Wiley
IET Software
Volume 2025, Article ID 5227350, 24 pages
https://doi.org/10.1049/sfw2/5227350

Review Article

A Systematic Literature Review on Application of Agile Software Development Process Models for the Development of Safety-Critical Systems in Multiple Domains

Hafiza Maria Maqsood[1], **Joelma Choma**[2], **Eduardo Guerra**[3] and **Andrea Bondavalli**[1]

[1]*Department of Mathematics and Informatics, University of Florence, Florence, Italy*
[2]*Laboratory for Applied Computing and Mathematics at the National Institute for Space Research (INPE), São José dos Campos, Brazil*
[3]*Faculty of Computer Science, Free University of Bozen-Bolzano, Bolzano, Italy*

Correspondence should be addressed to Andrea Bondavalli; andrea.bondavalli@unifi.it

Received 26 June 2023; Revised 18 September 2023; Accepted 22 July 2025

Academic Editor: Manuel Angel Serrano

This paper presents a literature review on using agile for safety-critical systems (SCSs). We have systematically selected and evaluated relevant literature to find out major areas of concern for adapting agile in the development of SCSs. In the paper, we have listed the most used Agile process models and reasons for their suitability for SCS, then we have outlined phases of the software development life cycle (SDLC) where changes are required to make an agile process suitable for the development of SCSs. Thirdly, we have elaborated on problems and other important aspects according to specific domains where agile is used for SCS. This paper serves as an insight into the latest trends and problems regarding the use of Agile process models to develop SCSs.

1. Introduction

The development of safety-critical systems (SCSs) employed in the aerospace, automotive, railway, nuclear, and medical sectors requires high standards of reliability, availability, and fault tolerance. SCSs are defined as those systems whose failure can cause harm [1]. The system is considered safety-critical if its failure can lead to unacceptable circumstances such as severe human, environmental, or economic losses [2]. Traditionally, this field relies on rigorous software engineering practices, including formal verification, extensive testing, risk analysis, and regulatory certification [3, 4].

The development of SCSs in an agile way can be very beneficial in terms of time and cost [5, 6]. The basic principles of agile are that there should be rapid development, strong communication among all stakeholders, and changes should be welcomed at any stage of development [7]. Since there is a strong focus on people, each team member should be motivated and provided with a suitable environment and adequate support to perform their job effectively [8]. An agile team is very responsive to changes since adapting to change is what agile software development is all about [9]. Moreover, it is very important for an agile team to understand that software is developed by teamwork, and collaboration is the heart of success [10].

Software engineers gathered forces and began to classify agile processes in early 2001 [7]. Agile Alliance stated the agile manifesto as "individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, and responding to change over following a plan" [7]. The agile manifesto states that it gives priority to working software over detailed documentation. Regarding this subject, it seems there is a conflict between heavy documentation and agile principles. Cohen et al. [11] argue that there are issues that must be looked upon in written communication, and we should not abandon documentation; instead, we should use the documentation at and for appropriate points, especially for the development of SCSs [12].

The critical nature of SCSs requires that, if not all, maximum risks are handled during the development of these systems [13]. However, it is not enough to just perform a risk analysis; certain safety standards must be incorporated during the development of these systems, and for this reason, they must have a testing strategy in place [3]. Holcombe et al. [14] state that formal verification methods and test-driven development can be used together, and there can be many enhanced benefits of this approach, ensuring the safety of systems along with reliability. The authors further argue that SCSs have to comply with certain safety standards [14]. Nevertheless, currently, there are not many techniques of testing that can fit into this scenario [4].

The use of agile methodologies to develop SCSs poses many challenges. There is a lot of room for improvement in the development process to fit perfectly the needs of SCSs, especially from an agile perspective [15]. For example, there is a gap to be filled about how organizations should arrange agile teams to achieve the right level of autonomy in any particular scenario, in our case, for SCSs [16]. Agile stresses cross-functional autonomous teams [17], whereas SCSs need specific experts who perform specific tasks. Moreover, it is a challenging task to create cross-functional teams and keep the size of the team small, for example, 10–15 persons on a team, which is a standard agile team size [18].

The existing literature on agile methods in SCSs is relatively sparse compared to the more extensive research available on nonagile approaches. However, to understand the challenges and problems in general between agile and SCSs, Heeager and Nielsen [19] have discussed areas of major concern. There are other papers that discuss the application of agile in different fields. Medical and pharmaceutical are the most discussed [20]. However, there is no major evidence to specify the particular problems of each field. Also, not much is discussed in terms of which agile methods are more used and what the reasons are. Some papers argue to use a hybrid approach like [21].

Agile has been used for the development of software in different fields for years now [7, 22]. However, there are still many concerns that need to be addressed for using agile in SCSs [23]. However, there is much to be established to incorporate agile in the development of SCSs [24, 25]. This study presents a literature review carried out to investigate the main gaps between agile and SCSs and also to find possible collaborations between the two. As a main contribution, this study highlights the main areas of concern and points of conflict between agile and SCSs. Along with that, it addresses some of the major problems for agile adaptation in the development of SCSs.

2. Methodology

There are many ways of performing a literature review, as explained by Templier and Paré [26]. It is also common to combine multiple approaches [27]. We have adopted an interpretive literature review approach based on grounded theory elements given by Wolfswinkel et al. [28], along with the steps and guidelines of [29]. We conducted the following steps: (1) research questions defining; (2) search strategy, which includes defining the keywords, selection of data sources, and refining the query for searching in selected databases; (3) study selection, which includes establishing the inclusion/exclusion criteria, execute the search for studies and apply selection criteria; (4) assess the quality of selected studies; and (5) data extraction and analysis. The summary of selected studies can be found in Table A1 and selected study categories can be found in Table A2 in Appendix section. In the next sections, we provide further details of these steps.

2.1. Research Question. In broader terms, our empirically researched questions have the primary goal of identifying concerns about agile in different fields of SCSs. We also try to identify the most used agile approaches and aspects that make them more suitable compared to others in this domain. Then, we further investigate the prospects of a hybrid approach. We started by formalizing our research questions so that we could review the selected literature with a clear notion of "what to look for" in mind. We formulated the following questions:

- RQ1: Which agile process model/models are most used or discussed for developing SCSs?
- RQ2: Which phases of software development life cycle (SDLC) are most discussed for adopting agile for the development of SCSs?
- RQ3: Which domains of safety systems report the highest adaption of agile for their systems?

2.2. Search Strategy and Data Sources. Considering the purpose of the study, we started with the basic keywords of "agile" and "safety-critical systems". However, soon we realized that we must form a proper string to include relevant studies. After doing multiple iterations by including different keywords and using clauses of "OR" and "AND," we were able to comprehensively formalize the following query string:

> (Agile OR agility OR Scrum OR XP OR Feature-driven development OR ASD OR Crystal OR DSDM) AND (safety software OR safety critical software OR safety-critical systems OR regulated software).

We have used a combination of search engines and different repositories to locate the relevant literature. In particular, we have used the following digital libraries: Google Scholar, ACM, IEEE Xplore, Scopus, and Web of Science.

2.3. Study Selection. We have chosen the following criteria for including studies in our review:

- Studies that relate to agile and SCSs.
- Studies that discuss any particular agile process model with an SCS.
- Studies that discuss issues between agile and safety.
- Studies that discuss safety and agile in a broad spectrum.

The studies that fall under the following criteria were not included in this review:

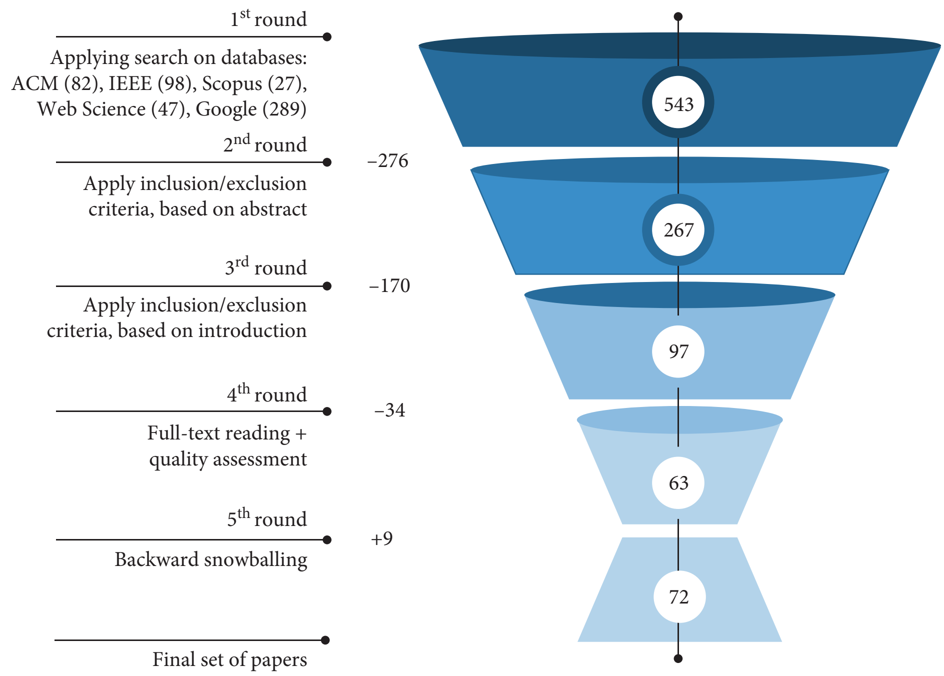

Figure 1: Relevant papers selection rounds.

- Studies written in languages other than English.
- Short papers or posters have a length of fewer than three pages.
- Studies that are unavailable on the mentioned platforms.
- Studies that focus on SCSs without discussing process models or methodology for development.

When we started our search after formalizing the query, it still yielded 1000s of results. We analyzed these results to select the most relevant ones with substantial information. In the first round, just by looking at the titles, we found 543 papers to be relevant. We performed a second round of selection on these 543 papers. We have read abstracts of all these studies and shortlisted them to 267 papers, based on predefined inclusion/exclusion criteria. In the third round, we began reading the introductions of the papers, which further reduced the number of selected papers to 97. Only these 97 papers met the inclusion criteria for this literature review. In the fourth step, we read the full text of selected papers and also applied two rubrics for quality assessment. The two rubrics used for the quality assessment were rigor and relevance. Further details on quality assessment are given in Section 2.4. After completing this stage, we had 63 papers. In the last step, we performed backward snowballing on these papers and that made us add 9 studies to the selected data. Backward snowballing is a technique in which one needs to look up the references of selected studies [89]. After going through the list of referenced papers, we found 9 more papers that were relevant to our interest. Hence, at the end of this selection process, we have a total of 72 papers. We further performed an analysis of this selected literature. Figure 1 summarizes the selection rounds of the paper.

2.4. Quality Assessment. There are different methods available to assess the quality of studies [90–92]. Bandara et al. [90] give a list of questions that should be answered for all papers to examine the quality of papers. Brereton et al. [91] formed a checklist of questions and based their quality criteria on answers to those questions. Their checklist can be divided into four parts. First, they ask questions about the criteria to design the study; second, they have points about the method used for the study setup; third are questions about how they performed the study; and fourth and final are points about how they come up with conclusions from the selected studies. Anwar and Pfahl [92] used a table of questions to assess the quality of studies. They assigned numbers to each question and further performed analysis on the basis of these numbers. The higher the numbering score, the higher the quality of the paper.

Two independent researchers have worked on quality assessment. We carefully looked into the above-mentioned various methods for performing quality analysis on our selected set of studies. We have used well-defined inclusion/exclusion criteria from the very start of the process. After selecting 72 papers, we applied metrics for calculating the rigor and relevance of the papers, as proposed by Ivarsson and Gorschek [93]. Ivarsson and Gorschek [93] have proposed three aspects to measure rigor: context description (*C*), study design (*S*), and study validity (*V*) (Table 1). For evaluating relevance, Ivarsson and Gorschek [93] have proposed a matrix on the basis of four aspects: subjects (*U*), context (*C*), scale (*S*), and research method (RM) (Table 2). They further assign scores from 0 to 1 to evaluate these aspects. If the aspect is well-defined, it is given a score of 1; if it is at a medium level, then a score of 0.5 is given, and for a weak description 0, is given. For a strong description, an aspect should be clearly defined and with all

Table 1: Scoring rubric for evaluating rigor based on [93].

Aspect	Score		
Context described (C)	Strong description (1)	Medium description (0.5)	Weak description (0)
Study design described (S)	Strong description (1)	Medium description (0.5)	Weak description (0)
Validity discussed (V)	Strong discussed (1)	Medium discussed (0.5)	Weak discussed (0)

Note: Quantification of rigor $= C + S + V$.

Table 2: Scoring rubric for evaluating relevance based on [93].

Aspect	Score		
Subjects (U)	Industry professional (1)	Students, researchers (0.5)	Subject not mentioned (0)
Context (C)	Industrial setting (1)	Laboratory (0.5)	Context not mentioned (0)
Scale (S)	Realistic size: industrial scale (1)	Down-scaled industrial (0.5)	Toy example (0)
Research method (RM)	Action Research, lessons learned case study, field study, interview, descriptive exploratory survey (1)	Conceptual analysis/mathematical, laboratory experiment (human), laboratory experiment (software) (0.5)	Not mentioned (0)

Note: Quantification of relevance $= U + C + S + \mathrm{RM}$.

Table 3: Scoring rubric of selected studies for evaluating rigor based on [93].

Study	Description of context (C)	Description of study design (S)	Validity discussed (V)	Rigor score
S59	0.5	0	0	0.5
S48	1	0.5	0	1.5
S26	1	1	0	2
S65	0.5	0.5	0	1
S41	0.5	0,5	0	1
S67	1	1	1	3
S33	0.5	0.5	0	1
S34	0.5	0	0	0.5
S60	1	1	0	2
S37	0.5	0.5	0.5	1.5

Note: Quantification of rigor $= C + S + V$.

required parameters in a way that is understandable by the user. If an aspect has some relevant parameters, but in a vague way, that is not understandable by the reader, then it is at a medium level. If there is no discussion of an aspect, then it is considered weak. Tables 1 and 2 present the rubrics of rigor and relevance, respectively.

In our study, two researchers assigned scores to all studies separately, and each of them calculated one final value of rigor and relevance for each paper. Papers were further scrutinized on the basis of these values. To further assess the quality of these attained values, we calculated Kendall's coefficient of concordance [94, 95]. Kendall's coefficient of concordance, commonly known as Kendall's W is a nonparametric statistical test. It is a normalized form of the Friedman test [96]. It is used for assessing agreement among raters. Kendall's W ranges from 0 (no agreement) to 1 (complete agreement). Kendall's coefficient of concordance (W), gives the degree of association of ordinal assessments made by multiple assessors on the same data [97]. We have used the following formula to calculate Kendall's coefficient of concordance:

$$W = 12 \times S(m^2) \times ((n^3 - n) - m \times T)),$$

where S is the sum of squared deviations (in our case sum of squared deviations of final values of rigor or relevance), m is the number of raters, n is the total number of objects (in our scenario, the number of selected studies), and T represents tied ranks.

We selected 10 random studies from our selected list of studies. These 10 studies have been assigned certain values of rigor and relevance by two authors, for details, please refer to Table 3 for rigor and Table 4 for relevance. After applying the above-mentioned formula of Kendall's coefficient (W), we got the values of 0.97 for rigor and 0.98 for relevance.

As 1 is considered the complete agreement and 0 represents no agreement between raters, our values represent that there is great consensus among assessors about the quality of selected studies. This gives empirical evidence for the quality of studies in terms of their relevance and rigor. Tables 3 and 4 show the quality assessment performed on selected studies as per above-described criteria.

2.5. Data Extraction and Analysis. In our analysis, we have used elements of the Grounded Theory Method [98] for coding our data. Coding the data refers to assigning keywords to data

Table 4: Scoring rubric of selected studies for evaluating relevance based on [93].

Study	Subject (U)	Context (C)	Scale (S)	Research method (RM)	Relevance score
S59	0	1	1	0	2
S48	1	1	1	1	4
S26	1	1	1	1	4
S65	1	1	1	1	4
S41	1	1	1	1	4
S67	1	1	1	1	4
S33	1	1	1	1	4
S34	0	0	0	1	1
S60	0	0	1	1	2
S37	0	0	0	0	0

Note: Quantification of relevance $= U + C + S + \mathrm{RM}$.

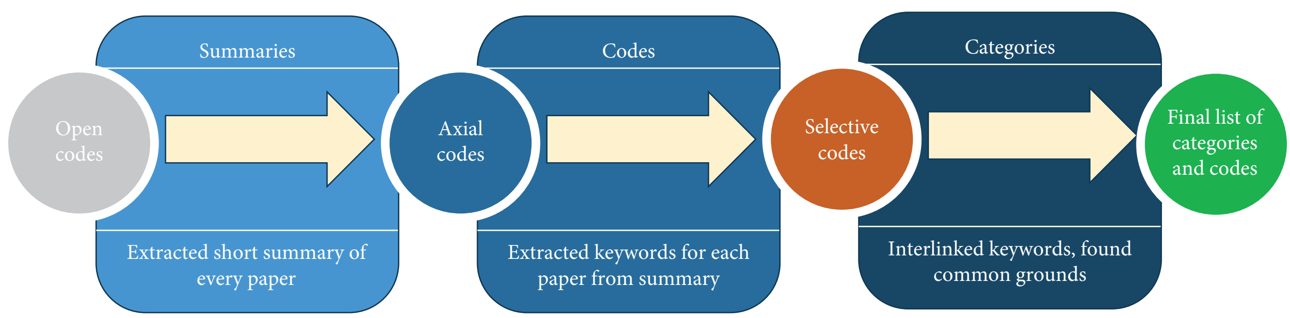

Figure 2: Steps of coding.

that best represent that paper. Specifically, we have followed three steps of assigning codes to our data: open code, axial code, and selective code [99]. Figure 2 displays the steps of coding used during our analysis.

Open codes refer to the bird's-eye view of data. We have read papers time and again to get the basic idea of each paper clearly. Then we created a one-line summary of each paper, writing the essence of the paper with major keywords. Open, selective, and axial coding can be done in back and forth manner [28]. This is the strategy we have used to ensure that we extract all required information from selected studies.

After performing the step of open coding, we moved to axial coding. Here, the idea is to read what has been extracted till now and narrow it down to more concrete keywords. We read the extracted summaries and extracted keywords from the summaries. At this stage, we had a relatively huge set of keywords extracted from all selected studies. The last step in coding is to interrelate the codes and find common ground. All the steps of coding are done by moving back and forth multiple times. We established some high-level categories and linked all codes to these categories. This led us to discover different aspects of agile and security-critical systems, which we will present in Section 3.

There are plenty of ways to organize data for analysis. It is widely used by many researchers. It provides very efficient support for the analysis of data. There are many tools available to make the process easy and efficient [100]. The choice of tool can vary according to the type of data, field, or the kind of analysis one wants to perform on data [90, 101]. We have used the "R" tool for coding our data [102]. We have carefully selected and performed each step of our chosen methodology. At this point, we had comprehensive and well-sorted data with the help of the tool "R" for further analysis.

3. Results and Analysis

After completing the stage of coding, we were able to establish results in terms of 3-tier categories and relevant codes. At the highest level, we have three groups, namely SDLC phases, application domains, and process models. At the second tier, we have requirements, testing, IV "I&" V, and traceability under SDLC, medical, automotive + rail, avionics, nuclear plants, and space under application domains, and finally Scrum, XP, hybrid model, ASD and DSDM under process models. It is represented in Figure 3.

All the codes that we have extracted from selected studies belong to the above-mentioned category and subcategories. A list of codes along with their referenced study is shown in Figure 4.

3.1. Studies Overview. In this section, we present an overview of selected studies along with associated details. We have selected 72 studies that make our final set of data. As shown in Figure 5. The data is represented in terms of time, type of publication, and the kind of research methodology adopted by papers.

It is visible that the peak was raised in 2013, 2017, and most recently also. This shows that this is an active field of research at present. Figure 5 also shows the type of publication of studies.

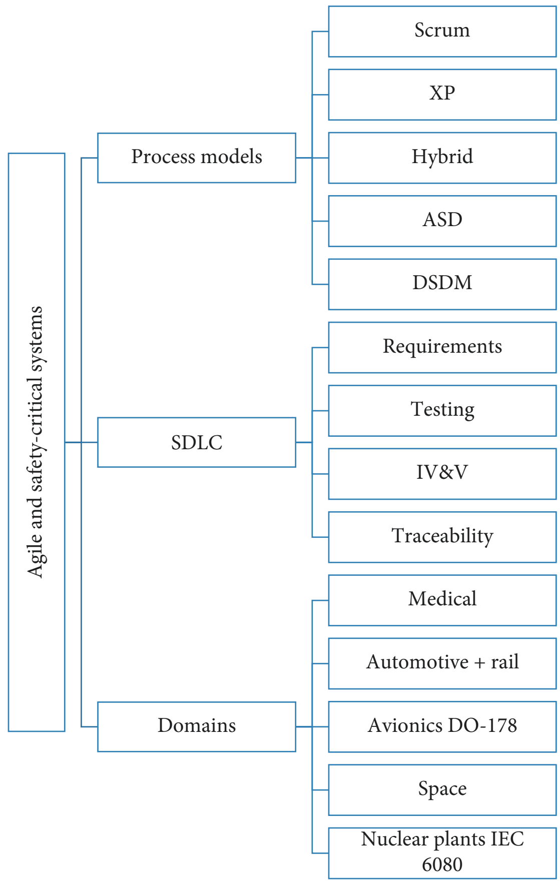

Figure 3: Agile and safety-critical systems categories.

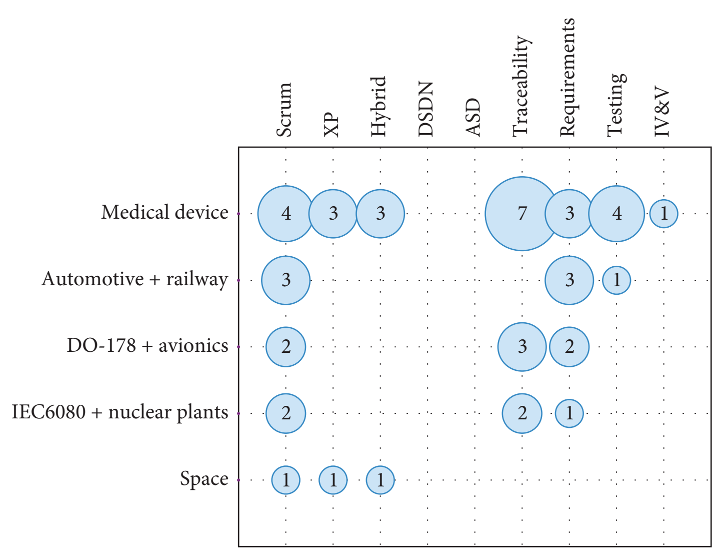

Figure 4: Mapping of studies per categories.

For example, most of the studies were presented at conferences, secondly in journals, and then in workshops. The most adopted research methodology in selected studies was example applications. From 2001 to 2013, the research was more focused on experiments and case studies, where researchers focused to learn and see by applying agile to SCSs. However, from 2014 onwards the research is more focused on lessons learned, challenges for the application of agile in SCSs, and possible approaches. Specifically, a hybrid approach is adopted with reports of many benefits. It is a combination of traditional approaches and agile. There are many reported benefits of the approach.

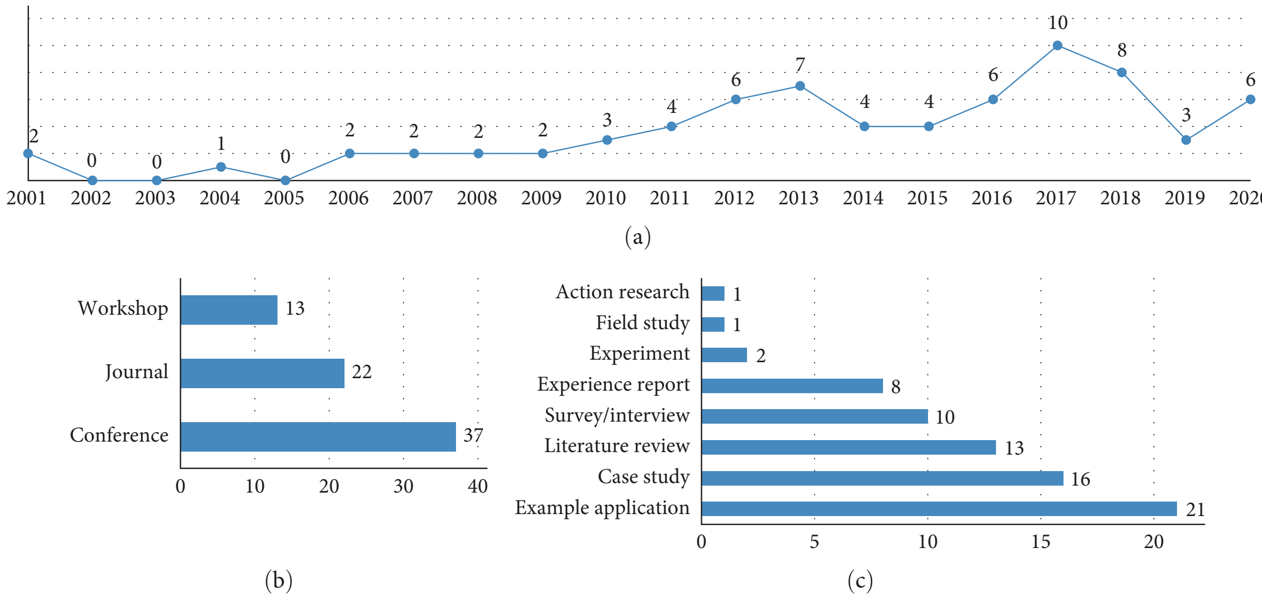

Figure 5: Types of studies selected. (a) Years of publication, (b) publication type, and (c) type of research.

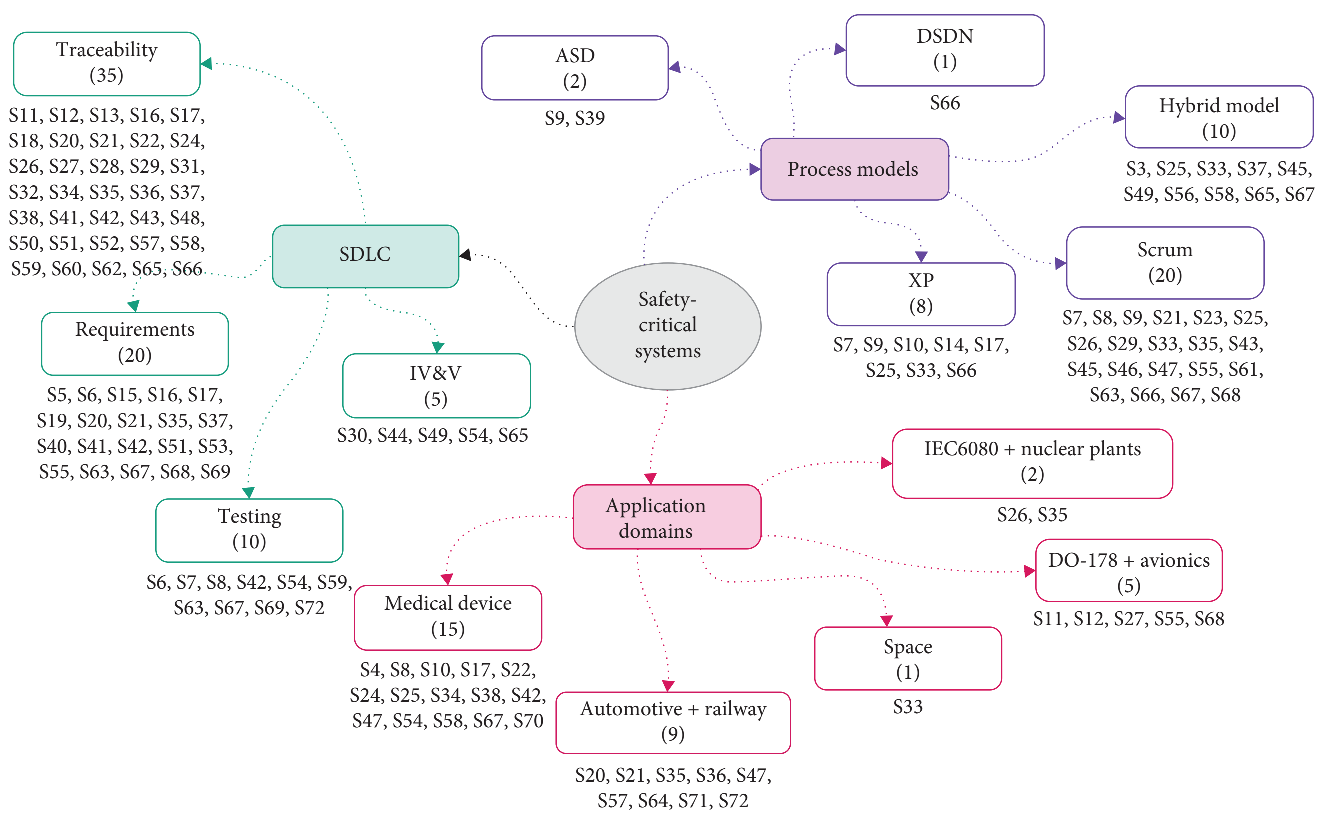

Figure 6: Results of studies categorization.

There are different research methodologies applied in our set of selected studies. An overview of the type of RMs presented by papers is shown in Figure 5. There are various techniques for categorizing RMs [103]. Depending on the research methodology used, the reliability of findings can vary. Generally, researchers give more value to the research performed in industrial settings, for example, case studies or example applications [104]. There are other categories of RMs defined by Easterbrook et al. [105, 106], such as action research, field study, survey, experience report, and literature review. We have selected categories most fitted to our studies and they include all major RMs as shown in Figure 6.

Table 5: Process models used for developing safety-critical systems.

Process model	SDLC-phases	SDLC-steps	Compliance to safety standards	Proof of industrial application
Scrum	Fully covered	Well-defined	Yes with small modifications	Yes
XP	Partially covered	Not defined	Needs modifications	Yes
Hybrid	Fully covered	Well-defined	Yes	Yes
ASD	Abstract	Abstract	Needs modifications	No
DSDM	Abstract	Abstract	Needs modifications	No

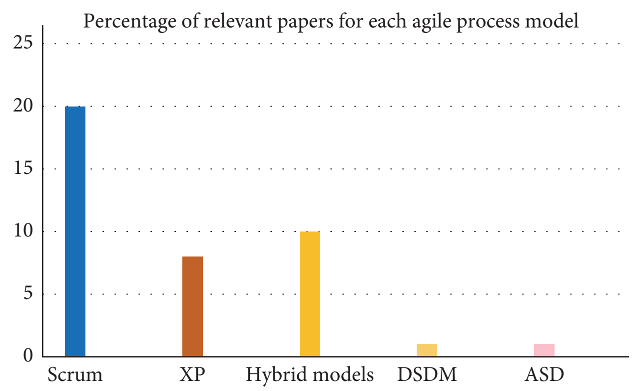

Figure 7: Process models.

In the Appendix, we have presented the selected studies with the following characteristics: distribution over time, venue of publication, type of study, and applied RM. Studies have discussed different aspects of agile for SCSs.

In early 2000, studies were more focused on case studies and experiments, while more recently, there is more focus on the challenges of using agile in different safety domains and possible solutions for the application of agile in the development of SCSs. Various topics are discussed in the literature regarding multiple aspects of using agile for the development of SCSs. Here we present our results from selected studies.

We have focused on all selected studies, in particular, to find answers to our research questions.

3.2. Process Models (RQ1). In this section, we present our analysis of studies in reference to our first research question stated as "Which agile process model/models are most used or discussed for developing SCSs?"

We found that Scrum and XP are the most discussed process models in our literature review [70]. The relevant data can be viewed through Figures 7 and 8. There are multiple studies that discuss Scrum and claim that it is one of the best and most suited agile models for the development of SCSs. Özcan and McCaffery [80] have applied Scrum and XP for Medical device development and found that Scrum partially or fully covers five aspects of a process model, which are project planning, system requirements analysis and software requirements analysis, stakeholder requirements definition, and project assessment and control. The authors have also looked into XP for the development of medical device software and propose that XP provides only partial coverage of project planning, software requirements analysis, project assessment, and control, software unit implementation and verification, software release and software problem resolution, software integration, and integration testing. They further combined XP and Scrum and found that some additional practices were still needed to develop medical device software. By extending their study and combining DSDM with XP and Scrum, the results were better than combining the two process models [85]. With three process models combined, the resulting process provided better coverage and support for the development of medical device software [32, 33].

Another interesting approach is shared by Cordeiro et al. [34], who find two aspects of agile process models very intriguing, namely incremental approach, adaptive planning, and flexibility. To incorporate these aspects, they combined XP and Scrum and proposed agile patterns that can be termed useful for the development of SCS.

Several studies have discussed only Scrum from the perspective of its usage for SCSs [36, 54]. It has been widely used on its own, or there is a new version called Safe Scrum [49] that has some modifications to better suit the needs of SCS. Scrum has shown some promising results in the domain of SCS [69]. Most studies report that a customized version of Scrum is more suitable for safety systems than the model in its pure form. Studies also argue that there are certain aspects that need to be tailored in Scrum or need modification. These are documentation, planning, proof of conformance, requirements specifically safety requirements, and evaluation, mitigation of risks, or simply put "risk analysis," verification, and validation, and last but most important, traceability. These ideas are very concretely explained by Doss and Kelly [66], who discuss four basic underlying principles for all safety standards. They say that although standards differentiate from each other in their details, there are $4 + 1$ principles that stay true. And based on these principles, Scrum can be integrated into the development of SCSs. The first three principles are about safety requirements. Safety requirements should be in line with safety standards; they must be maintained throughout the process, and there must be criteria to satisfy these requirements. The fourth principle says that the "hazardous behavior" of software should be mitigated. The last principle is about the balance between following these principles and software systems. Second most discussed model is eXtreme Programming/XP [40].

After Scrum, XP emerged as the most used agile process model [37]. Most studies have reported that there are key differences between the two, and XP in its original form cannot be used for the development of SCSs. By looking at each phase of the development life cycle, starting from requirements, design, code, and testing, there are significant differences, specifically in requirements and testing. Studies propose that changes must be made in XP, and then an extended model can be a solution.

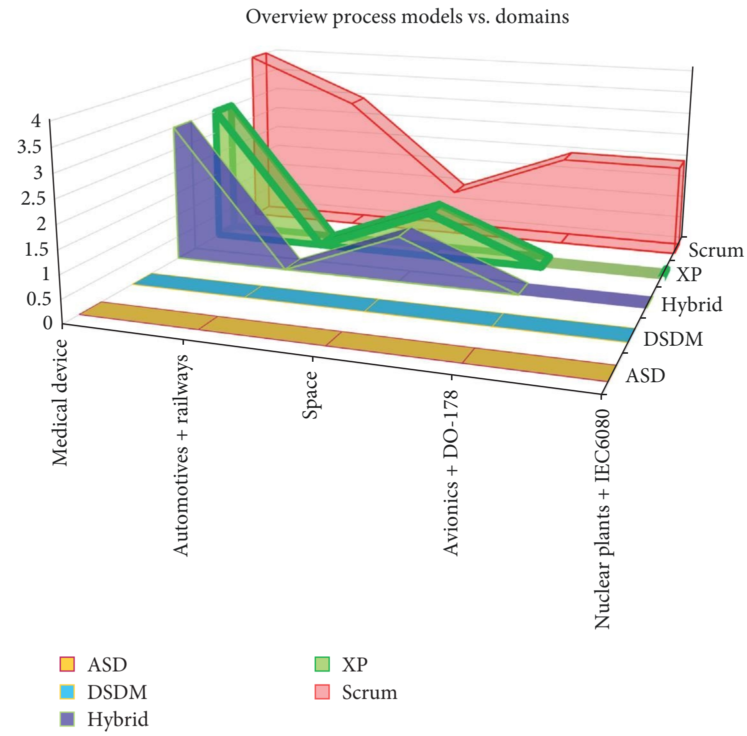

Figure 8: Overview process models vs. domains.

Scrum and XP have been used in combination, too, and provided good results. As agile itself is a way of doing things, it does not impose strict steps, so combining the practices from two agile models has shown success for the development of SCSs. Only one study has provided evidence of using ASD [63], and one has discussed DSDM [85]. DSDM was adopted with a combination of Scrum and XP. There is hardly any evidence of other agile process models being used, and there are no significant results or applications presented for other models.

Another interesting aspect discussed in studies is the human aspect. As agile puts a lot of focus on involving all stakeholders in the process as much as possible, hence there are certain facets that need more attention as compared to traditional approaches. People need to have more open acceptance toward the adaption of new approaches; otherwise, it becomes a huge hurdle to adapt to something as new as agile for SCSs [31]. Grenning [30] has used XP in an industrial setting. They state that one of the parties is always reluctant to use a new approach like XP, and this mindset is the biggest hurdle in the process.

A more recent trend shown in studies is the use of hybrid models. More specifically, after 2010, there has been a huge trend of using hybrid process models [68] have introduced a hybrid model with SAFE-Scrum and STPA (System-Theoretic Process Analysis). In this approach, researchers have combined the traditional process model approach with agile methods. This approach has shown the most promising results and is widely adopted. Still, Scrum or Safe Scrum remains the most popular one for the development of SCS. Safe Scrum is also a hybrid model, where people modify it a bit every time according to the project at hand. It takes basic concepts of Scrum and integrates traditional methods to form a process model that can develop SCS successfully with agile.

Arthur and Dabney [71] have taken a deep dig into hybrid process models. Their work discussed the approach in the context of validation and verification. It is stated that validation and verification are still daunting task even with hybrid approaches. They identified 30 practices of verification and validation and assessed them against a hybrid approach. They further say that some methods of the hybrid approach can be applied without any change, some need minimal modification, whereas others cannot be applied and need replacement. Dabney and Arthur [84] have further worked along the same lines and described some approaches to overcome these hurdles specific to validation and verification with hybrid process models for SCS.

Hybrid models are mostly reported as a combination of Scrum and traditional approaches. More popularly called "Safe Scrum". Studies report its wide usage and successful application industry. Some studies report the alteration of the traditional V-Model by combining it with agile, and the new model is termed as AV model [58]. McHugh et al. [61] have presented the application of a hybrid model called AV-model. It is documented that the approach was a huge success, and medical device software was developed 7% faster as compared to when developed with a traditional process model.

Our analysis of process models reveals (as shown in Table 5) that there are certain aspects due to which one model is reported or used more as compared to others. Here we outline some key findings, which are also summarized in Table 3.

Scrum has appeared as the most used model since it has the following features:

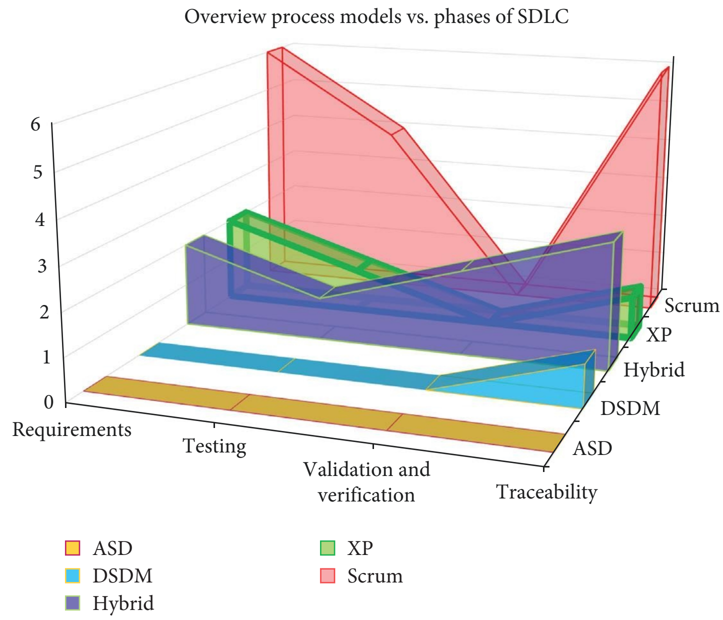

FIGURE 9: Overview of process models vs. phases of SDLC.

- Covers all phases of SDLC. It has well-defined phases and not just an abstract approach to be followed in each phase.
- It is the only agile model with concrete steps.
- Concretely defined steps need very few modifications to comply with safety standards.
- Industrial proof of good performance is available, hence more people are relying on this approach.

XP is the second most used model with the following pros and cons:

- Does not cover all phases of SDLC. Talks about principles that should be followed to define phases.
- Presents ideas of steps only for certain phases of SDLC and does not give concrete steps.
- Since it gives abstract ideas about phases and steps, hence needs major modifications to adapt to safety standards.
- Industrial proof of good performance is available, hence comparatively more studies have reported this approach.

The hybrid model is the most discussed in recent studies. We have found the following aspects regarding the hybrid model:

- Covers all phases of SDLC for SCS.
- Has well-defined phases, mostly taken from traditional SDLC.
- Has well-defined steps of each phase, mostly taken from Scrum.
- Adapts well to the needs of safety standards.
- Widely accepted in the industry with proof of application.

Future prospects, in our point of view, hybrid agile models are the most promising future for the development of SCS with agile. Still, there is a lot of room for research in this area but it can be deemed as the most promising current topic.

3.3. SDLC (RQ2). In this section, we present our analysis of studies in reference to our second research question, stated as "Which phases of SDLC are most discussed for adopting agile for development of SCSs?"

The debate has been ongoing for several years as to use agile process models as they are or with modifications [53]. There are many proposed models with changes in basic agile models or with concepts of SCS integrated into them at different stages of SDLC. We have found that agile SDLC models cannot be adapted without the needed alterations in different phases to meet the needs of SCSs. However, after these amendments, models have shown good results for SCS. The major areas of concern that need attention when one wants to adopt agile for SCS are traceability, requirements, testing, and validation-verification techniques as shown in Figures 9 and 10 and Table 6.

Eliciting and understanding requirements is a daunting task [64]. It becomes even more trivial with SCSs [87]. Safety requirements do not come from clients only; they can come from assessors, safety standards, and risk factors [107]. These requirements must be dealt with great care, hence making the whole process of requirement engineering one of the most discussed topics in our set of studies. Studies report that this phase involves all stakeholders, and hence, there are many aspects regarding people involved in the process. Acceptance of Agile is still an issue, as people (whether development team members, assessors, or clients) are not yet ready for this change. Then there are issues of awareness about agile among stakeholders. Alhubaishy and Benedicenti [72] have presented an

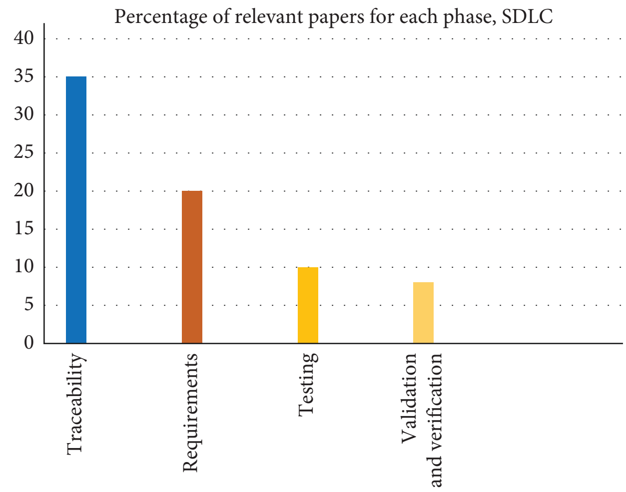

Figure 10: SDLC phases.

Table 6: Phases of SDLC or developing safety-critical systems.

Practices	Effects all stages	Heavy documentation	Huge time and cost	Imposed by safety standard[a]
Traceability	Yes	Yes	Yes	Yes
Requirements	Yes	Yes	Yes	Yes
Testing	Yes	No	Yes	Yes
IV&V	Yes	Yes	Yes	Yes

[a]Not much room for change.

approach to discuss the emotional contagion in agile teams. They propose that this is an important factor that has not been given due importance till now. As agile has a lot of focus on people hence when we try to analyze factors to increase positivity among the people of teams. They argue that this will ultimately increase the ratio of successfully completed projects.

Bartsch [41] presented the results of interviews about problems practitioners face while using agile for SCS. Authors have presented some major hurdles, which are customers' involvement for a better understanding of requirements, and awareness of developers regarding security features of the product. Formalization of requirements is very important, and one compelling approach is discussed by Boström et al. [10]. The authors introduced two extensions to Scrum in the phase of user stories. Authors have proposed "Abuser stories (threat scenarios) and Security-related User stories (security functionalities)" for dealing with requirements. They tested the approach in a student project and concluded by saying that agile can have many benefits if integrated thoughtfully for the development of SCSs.

The heavy documentation required by safety standards for detailed requirements and traceability is in conflict with the agile principle of less documentation [73, 75]. Wang et al. [74] proposed an approach using Scrum for SCS in which they categorized documentation in three ways, namely the safety story pattern, the safety epic pattern, and the agile safety plan. The safety story and epic safety patterns have shown promising results in finding a middle ground acceptable by agile and the standards of SCS, while the agile safety plan needs further investigation. Handling requirements in an iterative manner and adhering to safety standards is a tough task. Hence, this area needs special attention in SDLC for using agile for SCS development. Heavy documentation is used by traditional approaches to document requirements in detail, since agile asks for as little documentation as possible, hence making the task difficult.

Wang et al. [78] have outlined key problem areas between agile and SCSs with respect to requirements and traceability. They are acceptance of safety requirements, communication, time frame for safety requirements, and traceability. When the system is a large-scale system, the requirements phase becomes more critical to handle. Islam and Storer [88] have also presented a study on agile in avionics systems, which is a large-scale system. They comment that there is a need for tools for the traceability of requirements that can work in an agile manner while keeping security measures intact. The traceability of all artifacts in the process is extremely important. Specifically, with changing requirements, it is even crucial to have clear traceability management in place. This is another extremely important aspect of SDLC that needs special attention. After requirements and traceability, we observe that there is a humongous discussion about testing techniques for SCS with agile methods. In testing, there is a roam to include agile concepts of automation to reduce time and perform it in iterations. Duffau et al. [76] have discussed the issues related to increased testing and specification activities when developing embedded systems with agile. To overcome these, they proposed a continuous integration ecosystem. The paper presented end-to-end product tests and automatic production of justification documents. They tested the approach in medical device production companies, and concluded by saying that more research is required in the area, as they can see potential benefits in terms of time and cost. Kasauli et al. [5] have presented a study about mapping agile on SCS. The research catered to studies from 2001 to 2017. They discussed the issues with six Swedish company representatives. They proposed an interesting approach to handling testing. They argue that in testing, the helpful approaches can be acceptance and unit testing, continuous building, and following coding standards. To ensure safety practices, a new solution is proposed to define and implement certain SOPs that must be followed by all teams on the project. Basically, the measures elaborated by standards are so well defined that there is not much room for Agile to fit in. However, in requirement engineering, it can be used as an iterative method as it can help elicit and understand requirements in a better way. Also, the concepts of agile can be helpful in testing.

IV&V techniques are the next most discussed part of SDLC for adopting agile in the development of SCSs [55]. Validation-verification techniques have emerged as the most critical area that should be handled in a way different from typical agile process models. This is one of the most critical activities in the life cycle of SCSs, but still, there is not much evidence of agile amalgamation in this area. There are different approaches proposed in the literature to bridge this gap; however still industry endorsement is required by the proposed approaches. Validation and verification are still an area that requires further research to mature processes for adapting agile. With incremental development, it is challenging and extremely important to have proper validation and verification techniques and requirement management. Arthur and Dabney [71] have presented a study about the compatibility of traditional IV&V techniques in a hybrid model for developing SCS. The authors further stated that there can be three categories of IV&V techniques in a hybrid model. First is early lifecycle IV&V techniques that are fully compatible with a hybrid model. Second is partially compatible IV&V techniques, which require modification. These are present in parts of requirements and testing, and the third is IV&V techniques that are not compatible with the hybrid process. They cannot divert from traditional approaches. They conclude by saying that since two of the three categories have the potential to be used with a hybrid model, there are good prospects of such models for SCS.

McBride and Lepmets [67] say that there are agile methods that can be used efficiently for the development of SCS, but the issue is that there are no formal validation methods in place for such models. Till now, people have relied on developers and not on formal methods to perform these tasks. In the same context, a framework is presented called CYNEFIN. The framework proposes to divide a system or situation into four categories and proposes a way to deal with it. The four categories are Complex, complicated, chaotic, and simple. They conclude by saying that verification and validation are the key areas for future research for the successful integration of agile with SCS. Another interesting approach is presented by Dabney and Arthur [84] have categorized validation and verification techniques into three categories for hybrid agile processes. First is early life cycle techniques, which can be completely adapted with agile. The second category is for those techniques that require amendments but can be tailored and used with agile, and the third category is noncompatible techniques. They further investigated 7 techniques in the last category, noncompatible techniques, and proposed alternative approaches that can be used to have similar results with agile.

Table 4 summarizes our analysis of characteristics of the most discussed phases of SDLC for including agile in SCSs.

Traceability is the most discussed practice of SDLC since it is extremely critical for the following reasons:

- It is linked to all phases of SDLC.
- Heavy documentation is required for complete traceability, which is in contrast to agile.
- Demanded by assessor and safety standards cannot be compromised to adjust to agile.
- Adds time, whereas agile talks about as quick as possible deliverable.

Requirement engineering is the second most discussed phase of SLDC. The following points of concern are reported in studies:

- Heavy documentation is required by standards, but agile does not support the idea.
- Well-defined requirements are needed by safety standards whereas agile talks about accepting changing requirements during SDLC.
- Requirements not only come from the client, but the assessor is also an important factor, hence making it more complex.
- Safety standards impose restrictions-requirements that must be met, in contrast to agile.

Testing is also among the most discussed areas, but relatively lesser touched as compared to traceability and requirements.

- Detailed testing of every step is required by safety standards, which adds time to the project, hence against agile principles.
- Multiple types of tests from unit level to integration level need to be performed at every step, also after any change or new addition.
- Increased time and cost with traditional testing, which is against agile.
- Upfront planning, including testing, is required, while agile has no concept of upfront planning of any activity.

Validation and validation in light of Agile is the most newly discussed area; it is extremely critical for SCS and hence needs to be handled with great caution. It is one of the hot topics of research in current times. The major prospects reported by studies are as follows:

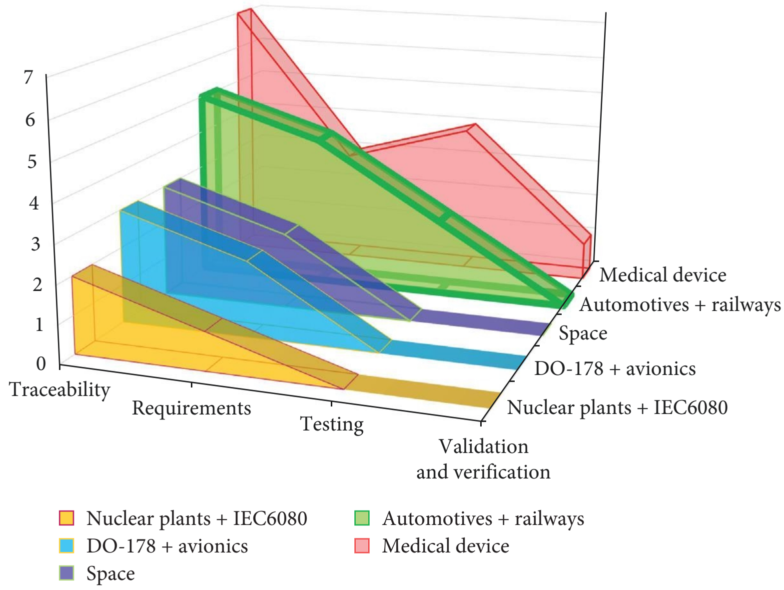

FIGURE 11: Overview of domains vs. phases of SDLC.

- Strict rules by safety standards for validation and verification hardly leave any room for a new approach.
- Required to be performed at every step of SDLC, hence adding more time to the timeline.
- Difficult to stay in line with traditional validation and verification techniques if one wants to accept change in requirements, a trait hugely supported by agile.
- Recently discussed area, currently hot topic, but not much proof of application in industry is available as yet.

For future work, there is a need for one concrete process model that can specifically address these concerns while finding a middle ground between agile and rules imposed by SCSs [45]. While doing so, the above-mentioned phases and practices are most critical to handle.

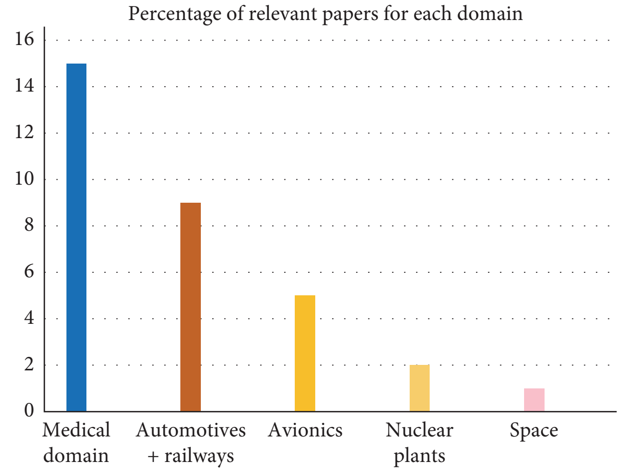

FIGURE 12: Domains of safety systems.

3.4. Domains of Safety Systems (RQ3). Here we try to answer our third research question, stated as "Which domains of safety systems report the highest adaption of agile for their systems?" Agile process models are widely adapted for the development of different software. However, their use for critical systems has seen a comparatively lower rise. The reason is the contrasting approach between agile and critical software development approaches [42]. However, there are some critical domains where the application of agile has seen more success and hence implementation as compared to other domains, as shown in Figures 11 and 12 and Table 7.

Medical devices have emerged as the most discussed domain for the development of SCSs with agile process models [43, 44, 50, 65]. In the medical domain, there are reported case studies of using agile process models for the development of device software. Hybrid process models have shown considerably good results in this case. Hybrid models refer to the combination of traditional approaches and agile approaches. The most used approaches are XP and Scrum, which are extended by including traditional models. The main problem is mapping agile principles to FDA regulations since almost all studies have discussed development in regulation with FDA standards. The major areas of concern for this domain are requirements, testing, and verification. In particular, [48] have outlined barriers to adopting agile for medical device software. The mentioned barriers are regulatory compliance, maintaining traceability, the process of managing multiple releases, lack of upfront planning, and lack of documentation. They also proposed ways of overcoming these five barriers. However, some studies have argued that it is the mindset of people that needs to change, and actually, there are no huge barriers. One such study is presented by McHugh et al. [58]. They performed a comparison between actual and perceived

Table 7: Domains of safety systems findings.

Domains	Size of software	System of systems	Complex	Detailed/diverse[a]	Areas of concern
Medical	Mostly small	Mostly no	No	No	Planning, requirement management
Automotive, railways	Huge	Yes	Yes	Yes	Testing, quality aspects, IV&V
Avionics	Huge	Yes	Yes	Yes	Change impact analysis
Nuclear plants, space	Huge	Yes	Yes	Yes	Documentation, proof of conformance

[a]Safety standards.

barriers and concluded by saying that actually, there are no external barriers, and it is a mindset and acceptance of a new approach that is needed. Rottier and Rodrigues [35] have presented another case study in a company called Cochlear, a medical device software development firm. They used Scrum instead of traditional approaches and tailored the approaches of requirement management, validation, quality metrics, and testing. They say that in these areas, they found a middle ground between agile and plan-driven approaches. The final findings showed that this approach is better than the traditional plan-driven approach for the development of medical device software.

The second most discussed domain is automotive and railways [46, 47]. Here we see that it is still in the emerging phase. We can find many proposed approaches, models, and practices deemed suitable for using agile for the development of these systems, but not much is available on the successful application of these strategies in industry [79]. The strict standards for development in this domain make it difficult to find companies that agree on trying new approaches. Still, we can safely say that this is the next generation for the development of these systems, and a lot of work is being done to find suitable agile approaches with promising results in the development of the automotive and railway domains. Here, testing and verification are the biggest challenges as they need detailed traceability of all artifacts, which increases documentation, time, and cost of a project. Wohlrab et al. [83] have presented an approach to discuss interfaces in the architecture of automotive systems. Authors identified three categories of interfaces, namely commodity interfaces, early-stage interfaces, and central vehicle interfaces. Commodity interfaces are mature interfaces that provide a platform for long-distance working teams, and changes take time for acceptance and approval. Early-stage interfaces are more easy and open to changes, although too many change requests are never desirable. Central vehicle interfaces are more critical to handle. They need more upfront planning, and ideally, there should not be any mid-project changes. Further, the paper suggests certain practices to manage interfaces. These practices are proposed on the basis of the experiences of interviewees and reasoning based on data. Myklebust et al. [60] have presented an intriguing approach. This paper on their previous work is based on using agile with standards like IEC 61508, IEC 60880, and IEC 61508, which resulted in a process called Safe Scrum. Here, they propose to use agile for standard EN 51208. Paper further pointed out the areas of concern for using agile for the development of systems with EN 51208, in which there must be a body to verify the process of an assessor, there must be a software quality assurance plan, verification and validation plan, and a software configuration plan. These can be made part of sprints of Safe Scrum, an agile methodology [82]. Testing by the developer is acceptable only if the assessor agrees to it before the start of the project. The responsibility to ensure that safety requirements are met must lie with RAMS after each sprint. Change impact analysis is an important activity for such systems [59]. This study has proposed a method to handle change impact analysis. They covered a wide range of topics from safety assessment for railway systems to software development for nuclear systems. In general, this work addressed SCSs. The main focus is on change impact analysis. The paper proposed an improved method for change impact analysis based on IEC 61508. In this paper, the particular focus and improvement are presented for IEC 60880 with the inclusion of agile.

Some studies suggest guidelines for better adoption of agile for automotive systems. One such work is done by Thawaba et al. [108]. The authors of this paper have developed guidelines that can help the SCSs development team to improve the processes. They discuss four major characteristics, which are development practices, failures, test techniques, and standards. The paper reviews some traditional and agile processes for SCSs development and provides insight into how to choose the most appropriate one. The failures are discussed as stories to learn from and avoid similar mistakes in the future. Third, they present testing techniques, both classical and agile ones, with their pros and cons to select the most appropriate one. Last, standards are discussed to make the whole approach in line with the required parameters of any standard. Here the authors have discussed 50128 in particular. They further discussed the four points in light of a case study of a railway system. In particular, they used agile techniques of PMBOK and SAFE and reported the possibility of positive outcomes by including these in the system.

We can also spot significant papers about avionics that have proposed agile for the development of such systems [39, 81]. Here, papers stress more on [77] including certain practices of agile in traditional way of development practices, like continuous integration, pair programing, refactoring, test-driven development are considered suitable agile practices for the development of artifacts under DO-178. There is also good scope for adapting agile in phases of requirements elicitation and verification. Coe and Kulick [52] have presented an agile-based model for development under standard DO-178, an avionic systems' standard. Authors have taken into account the incremental and iterative nature of agile and applied it specifically in the areas of requirements whole cycle from elicitation to verification. They further extended the model to perform testing of requirements in the same manner. However, after extracting safety requirements, they proposed making a detailed design with UML diagrams. After the detailed design and requirements are agreed upon by all stakeholders, testing can start in an agile manner. So they sliced the process and performed iterations to complete the design and requirements, and later they did the same for testing. They claim that this process satisfies all requirements of standard DO-178 and can be applied to another SCS with some amendments also. Some practices are considered more applicable in this domain from an agile set of practices, like pair programing, refactoring, test-driven development, and overall an iterative approach to developing the system. VanderLeest and Buter [38] report the use of such techniques and their benefits.

There are very few studies reported on Space and nuclear plant systems. The research in these areas is still in its initial phase for including agile for the development of such systems. Carpenter and Dagnino [57] have named Scrum and eXtreme Programing to be more promising than others; also, hybrid approaches are highly suitable for space systems. Stålhane et al. [51] have discussed important issues in the development of SCSs in this case, specifically for nuclear plants conforming to standard IEC 60880. The issues are planning, documentation, and proof of conformance.

Our analysis of studies revealed that there are certain aspects due to which medical device software has adapted agile more as compared to other domains. These aspects are as follows:

- Small size of software as compared to software of other safety domains.
- Less complex as compared to software of other safety domains.
- Most reported cases do not report software, which is a system of systems.
- Most studies report conformance to FDA guidelines. These guidelines remain unchanged for most kinds of medical device software, whereas in other safety domains, safety standards show vast diversity with respect to the product in hand.

The second most discussed domain is automotive and railways, followed by avionics and nuclear plants. Automotive and railways can also be termed as the most prevailing domains for including agile in development processes. We analyzed the same aspects for these domains as we did for the medical domain. The following aspects are common among all these domains:

- Huge size of software.
- Extremely complex.
- Most reported cases are system of systems.
- Diverse safety standards with respect to product in hand and safety level required.

Tables 6 and 7 present summary of this data.

4. Threats to Validity

As the world of critical systems is going through a rapid evolution, therefore, the data from very recent updates in industry may not be 100% in line with current systematic literature review. However, it will still be useful to see the trend of issues faced in the industry in adopting agile software development methodologies. Following, we present some major threats to validity and how they are addressed in our approach for this systematic literature review.

4.1. Internal Threats to Validity. In this systematic literature review, the internal threats to validity may arise from biases in study selection, data extraction, and interpretation. These threats include selection bias due to incomplete search strategies or restrictive inclusion/exclusion criteria that may omit relevant studies. To avoid researcher bias affecting how data is interpreted, two independent researchers have worked to assess the quality of studies. Hence, no subjective judgments are made when assessing the relevance or quality of primary studies. Consistent application of inclusion criteria across reviewers was ensured. All selected papers have gone through a rigorous process of scrutinizing by two independent people, and only relevant, good-quality studies were selected for analysis in this systematic literature review.

4.2. External Threats to Validity. In this systematic literature review, the external threats to validity concern the generalization of our findings beyond the included studies. These threats arise as the contexts, domains, or types of SCSs represented in the reviewed literature do not reflect the broader landscape of such systems, which may vary significantly across industries like aerospace, medical devices, or automotive. Additionally, the studies primarily focus on experimental or pilot projects rather than real-world, large-scale implementations; the applicability of conclusions to actual industrial settings may be limited. Variations in regulatory environments, organizational maturity, and team expertise across different contexts also affect how agile practices can be adopted safely and effectively. As such, caution has been exercised by both researchers when extrapolating results to all SCSs. However, the review explicitly acknowledges the contextual boundaries within which its conclusions are valid.

4.3. Credibility. In this systematic literature review, threats to credibility may arise from issues such as inconsistent data extraction, misinterpretation of study results, or insufficiently rigorous quality assessment of the included studies. This can happen if the primary sources lack methodological transparency or empirical rigor; then the credibility of the synthesized conclusions is weakened. In this work, we have used quantitative method to assess the quality of selected studies, which gives a clear idea in terms of rigor and relevance calculations (Section 2.4). To enhance credibility, we have employed well-defined protocols and multiple reviewers to cross-check key decisions and reduce subjectivity.

4.4. Limitations. We have considered Google Scholar, ACM, IEEE, Xplore, Scopus, and Web of Science for this literature review since many journals published in ScienceDirect (Elsevier) are indexed in Scopus, as Scopus is maintained by the same company (Elsevier), so ScienceDirect was not considered for this systematic literature review. Since the data is considered from a selected set of repositories only, this can be considered a limitation of our work, and this work can be enhanced by including data from other repositories. The development of SCSs is a highly competitive industry, and companies are never willing to share a large amount of data from their projects. Hence study is based on the available, documented data, experiences only. Validation of results presented in this systematic literature review needs to be done, preferably by industry personnel, to have better insight.

5. Conclusion

We have performed a systematic literature review of 72 studies relevant to SCSs and agile (details of selected studies is available in Appendix. We have used a systematic approach to provide answers to our research questions. We critically analyzed the selected studies and then extracted the information useful to answer the most relevant and important questions regarding the use of agile for SCSs. Our major findings are no agile model in its original form is deemed suitable for the development of SCSs; on the other hand, hybrid process models have been reported in studies to show great success. Secondly, a complete agile-based SDLC cannot be adapted for developing SCS; however, iterative and incremental nature of agile is useful for approaches to SCS development. Requirements and planning have to be done completely before the start of the project; the rest of the project can be built incrementally/iteratively. Then, for testing and verification, each project needs to seek an appropriate balance between agile and traditional approaches, as these have come out to be the most critical activities, especially in the presented scenario.

We have also highlighted the problems/challenges and experiences of using agile [86] according to particular domains [56, 62]. It is observed that agile has been applied the most in the medical domain, followed by automotive/railways, and then in avionics. We found some evidence for Space and nuclear programs as well. For the medical and automotive/railway domain, there is significant evidence to say that agile has shown promising results, with major fields of concern being requirements, testing, and verification. For avionics, also, the areas of major concern are the same however there is a growing trend of adapting certain agile processes, especially for continuous integration and traceability.

Overall, there is a huge trend of adapting agile for the development of SCSs with some amendments to typical agile process models. They have shown promising results in terms of lowering time, cost, and quality of products. We further plan to enhance this work further by developing an agile model suitable for the development of products under the safety standard ISO-26262.

For future work, we perceive automotive and railways to be the most emerging domains for including agile process models in the development phases. However, there is still a huge gap in terms of industrial implementation and formalization of process models that can fit best between agile and this domain. There is a lot yet to be explored and implemented.

Appendix A

Table A1: List of selected papers.

#	Year	Title of selected primary studies	Type venue	Reference
S1	2001	Using XP in a big process company: A report from the field.	Conference	[30]
S2	2001	Launching extreme programing at a process-intensive company.	Journal	[31]
S3	2004	Toward agile security assurance.	Workshop	[32]
S4	2006	Health modeling for agility in safety-critical systems development.	Conference	[33]
S5	2006	Extending XP practices to support security requirements engineering.	Workshop	[10]
S6	2007	Determining the applicability of agile practices to mission and life-critical systems.	Workshop	[24]
S7	2007	TXM: an agile HW/SW development methodology for building medical devices.	Journal	[34]
S8	2008	Agile development in a medical device company.	Conference	[35]
S9	2008	Comparing agile software processes based on the software development project requirements.	Conference	[36]
S10	2009	Adopting agile in an FDA-regulated environment.	Conference	[37]
S11	2009	Escape the waterfall: agile for aerospace.	Conference	[38]
S12	2010	Applying continuous integration principles in safety-critical airborne software.	Workshop	[39]
S13	2010	An iterative approach for the development of safety-critical software and safety arguments.	Conference	[23]
S14	2010	Investigating the capability of agile processes to support life-science regulations.	Conference	[40]
S15	2019	Lean/agile software development methodologies in regulated environments, state of the art.	Conference	[13]
S16	2011	Practitioners' perspectives on security in agile development.	Journal	[41]
S17	2011	Agile methods for open source safety-critical software.	Journal	[42]
S18	2011	The impact of regulatory compliance on agile software processes with a focus on the FDA guidelines for medical device software.	Conference	[43]
S19	2011	Med-trace: traceability assessment method for medical device software development.	Journal	[44]
S20	2012	Assessment of risks introduced to safety-critical software by agile practices, a software engineer's perspective.	Workshop	[45]
S21	2012	Agile practices in regulated railway software development.	Conference	[46]
S22	2012	The application of Safe Scrum to IEC 61,508 certifiable software.	Conference	[47]
S23	2012	Barriers to adopting agile practices when developing medical device software.	Workshop	[48]
S24	2012	Scrum goes formal: agile methods for safety-critical systems.	Conference	[49]
S25	2012	Medical device software traceability.	Workshop	[50]
S26	2013	An agile V-model for medical device software development to overcome the challenges with plan-driven software development life-cycles.	Conference	[20]
S27	2013	Scrum and IEC 60,880.	Conference	[51]
S28	2013	A model-based agile process for DO-178C certification.	Journal	[52]
S29	2013	Contextualizing agile software development.	Conference	[53]
S30	2013	Scaling agile methods to regulated environments: an industry case study.	Conference	[54]
S31	2013	A model-based framework for flexible safety-critical software development: a design study.	Conference	[55]
S32	2013	Challenges in flexible safety-critical software development: an industrial qualitative survey.	Conference	[56]
S33	2014	Effects of agile practices on predictors of system reliability.	Conference	[25]
S34	2014	Is agile too fragile for space-based systems engineering?	Journal	[57]
S35	2014	Adopting agile practices when developing software for use in the medical domain.	Conference	[58]
S36	2014	Change impact analysis in agile development.	Conference	[59]
S37	2015	Application of an agile development process for EN50128/railway conformant software.	Conference	[60]
S38	2015	Adopting agile practices when developing medical device software.	Journal	[61]
S39	2015	An insight into the difficulties of software development projects in the pharmaceutical industry.	Conference	[62]
S40	2015	Adaptive software development for developing safety critical software.	Conference	[63]
S41	2016	Challenges and opportunities in agile development in safety-critical systems: a survey.	Journal	[64]

Table A1: Continued.

#	Year	Title of selected primary studies	Type venue	Reference
S42	2016	Suitability of agile methods for safety-critical systems development: a survey of literature.	Journal	[2]
S43	2016	Agile medical device software development: introducing agile practices into MDevSPICE.	Workshop	[65]
S44	2016	Addressing the 4 + 1 software safety assurance principles within Scrum.	Conference	[66]
S45	2016	Quality assurance in agile safety-critical systems development.	Workshop	[67]
S46	2016	Toward integrating a system theoretic safety analysis in an agile development process.	Conference	[68]
S47	2019	AgileSafe: a method of introducing agile practices into safety-critical software development processes.	Journal	[6]
S48	2017	The dynamics of agile practices for safety-critical software development.	Conference	[69]
S49	2017	A study of quality assurance and unit verification methods in safety-critical environment.	Journal	[70]
S50	2017	Software engineering of safety-critical systems: themes from practitioners.	Conference	[4]
S51	2017	Applying standard independent verification and validation (IV&V) techniques within an agile framework: is there a compatibility issue?	Conference	[71]
S52	2017	Toward a model of emotional contagion influence on agile development for mission-critical systems.	Conference	[72]
S53	2017	An approach to support the specification of agile artifacts in the development of safety-critical systems.	Workshop	[73]
S54	2017	A study of safety documentation in a Scrum development process.	Workshop	[74]
S55	2017	Quality requirements in agile as a knowledge management problem: more than just-in-time.	Conference	[75]
S56	2017	Toward embedded system agile development challenging verification, validation and accreditation: application in a healthcare company.	Conference	[76]
S57	2017	An assessment of avionics software development practice: Justifications for an agile development process.	Conference	[77]
S58	2017	An exploratory study on applying a scrum development process for safety-critical systems.	Journal	[78]
S59	2018	Compliance of agilized (software) development processes with safety standards: a vision.	Conference	[21]
S60	2018	Waterfall is too slow, lets go agile: multidomain coupling for synthesizing automotive cyber-physical systems.	Journal	[79]
S61	2018	A hybrid assessment approach for medical device software development companies.	Journal	[80]
S62	2018	Safety-critical systems and agile development: a mapping study.	Conference	[5]
S63	2018	A conceptual model of agile software development in a safety-critical context: a systematic literature review.	Conference	[19]
S64	2018	How is agile development currently being used in regulated embedded domains?	Conference	[81]
S65	2018	Agile usage in embedded software development in safety-critical domain: a systematic review.	Journal	[22]
S66	2019	Safe Scrum: agile development of safety-critical software.	Journal	[82]
S67	2019	On interfaces to support agile architecting in automotive: an exploratory case study.	Journal	[83]
S68	2019	Applying standard independent verification and validation techniques within an agile framework: identifying and reconciling incompatibilities.	Journal	[84]
S69	2019	To what extent the medical device software regulations can be achieved with agile software development methods? XP–DSDM–Scrum.	Conference	[85]
S70	2019	Challenges of scaled agile for safety-critical systems.	Conference	[86]
S71	2020	Meshing agile and plan-driven development in safety-critical software: a case study.	Conference	[87]
S72	2020	A case study of agile software development for safety-critical systems projects.	Conference	[88]

Table A2: Study categories.

Study	Scrum	XP	ASD	DSDM	HibridS	Req	Test	IV&V	Trace	Medic	Auto	Space	Avion	Nuclea	Research method
S1	—	x	—	—	—	—	—	—	—	—	—	—	—	—	Case study
S2	—	x	—	—	—	—	—	—	—	—	—	—	—	—	Case study
S3	—	—	—	—	x	—	—	—	—	—	—	—	—	—	Example application
S4	—	—	—	—	—	—	—	—	—	x	—	—	—	—	Example application
S5	—	x	—	—	—	x	—	—	—	—	—	—	—	—	Example application
S6	—	—	—	—	—	x	x	—	—	—	—	—	—	—	Experience report
S7	x	x	—	—	—	—	x	—	—	—	—	—	—	—	Experiment (SW)
S8	x	—	—	—	—	—	x	—	—	x	—	—	—	—	Experience report
S9	x	x	x	—	—	—	—	—	—	—	—	—	—	—	Example application
S10	—	x	—	—	—	—	—	—	—	x	—	—	—	—	Experience report
S11	—	—	—	—	—	—	—	—	x	—	—	—	x	—	Experience report
S12	—	—	—	—	—	—	—	—	x	—	—	—	x	—	Example application
S13	—	—	—	—	—	—	—	—	x	—	—	—	—	—	Example application
S14	—	x	—	—	—	—	—	—	—	—	—	—	—	—	Example application
S15	—	—	—	—	—	x	—	—	—	—	—	—	—	—	Survey/interview
S16	—	—	—	—	—	x	—	—	x	—	—	—	—	—	Case study
S17	—	x	—	—	—	x	—	—	x	x	—	—	—	—	Example application
S18	—	—	—	—	—	—	—	—	x	—	—	—	—	—	Case study
S19	—	—	—	—	—	x	—	—	—	—	—	—	—	—	Case Study
S20	—	—	—	—	—	x	—	—	x	—	x	—	—	—	Example application
S21	x	—	—	—	—	x	—	—	x	—	x	—	—	—	Literature review
S22	—	—	—	—	—	—	—	—	x	x	—	—	—	—	Survey/interview
S23	x	—	—	—	—	—	—	—	—	—	—	—	—	—	Experience report
S24	—	—	—	—	—	—	—	—	x	x	—	—	—	—	Case study
S25	x	x	—	—	x	—	—	—	—	x	—	—	—	—	Mapping study
S26	x	—	—	—	—	—	—	—	x	—	—	—	—	x	Case study
S27	—	—	—	—	—	—	—	—	x	—	—	—	x	—	Example application
S28	—	—	—	—	—	—	—	—	x	—	—	—	—	—	Experience report
S29	x	—	—	—	—	—	—	—	x	—	—	—	—	—	Case study
S30	—	—	—	—	—	—	—	x	—	—	—	—	—	—	Example application
S31	—	—	—	—	—	—	—	—	x	—	—	—	—	—	Survey/interview
S32	—	—	—	—	—	—	—	—	x	—	—	—	—	—	Survey/interview
S33	x	x	—	—	x	—	—	—	—	—	—	x	—	—	Literature review
S34	—	—	—	—	—	—	—	—	x	x	—	—	—	—	Literature review
S35	x	—	—	—	—	x	—	—	x	—	x	—	—	x	Experiment (SW)
S36	—	—	—	—	—	—	—	—	x	—	x	—	—	—	Example application
S37	—	—	—	—	x	x	—	—	x	—	—	—	—	—	Action research
S38	—	—	—	—	—	—	—	—	x	x	—	—	—	—	Survey/interview
S39	—	—	x	—	—	—	—	—	—	—	—	—	—	—	Example application
S40	—	—	—	—	—	x	—	—	—	—	—	—	—	—	Survey/interview
S41	—	—	—	—	—	x	—	—	x	—	—	—	—	—	Literature review

Table A2: Continued.

Study	Scrum	XP	ASD	DSDM	HibridS	Req	Test	IV&V	Trace	Medic	Auto	Space	Avion	Nuclea	Research method
S42	—	—	—	—	—	x	x	—	x	x	—	—	—	—	Experience report
S43	x	—	—	—	—	—	—	—	x	—	—	—	—	—	Experience report
S44	—	—	—	—	—	—	—	x	—	—	—	—	—	—	Example application
S45	x	—	—	—	x	—	—	—	—	—	—	—	—	—	Example application
S46	x	—	—	—	—	—	—	—	—	—	—	—	—	—	Case study
S47	x	—	—	—	—	—	—	—	—	x	x	—	—	—	Case study
S48	—	—	—	—	—	—	—	—	x	—	—	—	—	—	Survey/interview
S49	—	—	—	—	x	—	—	x	—	—	—	—	—	—	Literature review
S50	—	—	—	—	—	—	—	—	x	—	—	—	—	—	Survey/interview
S51	—	—	—	—	—	x	—	—	x	—	—	—	—	—	Literature review
S52	—	—	—	—	—	—	—	—	x	—	—	—	—	—	Survey/interview
S53	—	—	—	—	—	x	—	—	—	—	—	—	—	—	Example application
S54	—	—	—	—	—	—	x	x	—	x	—	—	—	—	Example application
S55	x	—	—	—	—	x	—	—	—	—	—	—	x	—	Literature review
S56	—	—	—	—	x	—	—	—	—	—	—	—	—	—	Example application
S57	—	—	—	—	—	—	—	—	x	—	x	—	—	—	Case study
S58	—	—	—	—	x	—	—	—	x	x	—	—	—	—	Field study
S59	—	—	—	—	—	—	x	—	x	—	—	—	—	—	Mapping study
S60	—	—	—	—	—	—	—	—	x	—	—	—	—	—	Literature review
S61	x	—	—	—	—	—	—	—	—	—	—	—	—	—	Survey/interview
S62	—	—	—	—	—	—	—	—	x	—	—	—	—	—	Literature review
S63	x	—	—	—	—	x	x	—	—	—	—	—	—	—	Case study
S64	—	—	—	—	—	—	—	—	—	—	x	—	—	—	Case study
S65	—	—	—	—	x	—	—	x	x	—	—	—	—	—	Literature review
S66	x	x	—	x	—	—	—	—	x	—	—	—	—	—	Mapping study
S67	x	—	—	—	x	x	x	—	—	x	—	—	—	—	Case study
S68	x	—	—	—	—	x	—	—	—	—	—	—	x	—	Case study
S69	—	—	—	—	—	x	x	—	—	—	—	—	—	—	Example application
S70	—	—	—	—	—	—	—	—	—	x	—	—	—	—	Example application
S71	—	—	—	—	—	—	—	—	—	—	x	—	—	—	Example application
S72	—	—	—	—	—	—	x	—	—	—	x	—	—	—	Case study

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Disclosure

This work has been submitted as the final thesis of PHD [107] in the University of Florence, Italy.

Conflicts of Interest

The authors declare no conflicts of interest.

Funding

No funding was received for this manuscript.

References

[1] J. C. Knight, "Safety Critical Systems: Challenges and Directions', in *Proceedings of the 24th International Conference on Software Engineering* (2002): 547–550.

[2] M. W. Mwadulo, "Suitability of Agile Methods for Safety-Critical Systems Development: a Survey of Literature," *International Journal of Computer Applications Technology and Research* 5, no. 7 (2016): 465–471.

[3] F. Zimmermann, R. Eschbach, J. Kloos, and T. Bauer, "Risk-Based Statistical Testing: A Refinement-Based Approach to the Reliability Analysis of Safety-Critical Systems," in *12th European Workshop on Dependable Computing, EWDC 2009*., (2009)

[4] P. A. Laplante and J. F. DeFranco, "Software Engineering of Safety-Critical Systems: Themes From Practitioners," *IEEE Transactions on Reliability* 66, no. 3 (2017): 825–836.

[5] R. Kasauli, E. Knauss, B. Kanagwa, A. Nilsson, and G. Calikli, "Safety-Critical Systems and Agile Development: a Mapping Study', in *2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)* (2018): 470–477.

[6] K. Łukasiewicz and J. Górski, "Agilesafe-a Method of Introducing Agile Practices Into Safety-Critical Software Development Processes', in *Federated Conference on Computer Science and Information Systems (FedCSIS)* (2016): 1549–1552.

[7] K. Beck, M. Beedle, A. Van.Bennekum, A. Cockburn, W. Cunningham, M. Fowler, et al., "Manifesto for Agile Software Development," (2001).

[8] S. Stavru, "A Critical Examination of Recent Industrial Surveys on Agile Method Usage," *Journal of Systems and Software* 94 (2014): 87–97.

[9] I. Jacobson, "A Resounding" Yes" to Agile Processes-But Also to More," *Cutter IT Journal* 15, no. 1 (2002): 18–24.

[10] G. Boström, J. Wäyrynen, M. Bodén, K. Beznosov, and P. Kruchten, "Extending XP Practices to Support Security Requirements Engineering', in *Proceedings of the 2006 International Workshop on Software Engineering for Secure Systems* (2006): 11–18.

[11] D. Cohen, M. Lindvall, and P. Costa, "Agile Software Development," *Dacs Soaroar* 11 (2003): 2003.

[12] R. J. M. Job, M. Cailas, M. Liotine, J. Novotny, et al., "Evaluation of Critical Infrastructure Essential Businesses Amidst Covid-19 Using Network Optimization Models," *International Journal of Business Research and Management* 12, no. 5 (2021): 289–307.

[13] O. Cawley, X. Wang, and I. Richardson, "Lean/Agile Software Development Methodologies in Regulated Environments–State of the Art', in *International Conference on Lean Enterprise Software and Systems* (2010): 31–36.

[14] M. Holcombe, F. Ipate, and A. Grondoudis, "Complete Functional Testing of Safety Critical Systems," *IFAC Proceedings Volumes* 28, no. 25 (1995): 199–204.

[15] D. E. Strode, H. Sharp, L. Barroca, P. Gregory, and K. Taylor, "Tensions in Organizations Transforming to Agility," *IEEE Transactions on Engineering Management* 69, no. 6 (2022): 3572–3583.

[16] J. Perlak, "Selected Concepts of Leadership in Self-Organizing Teams', in *International Conference on Agile Software Development* (2024): 209–215.

[17] J. Chen, D. O. Neubaum, R. R. Reilly, and G. S. Lynn, "The Relationship Between Team Autonomy and New Product Development Performance Under Different Levels of Technological Turbulence," *Journal of Operations Management* 33-34, no. 1 (2015): 83–96.

[18] V. Stray, N. B. Moe, and R. Hoda, "Autonomous Agile Teams: Challenges and Future Directions for Research', in *Proceedings of the 19th International Conference on Agile Software Development: Companion* (2018): 1–5.

[19] L. T. Heeager and P. A. Nielsen, "A Conceptual Model of Agile Software Development in a Safety-Critical Context: A Systematic Literature Review," *Information and Software Technology* 103 (2018): 22–39.

[20] M. Mc.Hugh, O. Cawley, F. McCaffcry, I. Richardson, and X. Wang, "An Agile v-Model for Medical Device Software Development to Overcome the Challenges With Plan-Driven Software Development Lifecycles', in *2013 5th International Workshop on Software Engineering in Health Care (SEHC)*, (2013): 12–19.

[21] B. Gallina, F. U. Muram, and J. P. C. Ardila, "Compliance of Agilized (Software) Development Processes With Safety Standards: a Vision," in *Proceedings of the 19th International Conference on Agile Software Development: Companion* (2018): 1–6.

[22] S. Demissie, F. Keenan, Ö. Özcan.Top, and F. McCaffery, "Agile Usage in Embedded Software Development in Safety Critical Domain–a Systematic Review', in *International Conference on Software Process Improvement and Capability Determination*, (Springer, 2018): 316–326.

[23] X. Ge, R. F. Paige, and J. A. McDermid, "An Iterative Approach for Development of Safety-Critical Software and Safety Arguments', in *Agile Conference* (2010): 35–43.

[24] A. Sidky and J. Arthur, "Determining the Applicability of Agile Practices to Mission and Life-Critical Systems," in *31st IEEE Software Engineering Workshop* (2007): 3–12.

[25] L. S. M. Lago, M. F. D. Silva, and E. Spina, "Effects of Agile Practices on Predictors of System ReliabilityEffects of Agile Practices on Predictors of System Reliability," in *5th European Conference of Computer Science*., (2013)

[26] M. Templier and G. Paré, "A Framework for Guiding and Evaluating Literature Reviews," *Communications of the Association for Information Systems* 37, no. 1 (2015): 6.

[27] T. Dybå, B. Kitchenham, and M. Jørgensen, "Evidence-Based Software Engineering for Practitioners," *IEEE Software* 22, no. 1 (2005): 58–65.

[28] J. F. Wolfswinkel, E. Furtmueller, and C. P. M. Wilderom, "Using Grounded Theory as a Method for Rigorously Reviewing Literature," *European Journal of Information Systems* 22, no. 1 (2017): 45–55.

[29] B. Kitchenham, S. Charters, et al., "Guidelines for Performing Systematic Literature Reviews in Software Engineering Version 2.3," *Engineering* 45, no. 4ve (2007): 1051.

[30] J. Grenning, "Using Xp in a Big Process Company: A Report from the Field," in *XP Agile Universe*, (Citeseer, 2001).
[31] J. Grenning, "Launching Extreme Programming at a Process-Intensive Company," *IEEE Software* 18, no. 6 (2001): 27–33.
[32] K. Beznosov and P. Kruchten, "Towards Agile Security Assurance', in *Proceedings of the 2004 Workshop on New Security Paradigms* (2004): 47–54.
[33] Z. Stephenson, J. McDermid, and A. Ward, "Health Modelling for Agility in Safety-Critical Systems Development," in *2006 1st IET International Conference on System Safety* (2006): 260–265.
[34] L. Cordeiro, R. Barreto, R. Barcelos, M. Oliveira, V. Lucena, and P. Maciel, "TXM: An Agile HW/SW Development Methodology for Building Medical Devices," *ACM SIGSOFT Software Engineering Notes* 32, no. 6 (2007).
[35] P. A. Rottier and V. Rodrigues, "Agile Development in a Medical Device Company', in *Agile 2008 Conference* (2008): 218–223.
[36] M. Qasaimeh, H. Mehrfard, and A. Hamou.Lhadj, "Comparing Agile Software Processes Based on the Software Development Project Requirements', in *2008 International Conference on Computational Intelligence for Modelling Control & Automation* (2008): 49–54.
[37] R. Rasmussen, T. Hughes, J. Jenks, and J. Skach, "Adopting Agile in an FDA Regulated Environment', in *2009 Agile Conference* (2009): 151–155.
[38] S. H. VanderLeest and A. Buter, "Escape the Waterfall: Agile for Aerospace', in *2009 IEEE/AIAA 28th Digital Avionics Systems Conference*., (2009)
[39] A. B. Boralli and R. B. França, "Applying Continuous Integration Principles in Safetycritical Airborne Software," *Anais do* 5: 1–12.
[40] H. Mehrfard, H. Pirzadeh, and A. Hamou.Lhadj, "Investigating the Capability of Agile Processes to Support Life-Science Regulations: the Case of Xp and Fda Regulations with a Focus on Human Factor Requirements," in *Software Engineering Research, Management and Applications*, (Springer, 2010): 241–255.
[41] S. Bartsch, "Practitioners' Perspectives on Security in Agile Development', in *2011 Sixth International Conference on Availability, Reliability and Security* (2011): 479–484.
[42] K. Gary, A. Enquobahrie, L. Ibanez, P. Cheng, Z. Yaniv, K. Cleary, et al., "Agile Methods for Open Source Safety-Critical Software," *Software: Practice and Experience* 41, no. 9 (2011): 945–962.
[43] H. Mehrfard and A. Hamou.Lhadj, "The Impact of Regulatory Compliance on Agile Software Processes With a Focus on the FDA Guidelines for Medical Device Software," *International Journal of Information System Modeling and Design* 2, no. 2 (2011): 67–81.
[44] V. Casey and F. McCaffery, "Med-Trace: Traceability Assessment Method for Medical Device Software Development," in *European Systems and Software Process Improvement and Innovation Conference*., (2011)
[45] J. Górski and K. Łukasiewicz, "Assessment of Risks Introduced to Safety Critical Software by Agile Practices–a Software Engineer's Perspective," *Computer Science* 13, no. 4 (2012): 165.
[46] H. Jonsson, S. Larsson, and S. Punnekkat, "Agile Practices in Regulated Railway Software Development', in *23rd International Symposium on Software Reliability Engineering Workshops* (2012): 355–360.
[47] T. Stålhane, T. Myklebust, and G. Hanssen, "The Application of Safe Scrum to IEC 61508 Certifiable Software', in *11th International Probabilistic Safety Assessment and Management Conference and the Annual European Safety and Reliability Conference* (2012): 6052–6061.
[48] M. McHugh, F. McCaffery, and V. Casey, "Barriers to Adopting Agile Practices When Developing Medical Device Software', in *International Conference on Software Process Improvement and Capability Determination* (2012): 141–147.
[49] S. Wolff, "Scrum Goes Formal: Agile Methods for Safety-Critical Systems," in *2012 First International Workshop on Formal Methods in Software Engineering: Rigorous and Agile Approaches (Formsera)* (2012): 23–29.
[50] F. Mc.Caffery, V. Casey, M. Sivakumar, G. Coleman, P. Donnelly, and J. Burton, "Medical Device Software Traceability," in *Software and Systems Traceability*, (Springer, 2012): 321–339.
[51] T. Stålhane, V. Katta, and T. Myklebust, "Scrum and IEC 60880', in *Enlarged Halden Reactor Project meeting*., (2013)
[52] D. J. Coe and J. H. Kulick, "A Model-Based Agile Process for do-178c Certification', in *Proceedings of the International Conference on Software Engineering Research and Practice (SERP)*., (2013)
[53] P. Kruchten, "Contextualizing Agile Software Development," *Journal of Software: Evolution and Process* 25, no. 4 (2013): 351–361.
[54] B. Fitzgerald, K. J. Stol, R. O'Sullivan, and D. O'Brien, "Scaling Agile Methods to Regulated Environments: An Industry Case Study', in *35th International Conference on Software Engineering (ICSE)* (2013): 863–872.
[55] J. P. Notander, P. Runeson, and M. Höst, "A Model-Based Framework for Flexible Safety-Critical Software Development: a Design Study', in *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, (2013: 1137–1144.
[56] J. P. Notander, M. Höst, and P. Runeson, "Challenges in Flexible Safety-Critical Software Development–an Industrial Qualitative Survey', in *International Conference on Product Focused Software Process Improvement* (2013): 283–297.
[57] S. E. Carpenter and A. Dagnino, "Is Agile Too Fragile for Space-Based Systems Engineering?', in *2014 IEEE International Conference on Space Mission Challenges for Information Technology* (2014): 38–45.
[58] M. McHugh, F. McCaffery, and V. Casey, "Adopting Agile Practices When Developing Software for Use in the Medical Domain," *Journal of Software: Evolution and Process* 26, no. 5 (2014): 504–512.
[59] T. Stålhane, V. Katta, and T. Myklebust, *Change Impact Analysis in Agile Development'* (EHPG Røros, 2014).
[60] T. Myklebust, T. Stålhane, and N. Lyngby, "Application of an Agile Development Process for en50128/Railway Con-Formant Software," in *Esrel*., (2015)
[61] M. McHugh, F. McCaffery, and G. Coady, "Adopting Agile Practices When Developing Medical Device Software," *Journal of Computer Engineering & Information Technology* 9307 (2015).
[62] A. Hajou, R. Batenburg, and S. Jansen, "An Insight Into the Difficulties of Software Development Projects in the Pharmaceutical Industry," *Lecture Notes on Software Engineering* 3, no. 4 (2015): 267–274.
[63] A. A. Abdelaziz, Y. El.Tahir, and R. Osman, "Adaptive Software Development for Developing Safety Critical Software', in *2015 International Conference on Computing, Control, Networking, Electronics and Embedded Systems Engineering (ICCNEEE)* (2015): 41–46.

[64] O. Doss and T. P. Kelly, "Challenges and Opportunities in Agile Development in Safety Critical Systems," *ACM SIGSOFT Software Engineering Notes* 41, no. 2 (2016): 30-31.

[65] F. McCaffery, M. Lepmets, K. Trektere, O. Ozcantop, and M. Pikkarainen, "Agile Medical Device Software Development: Introducing Agile Practices Into Mdevspice," *International Journal On Advances in Life Sciences* 8, no. 1 & 2 (2016): 133–142.

[66] O. Doss and T. Kelly, "Addressing the 4+ 1 Software Safety Assurance Principles Within Scrum," in *Proceedings of the Scientific Workshop Proceedings of XP2016*, 2016: 1–5.

[67] T. McBride and M. Lepmets, "Quality Assurance in Agile Safety-Critical Systems Development', in *2016 10th International Conference on the Quality of Information and Communications Technology (QUATIC)* (2016): 44–51.

[68] Y. Wang and S. Wagner, "Toward Integrating a System Theoretic Safety Analysis in an Agile Development Process." *Software Engineering* 2016): 156–159.

[69] P. A. Nielsen and L. T. Heeager, "The Dynamics of Agile Practices for Safety-Critical Software Development', in *Proceedings of the XP2017 Scientific Workshops* (2017): 1–6.

[70] M. Taliga, "A Study of Quality Assurance and Unit Verification Methods in Safety Critical Environment," *International Journal of Electrical and Information Engineering* 9, no. 11 (2017): 2407–2411.

[71] J. D. Arthur and J. B. Dabney, "Applying Standard Independent Verification and Validation (IV&V) Techniques Within an Agile Framework: Is There a Compatibility Issue?', in *2017 Annual IEEE International Systems Conference (SysCon)* (2017): 1–5.

[72] A. Alhubaishy and L. Benedicenti, "Toward a Model of Emotional Contagion Influence on Agile Development for Mission Critical Systems', in *2017 International Conference on High Performance Computing & Simulation (HPCS)* (2017): 541–544.

[73] A. I. M. Leite, "An Approach to Support the Specification of Agile Artifacts in the Development of Safety-Critical Systems', in *IEEE 25th International Requirements Engineering Conference (RE)* (2017): 526–531.

[74] Y. Wang, I. Bogicevic, and S. Wagner, "A Study of Safety Documentation in a Scrum Development Process," in *Proceedings of the XP2017 Scientific Workshops* (2017): 1–5.

[75] E. Knauss, G. Liebel, K. Schneider, J. Horkoff, and R. Kasauli, "Quality Requirements in Agile as a Knowledge Management Problem: More Than Just-in-Time', in *IEEE 25th International Requirements Engineering Conference Workshops (REW)* (2017): 427–430.

[76] C. Duffau, B. Grabiec, and M. Blay.Fornarino, "Towards Embedded System Agile Development Challenging Verification, Validation and Accreditation: Application in a Healthcare Company', in *2017 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)* (2017): 82–85.

[77] G. K. Hanssen, G. Wedzinga, and M. Stuip, "An Assessment of Avionics Software Development Practice: Justifications for an Agile Development Process', in *International Conference on Agile Software Development* (2017): 217–231.

[78] Y. Wang, J. Ramadani, and S. Wagner, "An Exploratory Study on Applying a Scrum Development Process for Safety-Critical Systems', in *International Conference on Product-Focused Software Process Improvement* (2017): 324–340.

[79] D. Roy, M. Balszun, T. Heurung, S. Chakraborty, and A. Naik, "Waterfall is Too Slow, Let's Go Agile: Multi-Domain Coupling for Synthesizing Automotive Cyber-Physical Systems', in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)* (2018): 1–7.

[80] Ö. Özcan.Top and F. McCaffery, "A Hybrid Assessment Approach for Medical Device Software Development Companies," *Journal of Software: Evolution and Process* 30, no. 7 (2018): e1929.

[81] P. Diebold and S. Theobald, "How Is Agile Development Currently Being Used in Regulated Embedded Domains?" *Journal of Software: Evolution and Process* 30, no. 8 (2018): e1935.

[82] G. K. Hanssen, T. Stålhane, and T. Myklebust, *SafeScrum-Agile Development of Safety-Critical Software* (Springer, 2018).

[83] R. Wohlrab, P. Pelliccione, E. Knauss, and R. Heldal, "On Interfaces to Support Agile Architecting in Automotive: An Exploratory Case Study', in *2019 IEEE International Conference on Software Architecture (ICSA)* (2019): 161–170.

[84] J. B. Dabney and J. D. Arthur, "Applying Standard Independent Verification and Validation Techniques Within an Agile Framework: Identifying and Reconciling Incompatibilities," *Systems Engineering* 22, no. 4 (2019): 348–360.

[85] Ö. Özcan.Top and F. McCaffery, "To What Extent the Medical Device Software Regulations Can Be Achieved With Agile Software Development Methods? Xp–dsdm–scrum." *Journal of Supercomputing* 75 (2019): 8.

[86] J. P. Steghöfer, E. Knauss, J. Horkoff, and R. Wohlrab, "Challenges of Scaled Agile for Safety-Critical Systems," in *International Conference on Product-Focused Software Process Improvement*, (Springer, 2019): 350–366.

[87] L. T. Heeager and P. A. Nielsen, "Meshing Agile and Plan-Driven Development in Safety-Critical Software: A Case Study," *Empirical Software Engineering* 25, no. 2 (2020): 1035–1062.

[88] G. Islam and T. Storer, "A Case Study of Agile Software Development for Safety-Critical Systems Projects," *Reliability Engineering & System Safety* 200 (2020): 106954.

[89] S. Jalali and C. Wohlin, "Systematic Literature Studies: Database Searches vs. Backward Snowballing," in *Proceedings of the ACM-IEEE International Symposium on Empirical Software Engineering and Measurement* (2012): 29–38.

[90] W. Bandara, E. Furtmueller, E. Gorbacheva, S. Miskon, and J. Beekhuyzen, "Achieving Rigor in Literature Reviews: Insights From Qualitative Data Analysis and Tool-Support," *Communications of the Association for Information Systems* 37, no. 1 (2015): 8.

[91] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, "Lessons From Applying the Systematic Literature Review Process Within the Software Engineering Domain," *Journal of Systems and Software* 80, no. 4 (2007): 571–583.

[92] H. Anwar and D. Pfahl, "Towards Greener Software Engineering Using Software Analytics: A Systematic Mapping', in *2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA)* (2017): 157–166.

[93] M. Ivarsson and T. Gorschek, "A Method for Evaluating Rigor and Industrial Relevance of Technology Evaluations," *Empirical Software Engineering* 16, no. 3 (2011): 365–395.

[94] M. G. Kendall, *Rank Correlation Methods* (Oxford University Press, 1948).

[95] P. Legendre, "Species Associations: The Kendall Coefficient of Concordance Revisited," *Journal of Agricultural, Biological, and Environmental Statistics* 10, no. 2 (2005): 226–245.

[96] M. Marozzi, "Testing for Concordance Between Several Criteria," *Journal of Statistical Computation and Simulation* 84, no. 9 (2013): 1843–1850.

[97] E. W. Fager, "Determination and Analysis of Recurrent Groups," *Ecology* 38, no. 4 (1957): 586–595.

[98] A. Strauss and J. Corbin, *Basics of Qualitative Research* (Sage Publications, 1990).
[99] J. Saldana, *The Coding Manual for Qualitative Researchers* (SAGE, 2013).
[100] S. A. Mohammadi and B. M. Prasanna, "Analysis of Genetic Diversity in Crop Plants–Salient Statistical Tools and Considerations," *Crop Science* 43, no. 4 (2003): 1235–1248.
[101] Z. Ali and S. B. Bhaskar, "Basic Statistical Tools in Research and Data Analysis," *Indian Journal of Anaesthesia* 60, no. 9 (2016): 662.
[102] J. Chambers, *Software for Data Analysis: Programming With R* (Springer Science & Business Media, 2008).
[103] C. Wohlin, M. Höst, and K. Henningsson, "Empirical Research Methods in Software Engineering," *Empirical Methods and Studies in Software Engineering* 2765 (2003): 7–23.
[104] K. Petersen and C. Gencel, "Worldviews, Research Methods, and Their Relationship to Validity in Empirical Software Engineering Research," in *2013 Joint Conference of the 23rd International Workshop on Software Measurement and the 8th International Conference on Software Process and Product Measurement* (2013): 81–89.
[105] S. Easterbrook, J. Singer, M. A. Storey, and D. Damian, "Selecting Empirical Methods for Software Engineering Research," *Guide to Advanced Empirical Software Engineering* (2008): 285–311.
[106] I. de Sousa.Santos, R. M. de Castro.Andrade, L. S. Rocha, S. Matalonga, K. M. de Oliveira, and G. H. Travassos, "Test Case Design for Context-Aware Applications: Are We There Yet?" *Information and Software Technology* 88 (2017): 1–16.
[107] H. Maqsood, et al., *Agile Software Development Methodologies for Safety Critical Systems'* (University of Florence, 2022).
[108] A. A. Thawaba, A. A. Ramli, M. F. M. Fudzee, and J. Wadata, "Characteristics for Performance Optimization of Safety-Critical System Development (SCSD)," *Journal of Advanced Computational Intelligence and Intelligent Informatics* 24, no. 2 (2020): 232–242.