IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 23 March 2024, accepted 8 April 2024, date of publication 11 April 2024, date of current version 18 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3387730

TOPICAL REVIEW

Weaving Agility in Safety-Critical Software Development for Aerospace: From Concerns to Opportunities

J. EDUARDO FERREIRA RIBEIRO[1], (Member, IEEE), JOÃO GABRIEL SILVA[2], AND ADEMAR AGUIAR[1,3]

[1]Department of Informatics Engineering, Faculty of Engineering, University of Porto, 4200-465 Porto, Portugal
[2]CISUC, Department of Informatics Engineering, University of Coimbra, 3030-290 Coimbra, Portugal
[3]INESC TEC, 4200-465 Porto, Portugal

Corresponding author: J. Eduardo Ferreira Ribeiro (jose.eduardo.ribeiro@fe.up.pt)

This work was supported by the Component 5-Capitalization and Business Innovation of Core funding for Technology and Innovation Centers (CTI), integrated in the Resilience Dimension of the Recovery and Resilience Plan within the scope of the Recovery and Resilience Mechanism (MRR) of the European Union (EU), framed in the Next Generation EU, for the period 2021–2026.

ABSTRACT Domain-specific standards and documents heavily regulate safety-critical systems. One example is the *DO-178C* standard for aerospace, which guides organizations to achieve system safety and evidence for their certification. Under such regulated contexts, most organizations use traditional development processes, in contrast to the massive adoption of Agile in the software industry. Among other benefits, Agile methods promise faster delivery and better flexibility to address customer needs. Adopting Agile methods and practices are possible in aerospace because the *DO-178C* standard does not prescribe concrete software development methods. In spite of that, Agile development is not used in DO-178C contexts. To help change that, our research aims to understand whether and how organizations engineering safety-critical software systems for aerospace may benefit from Agile methods and practices. We analyzed the *DO-178C* standard and confirm that it is compatible with Agile methods. Then, we present a systematic literature mapping of adopting Agile in software development for aerospace, where we identified significant concerns, recurrent issues, and several challenges. Some real industry aerospace projects provided us with important data and the perspective of domain experts about the pros and cons of Agile methods in this context. We conclude by proposing an agenda of research opportunities to improve safety-critical software development towards agility that we consider worthy of further research, application and confirmation in wider contexts.

INDEX TERMS Agile, aerospace, DO-178C, FAA, safety-critical, software development.

I. INTRODUCTION

Safety-critical systems domains have always been heavily regulated by standards and documents that ensure confidence in the quality of such systems [1], [2]. Some examples are: *IEC 61508 Functional safety of electrical/electronic/programmable electronic safety-related systems* [3], a standard used in automotive, industrial control, and railway domains [4]; *EN 50128* [5], a standard for railway applications; and *DO-178C* [6], used for the development of airborne systems [1], [6].

The demanding requirements imposed by such standards and documents aim to eliminate the severe impact of functional failures of safety-critical systems in terms of death, injury, occupational illness, damage to or loss of equipment/property, or environmental damage [2].

Consequently, companies working in safety-critical environments must follow established standards and documents to ensure that their requirements are translated into precise specifications and included in all development lifecycle phases. The goal is to guarantee the quality and safety of the components and systems as a whole and obtain evidence that sustains a successful certification of the developed system [1], [7].

The traditional Waterfall model has been used for many years in safety-critical systems engineering (SCSE) organizations to satisfy these standards and documents. Among other aspects, standards and documents mandate that organisations

The associate editor coordinating the review of this manuscript and approving it for publication was Rosario Pecora.

produce safety-related evidence, a requirement for regulatory agencies to certify safety-critical products [7]. Standards and documents such as *DO-178C* do not specify in detail the development model that companies should use (e.g., Waterfall, Rational Unified Process, Scrum) to produce safety-critical systems [6], [8], [9]. Instead, these standards and documents provide guidelines and describe objectives for achieving the evidence, leaving space for working methods as long as sufficient evidence sustains that the objectives have been satisfied [6].

A. AGILE IN AEROSPACE

To gain a deep insight into how widespread Agile adoption is and how it is managed by organizations worldwide, CollabNet Version One launched the first State of Agile Survey in 2006 [10]. The main objective of the State of Agile Survey is to inquire about organizations and explore Agile adoption, trends, best practices, and challenges faced during the process. In the 2022 edition, as mentioned in the report, data from all the respondents was collected and analyzed, revealing that 80% of the organizations used Agile methods [11].

We analyzed the 2022 edition, which presents the use of Agile methods by industry type, to understand its adoption in SCSE. Even though the 2022 survey results show an 80% Agile adoption, there is a low adoption percentage in what is considered SCSE namely, healthcare and pharmaceuticals (8%), industrial manufacturing (5%), transportation (3%), and energy (3%), which represent a total of 19% of all industrial areas [11]. This 19% still represents only a small sample of technology companies worldwide.

Agile has become trendy, mostly because of its widespread successful adoption. Many safety-critical organizations turned to the Agile community asking 'How can we proceed and adopt Agile?' [12]. Furthermore, with the increasing demand for the aerospace sector to operate an Agile transformation to respond faster to customers' needs, many companies and researchers have begun to explore this field. Because aerospace is a complex and challenging environment, researchers want to understand how to achieve it and whether there are any limitations or incompatibilities [1], [13]. A few researchers have highlighted that even in the development of safety-critical software requirement changes are inevitable, despite the traditional initial exhaustive requirements analysis phase. However, in a Waterfall method, additional or changed requirements that appear during the development phase are quite difficult and costly to accommodate [1], [14].

B. RESEARCH QUESTIONS

Our research aims to understand whether the organizations engineering safety-critical systems adopt Agile methods and practices and why. We aimed to identify significant concerns, recurrent issues, and challenges, specifically in the aerospace domain regulated by the *DO-178C* standard.

We started with a Systematic Literature Mapping (SLM) to obtain a comprehensive overview of relevant literature and classify it into the following categories: number of publications per year, venue types, and targeted venues. We followed the guidelines of [15], that state that the outcome of a SLM is an inventory of publications mapped to a classification, allowing the discovery of research gaps and trends. We integrated the ensuing review of safety-critical software development employing Agile methods and practices in the aerospace domain with an analysis of industry data derived from real aerospace projects conducted by Critical Software SA (CSW).

The following research questions drove our work:

- **RQ1** – The Agile Manifesto has been around since 2001 [16], but research focusing on Agile for safety-critical systems appeared significantly later.
  What is the distribution of scientific publications about Agile methods within the aerospace domain over the years, the types of venues, and the venues?
- **RQ2** – While the software industry has been applying Agile methods and practices at large, they do not seem to be the focus for safety-critical software systems and their relevant standards and documents.
  What are the major concerns and challenges related to the adoption of Agile methods and practices for safety-critical software development, particularly in the aerospace industry?
- **RQ3** – The concerns and challenges above identified should be tackled.
  What are the main opportunities related to Agile methods and practices for the improvement of safety-critical software development, particularly in the aerospace industry?

These questions aim to capture the state of adoption of Agile practices and methods within the aerospace domain, characterize the known concerns and challenges, and identify potential opportunities for further research and development.

C. RESEARCH STRATEGY

To answer these research questions, we worked in three complementary phases: an overview of the key concepts, a literature search, and the study of post-mortem data from industry projects. A final analysis of all the findings concludes this study.

- **Literature review** – Section II overviews the most relevant Agile methods, their benefits and shortcomings, which are explored in more detail later together with the identification of major concerns and challenges (Subsection IV-F). Section IX introduces the concept of safety-critical systems and briefly analyses the *DO-178C* standard, showing that there is no incompatibility with Agile methods, and characterizes the software development process currently used by aerospace organizations.
- **Systematic literature mapping** – Section IV details the method used to conduct the literature search.

Subsection IV-E presents the spectrum of the existing literature on Agile adoption in the aerospace domain resulting from the SLM conducted, as well as major concerns and challenges identified. To ensure transparency and reproducibility of our SLM, we developed a replication package using a Replication Package Builder (RPB) [17]. The package includes the resulting datasets and offers comprehensive details about our search strategy, including the inclusion and exclusion criteria, data extraction forms, and results. These resources empower other researchers to replicate our study and build on its findings.

- **Industry project analysis** – Section V explores data from the post-mortem analysis of concrete aerospace industry projects, and the major concerns and challenges are gathered. Despite industry projects containing rich sets of information and data for research purposes, access to these projects' data is frequently not possible owing to confidentiality issues.
- **Findings and analysis** – Section VI details the opportunities identified for the possible improvement of software development for aerospace. Section VII summarizes the main findings and discusses the research questions. Section VIII briefly presents the main threats to validity identified during the study. Finally, Section IX identifies opportunities for further exploration, implementation, and assessment in pilot cases and summarizes the study outcomes.

II. STRENGTHS AND WEAKNESSES OF AGILE METHODS FOR AEROSPACE

In February 2001, 17 people gathered at the Snowbird Ski Resort in Utah and created the Agile Manifesto, a response to the need for an alternative to documentation-driven software development processes [16].

Islam and Storer [18] stated that Agile software development emerged as a response to the difficulties of the traditional Waterfall model and the Rational Unified Process (RUP) [19] to accommodate the highly volatile requirements for many software projects. A typical limitation of Waterfall and RUP processes is that software delivery is often slower than the pace of change in the problem-to-solve domain. The requirements for a particular project or available technology in the marketplace may change considerably over long iterations. Furthermore, in contrast to the traditional Waterfall method, in different Agile methods multiple software development activities can occur concurrently, including requirements analysis, design, implementation and testing within each iteration. A set of practices characterizes each Agile method to support development work and manage the complexity of concurrent activities [18].

Following the creation of the manifesto, several consultants independently developed methodologies and practices to grasp and react to the inevitable changes experienced by everyone, including organizational changes [20].

A. POPULAR AGILE METHODS

It is essential to understand that "Agile" can refer to several different methods comprising several practices [11]. The most well-known methods are Extreme Programming (XP) [21], [22], Scrum [23], [24], Kanban [25], and a few variations of these to support large-scale Agile, e.g., Large-Scale Scrum (LeSS) [26], Scrum@Scale (SaS) [27], Disciplined Agile Delivery (DAD) [28], and Scaled Agile Framework (SAFe) [29].

- **XP** – This is a method created by Kent Beck, comprising 12 practices. Some of the well-known XP practices are Pair Programming, Test-Driven Development (TDD), Refactoring, and Continuous Integration (CI) [21], [22].
- **Scrum** – Sutherland and Schwaber worked together to develop Scrum as a formal framework that employs various processes and techniques to improve the product continuously, team, and work environment [23], [24].
- **Kanban** – It was introduced and successfully used as a practice, as a flow control mechanism for pull-driven just-in-time manufacturing production by Toyota. This was introduced as a software development method by Anderson. Its goal is to minimize work in progress, producing a constant flow of released work items to customers, as the team focuses only on a few items at a given time [25].

B. POTENTIAL STRENGTHS FOR AEROSPACE

The authors of the Agile Manifesto stated the benefits of Agile in 2001. Some of these benefits include short iterations, frequent releases, flexibility to accept requirement changes, effective communication and relationship between customers and developers, increased visibility and predictability, and increased customer satisfaction [16].

VanderLeest and Buter conducted different pilot tests to implement Agile techniques in critical software system development [13]. These techniques are Test-Driven Development (TDD), Pair Programming, Continuous Integration (CI), Iterative Approach, Fixed Length Iterations and Client-Driven Adaptive Planning. They mentioned that these techniques helped aerospace software development teams deal with some difficulties, despite the specificity of the *DO-178B* standard. However, they needed to detail the concrete approach used to include these techniques and the produced results and even conclude that further demonstrations are needed.

The systematic literature review by Kasauli et al. [30] revealed several benefits of using Agile practices in critical software system development. Kasauli et al. elaborated on a list of the key benefits that were most discussed in the literature: improved stakeholder involvement, reduced costs, improved quality, efficient use of available information, improved safety culture, improved opportunities for reuse, improved management of changing requirements, improved prioritization, mapping of functional and safety requirements and better test cases.

In the analysis [31], Vuori also stated that Agile methods are expected to provide more control for the development process and to be able to deliver value to customers and developers earlier. The author also considers that Agile methods meet the challenges of changing requirements more efficiently than previous process lifecycle models.

In summary, Agile methodologies have a wide range of potential strengths in aerospace software development, including short iterations, frequent releases, flexibility with requirements, efficient use of information, improved stakeholder involvement, increased visibility, and adaptability to changing requirements.

C. POTENTIAL WEAKNESSES FOR AEROSPACE

However, there are also a few concerns regarding the adoption of Agile practices by safety-critical organizations. Heager and Nielsen [32] conducted interviews with critical software development teams and pointed out four different challenges to the implementation of Agile practices within the context of critical software development:

- **Use of documentation** – Keeping documentation to a minimum can be problematic because much of it must exist to comply with the standards.
- **Requirements engineering** – Dealing with requirements' uncertainty may negatively impact the developers.
- **Lifecycle** – Given that critical software development requires coordination with the produced hardware [33], owing to its embedded development, the iterations of the software development team are often interrupted by hardware teams, affecting the development process.
- **Testing** – Adopting an iterative testing strategy requires significant changes in work practices, which constitutes a challenge for testers. The need to learn new Agile testing skills can be problematic.

The above shortcomings and concerns related to adopting Agile methods and practices in the aerospace domain regulated by the *DO-178C* were the most relevant found in the literature review.

III. AEROSPACE AND THE DO-178C

Safety is closely linked to risk [2]. Safety-critical systems can be defined as systems whose functional failure can lead to death, injury, occupational illness, damage to or loss of equipment or property, or environmental damage [2].

As software plays a vital role in aerospace systems, it is essential to ensure high reliability. For example, in safety-critical systems, it is common to have ''embedded software'' as a vital component of the system. Embedded software is essential for managing overall equipment/system safety in combination with hardware. The system and software development processes and corresponding safety assurance are interrelated.

The most well-known examples of safety-critical system applications include aircraft flight control, landing gear, medical devices, rail control, weapons, satellites, rockets, and nuclear systems [2]. Future safety-critical systems will become more common, such as the auto driving system developed by Tesla, and more car brands that did or are doing the same (auto driving systems) [1], [2].

The traditional Waterfall model has been in use for many years in safety-critical systems engineering organizations. In addition to the Waterfall model, safety-critical systems have been developed using the V-Model, also known as the verification and validation model, which is considered an extension of the Waterfall model [2]. Covering formal verification and validation activities to assure the product quality, the V-Model was first proposed by Paul Rook in the late 1980s and is still actively used today in the domain of safety-critical systems [2].

Figure 1 highlights the relationships between primary standards and documents in the aerospace domain.

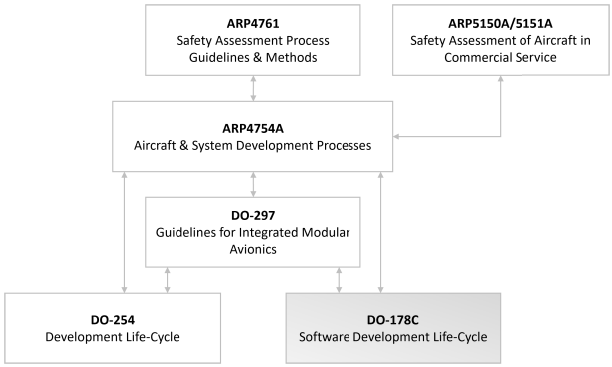

FIGURE 1. Safety-critical standards and documents for aerospace relationships, adapted from [2].

These standards and documents are pivotal in guiding safety assessment, electronic hardware and software lifecycle processes, and the overall system development process. Each development document provides essential guidelines to ensure the safety and reliability of aerospace systems, with *DO-178C* and *DO-254* being the most widely used.

Turning our attention back to Figure 1, we can see how it presents a concise overview of the interrelationships between key standards and documents governing avionics system. The diagram explicitly highlights the connections among the following standards and documents:

- **RP4761** focus on the assessment of system safety and provide guidelines to identify and mitigate potential hazards in avionic systems [35].
- **ARP4754A** deals with system development and address the activities and processes involved in the design, development, and validation of aircraft and systems [34].
- **DO-297**, known as the ''Integrated Modular Avionics (IMA) Development Guidance and Certification Considerations,'' offers guidance for developing integrated modular avionics systems [36].
- **DO-254**, or ''Design Assurance Guidance for Airborne Electronic Hardware,'' specifies the objectives

and activities necessary for designing and verifying electronic hardware used in airborne systems [37].

- **DO-178C** sets standards for software development assurance, ensuring that software for airborne systems is developed and tested to meet strict safety and reliability requirements [6].
- **ARP5150A/5151A** focus on electromagnetic environmental effects and guide the management and mitigation of the impact of electromagnetic interference on avionic systems [38], [39].

By adhering to these standards, documents and guidelines, aerospace developers and certification authorities can collaborate to create robust and reliable avionics systems that meet the industry's strict requirements [2]. This research focuses mainly on the *DO-178C* standard, also known as "Software Considerations in Airborne Systems and Equipment Certification," which is of utmost importance in the study of avionics system development and certification. The analysis conducted by [2] on the *DO-178C* standard establishes the essential guidelines and requirements that developers must adhere to when designing and implementing software for airborne systems.

A. DO-178 AND ITS EVOLUTION

In the past, software played a more straightforward role, primarily in enhancing the functions of mechanical and analog electrical systems. However, an important realization surfaced early: this approach could not ensure the reliability and safety required for intricate safety systems. This epiphany catalyzed the release of the inaugural *DO-178* certification document in 1980, as cited by Singh [40].

During this era, *DO-178* established a form of software certification grounded in "best practices," although its content remained abstract. Regulations were gradually refined through trial and error. *DO-178* introduced the concept of software verification requirements, with specifications contingent upon the software's safety criticality. This document also categorizes software applications into critical, essential, and nonessential sectors. Additionally, it intertwined the software certification process with relevant Federal Aviation Regulations (FARs), such as Type Certification Approval, Technical Standard Order (TSO) authorization, and Supplemental Type Certification [2], [40].

The maiden version of *DO-178* laid the foundation for the software certification. However, practical applications swiftly required revisions. As new needs emerged, more frequent updates ensued to enhance the guidelines. Table 1 illustrates the progression from *DO-178* to its latest incarnation, *DO-178C*.

The 1982 publication of *DO-178* established a "prescriptive set of design assurance processes for airborne software that focuses on documentation and testing" [2], [40], which underwent numerous refinements, each aimed at clarifying and advancing concepts. Iterations introduced novel ideas when necessary, as exemplified by the supplements in *DO-178C*. *DO-178*'s 1985 update, version A, introduced the concept of varied activities based on software component criticality (Subsection III-C). The subsequent version, B, introduced the concept of activities and associated objectives, leading to a comprehensive overhaul and fostering of development method flexibility in 1992. This iteration also delineates essential attributes that a design assurance process must possess [2], [40].

TABLE 1. ***DO-178*** **evolution, adapted from [2] and [6].**

Document	Basis	Year	Themes
DO-178	*498 & 2167A*	1982	Artefacts, documents, traceability, management and testing.
DO-178A	*DO-178*	1985	Process, testing, components, four critical levels, reviewers, Waterfall methodology.
DO-178B	*DO-178A*	1992	Integration, development methods, data (not documents), tools COTS.
DO-178C	*DO-178B*	2011	Reducing subjectivity, address modeling and reverse engineering.

The most recent version, *DO-178C*, emerged in 2011, retaining much of its predecessor's content while striving to clarify application nuances and eliminate inconsistencies. *DO-178C* also introduced the notion of supplements and supplemental documents addressing the application of cutting-edge technology without altering the core standard (Subsection III-B). The *DO-178* family of standards has long been viewed as the cornerstone of aviation safety. The literature affirms that its utilization has not been linked to significant aviation accidents [2], [40].

In summary, the journey from the early days of software's role in mechanical enhancement to the intricate certification processes of *DO-178C* showcases the evolution of software integration into safety-critical systems within the aerospace domain.

B. DO-178C AND RELATED SUPPLEMENTS

The *DO-178C* standard guides the aerospace community in developing an acceptable level of confidence in the software parts of airborne systems and equipment that must comply with the requirements of the standards and documents. It describes the complete system lifecycle process, including the safety assessment and validation processes; however, it does not describe the certification process for which we must refer to applicable regulations and guidance material issued by the certification authorities [2], [6]. Figure 2 shows the relationship between *DO-178C* and related supplements. As we can observe, *DO-178C* is what we can call a core document.

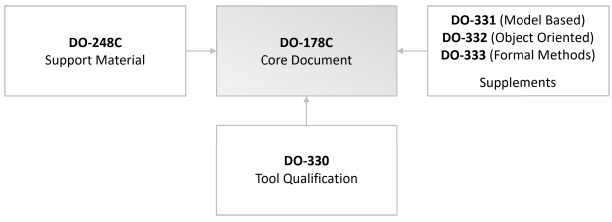

FIGURE 2. ***DO-178C*** **and related documents, adapted from [2].**

Each *DO-178C* supplement provides additional information on key topics, namely [2], [6], [8]:

- **DO-248C – Supporting Information for DO-178C and DO-278A**. Collection of Frequently Asked Questions (FAQs) and Discussion Papers (DPs) with applications of *DO-178C* and *DO-278A* in the safety assurance of software for aircraft [41].
- **DO-330 – Software Tool Qualification Considerations**. Standalone document referenced by *DO-178C* that provides guidance on tools qualification [42].
- **DO-331 – Model-Based Development and Verification Supplement to DO-178C and DO-278A**. Provides guidance for model-based development and verification used as part of the software lifecycle. Furthermore, it includes the outputs that would be used in the models and verification evidence that could be derived from them. Therefore, this supplement also applies to the models developed in the system process that define the software requirements or software architecture [43].
- **DO-332 – Object-Oriented Technology and Related Techniques to DO-178C and DO-278A**. Provides guidance when object-oriented technology or related techniques are used as part of the software development lifecycle (development and verification) [44].
- **DO-333 – Formal Methods Supplement to DO-178C and DO-278A**. Provides guidance when formal methods are used as part of a software lifecycle [45].

Each *DO-178C* supplement provides additional information on key topics [2], [6], [8]. While the current focus of this study is on *DO-178C*, these other documents may be considered in the future.

The main purpose of the *DO-178C* standard is not to explain ''how to develop'' but rather to ''support the development'' of software for airborne systems and equipment in order to ensure a level of confidence in safety that complies with specific airworthiness requirements/objectives [2], [6].

C. FAILURE CONDITION CATEGORY LEVEL

When developing an aerospace system, it is essential to identify the specific categories of associated failure conditions. This initial comprehension holds enormous importance in ensuring strict adherence to the mandated standards, documents and levels of certifying authority set during the validation process; otherwise, our systems will not be certified.

The *DO-178C* standard demonstrates confidence in a software component in proportion to its designated failure condition category level, which is often called its criticality. To achieve this, *DO-178C* categorizes the criticality of the components into five distinct Design Assurance Levels (DAL), as shown in Figure 3. This classification system acts as the foundation for determining the level assigned to a software component by considering its contribution to the potential failure conditions of the overarching system [2], [6].

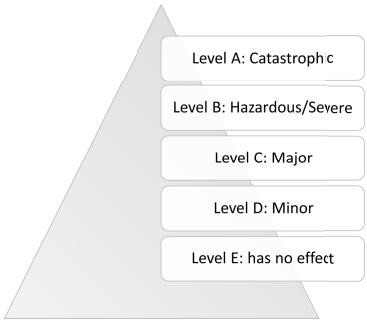

FIGURE 3. DAL pyramid, adapted from [2].

Below, we detail each DAL level caused by the software's anomalous behavior that determines or contributes to the failure of a system function [2], [6], [9].

- **Level A: Catastrophic** failure condition resulting in multiple fatalities and loss of the aeroplane.
- **Level B: Hazardous/Severe** is a failure condition that reduces the aeroplane or flight crew's capability to cope with adverse operating conditions. These effects can manifest as a large reduction in functional capabilities and safety margins, as fatal or serious injury to a small number of passengers besides the crew and can create physical distress or excessive workload of the flight crew that can no longer accurately or completely perform their tasks.
- **Level C: Major** failure condition that reduces the aeroplane or flight crew's capability to cope with adverse operating conditions. These failures can create a significant reduction in the functional capabilities, safety margins, discomfort or distress of the flight crew and passengers possibly including injuries or can significantly increase the workload of the crew impairing their efficiency.
- **Level D: Minor** failure condition would not have a significant effect on aeroplane safety; all crew members possess the knowledge needed to manage such incidents. These conditions may refer to a slight reduction in safety margins and functional capabilities, a slight increase in the crew workload, and some physical discomfort to passengers and cabin crews.
- **Level E: has no effect** on the criticality level, which does not impact the safety margins and does not affect the operational capabilities of the aircraft. While this error does not affect the functioning of the aircraft, justifications for this error (issue) must be provided to the Federal Aviation Administration (FAA).

DO-178C categorizes the DAL levels based on the required objectives, the impact of anomalies in the software component, and the failure rate of that component. These objectives encompass the process requirements outlined in the document, showcasing adherence to *DO-178C* when interacting with the certification authorities. Table 2 summarizes the overviews of these levels from the utmost criticality (level A) to the least (level E). A comprehensive set of

objectives and verification objectives are required to ensure the safety and reliability of the software. Notably, level E, positioned as the least critical, is the only level excused from requiring verification activities. We might still need to follow certain development practices if our software falls under level E, but the scrutiny and verification levels are lower than for the other levels.

TABLE 2. Number of objectives and verification objectives required by *DO-178C* for every DAL level, adapted from [2].

DAL	No. of Objectives	No. of Verification Objectives
A	71	43
B	69	41
C	62	34
D	26	9
E	0	0

Presenting all the objectives and verification objectives for each DAL summarized in Table 2 is not feasible owing to the extensive and complex information involved. For a thorough compilation of these objectives and verification objectives, directly referring to the *DO-178C* standard is recommended [2]. It offers comprehensive guidance on the requirements for each level, from A to E, elucidating the objectives, verification objectives, activities, and outcomes associated with the different software levels.

In addition, the *DO-178C* standard expands beyond failure conditions and outlines additional software-related considerations within the system lifecycle process [6]. These considerations encompass several aspects.

- **Parameter Data Items** - Consists of executable object code and/or data that can compromise one or more configuration items or data that can influence software behavior without modifying the executable.
- **User-Modifiable Software** - Consists of software or part that may be changed by the user within modification constraints without certification authority review.
- **Commercial-Off-The-Shelf Software** - Software that is already included in airborne systems.
- **Option-Selectable Software** - Consists of functions from some airborne systems and equipment that may be selected by software instead of hardware connector pins.
- **Field-Loadable Software** - Refers to software that can be loaded without removing the system or equipment from its installation.

In summary, the *DO-178C* standard surpasses the scope of merely addressing failure conditions. It embraces a wide array of software considerations incorporated within the context of the system lifecycle process. As highlighted in [2], at its core, the *DO-178C* standard is not intended to be a rigid and prescriptive rulebook for software development. Instead, its fundamental objective is to offer comprehensive guidance for software production in the aerospace domain. This approach recognizes the ever-evolving landscape of current software development while setting a clear path towards ensuring safety and reliability. By steering clear of a one-size-fits-all approach, the *DO-178C* standard actively promotes adaptability and innovation in the search for robust, Agile-friendly software solutions.

IV. SYSTEMATIC LITERATURE MAPPING

We conducted an SLM to provide an overview of the existing contributions related to the topic "Agile in safety-critical systems," specifically for the aerospace domain using the *DO-178C* standard. It should be noted that most engineering done for safety-critical systems is often not openly disclosed, primarily because of confidentiality concerns. The process we followed can be summarized in five main steps:

1) the definition of what to look for in the selected publications;
2) the planning of the publications search methodology, including the list of search terms;
3) the application of the search process and selection of publications where a filtered list of relevant publications is produced;
4) data extraction for the publications distribution;
5) publications search results validity evaluation by a second researcher.

The search protocol followed during the SLM is presented. Additionally, to ensure the reproducibility and transparency of our research, we assembled a comprehensive Replication Package [17] to enable other researchers to replicate our study or build upon our findings.

A. SEARCH PLANNING DEFINITION

We conducted a thematic analysis to ascertain the scope of publications regarding Agile adoption within the aerospace domain and standards/documents. Our objective was to identify if any relevant work had been conducted, and in which specific areas Agile adoption could be advantageous. Our analysis aimed to determine the overall distribution across years, types of venues, and specific venues and conducted a literature review. To structure our analysis, we followed the Population, Intervention, Comparison, and Outcomes (PICO) framework suggested by Petersen et al. [15].

- **Population** – Publications related to the aerospace domain, related standards and documents.
- **Intervention** – Agile methods researched within the aerospace domain and the related standards/documents and their wording variation used by the industry and researchers.
- **Comparison** – Although no empirical comparison has been made, different Agile methods or practices have been identified as alternatives to the existing traditional methods and practices (e.g., Waterfall and RUP).
- **Outcomes** – Trends of specific publications per year and the total number of publications related to Agile methods in the aerospace domain.

The keywords identified from the PICO criteria and variations used by different authors and organizations were grouped into the following sets to formulate the search strings:

- **Set 1** – terms related to the Agile methods, for example: Agile, Scrum, and Kanban.
- **Set 2** – terms directly related to the aerospace domain, for example: aerospace, avionic, and aeronautics.
- **Set 3** – terms related to safety-critical systems, for example: safety systems, high-integrity systems, and safety.
- **Set 4** – terms related to aerospace standards and documents, for example: *ARP 4761*, *DO-178B*, and *DO-178C*.

Furthermore, domain industry specialists (Aerospace and Agile) validated the keywords and search strings. This validation process involved a thorough analysis of the experimental results, leading to the refinement of the search strings. A RPB was developed and utilized to extract the results from the selected databases. This RPB effectively compiles scientific publications from multiple sources, including Scopus,[1] IEEE Xplore,[2] Engineering Village (Inspec),[3] Science Direct,[4] HAL Open Science,[5] Springer Nature,[6] and ACM Digital Library.[7] The replication package contains the tool, configuration settings, and result datasets. These components are provided to facilitate the accurate replication of the original study or serve as a foundation for further research, as detailed in [17].

We conducted and retrieved our SLM results using RPB Version 1.0.1 on Saturday, December 1, 2023 [17].

B. STUDY SELECTION

A simple database query can often yield hundreds or thousands of publications, many of which may not be relevant to the research topic [46]. To refine the focus of our search, we established and applied specific inclusion and exclusion criteria. This approach ensured that only publications pertinent to state-of-the-art analyses were included. The preliminary results of the publication search, based on the inclusion and exclusion criteria detailed in Subsections IV-B1 and IV-B2, were automatically processed using the RPB. For the final results, both the inclusion and exclusion criteria were applied through comprehensive full-text reading. In addition, snowball sampling was employed during the final step. This was done by analyzing the references of the selected publications to identify and include additional relevant publications in our sample, considering one level of reference. Finally, one author performed all the final publication reviews. Another researcher independently confirmed the final set of publications to mitigate the threat to the reliability of the SLM. Thus, the search was inclusive, taking a publication to a full-text reading in case of doubt.

[1]https://www.scopus.com/search/form.uri?display=basic
[2]https://ieeexplore.ieee.org/Xplore/home.jsp
[3]https://www.engineeringvillage.com/search/quick.url
[4]https://www.sciencedirect.com/
[5]https://hal.science/?lang=en
[6]https://www.springernature.com/gp
[7]https://dl.acm.org/

1) INCLUSION CRITERIA

The following inclusion criteria were applied to support the analysis:

(i) As the Agile Manifesto was created in 2001, only online publications with a starting date of 2001 or later were considered [16].
(ii) All online publications were from Scopus, IEEE Xplore, Engineering Village (Inspec), Science Direct, HAL Open Science, Springer Nature, and the ACM Digital Library.
(iii) Publications focused on Agile methods or practices in safety-critical software development, with a particular emphasis on standards and documents related to the aerospace domain, as outlined in the RPB configuration file named "query_terms.csv" [17].

2) EXCLUSION CRITERIA

The following exclusion criteria were applied to support the analysis:

(i) Studies unrelated to engineering and computer science, especially in the context of aerospace and safety-critical software development, as specified in the RPB configuration files "venues_to_exclude.txt", and terms to be excluded as detailed in the "query_terms.csv" [17].
(ii) Publications not presented in the English language.
(iii) Duplicated publications were not considered.

3) RESULTING PUBLICATIONS

As illustrated in Figure 4, the study selection process included and excluded several studies at each stage. These publications were essential for verifying the precision of our study's inclusion and exclusion criteria, culminating in a final validation by the first author. This method led to the identification of 45 publications.

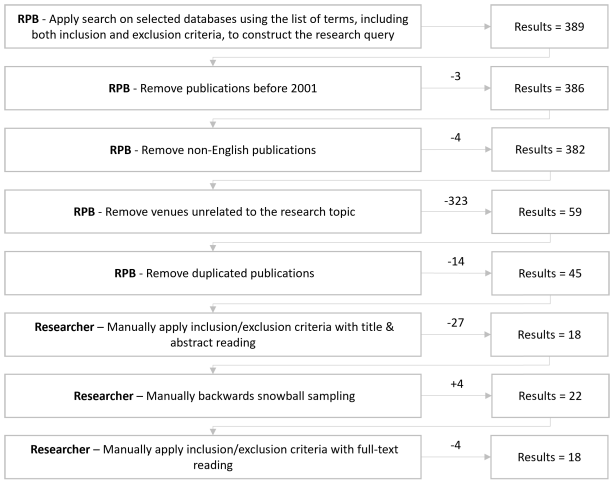

FIGURE 4. Number of publications included and excluded during the study selection process.

Subsequently, 18 publications were selected after manually implementing the inclusion and exclusion criteria by analyzing the titles and abstracts. This was followed by

snowball sampling, resulting in the discovery of 4 additional publications, totalling 22. The final selection, conducted through a full-text review, yielded 18 publications, with 4 excluded at this stage due to non-compliance with the inclusion criteria.

Repeatability is a fundamental aspect of research, emphasizing the need for meticulous documentation of the search process, as underscored in [15]. In line with this principle, we describe the SLM process in this Section. Furthermore, to address any potential threats to the validity of our study, both authors (first and third) conducted rigorous verification measures. We ensured reproducibility and transparency by developing and utilizing RPB [17]. This tool consolidates the generated datasets and offers comprehensive insights into search strategies. It includes detailed information regarding the inclusion and exclusion criteria, data extraction forms, and results obtained.

As shown in Figure 4, this process ultimately led to the selection of 18 publications for detailed analysis, listed as follows: [13], [14], [18], [47], [48], [49], [50], [51], [52], [55], [56], [57], [58], [59], [60], [61], [62], and [63].

C. DATA EXTRACTION

The initial step of data extraction was conducted by the first author, incorporating specific inclusion and exclusion criteria. This step involves executing the RPB for data extraction, as detailed in the template shown in Table 3 [17]. Moreover, 4 publications [13], [47], [49] and [50] served as benchmarks to automatically validate the search results, as specified in the "titles_to_validate.txt" file within the RPB configuration [17]. This procedure was followed by a review of the third author, who verified the accuracy of the extracted data and validated the outcomes, a practice commonly adopted in systematic reviews [15].

TABLE 3. Data extraction form.

Data Item	Value	RQ
Title	Name of the Article	
Publication Year	Year of Publication	RQ1
Venue	Publication Venue Name	RQ1
Venue Type	Publication Venue Type	RQ1
Authors	Name of the Author/s	
Link	Link for the Article	

D. VALIDITY EVALUATION

The data extraction process remained dynamic, allowing for continuous revisions and refinements by the first author. This flexible approach was applied to publications included after the third author's review. The third author validated the snowball sampling conducted, effectively minimizing the risk of overlooking relevant publications and mitigating the potential threat of missing key contributions.

E. SEARCH RESULTS

To better understand the state-of-the-art on the topic and the distribution of publications over the years, including the types and specific venues within the aerospace domain where these publications have been accepted, we used search results from the specified databases in Section IV. Based on this data, we created three charts. These charts provide a comprehensive overview of the spectrum of publications related to Agile adoption in the aerospace domain, along with relevant standards and documents [15].

Figure 5 shows the number of publications identified between 2001 and 2023. The first publication [52] focused on agility in the avionics software world. It was the only study until 2006. The second publication appeared two years later, in 2008. We observed that a moderate increase began in 2013. As shown in Table 4, our initial review of 389 studies found that only 4.6% of the publications were related to Agile methods and practices in aerospace software development.

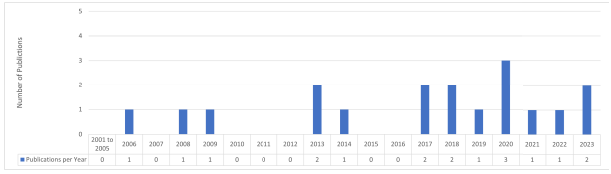

FIGURE 5. Number of publications per year.

Figure 6 provides an overview of the distribution of publications by venue type. We observed that a higher number of publications were accepted at conferences.

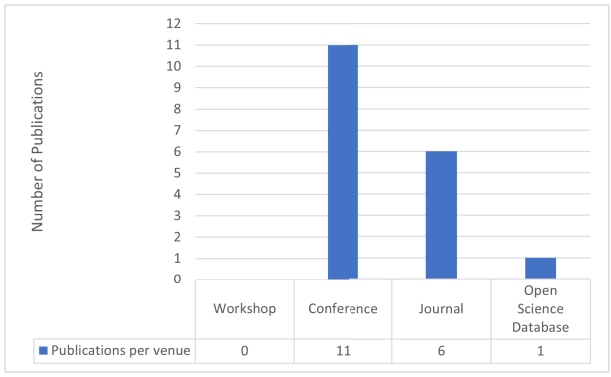

FIGURE 6. Number of publications per venue type.

An examination of the specific venues listed in Table 4 reveals the primary targets of the authors' publications. Predominantly, IEEE conferences stand out with four publications, closely followed by the AIAA/IEEE Digital Avionics Systems Conference, which contributed three publications.

TABLE 4. Number of publications per venue.

Venue	No. Publications	References
Computers in Industry Journal	2	[47], [57]
Communications in Computer and Information Science	1	[56]
AIAA/IEEE Digital Avionics Systems Conference	3	[49], [51], [58]
Advances in Intelligent Systems and Computing	1	[59]
Software - Practice and Experience	1	[60]
International Conference on Information Technology	1	[48]
Canadian Conference on Electrical and Computer Engineering	1	[61]
IEEE	4	[13], [50], [62], [63]
Integrated Communications, Navigation and Surveillance Conference	1	[14]
Reliability Engineering & System Safety	1	[18]
Extreme Programming And Agile Processes In Software Engineering	1	[52]
Hal Open Science	1	[55]

In summary, the data presented in Figures 5 and 6 and Table 4 indicate that the limited number of publications currently available in this field reveals a significant potential for future research by the scientific community.

F. LITERATURE REVIEW

The literature review encompassed the resulting 18 publications (Subsection IV-B3), providing valuable insights and contributing to our understanding of the topic.

1) EXPLORING AGILE METHODS

An academic project, described in [51], showcased the successful application of Scrum in the development of an Avionics Real-Time Embedded System (ARTES) with Cockpit Display Systems (CDS) for a fictional Unmanned Aircraft Vehicle (UAV). This project involved multidisciplinary collaboration and utilized cloud computing resources. This paper discusses the adaptation of Scrum, the application of Acceptance Test Driven Development (ATDD), and the use of a specific tool for modelling and source-code generation. While the study contributed to the extension of knowledge in this field, it is essential to note that Scrum applicability within the context of *DO-178C* is addressed at a high level without providing specific details on how each goal and output mandated by *DO-178C* was achieved. The study acknowledges the implementation of Scrum from the perspective of a software development team. However, there is a need to delve into the specific details of how each goal and output required by the *DO-178C* was accomplished.

Two related studies, [56] and [59], focused on applying Agile methodologies in complex, interdisciplinary, and technology-driven environments. The first study, by Marques et al., demonstrates the use of a modified Scrum framework called Q-Scrum within Interdisciplinary Problem-Based Learning (IPBL) at Brazil's Aeronautics Institute of Technology (ITA) [56]. Over seven years, Q-Scrum has been implemented in four courses, teaching students to create safety-critical prototypes while complying with software standards. This approach resulted in integrating Agile methods into software development, adopting Model-Based Development in a dedicated environment for reliable embedded software, developing checklists for walk-throughs and audits, enhancing collaboration in software quality education, and emphasizing multi-environment, standard-driven education. The second study, by [59], investigates the development and testing of a computer system for managing critical information during hypothetical crises to improve situational awareness and response coordination. Conducted in 2015, the project tackled challenges such as stringent specifications, Agile methodologies, embedded systems, software testing, and product evaluations. It emphasizes quality, reliability, safety, and testability, and incorporates the same interdisciplinary IPBL. The project employed hardware technologies like environment sensors, Radio Frequency Identification (RFID), and Unmanned Aerial Vehicles (UAVs) and developed a cloud-based, web-responsive platform and a mobile app for real-time resource management. The model-based development environment was vital for the embedded and safety-critical elements of the system. This system aims to improve the efficiency and coordination of response efforts in emergencies involving multiple public and potentially private entities with diverse cultures and management styles.

The research described in [60] explores the feasibility of applying agile methodologies to develop aircraft-embedded software. This paper presents the findings of an experiment involving the integration of Agile practices from Scrum with model-based development and distributed team collaboration. The experiment revolved around the development of an aircraft cockpit display system managed by five distributed teams. Three key aspects were examined and quantified based on the teams' deliverables: the quality of artifacts produced, adherence to Agile methodologies, and adherence to the *DO-178C* standard. The primary conclusion drawn from the experiment is the existence of a significant correlation between adherence to Agile methodologies and the quality of artifacts, indicating that Agile approaches hold promise for improving critical system development. However, it is also highlighted that Agile methods do not inherently address the challenges associated with integrating distributed teams and hardware/software integration, which can negatively impact artefact quality. The research results contribute to the ongoing debate regarding the suitability of Agile methods for safety-critical systems, particularly in the aerospace domain. The results indicate that teams with stronger adherence to Agile practices tend to work more efficiently, dedicating more time to hardware/software integration, and collaboration with other teams. Nevertheless, disparities in team performance in areas such as hardware and software integration suggest that Agile practices should be adapted to address these aspects from the outset of software development. This paper acknowledges potential limitations in the scope and duration of the experiment, emphasizing the need for future work to refine the experimental design.

2) INTEGRATING AGILE PRACTICES

In the field of aerospace, VanderLeest et al. explored the use of Agile practices in the context of *DO-178B* software development [13]. These practices include TDD, Pair Programming, and CI. VanderLeest et al. recommended integrating Agile practices into existing processes rather than pursuing a complete transition. However, while the study highlighted the importance of this integration, it did not provide detailed guidance on how to effectively implement these Agile practices to achieve the objectives and outputs outlined in the *DO-178B* standard. One notable research finding was the authors' emphasis on the collaborative efforts required to explore Agile methods and practices as a viable solution for complex systems that necessitate rigorous V&V. This reinforces the practical relevance of our research and its potential impact in the field. Nevertheless, further research and exploration are necessary to fully understand how to successfully apply Agile practices following the *DO-178C*

requirements, including overcoming potential challenges and ensuring that the desired outcomes are met. While challenging, the study showed that transitioning to Agile is achievable without abrupt changes to existing processes by gradually integrating Agile practices, methods, and automation to support a smooth transition.

In [48], the authors present the preliminary aspects of a doctoral research project that aimed to develop the Certifiable, Agile, Reusable, and Disciplined Reference Model (CARD-RM) for airborne software. The CARD-RM aimed to integrate Agile practices while maintaining compliance with the *DO-178C* standard. The study highlighted the importance of delivering early partial products, conducting informal testing, and performing formal verification. In a similar publication [49], summarized a doctoral research project focused on developing the CARD-RM as a Reference Method for safety-critical software requirements definition. The paper emphasised the intersection between discipline and agility in the CARD-RM and its compliance with *DO-178C* and *DO-331* standards. The authors highlighted the importance of writing requirements, prioritization, and functional modelling in the context of software architecture. Although both studies contributed to advancing knowledge in this field, more research is needed to explore how Scrum can be applied to the entire software development lifecycle in a context that adheres to the documentation and output requirements specified in *DO-178C*. This study does not present a detailed analysis of how each goal and output mandated by *DO-178C* was accomplished, including full traceability and V&V activities.

In [55], the authors explored the alignment and resolution of conflicts between Agile practices, the *ED-12C* [64], and the *DO-178C* standard in Airbus projects. This study investigated three distinct process approaches rooted in Scrum to enhance agility: Incremental, Functional, and Agile. The Incremental Process emphasizes sequential releases, each showcasing a functional increment. In the Functional Process, releases are structured around the functional packages. The Agile Approach furthers this by refining the granularity of implemented functions or features and executing each feature's software lifecycle individually. Quality gates are synchronized, with requirements, design, and test procedures being finalized only in the ultimate sprint. To address potential challenges, ''micro-reviews'' are utilized to conduct internal reviews for every process transition of each function. However, this approach requires careful consideration of certain aspects to prevent conflict. These considerations encompass elements such as user-oriented requirements, optimized configuration and change management, and highly automated CI and Certification.

In [52], the authors conducted a comprehensive investigation into applying XP and other Agile practices in aerospace software development, considering the rigorous regulations of *DO-178B*. The study explored the challenges of implementing an iterative development process and how Agile techniques could expedite development and effectively handle evolving requirements. The research highlighted the relevance and advantages of most Agile principles in the aerospace industry and recommended applying Agile practices across various phases of aerospace software development. Moreover, this study provided us with a holistic understanding of *DO-178B* and how Agile principles, along with selected Agile and XP practices such as Pair Programming and Automatic Testing, could prove beneficial in aerospace software development while adhering to stringent *DO-178B* regulations. This study demonstrates the need for continued research in the field of aerospace software development.

3) CULTURAL RESISTANCE IN AGILE SOFTWARE DEVELOPMENT FOR AEROSPACE

In [18], the authors investigated the introduction of Agile software development in an avionics company engaged in safety-critical system engineering. This study explored practitioners' experiences and challenges in adopting Agile methods through interviews. The findings emphasized the significance of addressing cultural resistance in effectively adopting Agile practices. Although this study offers valuable insights into identifying challenges and opportunities, it does not provide a comprehensive overview of the specific methods and practices implemented and how the goals and outputs outlined in *DO-178C* were successfully achieved.

4) DO-178'S IMPACT ON AEROSPACE SOFTWARE DEVELOPMENT

Kennedy's research examined the impact of the *DO-178* standard on aviation software development [14]. This research focused on addressing the challenges arising from the increasing complexity and costs associated with software development in the aerospace industry. These challenges stem from advancements in the processing power of embedded systems and expanding capabilities of software applications. Kennedy highlights the significance of adopting innovative techniques to effectively manage complexity. One approach is software reuse, which involves leveraging existing software components to minimize development efforts and enhance efficiency. Additionally, the study emphasizes the use of innovative software engineering techniques and Agile methods while ensuring adherence to safety standards. Although the study does not explicitly outline specific opportunities for improving safety-critical software development in the aerospace domain under the *DO-178C* guidelines, it significantly contributes to our understanding of the challenges involved. This serves as a valuable reminder of the importance of ongoing research efforts in this field, reinforcing the need for innovative approaches to address complexity while maintaining safety standards.

5) CERTIFICATION CHALLENGES IN AEROSPACE SOFTWARE DEVELOPMENT

The research carried out in [47] examined the development and certification of safety-critical software in the avionics

industry. The study highlighted the challenges that companies face in demonstrating compliance with certification requirements. The conclusions emphasized the importance of trust, collaboration, and incremental releases in achieving certification-ready software. Furthermore, this study enhances our understanding of how certification objectives and requirements affect the software development process. This study offers valuable feedback and an analysis of current postures and practices in the industry regarding certification activities. This overview also highlights the emerging industrial trend for implementing continuous development.

In [58], the authors discussed the challenges and requirements for developing embedded military software, particularly in the context of safety and mission-critical applications. As military software systems have grown increasingly complex, there has been a shift towards adopting more stringent software standards and processes, borrowing from successful practices in civil aviation, such as the *DO-178C* standard. This paper outlines a set of requirements to certify Airborne Military Software (AMS) to enhance software product quality for safety and mission success. The paper presents an approach for certifying AMS, considering various Software Development Lifecycles (SDLCs) and offering a defined set of 25 Process Requirements designed to elevate software product quality. These requirements are intended to be integrated into the processes established by the applicant. Additionally, the paper introduces five potential SDLCs that can be used to develop an AMS. This work proposes a framework for future applications in AMS development, potentially eliminating the need for specific standards such as *DO-178C*.

In [61], the authors addressed the challenge of certifying aerospace software, a process mandated by governmental agencies such as the FAA in the United States, typically following the *DO-178B* standard. Preparing *DO-178B* certification can be a complex task for companies seeking it for the first time, as it involves rigorous documentation and planning for high-integrity software. In response, this study explores the potential use of OpenUP, an open-source derivative of the Unified Process, as a foundational process model to facilitate the certification process. This study primarily focuses on software requirement activities, which are crucial within the *DO-178B* standard. This study examined how OpenUP can be applied to aerospace projects requiring *DO-178B* certification, particularly in the context of software design encompassing high and low-level requirements. Despite terminological differences in software architecture, requirement abstractions, and the absence of specific safety-critical software project support within OpenUP, the results suggest that OpenUP could still be beneficial for a first-time certification project. This paper outlines a future project to customize the OpenUP process to align with the *DO-178C* standard. This customization would entail the development of OpenUP plug-ins and experimentation on real software projects to enhance their suitability for certification under *DO-178B*.

The study [62] addressed the challenges faced by startups in the global urban air mobility and unmanned aerial systems market, which often lack experience and resources for traditional aerospace software and hardware development processes. To bridge this gap, this study introduces a tailored workflow based on a subset of objectives derived from safety-critical software standards *DO-178C*/*DO-331*. These objectives were selected based on factors such as importance, automation feasibility, and reusability. This custom workflow aims to establish an efficient and highly automated development cycle, resulting in higher-quality software with improved maintainability, primarily for research and prototyping. In addition, it can serve as a compliance framework for software in applications such as unmanned aerospace systems, urban air mobility, and general aviation. The custom workflow generates essential development and verification artifacts and provides a scalable foundation for potential future certification in alignment with *DO-178C* the *DO-331* standards. This paper illustrates the application of this custom workflow through a case study involving an Autopilot Manual Disconnection System.

6) ENHANCING DEVELOPMENT AND DOCUMENTATION THROUGH AUTOMATION

The [63] research explored the application of DevOps principles to aerospace software development as a means to address the challenges associated with strict safety processes, which often result in significant cost and schedule implications. DevOps principles, known for enhancing automation, providing faster feedback, and promoting continuous improvement in software development, have historically been applied to software implementation, but have not typically encompassed domain-specific requirements such as *DO-178C*. By automating the tasks and evidence generation required by *DO-178C*, this study aims to achieve notable speed advantages and reduce the risk of human errors in the development process. The preliminary findings of the research project indicate that over 75% of common defects, especially those prone to becoming open Problem Reports (PRs), can be effectively addressed through a robust automation infrastructure. Additionally, significant improvements have been observed in cycle time and a reduction in the use of regression-oriented testing, leading to the conclusion that a substantial shift in safety-driven software development, referred to as ''Cert DevOps,'' is feasible. This shift enables a more Agile and robust development lifecycle, which is crucial for accommodating the numerous new features needed in aerospace to support single pilot operations. The implications of this transformation include the potential for auditing the automation infrastructure rather than the resulting automated output, which could serve as the foundation for future standard development. In summary, automation of the *DO-178C* development process not only accelerates development by eliminating manual tasks but also reduces the probability of errors, including those related to compliance with development processes. This innovative

approach, labelled ''Cert DevOps,'' demonstrates the potential to address a significant portion of problem reports and offers the promise of increased project throughput and up to a 50% reduction in certification-related documentation in aerospace software development.

The authors in the [50] study focus on addressing the challenges of implementing Agile methods in the development of safety-critical software systems, particularly in the aerospace industry. The main challenge is documentation, which is traditionally seen as an obstacle to Agile methodologies in this context. To address this issue, this study presents the development and implementation of automated document processing and management tools. These tools are designed to improve the efficiency and effectiveness of documentation activities in safety-critical software system development, more precisely for the aerospace domain in compliance with *DO-178C*. They were developed collaboratively with industry professionals and were iteratively co-designed and validated within industrial projects. The research included interviews with professionals to validate the tools and to collect feedback. In addition, synthetic tests were conducted to confirm the feasibility and benefits of document automation in the safety-critical software development industry. However, these tools still need to be qualified to comply with the DO-330 standard, which is relevant for tool qualification in this domain. The ultimate aim is to develop a complete Documentation Management Tool (DMT) that can handle all aspects of project documentation, ensuring that it is automatically verifiable and certifiable, thereby improving the overall experience and performance of professionals in the safety-critical software industry.

Finally, the research [57] is an evolution from the previous one [47] by the same authors and addresses the challenge of improving quality while optimizing development costs in the creation of *DO-178C/ED-12C* safety-critical software. It challenges traditional V-Model software development practices and advocates adopting a ''continuous certification'' process grounded in Agile methodologies. This paper proposes a new framework to facilitate continuous certification in safety-critical software development. This framework is detailed, emphasizing its compliance with existing certification standards. This article also introduces a tooling solution based on open-source and off-the-shelf components to implement this framework. The effectiveness and efficiency of this approach are demonstrated through an industrial case study. The proposed framework and tooling were tailored to ensure compliance with the rigorous requirements of the certification process. They focused on creating engineering data such as specifications, codes, unit tests, and integration tests and maintaining traceability between all elements. This ensures that all components have undergone independent reviews, and that certain activities are carried out by different individuals, enhancing the reliability of the process. A distinctive aspect of the proposed framework is its hybrid nature, which incorporates elements from various certification-compatible methods. This includes frequent releases and automation of repetitive tasks, especially testing from DevOps and Scrum; Scrum-specific ceremonies such as Daily Meeting, Planning Poker, the concept of Minimal Valuable Product (MVP), and specific role distributions among team members; and the Pull/Merge Request mechanism for software engineering data to systematize independent reviews. The practical application of this framework, tested in developing an application for a safety-critical embedded computer in the aerospace industry, indicates a significant increase in productivity compared with similar initiatives. However, the article also acknowledges the challenges in adopting this new approach, particularly the paradigm shift it represents in the methods and tools for organizations accustomed to complex processes for certification compliance. It highlights initial investment and risk aversion during certification audits as potential obstacles to adopting this new approach. Additionally, the article notes the need for trained and competent resources familiar with these innovative solutions, which are rarely included in engineering curricula.

G. KEY CHALLENGES

These studies shed light on the challenges, opportunities, and practical considerations of applying Agile methods to develop safety-critical software. The insights gained from this literature review contribute to a better understanding of how Agile practices can be effectively integrated into the context of safety-critical systems, enabling organizations to enhance their software development processes while maintaining the required level of safety and compliance. Nonetheless, while reviewing the publications, we realized that they mainly relate to the applicability of Agile values, principles and methods, and requirements management. Overall, Agile requirements management is the most discussed topic in these publications. Therefore, based on the literature review, we identified and summarized four crucial topics in the subsequent subsections: requirements management, late requirement changes, documentation requirements, and Agile methods and practices. In addition to the literature review, Subsection IV-G5 addresses several unanswered aspects related to *DO-178C*. The primary focus is on the outputs of processes and plans, the possibility of reusing documentation, managing requirement changes, and emphasizing the significance of independence and change control.

1) CHALLENGES FOR REQUIREMENTS MANAGEMENT IN AEROSPACE

In various publications, there is consensus regarding the challenges in **requirements management**. For instance, [57] highlighted that software certification issues primarily revolve around the requirements and **traceability**. Similarly, [62] pointed out that product development faces obstacles, such as high costs, long development cycles, and extensive certification requirements. This opinion is reinforced by others, who note that recent aeroplane designs

have experienced significant delays and cost overruns, partly because of the rapid pace of technological evolution. To deal with the requirements and their complexity increases, companies rely on increasing the team size and extending the design cycle.

There are better solutions to increasing the **team size**. Let us consider a scenario in which new people without proper product knowledge or context are added to a team. In this scenario, they will need time to acquire that knowledge, and the current teams can slow down because they might need to help and support the newcomers. Potential solutions can be derived from the proposed opportunities for improvement; we are researching better alternatives and describing them in a follow-up publication. Requirement management in aerospace is very complex, as stated in [14]. The later a defect is discovered, the costlier is its correction, a principle that applies particularly to safety requirements. There is an essential need to improve the management and **validation** of requirements, which concerns the industry and regulators. **Poor requirements** can compromise the safety and lead to substantial correction costs. Wils et al. investigated how Agile techniques could accelerate development and adapt to changes in requirements [52]. The study examined improvements across different phases: software development, embedded systems (integration and testing), and certification. In the software phase, they found openness to changes in the requirements, including hardware alterations. The embedded phase lacks details regarding the flexibility for requirement changes. In the certification phase, changes are advised to be minimal because of the complexity and effort required for each software change, which involves tasks such as code coverage analysis, reviewing non-functional requirements, traceability, and manual testing.

Regarding documentation and traceability, the suggestion is to manage these aspects more like source code using tools such as DOORS.[8] Such tools can potentially reduce the time spent on document creation, management, and review. However, even with these tools, managing traceability remains a challenge, especially since manual **verification** of documents is often required by certification regulations when using uncertified tools. An alternative approach was explored by [56], [59], and [60], who investigated model-based methods combined with Agile practices. Marques et al. describes the implementation of a modified Scrum framework, Q-Scrum, for developing safety-critical prototypes in compliance with software standards [56]. Reference [59] focused on developing and testing a critical information management system for crisis situations, emphasizing the role of model-based development in handling embedded and safety-critical aspects. Reference [60] examined the use of Agile methodologies, particularly Scrum, combined with model-based development in creating an aircraft cockpit display system. These studies demonstrate the potential and challenges of integrating Agile methodologies into requirement management and model-based development. They highlighted the importance and applicability of these methods in complex, interdisciplinary, and technology-driven scenarios, especially in the safety-critical aerospace software development domain.

Although the literature review offers insights into the importance of managing requirement changes within the *DO-178C* context, more comprehensive guidelines and frameworks are required to implement effective change management processes. Further research should focus on identifying suitable change management strategies, defining clear change control procedures, and evaluating their impact on software quality, cost, and schedule.

2) CHALLENGES HANDLING LATE REQUIREMENT CHANGES IN AEROSPACE

Following the previously identified challenges in requirements management within aerospace software development, handling **late requirement changes**, as described in various publications, poses a considerable challenge owing to the strict requirements compliance by the applicable standards and documents, such as those outlined in the *DO-178C*, to achieve successful certification. Ribeiro et al. underscored that while the *DO-178C* standard is not meant to be overly rigid or prescriptive, it mandates specific outputs to achieve successful certification [2]. This complexity is further elaborated in Bertrand et al.'s research, which explores the difficulties of certifying aerospace software, requiring meticulous documentation and planning upfront, where any late requirement changes can disrupt established certification processes, necessitating additional documentation, validation, and potential re-certification efforts [61]. Kennedy emphasized the challenges arising from the increasing complexity and costs associated with software development in the aerospace industry. Any changes made late in the development process may require extensive rework, testing, and verification to ensure compliance with safety standards, leading to project delays and budget overruns [14]. As Silva Cardoso et al. noted, late requirement changes necessitate updates to documentation, including specifications, design documents, and test cases, to accurately reflect new requirements. This additional documentation overhead can strain the development resources and introduce delays in the software delivery process [50].

Furthermore, the *DO-178C* standard requires traceability between the requirements, design, implementation, and verification artifacts. Late requirement changes present challenges in maintaining traceability and ensuring compliance with certification standards, as discussed in Baron et al. 's research [57]. Any modifications to the requirements must be meticulously tracked and validated to demonstrate compliance, adding complexity to the development process [57].

[8]DOORS is an acronym for a Dynamic Object-Oriented Requirements System, a Requirements Engineering Management tool from IBM that allows the capture, tracing, analysis, and management of requirements changes [53].

3) CHALLENGES FOR ALIGNMENT OF AGILE PRACTICES WITH DOCUMENTATION REQUIREMENTS AND ITS POTENTIAL FOR REUSE IN AEROSPACE

In aerospace software development, aligning Agile methods and practices with **documentation** requirements creates considerable challenges owing to domain standards and documentation requisites, such as *DO-178C* [2], [6]. As outlined in the Agile Manifesto [16] and emphasized in a study by Silva Cardoso Rodrigues et al. [50], the Agile Manifesto prioritizes functional software over extensive documentation. However, in aerospace software development, adherence to regulatory standards and documents such as *DO-178C* necessitates detailed outputs, including documentation to ensure safety and compliance [2], [6]. Balancing documentation requirements with Agile principles of simplicity and flexibility can be challenging, leading to tension between development teams, regulatory compliance requirements, and authorities. Traditionally, the Waterfall model has been employed for years in SCSE organizations to fulfil these standards and documents.

Consequently, documentation often requires finalization and approval before development can proceed, as discussed in [2], [7], and [57]. This misalignment can restrict the Agile practice of delivering value early and consistently because development teams may need to await documentation approval before implementing changes or new features, resulting in project delivery delays. Additionally, the *DO-178C* standard requires traceability between the requirements, design, implementation, and verification outputs [2], [6]. Ensuring traceability while adhering to Agile practices of iterative development and frequent changes can be challenging, as noted by Kennedy and Towhidnejad [14]. Changes made during Agile iterations may necessitate updates to multiple documentation outputs, requiring careful coordination and validation to maintain traceability and compliance with regulatory standards and documents. However, in aerospace software development, documentation requirements can decelerate development velocity, redirect significant time and resources from actual software development and restrict the Agile principle of responding to change over strictly adhering to a plan [61].

Regarding **documentation and reuse**, the existing literature highlights the importance of proper documentation to ensure traceability, compliance, and maintenance of software systems. However, further research is required to address the challenges and strategies for effectively reusing documentation artifacts across different projects. Exploring successful reuse practices, identifying barriers to reuse, and proposing strategies for overcoming them are valuable areas for future research.

4) FAVORITE AGILE METHODS AND PRACTICES IN AEROSPACE

In addition to managing requirements, some authors have explored which Agile practices can be integrated into current processes in aerospace software development. These include TDD, Refactoring, CI, and Pair Programming, as referenced in several studies [13], [50], [51], [55], [57], [62], [63]. In [13], the authors stated that transitioning from the traditional Waterfall model to Agile is difficult but possible because this transition does not require a sudden sweeping change. Instead, it can be achieved by **incorporating Agile practices and methods** into existing processes. However, the aerospace sector has been **slow to adopt** practices and methods (e.g., Scrum) already proven outside the safety-critical industry. Other authors have also developed a reference model for safety-critical software requirements, known as the CARD-RM. This model, as discussed in [48] and [49], modifies Scrum to fit the specific phases of aerospace software development. Although still in the research phase and tested primarily in academic settings, CARD-RM aims to **blend Agile practices** with *DO-178C* compliance. Additional studies, such as [56] and [59] investigated the application of Agile methods in complex, interdisciplinary environments. Reference [56], for example, showcases the use of a modified Scrum framework, Q-Scrum, in IPBL. As stated by [13], a common agreement among researchers is the need to **tailor existing processes** to transition to Agile methods smoothly. This includes addressing the challenges associated with stringent safety processes, often leading to significant costs and scheduling issues. One proposed solution is to **increase process automation**, as explored in research such as [62] and [63], which looks at the application of **DevOps** principles in aerospace software development. Reference [62] explicitly addressed the challenges faced by startups in the aerospace sector, suggesting a tailored workflow based on *DO-178C*/*DO-331* standards. The study by [63] explores how DevOps principles can address safety-related challenges, enhance automation and continuously improve software development. Prominent research, including [47], [50], [55], [57], and [63], focuses on challenging traditional V-Model software development practices. They advocate for a "**continuous certification**" process grounded in Agile methodologies. They proposed **tailored Scrum frameworks** for **continuous certification** in safety-critical software development, particularly in the aerospace domain, while complying with the *DO-178C* standards. This approach emphasizes automation to ensure compliance with rigorous certification requirements, including traceability and documentation.

5) OTHER CHALLENGES FROM DO-178C

Based on the previous *DO-178C* analysis and reviewed publications, we highlighted various unanswered aspects of the processes outlined in *DO-178C*. These include the **outputs of the processes and plans**, **independence**, and **change control**.

While the literature provides comprehensive insights into the overall framework and guidelines for generating **process outputs**, a gap exists in understanding their practical implementation and the potential challenges faced during their development. Further research is required to explore the

best practices and real-world case studies that demonstrate the effective utilization of process outputs in different software development scenarios.

In terms of **plan outputs**, the literature review emphasizes the significance of well-defined plans to ensure successful software development under the *DO-178C* guidelines. However, limited information is available regarding the evaluation and assessment of plan outputs. Future studies could examine the criteria for evaluating plan outputs, their effectiveness in guiding development activities, and potential improvements that could enhance their practical application.

The issue of **independence** has also emerged as an unanswered aspect of the literature. While *DO-178C* emphasizes the need for independence in various activities, such as V&V, there is limited research on defining and quantifying independence measures. Future studies could investigate methodologies for assessing independence levels, establish metrics for evaluating independence, and examine the impact of independence on the overall software development process.

Finally, the topic of **change control** warrants further investigation. Although the *DO-178C* standard outlines the importance of change control in managing software requirements and design modifications, there is a lack of specific guidance for establishing robust change control mechanisms. Future research could focus on developing formal change control procedures, identifying effective change impact analysis techniques, and exploring ways to minimize the potential risks associated with changes.

In conclusion, while the literature review provides valuable insights into various aspects of *DO-178C* processes, many areas still require further exploration. These unanswered aspects provide opportunities for future research to enhance our understanding and implementation of these critical elements in the development of safety-critical software systems.

V. POST-MORTEM ANALYSIS OF AEROSPACE INDUSTRY PROJECTS

This section presents the collection and analysis of additional data gathered from three concrete aerospace industry projects selected and provided by CSW complementary to the SLM presented in Section IV.

This data allowed us to combine academic analysis with real-world industry insights, pinpointing and addressing crucial concerns, challenges, and opportunities associated with adopting Agile methodologies within the aerospace domain. Furthermore, this type of industry project data contains rich information for research purposes; however, access to detailed information from these projects is frequently not possible given their confidential nature.

As these projects were already closed, we proceeded with a post-mortem analysis. They all followed the Waterfall model. We had access to all the project activities, data, documents, and outputs required by the *DO-178C* standard to achieve successful certification. The post-mortem analysis helps identify which dependencies and risks the project teams faced and what lessons are learned. After all, the way to avoid past mistakes and keep improving is by understanding what went wrong and how issues could have been avoided [54].

Due to non-disclosure agreements (NDA), we need to preserve the identity and anonymity of the clients. For this reason, we name projects as A, D and C.

In the analysis, we investigated:

- Which were the project activities;
- Outputs and documents required by *DO-178C* standard;
- What issues did the team experience related to requirements, traceability, and other deliverables?

To perform the post-mortem analysis we followed the guidelines in [54], which consider the following four phases.

- **Data collection** – can be done in two ways: directly from team members through questionnaires or interviews or by gathering project documentation.
- **Workshop meetings** – they facilitate different types of structured discussions (e.g., fishbone diagrams) about the project or lead to a formal analysis of what happened.
- **Data Analysis** – can be performed during a workshop or separately. Moreover, it may include statistical methods or other types of analysis suitable for the data.
- **Reporting and publishing the results** – it culminates the process.

Because some of the teams were no longer entirely available, we deviated from the structure in [54] by not performing workshop meetings, although interviews were done, namely with the team leaders. Still, our post-mortem analysis focused on project documentation.

A. PROJECTS INSIGHTS AND CHARACTERIZATION

The CSW developed the three projects used in our post-mortem analysis under commercial contracts with separate suppliers and customer entities. As mentioned previously, all three projects followed the Waterfall method. Detailed insights from each project are provided as follows:

In **Project A**, as the first link in the chain of integrated flight control systems, sensors collect the aircraft's positioning data along three axes: roll, pitch, and heading. Combining millions of flight hours and the highest performance Directional Gyro Mode,[9] this reliable and precise input data for the automatic pilot computer and displays enables safe flight control movement with the best Size, Weight, Power & Cost (SWaP-C) optimization. Available in several versions (with a flux valve for the aeroplane version and a magnetometer for the helicopter version). The CSW, via Project A, has provided support to one of its clients in developing and validating this system and providing assistance with certification. Throughout Project A, the client took charge of all interactions with the certification authority, encompassing

[9] A Directional gyro, also referred to as "FREE" mode is used mode when magnetic heading references are not reliable (e.g. in polar regions). In this mode, the system supplies an inertial heading reference, with corrections introduced manually to offset earth rate and other errors [65].

activities across both Software Planning and Certification Liaison processes [2], [6]. As outlined in *DO-178C*, the remaining processes were collaboratively undertaken, with shared responsibilities between the client and the project team. For instance, the High-Level Requirements (HLR) definition fell within the client's scope in the Software Development process. Simultaneously, the project team handled subsequent activities, from Low-Level Requirements (LLR) definition to object code generation and updating the Software Configuration Management Plan (SCMP) [2], [6].

Project D was focused on the full lifecycle development of the Propeller Brake Control Unit (PBCU), which is a component of an Engine Propeller Brake Kit (PBK). The primary purpose of the PBK is to halt the rotation of the engine propeller when the aircraft is on the ground, thereby facilitating safe movement around the aircraft. The PBCU controls the motor responsible for physically stopping the propeller blades. Its two main functions are stopping propeller rotation using electrical power after landing and engine shutdown, such as during taxiing or towing, and preventing propeller rotation when the aircraft is parked and powered down, thereby preventing movement under windy conditions. The Project D team's focus was on HLRs and LLRs development besides V&V activities for the remaining process outputs, as described in the *DO-178C* [2], [6]. All activities from the other processes specified by the *DO-178C* standard were executed by the customer, including all interactions with the certification authority (Software Planning and Certification Liaison processes) [2], [6].

Finally, a client of the CSW develops actuators[10] which are well-known for their reliability in the drone sector. **Project C** was executed to achieve Software Level C for the software components from two of these actuators in compliance with the *DO-178C* standard by reviewing and correcting deficiencies across nearly all lifecycle phases, encompassing requirement specifications, design, source code, and the provision of testing and validation evidence. The assessment encompassed two actuators whose control software consisted of approximately 2000 lines of code, incorporating both the bootloader and the application code. It has been previously established that these actuators would not undergo certification by an aviation regulatory authority, such as the European Union Aviation Safety Agency (EASA), but would instead be subject to private evaluation, likely based on military standards. Regarding Project C, the team was responsible for almost all activities from the processes specified by the *DO-178C* standard, in addition to Software Planning and Certification Liasion [2], [6]. Furthermore, the team collaborated with the customer in the plan's creation during the Certification Liaison process for the Plan for Software Aspects of Certification (PSAC), Software Accomplishment Summary (SAS) and Plan for Hardware Aspects of Certification (PHAC) creation [2], [6].

For a more detailed understanding of the analyzed projects, some of the relevant raw numbers concerning the projects are shown in Table 5: the project software level (criticality), a generic evaluation of the project complexity (High or Low), and the total effort reported on all activities in each project. Additional data were introduced during our analysis. Furthermore, the projects can be considered similar in work (executed lifecycle phases) because they all cover software requirements specifications up to V&V of the integrated software in the aerospace solution.

TABLE 5. Projects Main Characterization Numbers.

Parameter	Project A	Project D	Project C
Software Level	A	D	C
Complexity	High	Low	Low
Real Effort (Hours)	13 020h	2 223h	3 927h

Several approaches were used to analyze the existing documentation for each project. However, the number of pages was so large that we had to perform document classification, as explained below. We identified critical activities and outputs that are extremely important for the certification described by the *DO-178C* standard and are pertinent to our research. We also identified the project milestones and organized them chronologically to understand the dependencies and risks associated with each artifact.

The documentation was classified into the following categories:

- Project Management Documents (e.g., minutes of meetings, memos);
- Project and Technical Documents (e.g., requirements documentation, testing documentation);
- Other documents (e.g., plans).

Documentation was classified according to the typical activities of a software engineering project: Requirements, Design, Coding, Testing and Project Management. Classification was performed to make the amount of documentation more manageable. The documents left unclassified were organized into two additional classes: Pre-project and Post-project.

The issues that the authors considered significant to impact the activities or outputs required to achieve successful V&V were considered interesting and analyzed.

B. PROJECTS ANALYSIS

CSW's involvement in a whole certifiable project is limited, as it is commissioned to deliver a particular set of activities, generally software development related, of the overall system project.

For example, in the project for the development of a new aeroplane by a manufacturer, to cope with such high complexity, a part of the development and V&V is performed by different third parties selected by the manufacturer. Each third party will only be working on a particular component

[10] A drone actuator is a key component within a drone that converts control signals from the drone's flight controller into physical movement. Essential for the drone's flight and manoeuvrability, actuators are integral to various subsystems of uncrewed aerial vehicles (UAVs) and other types of robotic vehicles [66].

or subset of the project while still being required to comply with the objectives and activities of the *DO-178C* standard. Although different third parties deliver different subsets of the overall project, they must all ensure that their deliveries will produce the required global outputs to be submitted and reviewed by the certification authority to achieve successful certification of the complete aeroplane. Thus, these projects are highly representative for the purpose of our research.

After analyzing all documents from each project and mapping them to the related activities and processes, as required by the *DO-178C* standard, we understood which activities were included in each project. Additionally, we identified specific efforts related to development and V&V activities. With the available data from CSW and support from project team leaders, we were able to conduct the analysis and obtain the results presented in Table 6.

TABLE 6. Development and V&V Effort per Project.

Variable	Project A	Project D	Project C
Development (DEV) Effort	1 926h	804h	1 804h
V&V Effort	11 094h	1 419h	2 123h
V&V Ratio (%) = (V&V Effort/Dev Effort-1)*100%	+476%	+76%	+18%

Table 6 shows actual effort data for each analysed project divided between software development and V&V activities. Development effort includes all technical tasks, from software requirements specification up to source code implementation. The V&V effort line presents the specific effort applied to the total development effort undertaken by CSW. Additionally, none of the Critical Software projects analysed (A, D, and C) incorporated requirements changes during the project execution, indicating the inability to introduce any requirements change successfully. Since the V&V effort is always superior to the DEV effort, we have more opportunities to save effort by researching how these activities can be optimised while having the possibility to incorporate requirements changes, always maintaining the mandatory compliance to the *DO-178C* standard.

C. RESULTS

All documentation was stored in a Bitbucket git repository. The variety and the massive number of interrelated documents, made it quite challenging to understand what they represented in the context of the project and how they were connected. This difficulty applies to documents such as plans, requirements and validation checklists, among other documents. Another challenge was the need for a tool to automatically link and manage the documentation. It created the need to keep maintaining several documents open and shift between them to follow a topic (e.g., requirements-related documents). However, the material's completeness from these projects allowed us to analyse them in detail to get a deeper understanding of what is needed to achieve a certifiable deliverable output and the challenges to achieve successful certification.

We confirmed that all *DO-178C* standard specified processes must be executed by the main contractor or by the third party, CSW; however, not all activities need to be carried out, as these would depend on the project DAL. We verified that in these three projects, some of the process activities were split between CSW and the main contractor, creating challenges during the project execution. This split was the first significant observation of our review.

The major challenges that we identified based on the detailed post-mortem analysis, where Agile methods can help, relate to requirements management, team synchronization, and automation.

1) CHALLENGES FOR REQUIREMENTS MANAGEMENT

Even with the requirements specified at the project's initial phase, the initial lack of complete knowledge of the system under development caused the need to mature more the initial requirements during the project execution. As more knowledge is acquired, the requirements are clarified, and we keep increasing the problem definition by adding new detailed requirements to the initial ones. However, despite this acknowledgement, given that the projects employed the traditional Waterfall model and the supporting evidence presented in Table 6 from the post-mortem analysis, none of the projects accomodated any requirements change, just refinements. The Agile methods can help by providing accelerated knowledge acquisition due to shorter customer deliveries and feedback loops, thus creating more flexibility to achieve the goal. This is possible by handling late requirements changes while maintaining traceability with fewer specification issues in case of missing details. In other words, it helps to be more customer-centric.

2) CHALLENGES FOR TEAM SYNCHRONIZATION

With most of the documentation defined in the initial phases of the project, during project execution the documentation and output interconnection (traceability and baselines) become complex and require significant effort to keep all documentation aligned. Agile methods provide practices that help to treat change management in documentation iteratively and incrementally, supporting documentation maintenance, such as iterative/incremental documentation production before each development iteration and refinement/review sessions, but in the Waterfall model the interactions between team members are much less structured, leading easily to undetected differences of understanding on how particular details should be handled.

During the project duration, owing to the vast number of activities that needed to be completed, a misalignment often existed between team members' interpretation of the required activities, creating challenges. For example, in one of the projects an issue related to independence forced the team to conduct a second review to reinforce it. The goal of a self-organizing team can be achieved because Agile methods provide events and practices to ensure daily team alignment and continuous improvement.

3) CHALLENGES FOR AUTOMATION

Most of the work delivered during the projects was carried out manually, potentially introducing human errors while running this complex environment. Examples include documents, different levels of requirements, source code and software release version management with inter-connectivity issues (traceability and baselines generation), reviews and validation activities, ownership misalignment, and independence validation issues. Manual effort to manage all required V&V activities is very high. The vision to achieve continuous delivery is almost impossible if validation activities require human intervention. This means that human involvement remains a crucial factor despite the potential for reducing the effort required for validation. Nonetheless, fully automated continuous delivery can be achieved for smaller parts of development. However, to effectively support and maintain these activities we need the right level of automation and qualified tools in place.

Currently, projects have some level of automation among IDEs, Bitbucket, and CI tools. However, there needs to be more automation to support the interconnection of different artifacts, providing the possibility for documentation reuse and continuous validation. For example, with a work management tool in place, we could automate a report that validates the activities' independence constantly throughout the software development process.

By implementing these improvement opportunities, we firmly believe that it is possible to significantly reduce potential human errors and efforts in the V&V activities while successfully introducing requirement changes.

VI. OPPORTUNITIES FOR IMPROVEMENT

The *DO-178C* analysis and state-of-the-art analysis described in Sections IX and IV helped identify one set of major concerns and challenges that safety-critical organizations face in Agile adoption, more precisely in the context of software systems development for aerospace regulated by the *DO-178C* standard. The post-mortem analysis of the real aerospace industry projects (Section V) helped confirm some of these findings and identified additional concrete concerns and challenges, mainly gathered from the documentation of those projects. Using the outcomes of Sections IV and V, we built a causal loop diagram (Figure 7) to show the possible causes and effects of behavior connections of the relevant system variables, as identified in this research. The variables of the system are shown as nodes (e.g., requirements maturity), and the connections between them are indicated by arrows. Arrows represent linkages or causal ties between the variables.

A blue line represents a positive causal link, which also depicts a positive correlation between two variables. This implies that if one variable changes, the other must follow suit, or that if one variable decreases, the other must also decrease. In this positive reinforcement loop, a change in one variable magnifies the change in the other, creating a self-reinforcing pattern.

Conversely, a red line shows a negative causal relationship or negative connection between two variables. This suggests that if one variable changes, the other must also change and vice versa. The pattern of stabilization or balancing represented by this negative feedback loop occurs when a change in one variable opposes or counteracts a change in the other variable.

The system variables depicted may be responsible for a significant portion of the identified challenges. Other variables exist and might affect safety-critical projects, such as the experience of the engineering teams, customer pressure, technologies involved, complexity of development, and V&V environments. However, these other variables are not listed and discussed in this work because we focused on those that are most likely mitigated by Agile methods and practices.

The causal loop diagram shown in Figure 7 starts with the requirements specification of the customer during the initial phase of the project. A typical initial lack of knowledge might cause requirements to be insufficiently mature. This lack of initial knowledge leads to the generation of new requirements during project execution and the change/adjustment of the initial requirements. A side effect is that new requirements might raise traceability and baseline complexity, and thus significantly increase the effort needed for requirements and output management. These effects might also lead to an increase in V&V efforts, which in turn will be solved by the engineering teams by having fewer deliveries with a larger scope. Likewise, feedback loops from customers related to deliveries are generally performed at predetermined moments (e.g., milestones meetings/review sessions) and might not be frequent enough. The long feedback loops seem to be a limitation of the knowledge acquisition process and on achieving high-maturity requirements, because requirements will be reworked and enriched along the lifecycle at a less frequent pace and only stabilized in the final delivery stages.

Given the above-mentioned concerns and challenges, the following improvement opportunities have been identified and deserve further exploration. Such opportunities are based on Agile methods, which are more flexible and customer-centric, and will accelerate knowledge maturity due to more frequent customer deliveries and shorter feedback loops.

A. AGILE METHODS & MINDSET

While the aerospace sector has looked into some Agile techniques and practices, as described in existing reports and publications, the adoption is still very low, as can be seen from the considerably low number of publications available on this topic. There is a lack of detail on how these proposals have been validated, as most analyses are just speculative. To improve the current processes and demonstrate its success, we need to identify and validate the right level of tailoring, techniques, and practices that can be implemented [13], [55]. These elements would also be used as inputs to initiate an incremental cultural change, including the certification authorities.

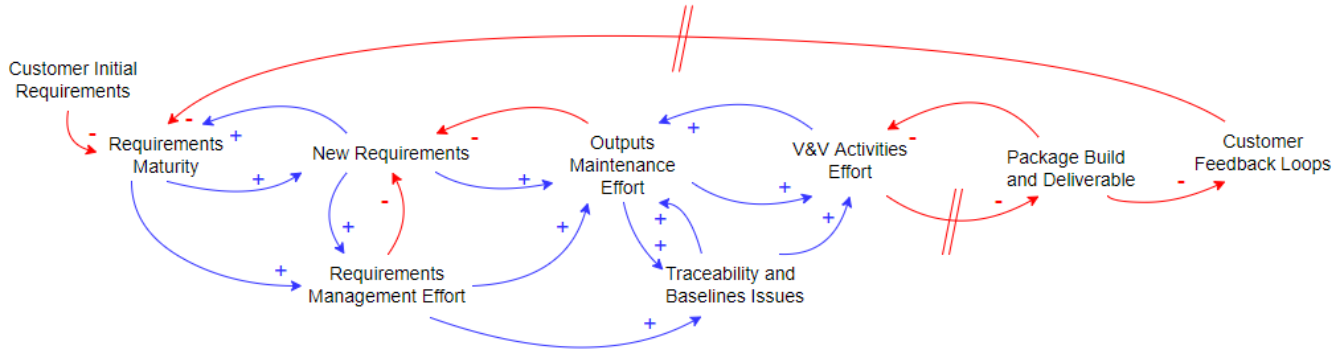

FIGURE 7. Current System Causal Loop Diagram.

B. LATE REQUIREMENTS MANAGEMENT

There have been some proposals to improve the management of requirements while ensuring traceability [52]. When used as a requirements list, some authors are currently experimenting with new methods to manage requirements [55]. An example is the creation of user stories that define a team's small requirements and are easier to implement using iterations. However, late requirement changes is still a highlighted issue depending on our development phase [52]. If we are in the validation stage, it is really difficult to incorporate a change because of the need to restart the entire validation process. Although some authors have proposed potential solutions, these are still theoretical and have not been fully validated. However, recognizing this problem has led researchers to focus on managing late requirement changes efficiently.

C. DOCUMENTATION AUTOMATION

When developing safety-critical systems, standards and documents require many outputs, from plans to documentation, to achieve successful certification. These outputs must follow strict processes that ensure traceability, baseline, and reviews until approval [6], [8]. To improve how outputs are managed we must identify approaches to automate parts of documentation production and reuse them. Currently, we have found only a few publications that provide examples of how some practices have met the requirements for document management. The analyzed publications do not provide a complete answer but rather discuss how to use flexible processes and how rigorous documentation management can be successfully tackled in an incremental and iterative way. Improving the automation level can simplify the current process by creating a requirement until it is considered to be certifiable. By improving automation, iterative deliveries and effort reductions can be achieved.

D. CERTIFICATION PROCESS

The certification process has rarely been discussed in research publications. They discuss certification complexity, but there needs to be more information on what is needed to clarify and simplify the certification process [47], [55], [57], [63]. The *DO-178C* standard states that besides the evidence that must be produced to achieve successful certification, to enable the certification authority's involvement in the certification process, additional regulations and guidance from those authorities are also applicable [6]. In conjunction with other opportunities for improvement, the certification process has room for improvement and optimization.

VII. ANALYSIS AND DISCUSSION

In addition to the analysis and synthesis of the *DO-178C* standard (Section IX) and the state of Agile and aerospace-related publications over the years, the contributions presented here from our research were to identify major concerns and challenges from the scientific community and industry regarding the use of Agile methods in aerospace (Sections IV and V), and to present a summary of opportunities for improvement (Section VI). The following sections aim to answer the research questions presented in Section I, which are crucial for identifying future possibilities to evolve Waterfall-based models towards a more efficient and flexible process with shorter feedback loops and more customer-oriented. Therefore, we revisit them, summarize our findings, and provide clear answers to each research question.

RQ1 – What is the distribution of scientific publications within the aerospace domain over the years, the types of venues, and the venues?

The first publication, published by [52], analysed speculatively the potential of agility in the avionics software world. This is the only study conducted in 2006. As detailed in Section IV (Figures 5 and 6 and Table 4), the publications distribution between 2001 and 2023 (Figure 5) shows that a moderate increase in publications started only in 2013.

According to Figure 6, more publications are being presented at conferences, indicating that Agile in safety-critical software development for aerospace is a growing topic but still not gathering much attention.

When examining the venues listed in Table 4, it is evident that IEEE conferences have the highest number of accepted publications, closely followed by the AIAA/IEEE Digital Avionics Systems Conference.

In summary, the adoption of Agile methods and practices in the aerospace domain in 2023 remains at an early stage

of maturity with high potential for research by the scientific community (our initial review of 389 studies found that only 4.6% of the publications related to Agile methods and practices in aerospace software development).

RQ2 – What are the major concerns and challenges related to the adoption of Agile methods and practices for safety-critical software development, particularly in the aerospace industry?

As observed in Section IV and supported by publications related to the applicability of Agile values, principles, and a few practices such as TDD, refactoring, CI and Pair Programming, requirements management is the most covered challenge for Agile adoption in aerospace. However, some aspects of *DO-178C* remain unanswered, including process and plan outputs, documentation reuse, requirement change management, traceability, independence, and change control. Addressing these gaps is crucial for enhancing the understanding and implementation of Agile methods and practices in safety-critical software development, particularly within the aerospace domain, while adhering to the *DO-178C* standard. Based on the analyzed industry data from the three projects provided by CSW (Section V), the major concerns and challenges identified were the management of requirements, maintainability of documentation and team synchronization, with the first two recognized as the most critical for the project's success.

As a result of combining Sections IX, IV, and V, Figure 7 (Section VI) depicts a possible causal loop diagram. This figure shows our understanding of the causes and effects associated with the behavior connections from the main system variables, highlighting the major concerns and challenges.

This result reinforces the conclusion that any sudden transition to follow an already predefined Agile method can create issues with documentation and requirements specifications. Even with a complete requirements specification at the initial phase of the project, there is a risk of a lack of requirement maturity caused by the limited knowledge of the system under development. Furthermore, the need for more automation to manage documents, different requirement levels, source code, tests, and software release versions can create inter-connectivity issues (traceability and baseline generation) because of potential human errors. These issues lead to a significant increase in the effort needed for requirements and output management, also causing a V&V effort increase, which engineering teams tend to solve by having fewer deliveries with a larger scope, thus being less frequent than desirable.

RQ3 – What are the main opportunities related to Agile methods and practices for the improvement of safety-critical software development, particularly in the aerospace industry?

As detailed in Section VI, different strategies to improve safety-critical software development in aerospace are proposed:

1) **Definition of a tailored Agile method that supports fine-grained requirements**. There is a significant opportunity to evolve the current software development process by defining a tailored Agile method that would allow for the breakdown and management of requirements into smaller, incremental, or iterative certifiable deliveries. The goal is to minimize the impact on the V&V effort while still meeting the comprehensive documentation and traceability requirements of safety standards such as DO-178C.
2) **Definition of a tailored Agile method for effectively managing and accommodating dynamic late requirements**. Another key opportunity lies in effectively evolving the tailored Agile method to manage and accommodate late changes in requirements effectively. This should be achieved while minimizing the impact on the V&V effort, thereby maintaining project integrity and compliance, even when late changes occur.
3) **Increase the level of tooling automation to support the defined tailored Agile method regarding documentation and outputs**. Developing and increasing the level of tooling automation to support the tailored Agile method represents a major opportunity. The aim is to produce and manage documentation and outputs required by standards such as *DO-178C* more efficiently, potentially allowing the reuse of these outputs. This automation could significantly reduce the engineering workload, particularly in V&V efforts, and mitigate risks associated with manual errors in documentation and compliance activities.
4) **Evolve the tailored Agile method to ensure the appropriate continuous involvement and engagement of the certification authorities**. There is an opportunity to develop a method that ensures the appropriate and continuous engagement of certification authorities throughout the software development lifecycle. This includes the V&V phases and certification phases, ensuring that the tailored Agile methods align with regulatory expectations and standards.

In summary, by adopting these strategies, the current safety-critical software development for aerospace can evolve towards a more customer-centric process with shorter feedback loops, that is, more Agile.

VIII. THREATS TO VALIDITY

Some limitations and threats to validity were identified during the research conducted for this study. These encompass both internal and external validity threats, which have the potential to impact the reliability and generalizability of research findings. To mitigate these concerns, the authors undertook the following measures to address these limitations.

A. INTERNAL THREATS TO VALIDITY

First, as the review of publications was conducted by a single author (first), there was a possibility of bias in the

selection process, which could have influenced the outcome of the mapping study. We developed and utilized an RPB to mitigate this internal threat by incorporating defined inclusion and exclusion criteria to achieve the desired results. We also made the replication package of our SLM available here [17] to ensure that our SLM is both reproducible and transparent. Additionally, to ensure a more balanced and objective evaluation, we included the third author in the process of assessing and validating publications.

Second, the potential for missing publications poses an internal threat to the validity of the study. To minimize this concern, we employed snowball sampling. This approach aims to enhance the completeness and reliability of results by identifying additional relevant publications through references and citations. The third author also independently assessed the results of the snowball sampling, further bolstering the credibility of the findings.

B. EXTERNAL THREATS TO VALIDITY

Owing to the heavy reliance on semantic analysis and expert assessment, achieving an exact replication of the results is challenging. A detailed and systematic mapping process, including an RPB, was implemented to address this challenge and mitigate potential threats to the validity of search results [17]. The authors aimed to enhance the reliability and reproducibility of the outcomes by developing and making the replication package available. Seven of the most relevant databases of scientific publications were carefully selected to ensure broad coverage of published results as specified in the replication package dataset [17]. These databases include the large majority of existing publications, effectively minimizing external threats to validity through their comprehensive content.

Another aspect affecting validity was the limited access to detailed information during the post-mortem analysis. To overcome this limitation, the authors obtained additional data on the three aerospace industry projects provided by CSW, besides the contained in the repositories, for instance by interview some of the projects participants. These projects serve as valuable complements to the analysis, enabling a deeper understanding of past mistakes and identifying opportunities for improvement.

Strict adherence to confidentiality constraints is crucial for maintaining privacy and protecting data. The authors primarily relied on project documentation to investigate the activities, outputs, and documents required for *DO-178C* certification. To safeguard the privacy of all parties involved, the project names were anonymized as A, D and C instead of being disclosed, as well as the data used and the conclusions were stated in an abstract way.

Regarding the last two external threats to validity, which pertain to limited access to detailed information during post-mortem analysis and the necessity of strict adherence to confidentiality constraints for safeguarding privacy and data protection, both authors (first and third) followed a well-defined process of reporting and publication. They conducted a comprehensive analysis using appropriate methods and techniques suitable for the available data. A comprehensive report meticulously documented the findings, lessons learned, and recommendations of this study. When presenting the results, utmost care was taken to preserve project confidentiality.

The authors acknowledge and addressed the above mentioned limitations and threats to validity to the best extent possible. However, it is essential to note that some uncertainties and potential biases may persist despite these measures. Researchers and readers should consider these limitations when interpreting this study's findings.

IX. CONCLUSION

Our analysis of the *DO-178C* standard in Section IX shows that it does not prescribe any particular software development method. Instead, it simply describes, at the software level, the activities and outputs that must be performed for successful certification, which makes it compatible with Agile methods. This conclusion confirms that the *DO-178C* standard allows the adoption of Agile methods and practices.

Because of the confidentiality of the majority of projects in the safety-critical domain, the scientific publications (Section IV) and the industry projects analyzed (Section V) only represent a limited sample of what has ever been done in the domain of Agile methods and practices applied to safety-critical systems. Still, the findings from the surveys of papers and projects that we present are consistent in the concerns and challenges that they express; we feel confident to use them as input for our research, to identify concrete opportunities for improvement in the adoption of Agile methods in safety-critical environments, specifically in the aerospace domain.

As outlined in Section VI, the identified concerns and challenges lead to the definition of four opportunities for improvement, resulting in future research areas.

First, the Agile Methods & Mindset (Subsection VI-A). There is no clear path to apply Agile practices in the safety-critical domain; they have to be significantly tailored. For instance, the current application of Agile methods to the task of breaking down requirements to achieve smaller, incremental, or iterative certifiable deliveries, has been shown to be complex for project teams. A recurring challenge is aligning that task with the comprehensive documentation requirements of safety standards such as *DO-178C*, which requires complete traceability.

Second, late requirement management (Subsection VI-B) currently presents challenges in maintaining independence and traceability, thereby increasing the overall engineering workload, especially in V&V efforts. A frequent challenge is aligning these late changes with the extensive documentation requirements set by safety standards, such as *DO-178C*. Additionally, ensuring independence in V&V activities and achieving the necessary certifications remain significant issues. Maintaining full traceability and effectively conducting V&V throughout the development lifecycle are critical concerns. For these two opportunities for improvement

(Subsections VI-A and VI-B), our ongoing research aims to leverage existing Agile methods and practices to define novel tailored methods that provide approaches to better requirements management, ensuring that support for late requirement change management is consistent with the certification process. Such tailored Agile methods should be iterative and incremental while minimizing the impact on the V&V effort and maintaining full compliance with the *DO-178C* and accompanying standards and documents. This future research area responds directly to both the first two opportunities for improvement, offering a pragmatic solution to the intricate balance between Agile flexibility and regulatory compliance and proposing structured yet flexible methods to maintain project integrity and compliance when a late change request is introduced during project execution.

Third, the current level of automation in generating required outputs (Subsection VI-C), especially in managing requirements with adequate traceability and documentation production and management, is significantly low. This shortfall notably increases the engineering workload, particularly in the V&V effort. This challenge is supported by both the scientific literature (Section IV) and the industry projects analyzed (Section V). Currently, these tasks are performed manually, significantly increasing the risk of compliance issues owing to potential errors from the extensive manual work involved. Implementing effective automation is crucial for improving the documentation, traceability, and verification processes. The recognized need for an increased level of automation in generating and reusing required outputs, particularly concerning documentation, has led our research to develop not only a novel tailored agile method but also an appropriate level of automated tools to support this tailored agile method. These tools aim to reduce the engineering workload and enhance the efficiency of the V&V efforts. By implementing automated requirements management, traceability, and documentation management, we aim to tackle the third challenge: streamlining the certification process while maintaining the rigor and thoroughness required by safety-critical standards, specifically *DO-178C*, for the aerospace domain. Traceability is critical in standards, documents, and regulations, such as the *DO-178C* standard. With this in mind, we must establish the right level of automation to support the proposed agile method, producing the required evidence that the artifacts remain consistent with the requirements, design, code, and tests to ensure reliable safety evaluations. By developing a tailored method that can automatically regenerate the outputs required by the *DO-178C* standard, allowing project teams to reuse the same outputs as requirement changes, we believe that we can reduce V&V effort. As a result, organizations move towards a more efficient and flexible model with shorter feedback loops.

Fourth, as discussed in Subsection VI-D, the certification process has rarely been addressed in research publications. As outlined in Section , certifying a complete system, such as an aeroplane, involves a primary system development process governed by various standards and documents. Each development document offers crucial guidelines for ensuring the safety and reliability of aerospace systems, with *DO-178C* for software and *DO-254* for hardware being the most prevalent. This underscores a potential area for future research focused on complete system development certification using Agile methods that can be used as input for each particular standard and document, such as our research, which is focused on software development for aerospace, specifically concerning the DO-178C standard. The *DO-178C* standard specifies that the involvement of the certification authority is essential in producing the necessary evidence for successful certification. This requires additional regulations and guidance from authorities, as noted in [6] and [2]. However, since our focus is software development and because the *DO-178C* requires at least four moments of certification authority involvement by focusing on the first three opportunities for improvement, we believe we can showcase the successful adoption of the tailored Agile method while minimizing the impact on the V&V effort and maintaining full compliance with the *DO-178C*.

The first opportunity for improvement identified, Agile Methods & Mindset (Subsection VI-A), refers to a shift in organizational culture. However, this requirement is not isolated for this opportunity; it applies equally to all four identified areas (Subsections VI-A, VI-B, VI-C and VI-D). Adopting Agile methods and practices and transforming an organization's culture and certification authorities are considerable challenges. This is especially true in safety-critical domains, which are naturally resistant to change. Evidence from publications and project post-mortem analyses within the aerospace sector highlights this challenge. A complementary work of our research is to address the cultural obstacles that slow the adoption of Agile methods and practices in such conservative environments. The success of this research focuses on creating a tailored Agile method, allowing late requirements management, implementing the right level of automation to support this method, and ensuring the commitment of teams in the case studies to embrace the new approach and collect applicable data. Our work involves developing strategies to foster a cultural shift towards Agile methods, demonstrated through case studies in real aerospace industry projects. This improvement effort targets not only the first opportunity but all four, aiming to validate and showcase the advantages of Agile methods in settings that are typically subject to strict compliance requirements and regulations.

In conclusion, four research areas can be defined as related to the identified opportunities for improvement (Section VI) while always ensuring a continuous inspection (V&V activities) as per the applicable standards and documents.

1) Evolving the current software development process and defining a tailored Agile method that allows requirements breakdown and management for smaller, incremental, or iterative certifiable deliveries while minimizing the impact on the V&V effort.

2) Evolve the defined tailored Agile method for effectively managing and accommodating dynamic late requirements while minimizing the impact on the V&V effort.
3) Develop and increase the level of tooling automation to support the defined tailored Agile method to produce the documentation and outputs required by the applicable standards and documents and the possibility of reusing these outputs.
4) Evolve the method to ensure the appropriate continuous evolvement and engagement of the certification authorities throughout the implementation, encompassing V&V and certification phases.

Our team is currently exploring these four research areas focused on the *DO-178C* standard within real industrial aerospace projects as case studies to support the assessment of an Agile development model for safety-critical systems and to validate an automation mechanism. Our objective is to ensure that our research results are theoretically robust and practically viable, leading to safety-critical software development for aerospace towards a more customer-centric process with shorter feedback loops, that is, more Agile.

ACKNOWLEDGMENT

The authors would like to thank Critical Software SA for supporting the access to data on concrete aerospace projects and the analysis of the outputs, Vitor Conceição for his continuous support and availability as a *DO-178C* Specialist, Nuno Silva who made contributions through comprehensive paper review, validation, and proofreading. Furthermore, they acknowledge Charles McGrath for his diligent proofreading efforts.

REFERENCES

[1] T. Myklebust and T. Stålhane, *The Agile Safety Case*. Berlin, Germany: Springer, 2018, pp. 1–235.
[2] J. E. F. Ribeiro, J. G. Silva, and A. Aguiar, "Beyond tradition: Evaluating agile feasibility in DO-178C for aerospace software development," 2023, *arXiv:2311.04344*.
[3] *Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems—Part 1: General Requirements*, Standard IEC 61508-1:2010, 2010.
[4] R. Bell, "Introduction to IEC 61508," in *Proc. Conf. Res. Pract. Inf. Technol.*, vol. 55, 2005, pp. 3–12.
[5] *Railway Applications—Communication, Signalling and Processing Systems—Software for Railway Control and Protection Systems*, document EN 50128, CENELEC, 2020.
[6] *Software Considerations in Airborne Systems and Equipment Certification*, document DO-178C, RTCA, 2011.
[7] D. J. Coe, J. S. Hogue, and J. H. Kulick, "Software safety engineering education," in *Proc. Int. Conf. Softw. Eng. Res. Pract. (SERP)*. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2011, p. 1.
[8] L. Rierson, *Developing Safety-Critical Software: A Practical Guide for Aviation Software and DO-178C Compliance*. Boca Raton, FL, USA: CRC Press, 2013.
[9] V. Hilderman and T. Baghai, *Avionics Certification: A Complete Guide to DO-178 (Software), DO-254 (Hardware)*. Avionics Communications Inc., 2007.
[10] (2006). *Version One Survey: The State of Agile Development*. [Online]. Available: http://www.versionone.com/pdf/2006-state-of-agile-survey.pdf
[11] Digital.ai Software. (2022). *16th State of Agile Report*. [Online]. Available: https://digital.ai/resource-center/analyst-reports/state-of-agile-report/
[12] A. Sidky, J. Arthur, and S. Bohner, "A disciplined approach to adopting agile practices: The agile adoption framework," *Innov. Syst. Softw. Eng.*, vol. 3, no. 3, pp. 203–216, Sep. 2007.
[13] S. H. VanderLeest and A. Buter, "Escape the waterfall: Agile for aerospace," in *Proc. IEEE/AIAA 28th Digit. Avionics Syst. Conf.*, Oct. 2009, pp. 6.D.3-1–6.D.3-16.
[14] J. D. Kennedy and M. Towhidnejad, "Innovation and certification in aviation software," in *Proc. Integr. Commun., Navigat. Surveill. Conf. (ICNS)*, Apr. 2017, pp. 3D3-1–3D3-15.
[15] K. Petersen, S. Vakkalanka, and L. Kuzniarz, "Guidelines for conducting systematic mapping studies in software engineering: An update," *Inf. Softw. Technol.*, vol. 64, pp. 1–18, Aug. 2015.
[16] K. Beck, M. Beedle, A. Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland, and D. Thomas. (2001). *Manifesto for Agile Software Development*. [Online]. Available: http://agilemanifesto.org/
[17] J. F. Ribeiro, G. Silva, and A. Aguiar, (Dec. 2023), "Replication package for a systematic literature mapping of agility in safety-critical software development within the aerospace industry," *Zenodo*, doi: 10.5281/zenodo.10354398.
[18] G. Islam and T. Storer, "A case study of agile software development for safety-critical systems projects," *Rel. Eng. Syst. Saf.*, vol. 200, Aug. 2020, Art. no. 106954.
[19] P. Kruchten, *The Rational Unified Process: An Introduction*. Reading, MA, USA: Addison-Wesley, 2004.
[20] M. Lindvall, V. Basili, B. Boehm, P. Costa, K. Dangle, F. Shull, R. Tesoriero, L. Williams, and M. Zelkowitz, "Empirical findings in agile methods," in *Proc. 2nd XP Universe 1st Agile Universe Conf.*, vol. 2418, 2002, pp. 197–207.
[21] K. Beck, *Extreme Programming Explained: Embrace Change*. Reading, MA, USA: Addison-Wesley, 1999.
[22] K. Beck and C. Andres, *Extreme Programming Explained: Embrace Change*, 2nd ed. Reading, MA, USA: Addison-Wesley, 2004.
[23] J. Sutherland and K. Schwaber, "Business object design and implementation workshop," in *Proc. 10th Annu. Conf. Object-Oriented Program. Syst., Lang., Appl.*, Oct. 1995, pp. 170–175. [Online]. Available: https://dl.acm.org/doi/pdf/10.1145/260094.260274
[24] K. Schwaber and J. Sutherland. (2017). *2017 Scrum Guide*. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S1051227609002210
[25] M. O. Ahmad, J. Markkula, and M. Oivo, "Kanban in software development: A systematic literature review," in *Proc. 39th Euromicro Conf. Softw. Eng. Adv. Appl.*, Sep. 2013, pp. 9–16.
[26] (2023). *Large Scale Scrum (LeSS)*. [Online]. Available: https://less.works/
[27] (2022). *Scrum@Scale (SaS)*. [Online]. Available: https://www.scrumatscale.com/scrum-at-scale-guide/
[28] S. Ambler, "Disciplined agile delivery meets CMMI," *Cutter IT J.*, vol. 25, p. 28, Nov. 2012.
[29] (2024). *Scaled Agile Framework (SAFe)*. [Online]. Available: https://scaledagileframework.com/
[30] R. Kasauli, E. Knauss, B. Kanagwa, A. Nilsson, and G. Calikli, "Safety-critical systems and agile development: A mapping study," in *Proc. 44th Euromicro Conf. Softw. Eng. Adv. Appl. (SEAA)*, Aug. 2018, pp. 470–477.
[31] M. Vuori, "Agile development of safety-critical software," Dept. Softw. Syst., Tampere Univ. Technol., Tech. Rep. 14, 2011.
[32] L. T. Heeager and P. A. Nielsen, "Meshing agile and plan-driven development in safety-critical software: A case study," *Empirical Softw. Eng.*, vol. 25, no. 2, pp. 1035–1062, Mar. 2020. [Online]. Available: https://link.springer.com/article/10.1007/s10664-020-09804-z
[33] J. Ronkainen and P. Abrahamsson, "Software development under stringent hardware constraints: Do agile methods have a chance?" 2017, *arXiv:1711.08637*.
[34] *Guidelines for Development of Civil Aircraft and Systems*, document ARP4754A, SAE Int., Dec. 2010.
[35] *Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment*, document ARP4761, SAE Int., Dec. 1996.
[36] *Integrated Modular Avionics (IMA) Development Guidance and Certification Considerations*, document DO-297, RTCA, 2005.
[37] *Design Assurance Guidance for Airborne Electronic Hardware*, document DO-254, RTCA, 2000.
[38] *Safety Assessment of Transport Airplanes in Commercial Service*, document ARP5150A, SAE Int., 2013.
[39] *Safety Assessment of General Aviation Airplanes and Rotorcraft in Commercial Service*, document ARP5151A, SAE Int., 2013.

[40] A. Singh. (2011). *RTCA DO-178B (EUROCAE ED-12B)*. [Online]. Available: https://www.researchgate.net/publication/315023556
[41] *Supporting Information for DO-178C and DO-278A*, document DO-248C, RTCA, 2011.
[42] *Software Tool Qualification Considerations*, document DO-330, RTCA, 2011.
[43] *Model-Based Development and Verification Supplement To DO-178C and DO-278A*, document DO-331, RTCA, 2011.
[44] *Object-Oriented Technology and Related Techniques To DO-178C and DO-278A*, document DO-332, RTCA, 2011.
[45] *Formal Methods Supplement To DO-178C and DO-278A*, document DO-333, RTCA, 2011.
[46] A. Kuckertz and J. Block, "Reviewing systematic literature reviews: Ten key questions and criteria for reviewers," *Manage. Rev. Quart.*, vol. 71, no. 3, pp. 519–524, Jul. 2021. [Online]. Available: https://link.springer.com/article/10.1007/s11301-021-00228-7
[47] C. Baron and V. Louis, "Towards a continuous certification of safety-critical avionics software," *Comput. Ind.*, vol. 125, Feb. 2021, Art. no. 103382.
[48] J. C. Marques, S. M. H. Yelisetty, A. M. Da Cunha, and L. A. V. Dias, "CARD-RM: A reference model for airborne software," in *Proc. 10th Int. Conf. Inf. Technol., New Generat.*, Apr. 2013, pp. 273–279.
[49] J. Marques and A. Cunha, "A reference method for airborne software requirements," in *Proc. IEEE/AIAA 32nd Digit. Avionics Syst. Conf. (DASC)*, Oct. 2013, pp. 1–29.
[50] J. M. S. C. Rodrigues, J. E. F. Ribeiro, and A. Aguiar, "Improving documentation agility in safety-critical software systems development for aerospace," in *Proc. IEEE Int. Symp. Softw. Rel. Eng. Workshops (ISSREW)*, Oct. 2022, pp. 222–229.
[51] R. A. da Silva Coelho, A. M. da Cunha, A. A. Gomes, E. R. Segeti, J. C. Marques, L. M. Vicente, L. A. V. Dias, M. L. Abrunhosa, R. N. Kamoi, S. Mirachi, T. J. Diedrich, and V. da Costa Guerra, "Developing a CDS with scrum in an interdisciplinary academic project," in *Proc. IEEE/AIAA 33rd Digit. Avionics Syst. Conf. (DASC)*, Oct. 2014, pp. 1–13.
[52] A. Wils, S. Van Baelen, T. Holvoet, and K. De Vlaminck, "Agility in the avionics software world," in *Proc. 7th Int. Conf. Extreme Program. Agile Processes Softw. Eng.*, in Lecture Notes in Computer Science, vol. 4044, 2006, pp. 123–132.
[53] (2020). *IBM Overview of DOORS*. [Online]. Available: https://www.ibm.com/docs/en/engineering-lifecycle-management-suite/doors/9.7.0?topic=overview-doors
[54] J. J. Ahonen and P. Savolainen, "Software engineering projects may fail before they are started: Post-mortem analysis of five cancelled projects," *J. Syst. Softw.*, vol. 83, no. 11, pp. 2175–2187, Nov. 2010.
[55] J. Marsden, A. Windisch, R. Mayo, J. Grossi, J. Villermin, L. Fabre, and C. Aventini. (2018). *ED-12C/DO-178C Vs. Agile Manifesto A Solution to Agile Development of Certifiable Avionics Systems*. [Online]. Available: https://hal.science/hal-02156357
[56] J. Marques, A. Cunha, and L. Dias, "Q-Scrum: A framework for quality in safety-critical development," in *Proc. Int. Conf. Quality Inf. Commun. Technol.*, 2020, pp. 238–245.
[57] C. Baron and V. Louis, "Framework and tooling proposals for agile certification of safety-critical embedded software in avionic systems," *Comput. Ind.*, vol. 148, Jun. 2023, Art. no. 103887. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166361523000374
[58] J. Marques and A. M. da Cunha, "A set of requirements for certification of airborne military software," in *Proc. IEEE/AIAA 38th Digit. Avionics Syst. Conf. (DASC)*, Sep. 2019, pp. 1–7.
[59] G. Goncalves, R. Shigemura, P. Silva, R. Santana, E. Silva, A. Dakwat, F. Miguel, P. Tasinaffo, A. Cunha, and L. Dias, "An agile developed interdisciplinary approach for safety-critical embedded system," in *Proc. 14th Int. Conf. Inf. Technol. New Generations*, 2018, pp. 947–949.
[60] S. Mirachi, V. da Costa Guerra, A. M. da Cunha, L. A. V. Dias, and E. Villani, "Applying agile methods to aircraft embedded software: An experimental analysis," *Softw., Pract. Exper.*, vol. 47, no. 11, pp. 1465–1484, Nov. 2017.
[61] C. Bertrand and C. P. Fuhrman, "Towards defining software development processes in DO-178B with OpenUP," in *Proc. Can. Conf. Electr. Comput. Eng.*, May 2008, pp. 851–854.
[62] K. Dmitriev, S. A. Zafar, K. Schmiechen, Y. Lai, M. Saleab, P. Nagarajan, D. Dollinger, M. Hochstrasser, F. Holzapfel, and S. Myschik, "A lean and highly-automated model-based software development process based on DO-178C/DO-331," in *Proc. AIAA/IEEE 39th Digit. Avionics Syst. Conf. (DASC)*, Oct. 2020, pp. 1–10.
[63] C. Hubbs and J. Myren, "Automating airborne software certification compliance using cert DevOps," in *Proc. IEEE/AIAA 42nd Digit. Avionics Syst. Conf. (DASC)*, Oct. 2023, pp. 1–6.
[64] *Software Considerations in Airborne Systems and Equipment Certification (Including Amendment N°1 19 October 1999)—With Corrigendum 1*, document ED-12C, Feb. 2021.
[65] *SKYbrary Aviation Safety: Heinrich Pyramid*, SKYbrary, 2019.
[66] F. Caliskan and C. Hacizade, "Sensor and actuator FDI applied to an UAV dynamic model," *IFAC Proc. Volumes*, vol. 47, no. 3, pp. 12220–12225, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474667016435597

J. EDUARDO FERREIRA RIBEIRO (Member, IEEE) received the degree in information systems management from the Polytechnic Institute of Càvado and Ave. He is currently pursuing the Ph.D. degree in informatics engineering with the Faculty of Engineering, University of Porto (FEUP), under the mentorship of Ademar Aguiar and João Gabriel Silva. With over 25 years in technology, his expertise spans software development, automation and quality engineering, project management, and Agile coaching. Since 2018, he has been with Critical Software SA, where he is currently the Quality & Agility Director. His significant contributions include leading the Agile transformation with the High Integrity Systems Division, positively impacting both the company and its clients. In addition, he authors the "Beyond Lean Agile Blog," showcasing his insights into Agile methodologies. His research focuses on applying Agile methods in safety-critical domains, more precisely, the aerospace domain, reflecting his passion.

JOÃO GABRIEL SILVA received the Ph.D. degree in electrotechnical engineering/informatics, in 1988. He was the Dean of the Faculty of Science and Technology, from 2006 to 2011. From 2011 to 2019, he was the Rector of the University of Coimbra, where he is currently a Full Professor with the Department of Informatics Engineering, Faculty of Science and Technology. He co-founded Critical Software SA, a well-known Portuguese software company. He participated in many Portuguese and European-funded research projects with many internationally strongly cited articles in dependable computing and software engineering. He directed many industrial projects, including the first Portuguese computer, Ener 1000, from 1981 to 1984. He chaired many international scientific conferences.

ADEMAR AGUIAR is currently an Associate Professor with the Faculty of Engineering, University of Porto (FEUP), where he is responsible for several course units related to software engineering. He is also a Researcher with INESC TEC, where he coordinates the HumanISE Research Center. He conducts several other activities related to software, from research, training, and consulting, to ideation, coding, and venture development. After more than 30 years of programming, he found a special interest in the architecture, design, and implementation of complex software systems, applying Agile methods (XP, Scrum), wikis, and open collaboration tools to better communicate and preserve the necessary software knowledge. He is the coauthor of *A Scrum Book: The Spirit of the Game* (2019). He often organizes scientific conferences; recently, was the General Chair of XP 2018, the Research Workshops Co-Chair for XP 2021, and the General Chair of <Programming> 2022.