find an equivalent objective in RTCA DO-178, as Figure 3 shows. In this case (i.e. EFB Function Operational Requirement Definition), it is been evaluated a method to write and validate this type of requirement. The method is still not defined.

7. Final Remarks and Next Steps

As Figure 2 shows, the current step of the research is the mapping creation from EUROCAE ED-273 to RTCA DO-178C. When this step is finished, the next is model development that can deal with EUROCAE ED-273's objectives not traceable to other standards and a proposal to deal with aspects related to legacy applications. The final step is to create a focal team of aviation software specialists to evaluate the development model.

The initial findings from this research further support the hypothesis that the final model will closely resemble RTCA DO-178C. Should the research validate this hypothesis, the possibility of establishing a new RTCA DO-178C DAL specifically for EFB development emerges.

It is important to mention that most parts of aviation manufacturers have their EFBs. Because those aviation manufacturers already have an organizational software development process to comply with RTCA DO-178C, the proposed model will benefit them because it would be much easier to adopt an RTCA DO-178C organizational process than to create a whole new process.

For EFB software developers who do not comply with RTCA DO-178C, the results obtained by this research will at least guide them in creating their process. The scientific and technological contributions of this research include: 1) The mapping with traceability between EUROCAE ED-273 and RTCA DO-178C; and 2) The proposed model. A video explanation of this paper is provided on https://youtu.be/AVsI-7jvS5w.

Acknowledgment

This research was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES).

Referências

Bhansali, P. V. (2005). Universal software safety standard. *Software Engineering Notes*, 30(2):1–4.

EUROCAE (2021). Ed-273 minimum operational performance standards for electronic flight bag (efb) application.

Fernandes, F. and França, R. (2015). Assessing the impact of scrum in airborne software quality assurance. In *Anais do XIV Simpósio Brasileiro de Qualidade de Software*, pages 204–211, Porto Alegre, RS, Brasil. SBC.

Ferreirós, A. and Dias, L. A. V. (2015). Evaluation of accomplishment of do-178c objectives by cmmi-dev 1.3. In *2015 12th International Conference on Information Technology - New Generations*, pages 759–760.

Marques, J. and Cunha, A. (2017). Verification scenarios of onboard databases under the rtca do-178c and the rtca do-200b. *36th AIAA/IEEE Digital Avionics System Conference (DASC 2017).*

Marques, J. and Yelisetty, S. (2019). An analysis of software requirements specification characteristics in regulated environments. *International Journal of Software Engineering & Applications*.

Rierson, L. (2013). *Developing Safety-Critical Software*. CRC Press, 1st edition.

RTCA (2011). Do-178c software considerations in airborne systems and equipment certification.

Yan, F. (2009). Comparison of means of compliance for onboard software certification. *4th International Conference on Computer Science & Education*, pages 1–4.

Annex A - Artigo DASC 2024

Artigo aprovado no Workshop de 43rd IEEE/AIAA Digital Avionics Systems Conference (DASC 2024), evento a ser realizado de 29 de setembro a 3 de outubro de 2024, na cidade de San Diego/EUA.

A development approach for Electronic Flight Bag software

1st Felipe Rodrigo Evangelista Matilde
Computer Engineering Department
Aeronautics Institute of Technology
São José dos Campos, Brazil
felipematildeita@gmail.com

2nd Johnny Cardoso Marques
Computer Engineering Department
Aeronautics Institute of Technology
São José dos Campos, Brazil
johnny@ita.br

***Abstract*—Operating an airplane is not an easy task. The pilots must check for weight and balance, communicate with towers, flight crew and passengers, fly the airplane, navigate from origin to destination, sometimes deal with system failures, and perform many procedures. Electronic Flight Bags (EFB) are applications pilots use inside the airplane cabin to make operating an airplane easier. This type of software application has been growing in the last twenty years. Most of these applications developed for tablets help pilots considerably operate airplanes by reducing workloads, optimizing airplane performance, and removing paper manuals from airplanes. In 2021, EUROCAE published the standard ED-273 for developing EFBs and introduced some requirements related to software development assurance, among other relevant points. The EUROCAE ED-273, as expected, made EFB manufacturers concerned about how to follow the new requirements. On the other hand, a 30-year-old software standard, the RTCA DO-178C, establishes considerations for developers, installers, and users when designing embedded equipment using software. Several aircraft manufacturers have well-established RTCA DO-178C software processes. We believe that would be very beneficial for organizations with an RTCA DO-178C compliant process to take advantage of it as a means of compliance for the new ED-273. Our work proposes to create an EFB development approach compliant with ED-273 guidelines but using original RTCA DO-178C processes defined by the aircraft manufacturer. A mapping between the two standards was performed. The development approach was thoroughly assessed by a panel of highly experienced experts in aeronautical software. The main contributions of this paper are (i) the mapping between EUROCAE ED-273 and RTCA DO-178C, (ii) the development approach, and (iii) the evaluation processes defined and executed with the participation of 5 experts in RTCA DO-178C or EFB software development.**

***Index Terms*—EFB, ED-273, electronic flight bag, software development**

I. Introduction

Electronic Flight Bags (EFBs) were developed to simplify pilots' tasks. Most of these applications are developed for iPad ®and adopted almost universally as an EFB device.

Some EFB application domains include takeoff and landing performance calculation, weight and balance, airport navigation, moving maps applications, briefing, fligh-planning , electronic QRH procedures, and replacement of paper manuals. Figure 1 presents a pilot interacting with an EFB.

It is important to highlight that when an airline decides to adopt an EFB as part of its operation, it has to apply for authorization from its civil aviation authority. This type of authorization normally evaluates flight crew training, hardware safety, and computed results correctness. Due to its benefits, EFBs are widely adopted worldwide. The growing popularity of EFBs stimulated the emergence of many EFB manufacturers and the increasing number of EFB application domains. Consequently, this scenario makes the EFB operational evaluation by civil aviation authorities even harder. Therefore, in 2021, EUROCAE published the EUROCAE ED-273 document entitled "Minimum Operational Performance Standards (MOPS) for Electronic Flight Bag (EFB) Application." The EUROCAE ED-273 addresses concerns like function eligibility, risk analysis, interface, databases, and security. One of these concerns in particular has not been addressed by any other document before: the EFB software development assurance

II. Motivation

EASA adopted EUROCAE ED-273 as a compliance alternative to the operational requirement CAT.SPA.EFB. 100 (b). Since a new EFB regulation is in place, EFB manufacturers are expected to be concerned about how to comply with it.

Fig. 1. A pilot interacting with an EFB software in an iPad®

In that regard, it would be very beneficial for organizations that already have an RTCA DO-178C compliant process to take advantage of it as a means of compliance for the new EUROCAE ED-273. After all, having only one development process instead of two brings efficiency, reduces waste, and keeps all organization employees informed about the process. By the time this dissertation is written, the scenario of an EFB manufacturer that is also an embedded software manufacturer compliant with RTCA DO-178C is the most frequent.

III. Related Work

s previously mentioned, the software development for embedded software is already standardized in aviation through the use of RTCA DO-178C [1] and its supplements. Recent works have discussed advances and new software development methods in this field. The authors of this work identified and grouped related works into 6 (six) themes:

- Impacts on the transition from RTCA DO-178B [2] to RTCA DO-178C [1], as explored in the works of 3 and 4;
- Model-Based Development, as explored in the works 5 and 6;
- Use of Agile Methods in Software Development, as explored in the works 7 and 8;
- Formal Verification, as seen in the works of 9 and 10;
- Aircraft Embedded Software Loading, as reported in the works of 11 and 12;
- Mapping between standards, models, and norms with a focus on safety, as reported in the work of 13 and 14.

The work of 3 explains the benefits of formal methods and object-oriented technology that RTCA DO-178C offers in conjunction with RTCA DO-332 [15] and RTCA DO-333 [16]. It also focuses specifically on modeling in software development and the qualification of tools that automate or facilitate the verification and validation of avionics applications built from models to ensure there are no unintended functions.

The work of 4 presents an overview of the guidelines for aeronautical software contained in RTCA DO-178C and supplementary documents. It also addresses the similarity between RTCA DO-178B and DO-178C, reviewing the fundamentals of verification philosophy and an overview of crucial guidance included in RTCA DO-178C.

The work of 5 presents a framework for using models for compliance with RTCA DO-178C. They also analyzed other approaches compared to the proposed framework, highlighting similarities, differences, strengths, and weaknesses.

The work of 6 presented a set of guidelines for development based on Aeronautical Embedded Software models, ensuring compliance with RTCA DO-178C and RTCA DO-331 17. In addition to the drivers, a case study is presented.

The work of 7 provides a detailed analysis of the main agile practices, with a preliminary assessment of their ease of implementation. The authors highlighted that the transition to agile development does not require sudden and radical changes but can be accomplished by incorporating agile methods into an existing process.

The work of 8 shows how apparent contradictions between agile practices and aeronautical software certification objectives were resolved in several Airbus projects and quantifies the resulting financial gains.

The work of 9 describes some of the goals and activities in the area of formal methods, explaining how these methods can be used instead of testing in an RTCA DO-178C context. The work summarizes the practical experience of Dassault-Aviation and Airbus in successfully applying formal methods for developing aeronautical embedded software.

The work of 10 provides some scenarios for database verification using the RTCA DO-178C and RTCA DO-200B [18] standards, including the use of tool qualification when processes are eliminated, reduced, or automated by the use of software tools without reviewing the output produced by such tools.

The work of 11 characterizes the scenarios of software loading on aircraft and treatments for possible threats involving information security in this process. This work was later improved to a framework in the work of 12 that presents a set of reusable requirements and general testing procedures for software loading involving manual and automatic checks. The authors believe that the framework can help smaller companies, especially those entering the market, to incorporate software loading capabilities into systems development.

In their study, 14 proposed a method to estimate the distance a CMMI-DEV compliant team would have to overcome to be an embedded software provider compliant with RTCA DO-178C.

The method was based on ten tables - one for each RTCA DO-178C Appendix A table - whose columns represented each RTCA DO-178C objective and activity and whose rows represented each CMMI-DEV practice. Figure 2 presents an illustration of these tables.

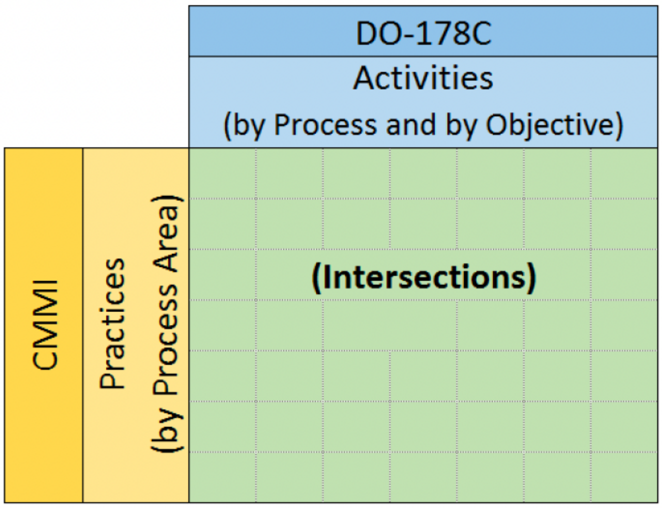

Fig. 2. Intersection table scheme proposed by [14]

The authors assessed the intersection of both standards in each of the ten tables. By doing so, they could estimate the level of accomplishment of RTCA DO-178C by a CMMI-DEV-compliant team.

The authors concluded that, despite similar philosophies and concerns in both standards, it is impossible to adopt CMMI-DEV to comply with RTCA DO-178C. This statement of non-compliance is presented by the authors, especially when they point to the Verification of Verification Process Result and Certification Liaison Process,

A comparison of four compliance means for embedded software was performed by 19. The author evaluated RTCA DO-178B, Safety Engineering Approach for Software Assurance, Capability Maturity Model (e.g. CMMI), and Alternative Methods (e.g. formal methods and service history).

The author recommends checking first with local authorities to determine whether the alternative means of compliance are acceptable. However, the author performed a survey and concluded that it would be hard to convince aviation players to deviate from RTCA DO-178C. According to the author, the industry and aviation authorities widely adopted the RTCA DO-178C.

In his study, 13 wanted to find a subset of common attributes or objectives among 16 safety-related development standards. He concluded that there is a subset of common attributes or objectives (e.g. System Safety Assessment, Software Requirement Validation, and Traceability Analysis). However, it can be deduced from his work that different safety-related areas emphasize the software development process differently.

IV. Research method

To achieve the objective, a set of five steps were identified. Figure 3 represents the five steps.

Step 1 was focused on reading and studying aviation software development standards. Three main documents were selected to achieve this step: RTCA DO-178C, RTCA DO-330, and EUROCAE ED-273. Each document was studied carefully to comprehend the context it is applied to and to identify commonalities and differences between them. Step 2 was to read and study related works, especially those that mapped two software standards. The related works were useful in identifying the adopted methodologies before performing the mapping to RTCA DO-178C.

Step 3 was the mapping itself. After a deep reading of both EUROCAE ED-273 and RTCA DO-178C, it was time to identify objectives and activities from RTCA DO-178C that could be used to show compliance with some of its development assurance requirements. Step 4 focused on building the development approach, which is the main contribution of the present dissertation. Step 5 was the evaluation via the focal team. In this step, qualified experts in the domain were invited to evaluate the proposed approach. After their evaluation, if the proposed development approach is not approved, then it shall be revised.

V. Development approach

The development approach, which is the main contribution of this research, is presented in Figure 4. It represents Step 4 of Figure 3.

Before executing any Task from Figure 4, a question about having or not an RTCA DO-178C compliant process must be answered. All the Tasks from 1 to 6 presented in Figure **??** and in this Section are to be executed only when there is an RTCA DO-178C compliant process. When the EFB applicant does not have an RTCA DO-178C compliant process, he/she must submit to his/her regulator an EUROCAE ED-273 compliant process. In this case, creating such a process is out of the scope of this research, even though the applicant could be inspired by the content presented here.

The approach and its Tasks presented in Figure 4 assume the applicant already has a RTCA DO-178C compliant process.

Task 1 simplifies the RTCA DO-178C compliant process according to EUROCAE ED-273 mapping to RTCA DO-178C. Assuming the applicant already has an RTCA DO-178C compliant process, it addresses all applicable objectives and activities from the standard. Therefore, this dissertation proposes tailoring the RTCA DO-178C compliant process so that only objectives and activities from 5 are effectively executed.

Observations from Figure 5 reveal two key insights. Firstly, an extensive mapping exists from EUROCAE ED-273 to RTCA DO-178C, indicating a substantial alignment between the two standards. However, several requirements from EUROCAE ED-273 lack a direct equivalent objective in RTCA DO-178C.

As depicted in Figure 5, the requirements from EUROCAE ED-273 that remain unmapped to RTCA DO-178C include 2.4.2.2.1 (EFB Function operational requirements definition), 2.4.2.2.3 (EFB Function Operational Requirements validation), and 2.4.2.2.4 (EFB Function compliance with operational requirements). The absence of equivalence in RTCA DO-178C for these requirements is primarily due to their focus on operational requirements.

EUROCAE ED-273 defines operational requirements, outlined in item 2.4.2.2, as functionalities to assist pilots in performing their duties. This aspect is not addressed within the scope of RTCA DO-178C.

It is worth emphasizing that RTCA DO-178C pertains specifically to embedded software. Embedded software operates within a system; therefore, it is nonsensical for the software component to introduce functionalities beyond what the system originally designed for.

In essence, requirements are formulated and validated during system development at the system level, rather than at the software level. This is one of the fundamental reasons why RTCA DO-178C does not validate system requirements. In RTCA DO-178C, all development activities originate from and align with system requirements.

An EFB application cannot be viewed as a system component; it exists within an environment where pilots utilize the EFB to carry out their tasks. Therefore, the concept most closely resembling EUROCAE ED-273 operational requirements would be system requirements, which fall outside the scope of RTCA DO-178C.

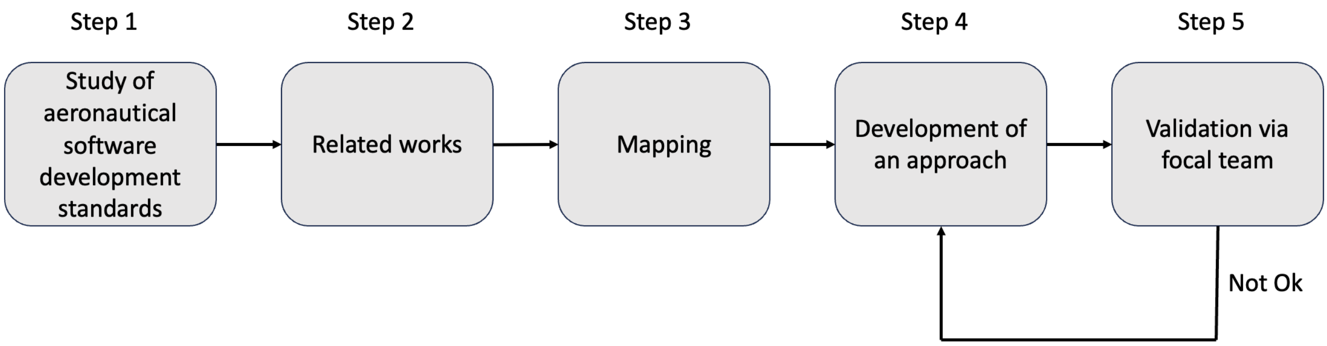

Fig. 3. Research method flowchart

Task 2 consists of writing the Software Development Plan (SDP). This is a plan from RTCA DO-178C that contains a description of the software development procedures and software life cycle(s) [1]. RTCA DO-178C requires more plans than SDP. However, EUROCAE ED-273 objective 2.4.2.1.1 requires only one, called Application Development Plan.

SDP shall contain two important components:

- It shall present this dissertation's proposed approach so the Certification Authority can evaluate it
- It shall contain the service history credit, in case the applicant intends to take credit of it (explained in Task 5.2).

The only EUROCAE ED-273 objectives that are not mapped to RTCA DO-178C objectives are 2.4.2.2.1 (EFB Function operational requirements definition), 2.4.2.2.3 (EFB Function Operational Requirements validation) and 2.4.2.2.4 (EFB Function compliance with operational requirements). It was also stated that the reason for this lack of mapping is the fact that system requirements are out of EUROCAE DO-178C's scope.

In this case, the EFB applicant has to create a method to define, validate, and verify the operational requirement to fulfill EUROCAE ED-273 objectives 2.4.2.2.1 (EFB Function operational requirements definition), 2.4.2.2.3 (EFB Function Operational Requirements validation) and 2.4.2.2.4 (EFB Function compliance with operational requirements).

When the applicant has a new EFB application (i.e. without service history) then Task 4 shall be executed. Task 4 shall also be executed by a legacy EFB application when its SDP is not approved by the certification authority for service history credit. Task 4 consists of conducting the full Software Development using the RTCA DO-178C compliant process and Operational Requirements. In other words, to develop his/her EFB application the applicant has to execute for the whole EFB the simplified process from Task 1, write the Software Development Plan from Task 2 and write the operational requirements from Task 3 for the whole EFB.

Figure 4 shows that when the application is considered a legacy EFB (i.e. with service history) then Task 5 shall be executed. It consists of preparing service history data to take credit for unmodified parts of the EFB Software.

Given the relatively recent introduction of EUROCAE ED-273 and the prevalence of EFB applications developed before its publication, this article proposes a methodology for developing new software increments within the framework outlined in Sections 4, specifically tailored for EFBs with approved service history.

it is beyond the scope of this research to propose an objective methodology for evaluating the relevance and sufficiency of legacy EFB service history. Hence, the flowchart depicted in Figure **??** operates under the assumption that the ACO retains the authority to accept or reject the collected service history data.

VI. Validation

After the development approach proposed in Chapter 3 was created, it was evaluated by a focal team. The focal team was composed of five experts in some software domain areas like RTCA DO-178C, software certification, and EFB development. Before the approach was evaluated, Step A was to pre-define criteria for selecting the focal team experts. The criteria are presented in Section ...

Each one of the five focal team experts answered five questions related to the proposed development approach. Therefore, a total number of 25 answers were provided by the focal team. The answered questions were the following:

- Did you understand the research context?
- Do you consider that you received enough information to evaluate the approach with strict criteria?
- Is the approach correct? In other words, is the approach defectless?
- Is the approach complete? In other words, does it need nothing more to fulfill the approach reason for existing?
- Is the approach reasonable? In other words, is it not too much onerous compared to any other alternative means to fulfill the approach reason for existing?

It was pre-defined that if all answers were Strongly Agree or Partially Agree, then the focal team could consider the development approach valid.

On the other hand, if at least one of the 25 answers was Neither Agree nor Disagree, Partially Disagree, or Strongly Disagree, then a modification in the proposed development

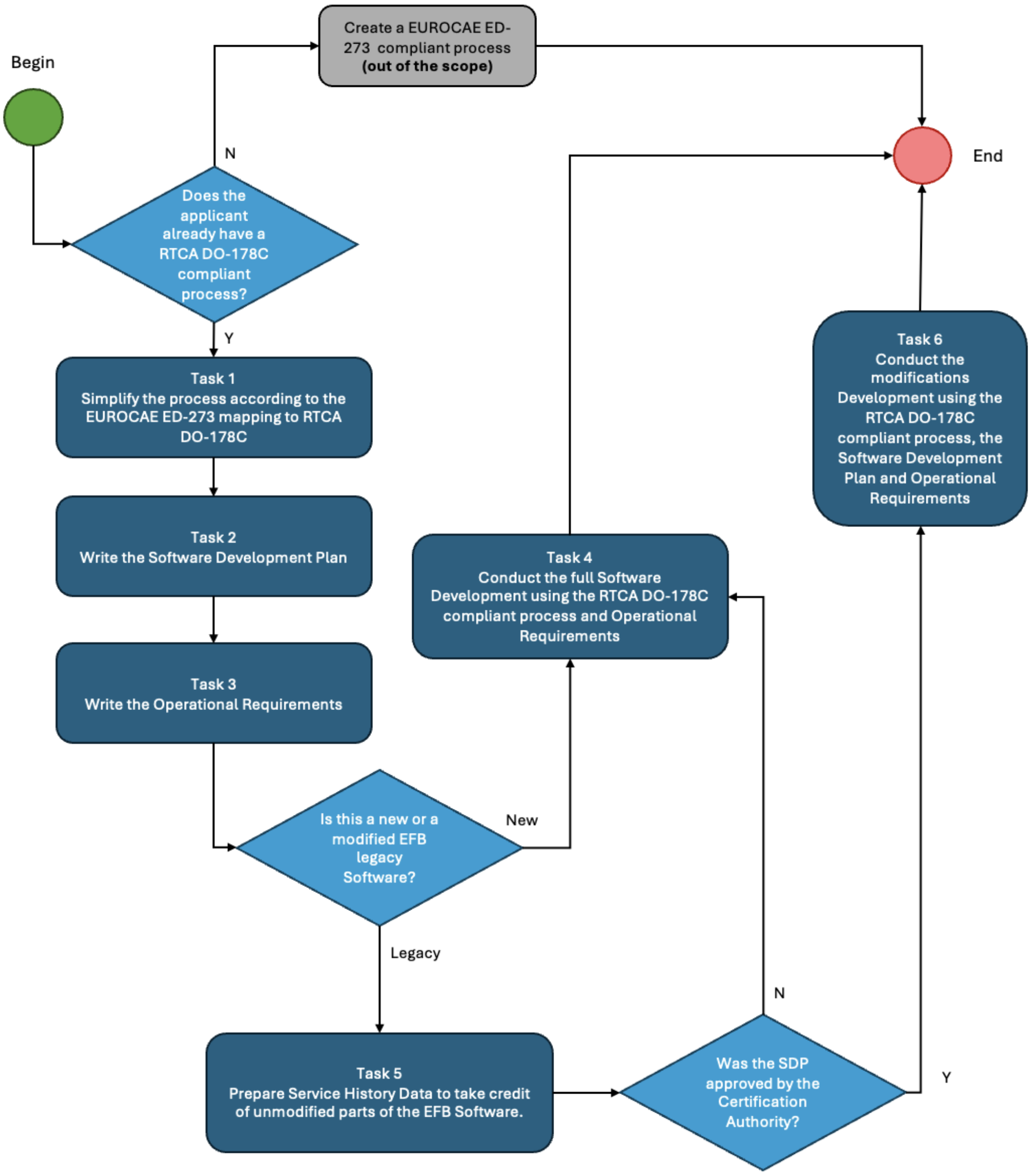

Fig. 4. Proposed development flowchart

approach would be required. If that is the case, a new evaluation by a focal team would be necessary.

If the proposed development approach is considered accepted by the focal team but at least one answer is Partially Agreed, then the author evaluates the comment provided by the respondent. In this case, the author could change the approach according to the feedback received.

All respondents Strongly or Partially agree with the statements. The results obtained by the focal team experiment suggest the experts accepted the approach as valid.

VII. Conclusion

This dissertation addressed a relevant topic for EFB manufacturers: compliance with EUROCAE ED-273. This relatively new standard brought unprecedented development assurance objectives for EFB development. It is a standard that brings quality to the software through the quality in the software process.

The proposed development approach brings benefits to applicants. It was shown that some parts of EFB manufacturers already have a RTCA DO-178C-compliant process. Therefore, it would be much easier to adjust this process and follow only one process for the entire company. On the other hand, the EFB manufacturers that do not have a RTCA DO-178C compliant process will benefit from the proposed development approach because more consolidated literature is available for RTCA DO-178C compared to EUROCAE ED-273.

REFERENCES

[1] RTCA, *DO-178C Software Considerations in Airborne Systems and Equipment Certification*, SC-205 Std., 2011.

[2] ——, *DO-178B Software Considerations in Airborne Systems and Equipment Certification*, RTCA Std., 1992.

[3] L. Marcil, "Realizing do-178c's value by using new technology: Oot, mbdv, tqc fm," in *2012 IEEE/AIAA 31st Digital Avionics Systems Conference (DASC)*, 2012.

[4] W. Youn, S. Hong, K. Oh, and O. Sung, "Software certification of safety-critical avionic systems: Do-178c and its impacts," *IEEE Aerospace and Electronic Systems Magazine*, vol. 30, no. 4, pp. 4–13, 2015.

[5] A. Paz and G. Bousaidi, "On the exploration of model-based support for do-178c-compliant avionics software development and certification," in *IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, 2016.

[6] A. Sarkis, J. Marques, and L. A. V. Dias, "Direcionadores para o desenvolvimento baseado emmodelos de software embarcado aeronáutico," *Cadernos do IME : Série Informática*, vol. 44, no. 1, 2020.

[7] S. VanderLeest and A. Buter, "Escape the waterfall: Agile for aerospace," in *2009 IEEE/AIAA 28th Digital Avionics Systems Conference*, 2009.

[8] J. Marsden, A. Windisch, R. Mayo, J. Grossi, J. Villermin, L. Fabre, and C. Aventini, "Ed-12c/do-178c vs. agile manifesto – a solution to agile development of certifiable avionics systems," in *9th European Congress of Embedded Real-Time Software and Systems*, 2019.

[9] Y. Moy, E. Ledinot, H. Delseny, V. Wiels, and B. Monate, "Testing, or formal verification: Do-178c alternatives and industrial experience," *IEEE Software*, vol. 30, no. 3, pp. 50–57, 2013.

[10] J. Marques and A. Cunha, "Verification scenarios of onboard databases under the rtca do-178c and the rtca do-200b," *36th AIAA/IEEE Digital Avionics System Conference (DASC 2017)*, 2017.

[11] J. Marques, S. Hayashi, and L. Barros, "Caracterização de cenários e garantias de segurança em carregamentos de software aeronáutico," in *Anais do V Workshop de Regulação, Avaliação da Conformidade e Certificação de Segurança*. Porto Alegre, RS, Brasil: SBC, 2019, pp. 1–10. [Online]. Available: https://sol.sbc.org.br/index.php/wrac/article/view/14031

[12] J. Marques, S. Yelisetty, and L. Barros, "A framework for loadable airborne systems," *IEEE Aerospace and Electronic Systems Magazine*, vol. 36, no. 5, pp. 38–47, 2021.

[13] P. V. Bhansali, "Universal software safety standard," *Software Engineering Notes*, vol. 30, no. 2, pp. 1–4, 2005.

[14] A. Ferreirós and L. A. V. Dias, "Evaluation of accomplishment of do-178c objectives by cmmi-dev 1.3," in *2015 12th International Conference on Information Technology - New Generations*, 2015, pp. 759–760.

[15] RTCA, *DO-332 Object-Oriented Technology and Related Techniques Supplement to DO-178C and DO-278A*, RTCA Std., 2011.

[16] ——, *DO-333 Formal Methods Supplement to DO-178C and DO-278A*, RTCA Std., 2011.

[17] ——, *DO-331 Model-Based Development and Verification Supplement to DO-178C and DO-278A*, RTCA Std., 2011.

[18] ——, *DO-200B Standards for Processing Aeronautical Data*, SC-217 Std., 2015.

[19] F. Yan, "Comparison of means of compliance for onboard software certification," *4th International Conference on Computer Science & Education*, pp. 1–4, 2009.

EUROCAE ED-273					DO-178C		
	Development process objective		FQL allocation		Table	Objective	Activity
	Section	Description	High	Low			
Development plan	2.4.2.1.1	Minimum considerations	x	x	Table A-1 Software Planning Process	1 - The activities of the software life cycle processes are defined.	4.2.a, 4.2.c, 4.2.d, 4.2.e, 4.2.g, 4.2.i, 4.2.l, 4.3.c
						3 - Software life cycle environment is selected and defined.	4.4.1, 4.4.2.a, 4.4.2.b, 4.4.2.c, 4.4.3
	2.4.2.1.2	Additional considerations	x			4 - Additional considerations are addressed.	4.2.f, 4.2.h, 4.2.i, 4.2.j, 4.2.k
					Table A-2 Software Development Process	5 - Derived low-level requirements are defined and provided to the system processes, including the system safety assessment process	5.2.2.a, 5.2.2.b, 5.2.2.c
					Table A-8 Software Configuration Management Process	1 - Configuration items are identified.	7.2.1.b
					Table A-1 Software Planning Process	4 - Additional considerations are addressed.	4.2.k
Operational requirements	2.4.2.2.1	EFB Function operational requirements definition	x	x	NOT APPLICABLE	NOT APPLICABLE	NOT APPLICABLE
	2.4.2.2.2	EFB application architecture definition	x	x	Table A-2 Software Development Process	3 - Software architecture is developed	5.2.2.a, 5.2.2.d
	2.4.2.2.3	EFB Function Operational Requirements validation	x	x	NOT APPLICABLE	NOT APPLICABLE	NOT APPLICABLE
	2.4.2.2.4	EFB Function compliance with operational requirements	x	x	NOT APPLICABLE	NOT APPLICABLE	NOT APPLICABLE
Software development	2.4.2.3.1	EFB Function software requirements definition	x		Table A-2 Software Development Process	1- High-level requirements are developed	5.1.2.a, 5.1.2.b, 5.1.2.c, 5.1.2.d, 5.1.2.e, 5.1.2.f, 5.1.2.g, 5.1.2.j 5.5.a
						2-Derived high-level requirements are defined	5.1.2.h, 5.1.2.i
	2.4.2.3.2	EFB function software requirements validation	x		Table A-3 Verification of Outputs of Software Requirements Process	1- Software high-level requirements comply with system requirement	6.3.1
						2-High-level requirements are accurate and consistent	6.3.1
						3-High-level requirements are compatible with target computer	6.3.1
						4-High-level requirements are verifiable	6.3.1
						6-High-level requirements are traceable to system requirements	6.3.1
	2.4.2.3.3	EFB function compliance with software requirements	x		Table A-6 Testing of Outputs of Integration Process	1- Executable Object Code complies with high-level requirements	6.4.2, 6.4.2.1, 6.4.3, 6.5
						2-Executable Object Code is robust with high-level requirements	6.4.2, 6.4.2.2, 6.4.3, 6.5
					Table A-7 Verification of Verification Process Results	1-Test procedures are correct.	6.4.5
						2-Test results are correct and discrepancies explained.	6.4.5
						3-Test coverage of high-level requirements is achieved.	6.4.4.1
Configuration Management	2.4.2.4.1	Configuration identification	x	x	Table A-8 Software Configuration Management Process	1 - Configuration items are identified	7.2.1
	2.4.2.4.2	Baselines establishment	x			2 - Baselines and traceability are established.	7.2.2
	2.4.2.4.3	Problem reporting	x	x		3 - Problem reporting, change control, change review, and configuration status accounting are established	7.2.3
	2.4.2.4.4	Change control	x			3 - Problem reporting, change control, change review, and configuration status accounting are established	7.2.4
	2.4.2.4.5	Archive	x	x		4 - Archive, retrieval, and release are established	7.2.7
Application Release	2.4.2.5.1	EFB Application conformity	x	x	Table A-9 Software Quality Assurance Process	1- Assurance is obtained that software plans and standards are developed and reviewed for compliance with this document and for consistency	8.2.b
						2 - Assurance is obtained that software life cycle processes comply with approved software plans.	8.2.f, 8.2.d
						5 - Assurance is obtained that software conformity review is conducted.	8.3
	2.4.2.5.2	Impact analysis of known issues	x	x	Table A-9 Software Quality Assurance Process	2 - Assurance is obtained that software life cycle processes comply with approved software plans.	8.2.d
						5 - Assurance is obtained that software conformity review is conducted.	8.3
					Table A-8 Software Configuration Management Process	3 - Problem reporting, change control, change review, and configuration status accounting are established.	7.2.3
Quality Assurance Process	2.4.2.6	Quality assurance	x	x	Table A-9 Software Quality Assurance Process	1- Assurance is obtained that software plans and standards are developed and reviewed for compliance with this document and for consistency	8.2.b
						5 - Assurance is obtained that software conformity review is conducted.	8.3

Fig. 5. Mapping between EUROCAE ED-273 and RTCA DO-178

FOLHA DE REGISTRO DO DOCUMENTO

1. CLASSIFICAÇÃO/TIPO	2. DATA	3. DOCUMENTO Nº	4. Nº DE PÁGINAS
DM	23 de julho de 2024	DCTA/ITA/DM-057/2024	72

5. TÍTULO E SUBTÍTULO:

A RTCA DO-178C oriented method to develop Electronic Flight Bag Software

6. AUTOR(ES):

Felipe Rodrigo Evangelista Matilde

7. INSTITUIÇÃO(ÕES)/ÓRGÃO(S) INTERNO(S)/DIVISÃO(ÕES):

Instituto Tecnológico de Aeronáutica – ITA

8. PALAVRAS-CHAVE SUGERIDAS PELO AUTOR:

Electronic Flight Bag, EFB, RTCA DO-178C, EUROCAE ED-273, Software Engineering

9. PALAVRAS-CHAVE RESULTANTES DE INDEXAÇÃO:

1. Pilotos de aeronaves 2. Sistemas de computadores embarcados 3. Engenharia de software

10. APRESENTAÇÃO: (**X**) **Nacional** ( ) **Internacional**

ITA, São José dos Campos. Curso de Mestrado. Programa de Pós-Graduação em Engenharia Eletrônica e Computação. Área de Informática. Orientador: Prof. Dr. Johnny Cardoso Marques. Defesa em 10/07/2024. Publicada em 2024.

11. RESUMO:

The development of EFBs - applications used by pilots inside the airplane cabin - has been growing in the last twenty years. These applications, most of them developed for iPad ®, help the pilots considerably to operate the airplane by reducing the workload, optimizing the airplane performance, and removing paper manuals from the airplane. In 2021 EUROCAE published the standard EUROCAE ED-273 for the development of EFBs and introduced, between some other relevant points, some requirements related to software development assurance. The standard, as expected, made EFB manufacturers concerned about how to follow the new standard. This work proposes to create an EFB development method that is compliant with EUROCAE ED-273 guidelines. The adopted method was to point to RTCA DO-178C - a widely recognized standard in aviation for embedded software development - and try to identify similarities and differences between the two standards. Therefore, a mapping between the two standards was performed. Furthermore, in what concerns legacy applications (i.e., applications that already have service history before the standard publication), this work proposes to create a method for new increments in legacy software to be compliant with the standard.

12. GRAU DE SIGILO:

(**X**) **OSTENSIVO** ( ) **RESERVADO** ( ) **SECRETO**