# Challenges in Teaching Aeronautical Software Quality Standards RTCA DO-178C and RTCA DO-200B

Johnny Marques
johnny@ita.br
Instituto Tecnológico de Aeronáutica
São José dos Campos, São Paulo, Brazil

Sarasuaty Yelisetty
sarasuaty@ita.br
Instituto Tecnológico de Aeronáutica
São José dos Campos, São Paulo, Brazil

## ABSTRACT

Teaching aeronautical software quality standards, such as RTCA DO-178C and RTCA DO-200B, presents significant challenges, mainly when delivered to students who do not work directly in this domain. This paper explores the complex characteristics of conveying these specialized standards to a diverse audience. We identify the primary challenges in the teaching-learning process through a detailed analysis of the difficulties faced. These challenges include the need to contextualize aeronautical quality standards with practical examples, the abstract nature of the content, and the necessity for interactive methods and teaching resources to facilitate comprehension. This study provides insights for educators and educational institutions to improve their approaches to teaching aeronautical software quality standards.

## KEYWORDS

aviation, software, audit, standard, DO-178C, certification

## 1 Introduction

Safety-critical systems develop within regulated environments, where even a small software error can lead to the loss of human lives or result in catastrophic consequences. Some prominent examples of such systems include aircraft control systems, nuclear reactors, and medical devices. The utmost priority lies in demonstrating the correctness of these software solutions with the highest level of assurance [14].

Safety-critical products must adhere to rigorous standards to meet stringent certification requirements imposed by regulatory agencies. This includes embedded software, which is crucial in ensuring system safety. The entire process of developing safety-critical software requires robust development, verification, configuration management, and quality assurance procedures [13].

As society becomes increasingly dependent on software and database systems, the need for reliability and safety intensifies. While the aviation industry has maintained a commendable track record, systems' growing complexity and criticality demand standardized approaches [25].

Software in regulated environments is critical in integrating various safety-critical products such as medical devices, nuclear power equipment, satellites, aircraft, vehicles, and more. The reliability of software is a paramount factor affecting the overall system reliability, as highlighted by Behera *et al.* [3]. Several approaches have been proposed to assess software reliability.

As noted by Wiegers & Beatty [40], software development is a challenging and resource-intensive task, especially with software projects' increasing size and complexity. Influential standards are established by committees, technical entities, or regulatory agencies to guide software development in regulated environments and address the associated risks [20]. Different domains have their specific software standards like RTCA DO-178C [31] in aviation and IEC 62304 [9] in medical devices.

This paper will delve into executing a comprehensive six-year program focused on teaching aeronautical software standards (RTCA DO-178C and DO-200B) to a group of professional master's students at a leading aerospace industry company. Over the duration of these courses, students were exposed to the critical principles and practices associated with these standards, which are essential for ensuring software reliability and compliance in the aviation sector.

The objective of this work is to assess the effectiveness and impact of the training program in software quality standards for the aeronautical domain (RTCA DO-178C e DO-200B) over six years (2018-2023) to professional master students. To achieve this, we administered a survey at the end of each class. These surveys were designed to capture the students' feedback in specific parts of our training. We aim to provide a comprehensive overview of the training's outcomes by analyzing the responses collected over six years.

Additionally, we conducted a thorough literature review to contextualize our challenges within the broader field of aeronautical software education. The combined insights from the questionnaires and literature review enabled us to identify key challenges and develop strategic approaches to enhance the training program. Our findings highlight areas of success and pinpoint challenges faced during the training.

Ultimately, this study offers valuable insights into the continuous development and refinement of educational programs in aeronautical software standards, contributing to enhancing safety and efficiency in the aerospace industry.

The findings of this work are of primary interest to the aeronautical software quality community, as Brazil is home to the third-largest aerospace industry, where software development for these products (airplanes, helicopters, drones, etc.) plays a crucial role. Reporting on the characteristics of a specific market may, as a consequence, encourage further research in the field of critical software development, not only in the aeronautical sector but also in other domains such as medical, automotive, railway, and nuclear, all of which have their own standards and regulations to meet. By showcasing a niche market in which Brazil plays a prominent role, it can open up avenues of interest for our undergraduate and postgraduate students.

This work contains seven other sections in addition to this introductory section. Section 2 presents the background. Section 3 presents the Professional Master's Program in Aeronautical Systems. Section 4 presents our Software Quality Standards Training. Section 4 presents our methodology. Sections 5, 6, and 7 present each stage of our methodology. Section 8 presents the conclusion and future work.

## 2 Background

### 2.1 Aircraft and Software Certification

In many safety-critical environments, obtaining approval or certification is a crucial step. The aviation industry, in particular, faces a demanding certification process that requires constant attention. Failing to meet the certification requirements during the project can lead to costly consequences, such as certification delays or even the inability to obtain certification at all [22].

Certification authorities like the FAA (Federal Aviation Administration), EASA (European Aviation Safety Agency), and ANAC (National Agency of Civil Aviation) are responsible for certifying aircraft in aviation.

Software plays a significant role in the systems or equipment installed in aircraft. It is not considered an isolated product but an integral component of the entire system. For this reason, both systems and equipment, including embedded software, must undergo an approval process to be accepted as part of the aircraft certification [16].

### 2.2 RTCA DO-178C

The RTCA DO-178C [31] document outlines comprehensive processes encompassing system aspects, software planning, development, verification, configuration management, quality assurance, and certification, as well as life cycle data.

The Radio Technical Commission for Aeronautics (RTCA) initially released RTCA DO-178 [28] in January 1982. This was followed by a revision, RTCA DO-178A [29], in 1985, which focused mainly on enhancing documentation. In December 1992, RTCA DO-178B [30] was issued, emphasizing objectives and life cycle data more.

The latest version, RTCA DO-178C, published in December 2011, includes supplements such as RTCA DO-330 (Software Tool Qualification Considerations) [32], RTCA DO-331 (Model-Based Development and Verification) [33], RTCA DO-332 (Object-Oriented Techniques) [34], and RTCA DO-333 (Formal Methods) [35].

RTCA DO-178C's 71 objectives are distributed across 10 specific tables (A-1 to A-10) within the standard. The processes described are structured around the objectives that must be achieved. Figure 2 illustrates the interconnections between the various tables (A1 to A-10) of RTCA DO-178C.

Each table includes guidance for 1) The process objectives applicable for each software level; 2) The independence required by the software level; and 3) The configuration control category applied by the software level for each software life cycle data. Table 1 presents the total of RTCA DO-178C objectives per table.

Figure 2 shows an example of the DO-178C table of objectives, presenting how the objectives and outputs are applicable for each software level. Each software level then corresponds to a failure condition category (related to the effect at the aircraft level), where:

**Table 1: RTCA DO-178C Objectives per Table.**

| Table | Title | Objectives |
|---|---|---|
| **A-1** | Software Planning Process | **7** |
| **A-2** | Software Development Processes | **7** |
| **A-3** | Verification of Outputs of Requirements Process | **7** |
| **A-4** | Verification of Outputs of Design Process | **13** |
| **A-5** | Verification of Outputs of Coding & Integration Processes | **9** |
| **A-6** | Testing Process | **5** |
| **A-7** | Verification of Test Results Process | **9** |
| **A-8** | Software Configuration Management Process | **6** |
| **A-9** | Software Quality Assurance Process | **5** |
| **A-10** | Certification Liaison Process | **3** |

A corresponds to *Catastrophic*; B to *Hazardous*; C to *Major*; D to *Minor*.

The levels defined in RTCA DO-178C are used to classify failure conditions based on their potential consequences:

- Level A (catastrophic) pertains to failure conditions that could lead to multiple fatalities, often resulting in the loss of the entire aircraft;
- Level B (hazardous) denotes failure conditions that cause a significant reduction in safety margins or functional capabilities;
- Level C (major) indicates failure conditions that result in a notable reduction in safety margins or functional capabilities;
- Level D (minor) represents failure conditions that lead to a slight reduction in safety margins or functional capabilities; and
- Level E (no effect) is assigned to failure conditions that would not impact the operational capability of the aircraft or increase crew workload.

In the context of RTCA DO-178C, Level E is not applicable, as it addresses scenarios where failures do not affect the aircraft's operations or crew workload. The software life cycle is defined within RTCA DO-178C in three processes:

- The Software Planning Process defines and coordinates the activities of software development and verification processes for a project;
- The Software Development Process produces the software product;
- The Software Verification Process ensures the correctness, control, and confidence of the software life cycle process and their outputs. The RTCA DO-178C does not use the term validation. The term verification includes the definition of validation; and
- Software Integral Processes that include configuration control, quality assurance, and certification.

The Software Development Process is divided into other four processes:

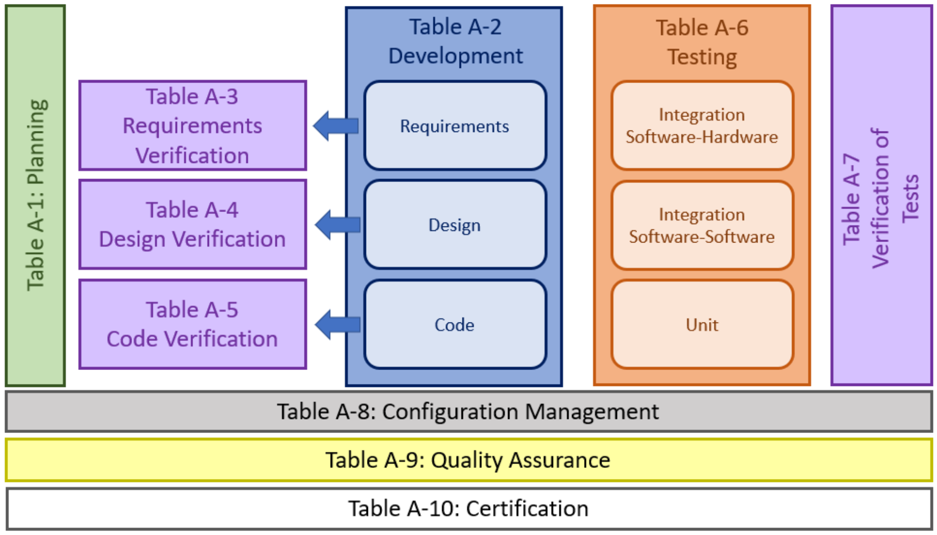


**Figure 1: Organization of Tables in RTCA DO-178C**

**Table A-3 Verification of Outputs of Software Requirements Process**

| | Objective | | Activity | Applicability by Software Level | | | | Output | | Control Category by Software Level | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Description | Ref | Ref | A | B | C | D | Data Item | Ref | A | B | C | D |
| 1 | High-level requirements comply with system requirements. | 6.3.1.a | 6.3.1 | ● | ● | ○ | ○ | Software Verification Results | 11.14 | ② | ② | ② | ② |
| 2 | High-level requirements are accurate and consistent. | 6.3.1.b | 6.3.1 | ● | ● | ○ | ○ | Software Verification Results | 11.14 | ② | ② | ② | ② |
| 3 | High-level requirements are compatible with target computer. | 6.3.1.c | 6.3.1 | ○ | ○ | | | Software Verification Results | 11.14 | ② | ② | | |
| 4 | High-level requirements are verifiable. | 6.3.1.d | 6.3.1 | ○ | ○ | ○ | | Software Verification Results | 11.14 | ② | ② | ② | |
| 5 | High-level requirements conform to standards. | 6.3.1.e | 6.3.1 | ○ | ○ | ○ | | Software Verification Results | 11.14 | ② | ② | ② | |
| 6 | High-level requirements are traceable to system requirements. | 6.3.1.f | 6.3.1 | ○ | ○ | ○ | ○ | Software Verification Results | 11.14 | ② | ② | ② | ② |
| 7 | Algorithms are accurate. | 6.3.1.g | 6.3.1 | ● | ● | ○ | | Software Verification Results | 11.14 | ② | ② | ② | |

**Figure 2: Example of a Table of RTCA DO-178C Objectives [31]**

- The Software Requirements Process uses the outputs of the system life cycle process to develop the software high-level requirements;
- The Software Design Process uses the outputs of the Software Requirements Sub Process to develop software architecture and software low-level requirements that can be used to implement source code;
- The Software Coding Process uses the outputs of the Software Design Sub Process to implement source code; and
- The Software Integration Sub Process uses the source code for generating the executable code to be loaded in the hardware.

The Software Configuration Process establishes the methods to achieve the objectives of configuration management throughout the software life cycle. It includes the environment, baselines, configuration items, archive, and release.

The Software Quality Assurance Process establishes the methods to achieve quality assurance objectives. It includes descriptions

**Table 2: Failure Condition Categories and DPALs**

| Failure Condition Category | DPAL |
|---|---|
| Catastrophic | 1 |
| Hazardous | |
| Major | 2 |
| Minor | |
| No Safety Effect | 3 |

of process improvement, metrics, and progressive management methods.

The Certification Liaison Process establishes communication and understanding with the certification authority throughout the software life cycle to assist the certification process.

## 2.3 RTCA DO-200B

RTCA DO-200B is a crucial document in aeronautics, especially concerning aeronautical databases' management and quality assurance. It provides standards and guidelines for processing aeronautical data to ensure accuracy, reliability, and suitability in aviation systems.

It addresses several key components, including data quality requirements, processing, organizational responsibilities, and certification and compliance. Regarding data quality requirements, RTCA DO-200B emphasizes accuracy, ensuring that the data correctly represents real-world entities and conditions; integrity, with measures to prevent data corruption and unauthorized alterations; resolution, with the appropriate level of detail necessary for specific aviation applications; and timeliness, ensuring that the data is up-to-date and relevant at the time of use.

In terms of data processing, the document describes standardized procedures and protocols for data handling from initial collection to final dissemination, as well as systematic methods to validate and verify the data against predefined criteria to ensure its correctness and reliability. Concerning organizational responsibilities, RTCA DO-200B defines the roles of different entities, such as data originators, who are responsible for collecting and initially processing the data following stringent protocols to ensure data quality; data processors, who must adhere to prescribed methods for further processing, including transformation and aggregation; and data users, who must follow guidelines to ensure proper application and adherence to data standards.

RTCA DO-200B establishes three Data Process Assurance Levels (DPAL). It represents the verification and validation tasks performed during data processing to assure data quality. Preliminary system safety assessment defines the DPAL. Table 2 presents the correlation of Failure Condition Categories and DPALs.

All data used to generate the Aeronautical Database must meet Data Quality Requirements (DQRs) specified by Data Processors. Data Quality Requirements shall characterize the data by Accuracy, resolution, confidence that data have not been corrupted while stored, processed, or transmitted, ability to determine the origin of data (traceability), level of confidence that data apply to the period of the intended use (timeliness), and format.

Figure 3 presents a typical Aeronautical Data Chain with the following 5 phases: Receive; Assemble; Translate and Select; Format; and Distribute.

The Receive phase involves identifying and selecting data sources that can support the development of the Aeronautical Database. The Assembly phase involves collecting and collating data from one or more suppliers. It should result in a database that meets the DQR of the next activity in the chain.

The Translate/Select phase involves the changes in how information is expressed. Checks are carried out to ensure that the integrity of the original data is maintained after translation. The appropriate data is also selected if needed.

The Format phase involves converting the selected data subset into a format that is acceptable to the following functional link in the chain. This may take the shape of a published exchange standard format for the transmission of data, a proprietary format for loading in a target application, or another agreed format. Checks are made to ensure that data elements are compatible with the selected format. The source of every error is identified, so that appropriate corrective actions can be taken.

The Distribute phase completes the processing data model and forms part of the transmission link in an aeronautical data chain. It involves delivering the formatted data set to users. Checks are carried out to ensure that the distributed data meets or maintains the DQR and that no media errors exist. If errors or omissions are identified, they are reported to the appropriate participant within the data chain, and procedures are followed to ensure that deficiencies are corrected and recorded for potential notification to data end-users.

# 3 Professional Master's Program in Aeronautical Systems

The professional Master's program in Aeronautical Systems is structured into three phases, each designed to provide comprehensive knowledge and expertise. As presented in Figure 4, the program contains three phases:

- Phase 1 - Fundamentals of Aeronautics: This phase covers the essential principles of aeronautics, including the basics of flight mechanics, aerodynamics, aircraft structures, and propulsion systems. Students will gain a solid foundation in the scientific and engineering concepts that underpin the design and operation of aircraft;
- Phase 2 - Systems Engineering: In this phase, students delve into the principles and practices of systems engineering, which is critical for managing complex aeronautical projects. Topics include systems integration, lifecycle management, risk assessment, and quality assurance. A key component of this phase is the course on Aeronautical Software Quality Standards (RTCA DO-178C & DO-200B), which provides in-depth knowledge on the guidelines and processes necessary to ensure the safety and reliability of aeronautical software; and
- Phase 3 - Master's Research: The final phase is dedicated to conducting original research in a specific area of aeronautical systems. Under the guidance of faculty advisors, students will develop and complete a research project that contributes to

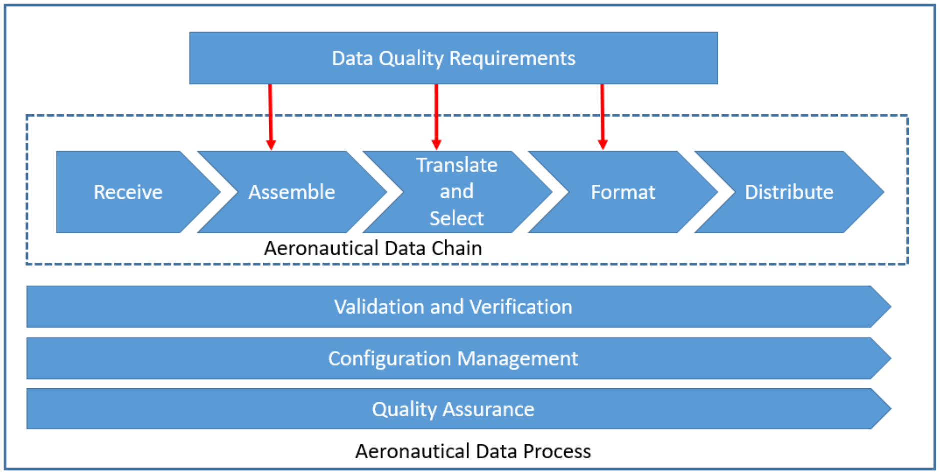


**Figure 3: The Aeronautical Data Process [14]**

advancing aeronautical science and technology. This phase emphasizes critical thinking, problem-solving, and applying theoretical knowledge to practical challenges.
This program is designed to equip students with the skills and knowledge necessary to excel in the aeronautical industry. It combines rigorous academic training with practical, hands-on experience.

## 4 Software Quality Standards Training

This summary outlines a comprehensive 40-hour course on Software Standards for the Aeronautics Sector, explicitly focusing on RTCA DO-178C and DO-200B. For the past six years, this course has been a vital component of a professional master's program, attracting newly graduated engineers from diverse disciplines who are specializing in the aeronautics industry.

The course was designed to provide in-depth knowledge and practical application of these critical standards, which are essential for ensuring aviation software's safety, reliability, and compliance. Students are introduced to the fundamental principles of RTCA DO-178C, which addresses software considerations in airborne systems and equipment certification, and RTCA DO-200B, which pertains to standards for processing aeronautical data.

Throughout the course, students engage in various learning activities, including lectures, hands-on exercises, and case studies, which collectively enhance their understanding of the standards' requirements and their implementation in real-world scenarios. The curriculum covers software planning, development, verification, configuration management, and quality assurance, all tailored to meet the stringent demands of the aeronautics sector.

By the end of the course, participants are equipped with the skills and knowledge necessary to contribute effectively to projects that require adherence to these standards. This training not only bolsters their technical expertise but also enhances their ability to navigate the complex regulatory environment of the aeronautics industry.

## 5 Methodology

Figure 5 presents the methodology used for improving the training program in aeronautical software standards. It consists of five main stages:

- Stage 1 - Literature Review: At this stage, a thorough literature review is conducted to gather insights and identify best practices related to aeronautical software standards. This review helps understand the current landscape, the related works, emerging trends, and existing challenges in the field;
- Stage 2 - Survey with students: The feedback collected through surveys is analyzed to gain valuable insights. These insights inform the refinement and improvement of the course, aiming to better align with the needs and expectations of students; and
- Stage 3 - Identify Challenges: Based on the insights from the literature review, the next step is to identify the specific challenges faced in training for aeronautical software standards. This could include issues such as regulatory compliance, resource limitations, cultural barriers, and the standards' complexity.

## 6 Stage 1

The development of aviation software adheres to standardization in aviation through the implementation of RTCA DO-178C [31] and RTCA Do-200B. Recent years have seen numerous papers discussing advancements and methodologies in aviation software development. The authors of this work have identified and categorized related research into six key themes:

(1) Impacts of transitioning from DO-178B [30] to DO-178C [31], as explored by Youn *et al.* (2015) [42], and Jimenez *et al.* (2017) [2].
(2) Experience reports and relation to maturity models, as investigated by Ferreiros & Dias [7], Machado & Marques (2023) [12], Marques *et al.* (2023) [17].

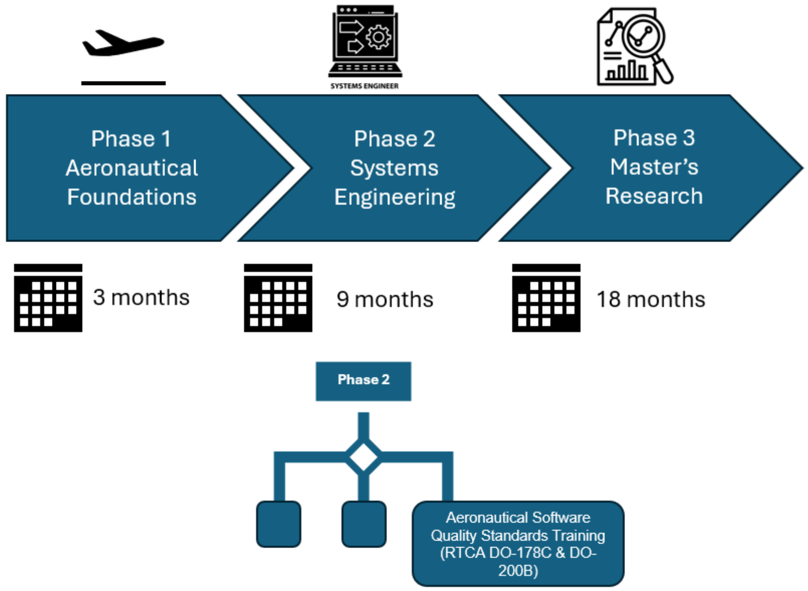


**Figure 4: Professional Master's Program in Aeronautical Systems**

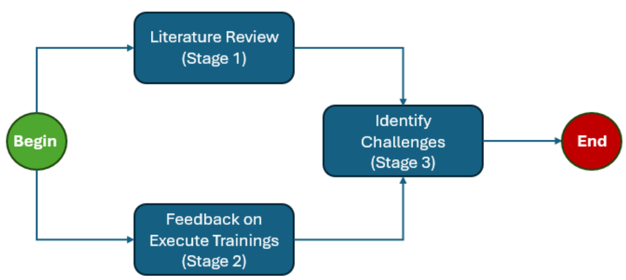


**Figure 5: Methodology**

(3) Model-Based Development, as investigated by Paz & Bousaidi (2016) [24], Sarkis *et al.* (2020) [36], and Panchal *et al.* (2022) [23].
(4) Use of Agile Methods in Software Development, examined by VanderLeest & Buster (2009) [38], Fernandes & França (2015) [6], and Marsden *et al.* (2019) [18].
(5) Formal Verification, discussed in the works of Moy *et al.* (2013) [19], Marques & Cunha (2017) [14], and Rodrigues *et al.* (2022) [27].
(6) Aeronautical Embedded Software Loading, addressed by Marques *et al.* (2019) [15], and Marques *et al.* (2021) [16].
(7) Focus on training on RTCA DO-178C and/or DO-200B, addressed by White (2017) [39], Anderson (2018) [1], Brown (2018) [4], DeLong (2019) [5], Smith (2020) [37], and Green (2021) [8].

We will summarize the works that focus on training on RTCA DO-178C and/or DO-200B

White (2017) [39] focuses on incorporating practical applications into aeronautical software training. The paper presents case studies and examples of how hands-on experience and real-world scenarios can significantly improve learning. White advocates for integrating simulations and project-based learning to bridge the gap between theoretical knowledge and practical application.

Anderson (2018) [1] emphasizes the importance of customized training solutions for aeronautical standards' compliance. The article discusses how tailored training programs can address the specific needs of different roles within an organization. Anderson highlights the benefits of personalized learning paths and modular training approaches in enhancing the effectiveness of training programs and ensuring compliance with standards like RTCA DO-178C and DO-200B.

Brown (2018) [4] discusses the challenges of training a diverse audience in aeronautical standards, noting the varying levels of expertise and backgrounds among trainees. The article emphasizes the need for customized training approaches for different roles, such as software engineers and quality assurance professionals. Brown highlights the effectiveness of tailored training programs in enhancing understanding and compliance across the board.

DeLong (2019) [5] delves into the multifaceted challenges faced in the development of aeronautical software, focusing on the complexity of adhering to stringent safety standards such as RTCA DO-178C. The paper highlights issues such as maintaining software reliability, managing extensive documentation, and ensuring compliance with evolving regulatory requirements. It underscores the need for robust training programs to equip engineers with the necessary skills and knowledge to navigate these challenges effectively.

Smith (2020) [37] explores the dynamic nature of aviation software standards and the implications of their continual evolution.

The article examines how updates to standards, including the transition from RTCA DO-178B to DO-178C, introduce new methodologies and tools that professionals must learn. Smith argues for the importance of adaptive training programs that keep pace with these changes to ensure that aviation software professionals remain compliant and proficient.

Green (2021) [8] examines the critical aspect of regulatory compliance in aviation software development. The article outlines the stringent requirements of standards like DO-178C and DO-200B and discusses the challenges developers face in achieving compliance. Green emphasizes the role of comprehensive training and continuous learning in ensuring that aviation professionals understand and adhere to these regulations, thereby avoiding costly compliance failures.

# 7 Stage 2

Implementing a system for continuous updates and incorporating feedback from past training sessions can ensure that the training material remains current and effective. Engaging with industry forums and standard bodies can provide insights into upcoming changes [26].

Between 2018 and 2023, spanning six-course editions, a survey was conducted to gather student feedback on various aspects. The survey aimed to identify the impact of diverse educational backgrounds, as the students came from different engineering disciplines. It also explored the challenges of teaching software quality standards to students without a computing background.

Additionally, the survey assessed the students' perceptions of the course's balance between theoretical and practical components. Another key focus was understanding the students' preferences regarding the course's schedule, precisely whether they preferred an intensive 40-hour week or a more spread-out schedule of 40 hours over five weeks, with classes held twice a week for four hours each.

The feedback collected through these surveys provided valuable insights into how the course could be refined and improved to better meet the needs and expectations of the students, ensuring a practical and engaging learning experience. Table 3 presents the three items used for the Survey.

# 8 Stage 3

Aeronautical software standards are not static; they evolve with technological advancements and emerging safety requirements. Keeping training material up-to-date with the latest revisions and best practices is a continuous challenge. Frequent updates mean that training materials can quickly become outdated, necessitating constant revisions and updates [37].

In 2011, the shift from RTCA DO-178B to DO-178C introduced new elements like model-based development and formal methods, which required comprehensive updates to training programs. Trainers must stay informed about changes and promptly incorporate them into their curriculum. In 2017, the RTCA DO-200B changed from RTCA DO-200A, introducing new elements. Both standards were used in its last version. So the need to continuously update training when standards evolve allowed us to identify Challenge 1.

*Challenge 1: Evolving Nature of Standards*

The survey instruments were developed following the guidelines provided by Kasunic (2005) [11], and the questions were specifically designed for this study based on the informed judgment of the authors responsible for structuring the training program on aeronautical software standards.

The audience for training in aeronautical software standards is highly diverse, encompassing software engineers, quality assurance professionals, project managers, and regulatory compliance officers. Each group has different levels of understanding and distinct training needs. For example, Engineers might need a deep dive into software coding standards, while quality assurance professionals must thoroughly understand verification and validation processes. This diversity necessitates a tailored approach to training that can effectively address the participants' varied backgrounds and expertise levels. Additionally, varying prior knowledge and experience can make it challenging to design a one-size-fits-all training program [4].

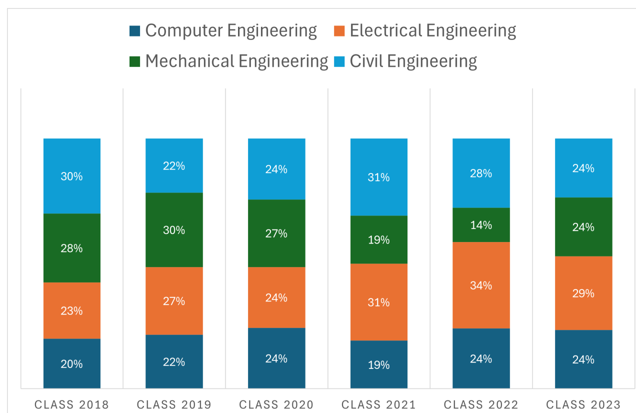


**Figure 6: Survey Results of Item 1, presented in Table 3**

*Challenge 2: Diverse Audience*

RTCA DO-178C and DO-200B are among the most detailed and stringent standards in the aviation industry, designed to ensure the highest levels of safety and reliability. These standards' sheer depth and breadth make them challenging to teach and learn. The complexity lies in understanding and applying the numerous requirements, such as software verification and validation processes and compliance with various criticality levels. For instance, differentiating and applying the objectives for different Design Assurance Levels (DALs) in RTCA DO-178C can be daunting for trainees who are not well-versed in these standards [5].

The intricate nature of these standards often results in significant knowledge gaps among developers and engineers, leading to potential non-compliance issues. In addition, the standards require meticulous documentation and detailed record-keeping, which can be overwhelming without proper training and experience. Over the

**Table 3: Survey items**

| ID | Item | Justification | Answer Type |
|---|---|---|---|
| 1 | Inform your engineering background. | This question aims to ask the respondent to indicate their undergraduate engineering degree. | Multiple choice |
| 2 | I believe that teaching standards like RTCA DO-178C and DO-200B is complex due to the amount of detail involved in meeting these regulations. | This item aims to gather perceptions on how complex it is to learn about these standards. | Likert scale (1 to 5) |
| 3 | I believe the course balanced theory with practice. | This item seeks to obtain the student's perception of the balance between theory and practice. | Likert scale (1 to 5) |
| 4 | Which schedule configuration do you prefer? | The proposed course can be delivered in two different formats to accommodate diverse learning needs and schedules: Intensive Week-long Format (5x8h/day in 1 week) and Extended Weekly Format (2x4h/day in 5 weeks) | Multiple choice |

course of six training sessions conducted between 2018 and 2023, a significant number of students perceived teaching both RTCA DO-178C and DO-200B standards as highly complex due to the extensive details involved in complying with these regulations, as presented in Figure 7. Specifically, the percentage of students who agreed with this sentiment increased from 15% in 2018 to 24.3% in 2020. Additionally, the percentage of students who strongly agreed with this assertion ranged from 13.5% to 17.5%.

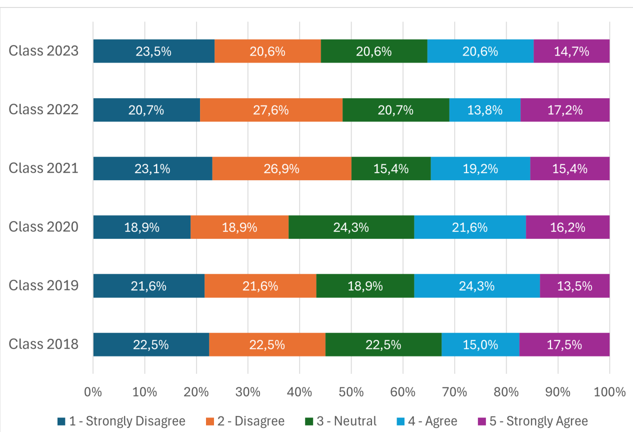


**Figure 7: Survey Results of Item 2, presented in Table 3**

This feedback is associated with the considerations detailed in [5], which offered a comprehensive understanding of the complexities and intricacies involved. By analyzing these insights alongside the feedback, we could pinpoint and clearly define Challenge 6, highlighting the specific areas that require focused attention and improvement.

*Challenge 3: Complexity of Standards*

Practical training in complex standards requires more than just theoretical knowledge. Trainees need practical, hands-on experience to understand how to apply these standards in real-world scenarios. For example, implementing real-world simulations and hands-on projects where trainees can apply DO-178C and DO-200B principles helps bridge the gap between theory and practice. Incorporating practical exercises has significantly enhanced the retention and understanding of complex technical material. Practical application also helps trainees develop problem-solving skills and gain confidence in their ability to comply with the standards [39].

Incorporating interactive elements such as case studies, simulations, and practical exercises can bridge the gap between theoretical knowledge and real-world application. Partnering with organizations specializing in aeronautical simulations can provide valuable hands-on experience [10].

Over the course of six years, student feedback has consistently highlighted the importance of adopting a more practical, hands-on approach in teaching software quality standards such as RTCA DO-178C and DO-200B. This approach allows students to better understand how to meet these requirements through examples and case studies. As illustrated in Figure 8, this perception has significantly increased: in the 2018 class, 55% of students found the course practical (with 25% responding as Agree and 30% as Strongly Agree). By 2023, an impressive 96.6% of students agreed that the course was hands-on (with 27.6% responding as Agree and 69% as Strongly Agree). This shift underscores the growing recognition of the value of practical learning experiences in mastering these complex standards.

The data feedback presented in Figure 8, associated with the considerations detailed in [39], offered a comprehensive understanding of the need to include more and more hands-on activities. By analyzing these insights alongside the feedback, we could pinpoint and clearly define Challenge 4.

*Challenge 4: Hands-On Experience and Practical Application*

Ensuring that trainees understand and can apply the stringent regulatory compliance requirements of RTCA DO-178C and DO-200B is crucial. Misunderstanding these requirements can lead to non-compliance, which has severe repercussions, including certification delays and increased costs. Detailed documentation and

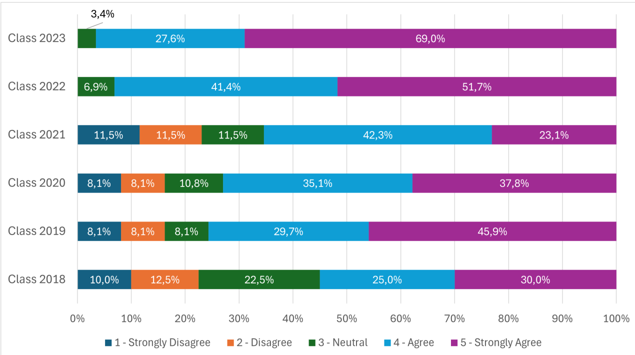


**Figure 8: Survey Results of Item 3, presented in Table 3**

meticulous verification processes are critical components of regulatory compliance that need thorough training. Regular assessments and practical case studies on past regulatory issues can help reinforce the importance of compliance and prevent costly mistakes. Understanding the implications of non-compliance and the potential risks involved is essential for all stakeholders [8].

Emphasizing the importance of regulatory compliance through dedicated sessions and real-life examples can enhance understanding. Regular assessments and quizzes can reinforce key regulatory concepts [21]. Although our survey does not cover the students' perception of regulatory compliance, the Literature Review has confirmed this as a challenge. So, we decided to include Challenge 5.

*Challenge 5: Regulatory Compliance*

Balancing the intensive training required to understand these standards with the regular duties of professionals can be challenging. Training sessions that are too long or too frequent can be difficult to accommodate. Intensive multi-day training sessions might be impractical for professionals who are also managing ongoing projects and responsibilities. Effective time management and integrating training into regular workflows can help alleviate this issue. Providing flexible training schedules and options for self-paced learning can also help professionals manage their time more effectively [41].

Tailoring training programs to the specific needs of different organizational roles can enhance effectiveness. Modular training sessions that allow participants to focus on relevant areas can help manage the diverse audience challenge [1].

As presented in Figure 9, most students prefer the course to be conducted over five consecutive weeks, with two 4-hour sessions per day, totaling 40 hours of training. The 2018 class delivered the course over five consecutive days with 8-hour daily sessions. The first survey conducted with this class revealed that students preferred a more distributed schedule over five weeks. Starting with the 2019 class, the course has been consistently offered over five weeks with two 4-hour daily sessions. However, at the end of each course, we continued to survey students to gauge their perception of this format. So, we decided to include Challenge 6.

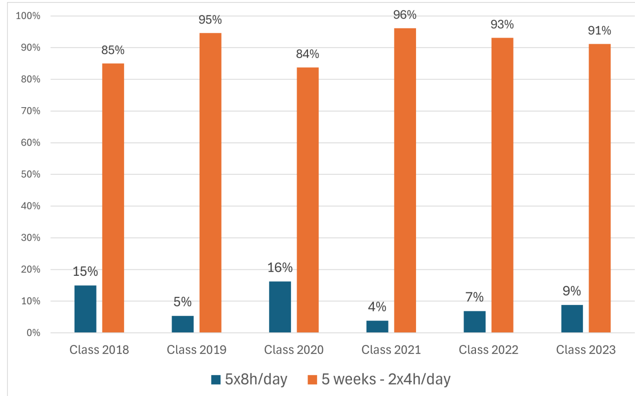


**Figure 9: Survey Results of Item 4, presented in Table 3**

*Challenge 6: Time Constraints*

## 9 Conclusion

In conclusion, the professional Master's program in Aeronautical Systems has been meticulously structured to provide a comprehensive education through its three phases: Fundamentals of Aeronautics, Systems Engineering, and Master's Research. Each phase builds upon the previous one, ensuring students acquire theoretical and practical knowledge. Including the Aeronautical Software Quality Standards (RTCA DO-178C & DO-200B) course within the Systems Engineering phase is particularly significant, as it equips students with essential competencies in software quality assurance, crucial for the aviation industry.

The course was offered within the scope of the Professional Master's Program in Aeronautical Engineering, during the Systems Engineering phase. It consisted of a 40-hour course held annually between 2018 and 2023, with the purpose of providing in-depth knowledge of the RTCA DO-178C (embedded software certification) and DO-200B (aeronautical data processing) standards, both essential for safety and compliance in aviation.

The target audience comprised newly graduated engineers from various fields undergoing specialization for work in the aeronautical industry. The methodology combined lectures, practical exercises, case studies, and interactive activities.

Regarding the format, during the pandemic, the course was conducted online in an intensive format (five days of eight hours) in 2020 and 2021, while in 2018, 2019, 2022, and 2023, it was delivered in person over five weeks (two four-hour classes per week).

Student evaluation was carried out through questionnaires administered at the end of each edition, aiming to capture perceptions about the balance between theory and practice, content complexity, workload, and course applicability. Over the six editions, a total of 245 students participated: 40 in 2018, 42 in 2019, 38 in 2020, 44 in 2021, 39 in 2022, and 42 in 2023.

Not all challenges emerged from the survey. Some were derived from the nature of the aviation sector itself or from the characteristics of the professional master's program. For example, the challenge Evolving Nature of Standards reflects the difficulty of keeping the training constantly updated in response to evolving standards. The challenge Diverse Audience stems from the fact that our professional master's program includes students with bachelor's degrees in different engineering fields (electrical, mechanical, civil, computer, naval, production, etc.) who are then trained in the aeronautical engineering domain. This diversity of backgrounds can complicate the understanding of aviation software regulations. The challenge Hands-On Experience and Practical Application is also connected to this diversity of backgrounds, combined with the inherent difficulty of teaching standards, since standards usually prescribe what must be done but not how. A training course, however, must explore examples, which are not always easy to find in the literature and may depend on the professional experience and expertise of the instructors.

The lessons learned are:

- Expand the use of real cases, projects, and audit simulations to better connect theory with certification scenarios.
- Modularize the content to address different profiles (software engineers, quality engineers, managers, etc.), enabling personalized learning paths.
- Integrate emerging topics (e.g., AI, advanced verification tools) to contextualize the standards in light of new technologies.
- Strengthen formative assessment mechanisms through quizzes to monitor individual progress throughout the course.

We identified some threats to validity:

- The study relied on questionnaires to capture students' perceptions, which may not fully reflect actual learning outcomes or the transfer of knowledge to professional settings.
- The heterogeneous profile of students (different engineering backgrounds, some without computing experience) may have influenced responses, introducing uncontrolled biases.
- The change in delivery format (face-to-face vs. online, intensive vs. distributed) over the years may also have affected students' perceptions, making direct comparisons across cohorts difficult.
- The study was conducted within a single professional master's program at a Brazilian institution, which may limit the generalizability of the findings to other academic or industrial contexts.
- Although the sample size was significant (245 students), it was limited to early-career professionals in the aeronautical field and may not reflect more experienced professionals or those from other regulated sectors.
- There is a risk of social desirability bias: students may have responded positively to please the instructors, thus inflating perceptions of the course's effectiveness.

Future work will focus on further refining and enhancing the program to address the evolving needs of the aeronautical industry. Key areas for future development include:

- Curriculum Enhancement: Continuously updating the curriculum to include the latest advancements and technologies in aeronautics and software engineering. This ensures that graduates are well-prepared to meet current and future industry demands;
- Integration of Emerging Technologies: Incorporating emerging technologies such as artificial intelligence, machine learning, and advanced data analytics into the program. This will give students cutting-edge tools and techniques to tackle complex aeronautical challenges;
- Industry Collaboration: Strengthening partnerships with leading aerospace companies and regulatory bodies to provide students with real-world experience through internships, collaborative projects, and guest lectures from industry experts.

## ARTIFACT AVAILABILITY

The training materials are available at https://www.gimps.info/materiais.

## REFERENCES


[1] P. Anderson. 2018. Customized Training Solutions for Aeronautical Standards. *Training and Development in Aviation* 22, 1 (2018), 12–18.

[2] J. Andres-Jimenez, J. Medina-Merodio, L. Fernandez-Sanz, J. Martinez-Herraiz, and J. Gonzalez-De-Lope. 2020. A Framework for Evaluating the Standards for the Production of Airborne and Ground Traffic Management Software. *IEEE Access* 8 (2020), 149142–149161. https://doi.org/10.1109/ACCESS.2020.3014794

[3] R.K. Behera, S. Shukla, S.K. Rath, and S. Misra. 2018. Software Reliability Assessment Using Machine Learning Technique. In *Computational Science and Its Applications – ICCSA 2018*. Springer International Publishing, Cham, 403–411.

[4] A. Brown. 2018. Training for Aeronautical Standards: A Diverse Audience. *Aviation Training Journal* 15, 1 (2018), 30–36.

[5] R. DeLong. 2019. Challenges in Aeronautical Software Development. *Journal of Aviation Technology* 23, 2 (2019), 45–57.

[6] F. Fernandes and R. França. 2015. Assessing the impact of Scrum in airborne software quality assurance. In *Anais do XIV Simpósio Brasileiro de Qualidade de Software* (Manaus). SBC, Porto Alegre, RS, Brasil, 204–211. https://doi.org/10.5753/sbqs.2015.15224

[7] A. Ferreirós and L. Dias. 2015. Evaluation of Accomplishment of DO-178C Objectives by CMMI-DEV 1.3. In *2015 12th International Conference on Information Technology - New Generations*. 759–760. https://doi.org/10.1109/ITNG.2015.132

[8] M. Green. 2021. Regulatory Compliance in Aviation Software. *International Journal of Aviation Management* 8, 2 (2021), 40–48.

[9] IEC. 2015. IEC 62304 Medical Device Software - Software Life-Cycle Processes.

[10] M. Johnson. 2021. Interactive Learning in Aviation Training. *Educational Technology in Aviation* 9, 3 (2021), 23–30.

[11] Mark Kasunic. 2005. *Designing an Effective Survey.* Technical Report CMU/SEI-2005-HB-004. Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA. https://doi.org/10.1184/R1/6573062.v1

[12] E. Machado and J. Marques. 2023. ME-MPS: An MR-MPS-SW extension model for critical software in regulated environments. In *Proceedings of the XXII Brazilian Symposium on Software Quality* (Brasília, Brazil) *(SBQS '23)*. Association for Computing Machinery, New York, NY, USA, 52–61. https://doi.org/10.1145/3629479.3629508

[13] J. Marques and A.M. Cunha. 2015. Use of the RTCA DO-330 in aeronautical databases. In *2015 IEEE/AIAA 34th Digital Avionics Systems Conference (DASC)*. 8D1–1–8D1–6. https://doi.org/10.1109/DASC.2015.7311479

[14] J. Marques and A.M. Cunha. 2017. Verification Scenarios of Onboard Databases under the RTCA DO-178C and the RTCA DO-200B. *36th IEEE/AIAA Digital Avionics Systems Conference.*

[15] J. Marques, S. Hayashi, and L. Barros. 2019. 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* (São Paulo). SBC, Porto Alegre, RS, Brasil, 1–10. https://doi.org/10.5753/wrac.2019.14031

[16] J. Marques, S. Yelisetty, and L. Barros. 2021. A Framework for Loadable Airborne Systems. *IEEE Aerospace and Electronic Systems Magazine* 36, 5 (2021), 38–47. https://doi.org/10.1109/MAES.2020.3040511

[17] J. Marques, S. Yelisetty, T. Slavov, and L. Barros. 2023. Enhancing Aviation Software Development: An Experience Report on Conducting Audits. In *Proceedings of the XXII Brazilian Symposium on Software Quality* (Brasília, Brazil) *(SBQS '23)*. Association for Computing Machinery, New York, NY, USA, 198–207. https://doi.org/10.1145/3629479.3629505

[18] J. Marsden, A. Windisch, R. Mayo, J. Grossi, J. Villermin, L. Fabre, and C. Aventini. 2019. 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.*
[19] Y. Moy, E. Ledinot, H. Delseny, V. Wiels, and B. Monate. 2013. Testing, or Formal Verification: DO-178C Alternatives and Industrial Experience. *IEEE Software* 30, 3 (2013), 50–57.
[20] J. Munch, O. Armbrunt, M.Kowalczyk, and M. Soto. 2012. *Software Process Definition and Management.* Springer-Verlag, Berlim, Germany.
[21] D. Nelson. 2020. Ensuring Regulatory Focus in Training Programs. *Compliance Today* 16, 4 (2020), 45–51.
[22] R. Oshana and M. Kraeling. [n. d.]. *Software Enineering for Embedded Systems: Mehtods, Practical Techniques, and Applications* (2 ed.). Elsevier.
[23] P. Panchal, S. Myschik, K. Dmitriev, P. Bhardwaj, and F. Holzapfel. 2022. Handling Complex System Architectures with a DO-178C/DO-331 Process-Oriented Build Tool. In *2022 IEEE/AIAA 41st Digital Avionics Systems Conference (DASC).* 1–8. https://doi.org/10.1109/DASC55683.2022.9925871
[24] A. Paz and G. Bousaidi. 2016. 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).*
[25] L. Rierson. 2013. *Developing Safety-Critical Software: A Practical Guide for Aviation Software and DO-178C Compliance.* CRC Press.
[26] K. Roberts. 2019. The Importance of Continuous Feedback in Training. *Journal of Continuous Improvement* 14, 2 (2019), 28–34.
[27] J.M. Silva Cardoso Rodrigues, J.E. Ferreira Ribeiro, and A. Aguiar. 2022. Improving Documentation Agility in Safety-Critical Software Systems Development For Aerospace. In *2022 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW).* 222–229. https://doi.org/10.1109/ISSREW55968.2022.00071
[28] RTCA. 1982. DO-178 Software Considerations in Airborne Systems and Equipment Certification.
[29] RTCA. 1985. DO-178A Software Considerations in Airborne Systems and Equipment Certification.
[30] RTCA. 1992. DO-178B Software Considerations in Airborne Systems and Equipment Certification.
[31] RTCA. 2011. DO-178C Software Considerations in Airborne Systems and Equipment Certification.
[32] RTCA. 2011. DO-330, Software Tool Qualification Considerations.
[33] RTCA. 2011. DO-331, Model-Based Development and Verification Supplement to DO-178C and DO-278A.
[34] RTCA. 2011. DO-332, Object-Oriented Technology and Related Techniques Supplement to DO-178C and DO-278A.
[35] RTCA. 2011. DO-333, Formal Methods Supplement to DO-178C and DO-278A.
[36] A. Sarkis, J. Marques, and L.A.V. Dias. 2020. Direcionadores para o Desenvolvimento Baseado emModelos de Software Embarcado Aeronáutico. *Cadernos do IME : Série Informática* 44, 1 (2020).
[37] J. Smith. 2020. Evolving Standards in Aviation Software. *Aerospace Today* 30, 4 (2020), 22–28.
[38] S.H. VanderLeest and A. Buter. 2009. Escape the waterfall: Agile for aerospace. In *2009 IEEE/AIAA 28th Digital Avionics Systems Conference.*
[39] L. White. 2017. Practical Applications in Aeronautical Software Training. *Software Engineering in Aviation* 12, 3 (2017), 19–25.
[40] K.E. Wiegers and J. Beatty. 2013. *Software Requirements 3.* Microsoft Press, Washington, EUA.
[41] J. Wilson. 2021. Time Management in Aviation Projects. *Project Management in Aviation* 10, 4 (2021), 55–61.
[42] W.K. Youn, S.B. Hong, K.R. Oh, and O.S. Sung. 2015. Software certification of safety-critical avionic systems: DO-178C and its impacts. *IEEE Aerospace and Electronic Systems Magazine* 30, 4 (2015), 4–13.