CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Automotive SPICE compliance in an Agile Software Development Process

A case study on optimization of the work products

Bachelor of Science Thesis in Software Engineering and Management

Nuria Cara Navas

Department of Computer Science and Engineering
UNIVERSITY OF GOTHENBURG
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2020

CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Automotive SPICE compliance in an agile software development process
A case study on optimization of the work products
The goal of this Bachelor thesis is to find a way to combine both A-SPICE and agile methodologies in a large automotive company with cross-functional teams in order to produce and record all the necessary documentation, making sure that quickly produced deliverables comply with A-SPICE.

Supervisor: Jan-Philipp Steghöfer
Examiner: Richard Berntsson Svensson

University of Gothenburg
Chalmers University of Technology
Department of Computer Science and Engineering
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Department of Computer Science and Engineering
UNIVERSITY OF GOTHENBURG
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2020

Automotive SPICE Compliance in an Agile Software Development Process

A case study on optimization of the work products

Nuria Cara Navas
DIT837 Bachelor Thesis in Software
Engineering and Management
Gothenburg University
Gothenburg, Sweden
nuriacnavas@gmail.com

***Abstract*— Automotive SPICE is used in the automotive industry to comply with functional safety. This guideline uses the waterfall/v-cycle model for software development. However, automotive companies are shifting their way of working into iterative software production using agile methodologies. One of the remaining challenges that persists when combining these two development methodologies is ensuring the critical-safety of the work products with their corresponding documentation, as the Scaled Agile Frameworks (SAFe) used in large companies do not consider this. Therefore, this study takes a deeper look into the software processes of the Aurora team at Volvo Cars ART steering department to propose a new document management strategy that allows the combination of agile software development while still complying with A-SPICE. To achieve this, active observations and interviews were conducted for data collection. The creation of the adapted document management strategy shows that it is possible to adopt agile development practices in large-scale automotive companies and still comply with A-SPICE. Further studies should be conducted to adapt the document management strategy to other automotive companies, as well as to evaluate the short and long-term effects that the suggested document management strategy has on the software development process and the team.**

Keywords—ASPICE, Agile, Software process improvement, document management strategy, Case Study, IEEE.

I. INTRODUCTION

The usage of electronic systems in the automotive industry has been continuously increasing in the past few years, as most of the functions of a modern car are now controlled by complex software. This gradual increase in software complexity also contributes to a gradual increase of the project complexity in the automotive industry. As more complex projects are being executed in parallel with shorter model life cycles and customers' expectations are getting more demanding in terms of comfort and safety, a stable development process needs to be followed when delivering software products in the automotive domain. Therefore, the VDA (Verband der Automobilindustrie – Association Of Automotive Industry) agreed to set Automotive SPICE® [4] as the standard process model. This model provides guidelines for defining, managing, and improving the system and software development process, specifically meant for the automotive industry.

Nowadays, more and more development companies are shifting their way of working into a more agile iterative software development way of working. By adopting this methodology, the developer teams aim to benefit from faster delivery, greater quality of the deliverables, and greater aptitude to respond to change. Besides, by working agile, the requirements and solutions evolve through collaboration between the organizations, the cross-functional teams, and their customer(s). [19]

In large automotive companies, many teams work together to develop and deliver a product. Thus, the coordination and combination of these different engineering disciplines for achieving the product's final delivery can be troublesome. [24] So, to manage this complexity, many automotive companies are adopting scaled agile frameworks such as SAFe [16], among others, to speed up the development of the product and manage the different teams and artifacts' exchange. However, these frameworks do not consider either risk management, safety analysis, or generating the corresponding documentation for the creation of safety-critical systems, and therefore, do not provide Automotive SPICE (A-SPICE) compliance. [24]

One of the many remaining challenges in combining agile and A-SPICE methodologies consists of adapting Scaled agile frameworks to fulfill both the need for agility and produce the necessary documentation (work products) for complying with A-SPICE beyond individual teams. [24,25]

This still remains a challenge, as agile practices focus more on faster delivery of the product and cross-functional team collaboration than documentation. Contrary to A-SPICE where documents provide a basis for certification for product release and are necessary to ensure functional safety of the products. A large number of organizations involved in producing vehicles, as well as many physical locations and disciplines with specifically established toolchains for the automotive domain, create this particular problem. [7,25]

To the best of our knowledge, no study has attempted to make such an adaptation for larger companies, specifically in the automotive domain.

Therefore, the purpose of this study is to find a way to combine both A-SPICE and agile methodologies in order to produce and record all the necessary documentation, making sure that quickly produced deliverables comply with safety standards. To achieve this, it is necessary to know beforehand which information shall be produced and recorded according to A-SPICE, how that information is going to be recorded and produced, and when in an agile iterative development process that information is produced and recorded. This is why the research questions are formulated as follows:

RQ1. Which information needs to be produced to comply with automotive SPICE while following an agile development process in the automotive industry?

RQ2. How is the information produced in an agile development process going to be collected and documented to assure A-SPICE compliance?

RQ3. When during the different stages of an agile development process does the information need to be produced, collected, and documented?

As the goal is to provide an example of the SAFe adaptation with A-SPICE work products, a document management strategy needs to be produced in order to answer RQ2 and RQ3. Therefore, this document management strategy acts as a recommendation on how to improve the previously mentioned issue found when combining SAFe and A-SPICE.

The creation of this document management strategy is done in collaboration with Volvo Cars' ART steering department, concretely with the Aurora team.

The structure of this paper provides an introduction to the topic, background information, and case-related description in Section II, followed by the related literature presented in Section IV, and a description of the research questions, data collection methods and data analysis strategies in Section V. Section VI presents the results and Section VII discusses the major findings from the results according to the research questions. The conclusion of the study is presented in Section VIII.

II. Background

A. Automotive SPICE

Automotive SPICE defines the base practices and processes needed to conform to functional safety. It serves as a guideline for ensuring a mature, systematic, and well-documented system and software development.

This model follows a V-model representation of the development process and it is considered by Volvo Cars to be an extension of the Waterfall development model. This means that each development phase will not start until the previous phase has been completed (see Figure 7 in the Appendix). The relation between the A-SPICE V-model and the Waterfall model can be interpreted differently in other companies.

The V-model splits the software development process into two sides of the main process. The left side of the V-model is about software requirements analysis, design, and unit construction while the right-side concentrates on the main verification and validation of the software where the testing is associated with each corresponding development stage. [1]

Automotive SPICE is also used for capability determination. The goal of establishing a capability level is to set process attributes, which are features of a process that can be evaluated on a scale of the achievement of a process, providing a measure of the capability of the process. The capability levels characterized by the set of attribute(s) work together to provide an enhancement in the capability to perform a process. [4, 15, 18].

According to ISO/IEC 33020, there are six capability levels incorporating nine process attributes being Level 0 "incomplete process" and Level 5 "Innovating process" respectively. Therefore, certain strategies need to be created, managed, applied, and maintained in order to achieve the highest capability level, where the created processes are being continually improved to respond to the organizational change. [1, 4, 15]

Another A-SPICE characteristic of base practices is that most of its work products, such as SWE.1-6, SYS.2-5, and SUP.10, have to be traceable both forward, from the requirement to test case, and backward, from test case to requirement. The bi-directional traceability is recorded in a single document, called Requirement Traceability Matrix. It makes sure that all the specified requirements have their corresponding test cases and vice versa. This is needed for embedded systems that combine hardware and software, to ensure safety and quality of the products. [4, 26]

B. Agile framework SAFe

SAFe is an iterative approach of developing software and project management methodology that helps teams deliver value to their customers fast and incrementally. Requirements, plans, and results are evaluated continuously (Continuous increment), so the teams can respond to change quickly. This methodology focuses on providing cross-functional teams with support and continuous communication and feedback, minimizing the number of artifacts in the development process reducing, therefore, documentation for software development and design. [27]

Full SAFe framework provided by Scaled Agile is the one being adopted by this specific automotive company, as it is specifically created for larger organizations. It combines lean product development principles and agile principles. [6, 16] This framework used to implement agility, can be adapted according to the company's needs and situation (see Figure 8 in the Appendix for an overview of SAFe's framework).

The different levels that are included in full SAFe are team level, program level, large solution level, and portfolio level.

1) Team level

Operates like Scrum and/or Kanban. The development teams consist of five to nine developers and testers who work cross-functionally to deliver the expected products at the end of a 2 weeks sprint-interval. The product owner (PO) is the person in charge of the sprint backlog and the one that conducts sprint plannings, program increments, backlog refinements, and progress reports or demos. It also interacts with the team in the daily meetings to discuss progress and on sprint retrospectives on how to improve the upcoming sprints. All this is guided by the scrum master who makes sure that the team works effectively without restrictions. [6, 16]

2) Program Level

This level is similar to the team level but scaled up to five to twelve teams who work together in delivering fully working solutions to the product. This team collaboration into the delivery of scheduled features for planning and delivery of the product is called an Agile Train Release (ART), due to its continuous delivery practices. [16]

A Program Increment (PI) is to an ART, as an iteration is to the Agile team. Every PI consists of 6 sprints. During a PI, the different teams build and validate a full system increment, demonstrating value, and getting fast feedback. Besides, they plan together with the ART's next increment work. [16]

In this level, the product management acts similar to the PO on the team level, as it determines what should be delivered to each PI and the content of the program backlog.

The release train engineer acts as the scrum master for the ART, making sure that everything runs according to plans.

Each PI planning starts with a planning meeting with all the teams, where they discuss the *features* to be completed at the end of the PI.

3) Large solution level

On this level, the content of the backlog is called *capability*, which consists of several *features.*

Here, the solution management person, who is the one that has the highest authority, works with the Solution Train engineer to make sure that the right architecture is being used in the ARTs. [16]

4) Portfolio level

The Lean Portfolio Management (LPM) is the group responsible for support and budget allocation for investments and resources. The product backlog contains *epics*, checked by the ARTs product management to address them during each PI.

III. Case description

In this section, I will explain the problem that the Aurora team currently faces when using both agile and A-SPICE.

The in-house software development team of the ART steer department at Volvo Cars, called *Aurora*, has recently shifted their way of working into a more agile-oriented software development process. By working agile, the team aims to benefit from shorter delivery times and respond faster to change, as well as improve customer collaboration and produce better quality software [3]. By acting as an in-house supplier for an automotive company, they have to make sure that their practices comply with A-SPICE, in order to ensure a mature, systematic, and well-documented software development system. For this, a capability of at least level 3 has to be achieved.

The agile methodologies and practices that the in house-team is currently using are Scrum in automotive (see Figure 9 in the Appendix) [14], continuous integration (CI), and test-driven development (TDD). Some of these agile practices are included at team level in the company's adopted framework called "Scaled Agile Framework" or SAFe [14,16]. The team starts the two weeks sprint with sprint planning, followed by a development phase and a testing phase. At the end of the sprint, activities such as sprint demos and sprint retrospectives are held to get feedback and show progress. The backlog refinement will be held in the middle of the second sprint to keep the sprint backlog updated.

The Aurora software development team has realized, after working over a year using the SAFe framework, that it is hard to adapt A-SPICE into their agile methodologies in order to produce the necessary documentation to comply with A-SPICE safety standards. They lack a document management strategy that serves as a guideline to the adaptation for both software development methodologies. Without the necessary strategy that maps all A-SPICE work products to their agile practices, providing bi-directional traceability required by most of the A-SPICE practices, becomes a hazardous matter, especially when working in a cross-functional team environment. The reason for this is when in a large automotive company that uses an agile iterative development process, the documentation has to be modified often by the team members involved in the changes. Besides, those changes in the documentation might not be perceived by providers and other team members.

To maintain this bi-directional traceability, A-SPICE requests that the software development team fills out a requirement traceability matrix document among others. [4] It records which requirement is tested by which test, establishing a trace link between the requirement and the test. This document needs to be filled by the person responsible for such a task, which, as mentioned before, becomes hard to track when the changes were performed and if the document is up to date with the last information.

IV. Related literature

In this section, we present the literature related to our study: document management strategy while combining both agile software methodologies and A-SPICE.

There are a limited number of studies that investigate the document management strategy while combining A-SPICE and agile methodologies. Existing studies about agile and A-SPICE focus mostly on software traceability issues in the automotive domain and on identifying their challenges and solutions. Contrarily, other studies tend to focus on how agile practices support A-SPICE compliance, but they do not focus on large-scale agile frameworks.

Diebold et al. [8] provide an evaluation of how agile practices support automotive SPICE requirements based on literature reviews and expert opinion, as well as including 722 mappings of agile practices, including Scrum, and of Automotive SPICE requirements. They found that 103 of 155 agile practices covered 173 out of 185 of A-SPICE requirements, meaning that most of the A-SPICE base practices were supported (96%) while work products (87%) were less supported. They concluded that companies in the automotive industry can benefit from an agile development process without compromising A-SPICE and functional safety. Similar studies to Diebold et al. support these findings [10, 11]. Only one study conducted by Kähkönen and Abrahamsson [12] contradicted the findings in Diebold et al., stating that the two methodologies contradict each other and can therefore not be combined.

Regarding the challenges with the requirements traceability matrix, the study by Maro et al. [7], provided a combination of an exhaustive literature review and a case study at an automotive company about the challenges and solutions experienced in practice for software traceability in the automotive domain.

They identified 17 challenges in the case study of which six remained unsolved. Some of the most reported ones, by both the literature review and case study, were manual link creation and maintenance. The solutions that the authors proposed to these challenges were to implement the use of an integrating tool platform and automatization of the documentation. Link creations were also perceived as an overhead by both the case study at the company and by the literature review, as links need to be created manually by the developers. The solution to this challenge was once more to

automate the documentation and use tools that provide quick navigation from one artifact to another. Visualization techniques were also proposed by Amalfitano et al [9] as a solution to this challenge.

Similar challenges to the ones found in the literature were identified by the Aurora team at the ART steer department. Manually creating traceability links and maintaining them is found to remain a tedious task for the team, and a partial automatization solution of the produced documentation shall be provided.

In contrast to the other literature found, only three articles were found to be similar to the purpose of this research where one of them was the previously mentioned study from Diebold et al. [8]. Another one is the study made by Hantke [21]. He provided a coverage matrix of Scrum practices and SPICE base practices for two A-SPICE processes software testing and software requirement analysis. He found that most of the engineering group processes ("ENG" in A-SPICE [4]) are carried out on project-specific tasks in the sprints by the team, based on their specific business requirements.

However, the most relevant article found for this specific research was a case study by Komiyama et al. [22]. In their article, they faced and resolved issues related to process improvement, such as integrating scrum and A-SPICE and showing visually how and when the practices have to be applied and adjusted during the projects. They clarified the issues in the application of A-SPICE and agile software development, described an adapted strategy and approach to resolving the identified issues, and provided practical examples of the implementations considering agile and A-SPICE compliance. Nevertheless, their approach only works for small companies that were not using scrum before.

Therefore, a similar approach to Komiyama et al. will be taken in this study, but it will focus on a more practical solution that suits the company's needs in the adaptation of both agile and automotive SPICE methodologies.

V. Research Methodology

The research methodology chosen for this study is case study. [13] This is thought to be suitable for this study as the research is of a contemporary case in its natural setting [5]. Therefore, there is a collaboration between the researcher and the development team Aurora, to develop a solution for the diagnosed problem. More specifically, as this study is trying to improve a certain aspect of a studied phenomenon, such as finding a way to document the information produced, this research can be considered an improving case study. [13, 17]

A. Research Questions

In order to answer the research questions, we need to propose, diagnose, and investigate what data is currently missing in the team's processes. To achieve this, we need to know which information should be collected and produced before creating the document management strategy; how they are going to collect that information, and when it is going to be produced in an agile development process with an A-SPICE compliance.

Therefore, the research questions that this case study intends to answer are the following:

RQ1. Which information needs to be produced to comply with automotive SPICE while following an agile development process in the automotive industry?

RQ2. How is the information produced in an agile development process going to be collected and documented to assure A-SPICE compliance?

RQ3. When during the different stages of an agile development process does the information need to be produced, collected, and documented?

In addition to the design of the document management strategy process, observations of the working process will be done on-site at the ART steering department at Volvo Cars. The purpose of these observations is to answer the previously mentioned research questions and aid in the creation of the new document management process in order to see if the new strategy can fit the needs of the organization and the standards it follows.

B. Research timeline

The research timeline consists of a cycle of five main phases and one initial phase. The purpose of the initial phase is to get a better understanding of the information provided in the A-SPICE gap analysis by the team and its missing agile development processes. The other phases involve the steps needed to create the adapted documentation strategy to fulfill the team's needs.

Figure 1 shows the cycle of the artifact's creation process.

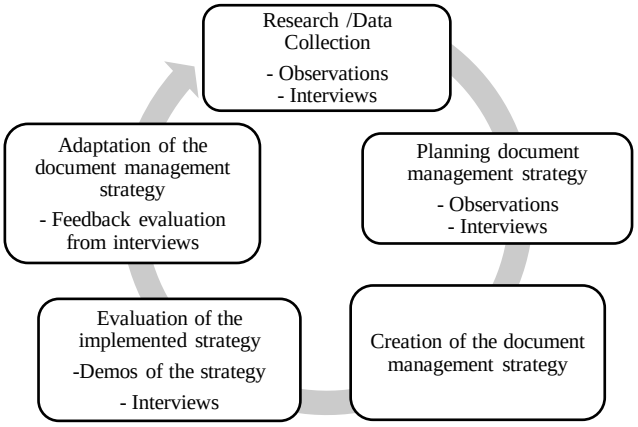


Fig. 1: Artifact's creation timeline.

1) Research and Data collection phase

In the first iteration of the research, the goal of this phase was to understand the gaps between the requirements of A-SPICE and the process the team currently applies. For this purpose, I used a mapping between A-SPICE practices and how the activities currently executed as part of the process as a foundation for a number of semi-structured interviews. [31] I analyzed the mapping by comparing the descriptions of the A-SPICE practices to the descriptions of the process steps. Whenever I found a mismatch, I added corresponding questions to the interview guide.

After the first iteration, this phase consisted of getting the necessary data needed to later produce the adapted document strategy for the identified missing processes in the initial phase. This was done by conducting semi-structured

interviews with the team and by actively observing the team's agile practices.

2) Planning phase:

This phase focused on the planning of a new document management strategy. The new strategy prototype was made in collaboration with the team to provide an adaptation of their agile processes and to assure compliance with A-SPICE. The information from the previous phase was taken into consideration to plan the strategy. Besides, semi-structured interviews for question clarification and observations of the team's practices were performed.

3) Creation phase:

The creation phase consisted of creating the proposed and previously planned document management strategy. This was achieved by using the necessary tools to properly document the information produced by the team during their working practices, and using the A-SPICE recommended base practices adapted into their agile software development processes.

4) Evaluation phase

During the evaluation phase, the previously created strategy was shown to the team. In this phase, the corresponding figures and tables containing the features and solutions for the new document strategy were shown to the team and evaluated by interviewing the team members.

The purpose of the evaluation and demonstration of the artifact is to get the necessary information and feedback to be able to later adapt the strategy.

Semi-structured interviews were conducted during this phase to get the necessary feedback from the team.

5) Learning and Adaptation phase

In this phase, the data collected from the previous phases' interviews was evaluated. This data was used to further adapt their agile practices to the presented document management strategy and make the necessary changes to fit their needs and still assure A-SPICE compliance.

C. Data Collection strategy

1) Subjects

The subjects for data collection were the members of the Aurora team of the in-house ART Steer department at Volvo Cars. The team consisted of six Software Developers, the Product Owner, and the Scrum Master. The majority of the team members had an average of working experience ranging from four to six years as software developers, and they have been working for Volvo Cars from a year to five years. Regarding the experience in agile software development process, all the team members had on average three years of experience working. Concerning A-SPICE, all of the team members were familiar with this methodology and had worked with it for roughly two years, except for one team member who did not have any previous working experience with A-SPICE.

2) Data type

The main data type to be collected is of type qualitative, as the main methods for data collection are interviews and active and passive observations.

3) Data collection methods

The methods used to gather the data were semi-structured interviews which were conducted on-site, documentation analysis of archival data, and active and passive observations of their processes.

a) Interviews

The conducted interviews were of semi-structured type. The interviews included prepared questions that were relevant to the process of the planning and implementation of the new document strategy. Besides, they were used to get direct feedback and to resolve the questions needed to gather data and information on the team's agile practices. Each of these interviews lasted an average of 30 minutes.

A total of two semi-structured interviews were conducted during the initial phase of research and data collection and a further fourteen interviews were performed during the remaining phases of the artifact's timeline creation.

Table. 2 shows the number of interviews that were conducted during the different phases of the artifact's creation cycle, how many team members were interviewed and their roles in the team, and when did the observations were performed.

The column "interviews conducted" shows the total number of interviews conducted during each of the artifact's timeline creation.

The column "interviewees role" shows the roles of the interviewees which the interviews were conducted with. Some of the software developers were interviewed twice or more times during the different phases of the artifact's timeline creation, e.g. during the evaluation phase, one software developer was interviewed three times while the other one was interviewed only twice. The same applies during the research and data collection phase, where the scrum master was interviewed once, the product owner twice and one of the software developers twice, and the remaining software developer was interviewed only once.

TABLE I. OVERVIEW OF CONDUCTED INTERVIEWS

Artifact's creation timeline	Interviews conducted	Interviewees role	Observations
First iteration - Research and Data Collection	2	Product Owner Software developer	- Daily stand-up - Sprint Planning - Sprint Retrospective - Sprint demonstration
Research and Data collection	6	Scrum Master Product Owner Software developers	- Daily stand-up - PI and Sprint Planning - PI and Sprint Retrospective - PI and Sprint demonstrations - Backlog Refinement
Planning	3	Software developers	
Evaluation	5	Software developers	

The interviews with the different team members were conducted in person and with each team member individually, in which the relevant team member was asked several questions. In case that a team member was not

physically available to participate in the interviews, the questions were asked via email, phone, or via video chat.

The information was recorded using notes and audio recordings. The subjects were asked for their verbal consent to be recorded before the beginning of the interviews.

b) Observations

Unstructured observations [23] were conducted and the role of the observer was assumed as a participant. The information was recorded in the form of field notes [23] containing the date, a unique identifier, a summary of the observation, and comments made on it.

The observations were divided into two groups:

- Active observations: participation was taken during the team's daily stand-up meetings, sprint-plannings, and retrospectives. Questions were asked during the meetings to get a better understanding of their agile practices and processes. Reporting of the document management strategy's progress was made during the daily-stand-ups. During the retrospective, participating in answering what went well or what can be improved for the next sprint was also done actively together with the team.
- Passive observations: Observations of two of the team Program Increment (PI) processes were performed. The meetings and demos performed by the team during the program increment were attended without actively taking part in them, only acting as an observer.

D. Data Analysis

The data analysis method used for analyzing the qualitative data collected was conventional content analysis. [28] The objective of the content analysis is to transform a large amount of text into an organized summary of key results in order to gain a better understanding of the topic and aid in the creation of the artifact. It consisted of a systematic categorization of verbal and text data from the interviews, observations, and documents provided by Volvo Cars. The coding and labeling of the data into categories were defined during the data analysis and were based on the research questions.

The steps followed to do the content analysis of the data were the following:

- Selection of the content to be analyzed. In this case, the transcriptions and notes taken during the interviews and observations.
- Read through the interview transcripts. Select the relevant data to the research questions. Then, create and categorize it into different categories depending on which research question they were related to. This was done with every field note and transcript.
- Once the data were categorized into major categories, such data answering any of the *which*, *when*, and *how* questions, the list was reviewed to check if the data included in those categories were relevant to the research questions and the creation of the artifact.
- In case that the data did not fit into any identified category, a new sub-category was created and reviewed if it was relevant for the study.
- Once the coding into categories was completed, the collected data were examined to find patterns and draw conclusions in response to the research questions.

Figure 2 provides an overview of the data analysis process.

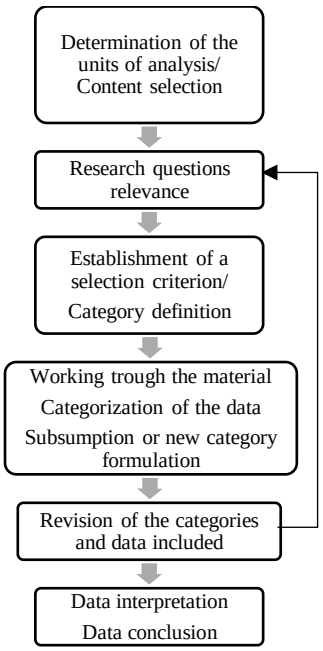


Fig. 2. Overview of the data analysis process

VI. Results

The main goal of this study was to produce a document management strategy that mapped which information was necessary for adapting agile software development practices to the compliance of A-SPICE functional safety, as well as how the information is going to be produced during the agile software development cycle.

A. Issues with the adoption of A-SPICE for Agile development

Table 2 shows the issues that the team faces when applying A-SPICE to agile development methodologies. This table is based on the data collected from the interviews done during the first iteration of the research and data collection, as well as the background information on A-SPICE and agile methodologies covered in Section II. Supplier Monitoring (ACQ.4) was excluded since it doesn't apply in this case, as the team "*only do in-house software[…]*" (Software developer, March 23, 2020). While Product Release (SPL.2) was added as they "perform releases of the products to the customer, which in this case is also Volvo Cars" (Software developer, March 23, 2020).

Some of the issues mentioned during those interviews are the difficulty of keeping the documentation updated and

notifying the necessary parties involved when working with different teams. Maintaining the consistency between the work products was also considered an issue as different teams might fill the document templates using different content. Nevertheless, the team members interviewed identified that their most relevant issues were establishing and maintaining traceability of their work products, producing and updating the right documents for A-SPICE compliance, and having the tool-chain which generates them.

The field notes that were taken during the observations and participation during the team's Sprint Plannings/PI Plannings, daily stand-ups, and sprint retrospectives helped uncovered the rest of the issues presented in Table 2. The uncover issues from the observation were the difficulty to delegate role responsibilities in large companies in case of a problem resolution management, to monitor and control daily progress within a sprint, and managing short-term repetitive product releases while producing the necessary documents for their customer(s).

TABLE II. OVERVIEW OF ISSUES FOUND WHEN APPLYING A-SPICE TO SCALED AGILE SOFTWARE DEVELOPMENT

Automotive SPICE processes	Overview [a]	Issues in Agile development
SWE.1 Software Requirements Analysis	To transform the software related parts of the system requirements into a set of software requirements	- Establishing and maintaining traceability among work products due to agile development responding with flexibility to changing requirements. - Producing and updating the necessary documentation - Having the necessary toolchain to produce documentation
SWE.2 Software Architectural Design	To establish an architectural design and to identify which software requirements are to be allocated to which elements of the software, and to evaluate the software architectural design against the defined criteria	
SWE. 3 Software detailed design and Unit construction	To provide an evaluated detailed design for the software components and to specify and to produce the software units	
SWE. 4 Software Unit Verification	To verify software units to provide evidence for compliance of the software units with the software detailed design and with the non-functional software requirements	
SWE. 5 Software Integration and Integration Test	To integrate the software units into larger software items up to a complete integrated software consistent with the software architectural design. To ensure that the software items are tested to provide evidence for compliance of the integrated software items with the software architectural design, including the interfaces between the software units and between the software items	
SWE. 6 Software Qualification Test	To ensure that the integrated software is tested to provide evidence for compliance with the software requirements	
SUP. 1 Quality Assurance	To provide independent and objective assurance that work products and processes comply with predefined provisions and plans and that non-conformances are resolved and further prevented	- Organizational structure and procedures for ensuring the quality of the activities and work products implemented by self-organizing teams - Setting quality assurance criteria for cross-functional teams
SUP. 2 Verification	To confirm that each work product of a process or project properly reflects the specified requirements	Developing the criteria verification and including them in agile practices. Communicating the verification results to all affected parties on a large company
SUP. 7 Documentation	To develop and maintain the recorded information produced by a process.	Producing and maintaining updated the necessary documentation. Not agile main focus
SUP. 8 Configuration Management	To establish and maintain the integrity of all work products of a process or project and make them available to affected parties	Maintaining consistency between work products among cross-functional teams and sprints
SUP. 9 Problem Resolution Management	To ensure that problems are identified, analyzed, managed and controlled to resolution	Issue sharing across teams and sprints with the necessary recorded information to all affected work products. Role responsibilities delegation in a large company
SUP. 10 Change Request Management	To ensure that change requests are managed, tracked and implemented	Flexible response mechanism for managing change requests. Providing traceability between change requests and their affected work products
MAN. 3 Project Management	To identify, establish, and control the activities and resources necessary for a project to produce a product, in the context of the project's requirements and constraints	- Defining a consistent product roadmap and defining functional growth of the product consistently - Realistically plan each sprint - Monitoring and controlling daily progress within a sprint.
SPL. 2 Product Release	To control the release of the product to the intended customer	- Short-term product releases - Repetitive releases - Organizational structure and suitable tools for documentation

[a]Taken from the VDA A-SPICE 3.1 [4]

B. *Results for RQ1. Which information needs to be produced to comply with automotive SPICE while following an agile development process in the automotive industry?*

Two tables were created to present a mapping of what information needs to be produced to answer RQ1. Table 3 presents a mapping of the documents that need to be produced according to A-SPICE with their corresponding base practices, as well as a description of the information that needs to be included in such documents.

The column about SAFe agile practices and a suggestion of continuous integration tools contains the agile practices that produce such information and the tools that contain it. This answers the question of which information needs to be produced during an agile development process.

TABLE III. A-SPICE INFORMATION TO BE PRODUCED AND THEIR SAFE PRACTICES – OVERVIEW MAPPING

Documents to be produced	**Corresponding A-SPICE base practices (BP)**	**Which information needs to be produced according to A-SPICE**	**Which SAFe practices and Continuous integration tools produce the information**
Software Requirement Specification (SWRS)	SWE.1 BP.1 ACQ.11 ACQ.13 SYS. 2 BP.1-5,8	Specification of functional and non-functional software requirements for the system requirements and system architecture	Program Increment (PI) planning Sprint Planning Tools: Software Configuration Management tools, e.g. JIRA - Product backlogs + development platform for embedded systems e.g. SystemWeaver
Software architectural design specification (SWADS)	SWE.2 BP.1 SWE.3 BP.1 SYS.3 BP.1-5,8	-Software architectural design specifying functional and non-functional requirements - Detailed design of each software component for functional and non-functional components - System architectural design specifying the elements of the system for functional and non-functional system requirements	PI planning Sprint development phase
Software Development Plan (SWDP) Containing**:** -Project Plan -Stakeholders list -Change Request -Project status report -Communication Record -Schedule -Project status	MAN.3	- A document defining project scope; life cycle; feasibility evaluation; activities for defining, monitoring and adjusting project activities + project estimates and resources - Project schedule/ roadmap of the product -Review and progress report of the project - Identify, monitor and adjust project interfaces and agreed commitments with stakeholders	Information from PI planning, Sprints planning, daily stand-ups, system demos, and retrospectives. The product owner uses Software Configuration Management (SCM) tools to produce the necessary information and manage the project
Risk Management Plan (RMP)	MAN.5 SYS.1 BP.2,5	A defined action plan for identified risks on both project and organizational level. Risk treatment actions and monitoring. Risk acceptability levels and prioritization strategy.	Scrum practices produce this information: -PI planning with method R.O.A.M (Resolve, Owned, Accepted, Mitigated) for Risk scope, risk identification, analysis, monitoring, and risk treatment actions. – Sprint planning: for corrective actions in the project development. The product owner is responsible for action-taking. SCM such as JIRA used for managing the risks.
Requirement traceability matrix (RTM)	SWE.1 BP.1,6,7 SWE.2 BP.6-8 SWE.3 BP.4-6 SWE.4 BP.5,6 SWE.5 BP.7,8 SWE.6 BP.5,6 SYS. 2 BP.6,7 SYS. 3 BP.6,7 SYS.4 BP.7,8	Bi-directional traceability of the requirements and their corresponding test cases throughout all phases of the life cycle	Map of user requirements and test cases. Scrum practices, such as PI planning, sprint development, and test phases produced this information using SCM tools and testing tools. Example. JIRA can produce a RTM document using a plugin.
Software Test description (SWTD)	SWE.4,5	Document with the result of functional test, integration test, and qualification test reports	Produced in the testing phase of the sprint using specific testing tools to automate the document's production of the test results
Software functional test specification and report	SWE.4 BP.2,4	- Results from functional test - Issues encountered record	
Software integration test specification and report	SWE.5 BP. 3,6 SYS.4 BP. 1-6,9	- Results from test integration - Issues encountered record	
Software qualification test specification and report	SWE.5 BP.2 SWE.6 BP.4 SYS. 5 BP. 1-4, 7	- Results from qualification tests - Issues encountered record	
Software configuration management plan (SWCP)	SUP.8 BP.1	-Team responsibilities and resources -Tools and repositories - Identify the configuration items and naming conventions	The Scrum team decides how to state the different configuration activities in a meeting.

		- Access rights - Merge and branching strategy Revision of configuration items	A standard document should be produced and stored on an accessible platform, such as SharePoint or similar content sharing tools. It should be defined just once in the project life cycle.
Problem Resolution management Strategy (PRMS)	SUP.9 BP.1	- Verification problems identified.	- Scrum activities such as daily stand-ups and sprint planning. - Product owner responsible for monitoring and updating the problem management activities using a SCM, e.g JIRA - Guideline document of the steps and team responsibilities shall be recorded in SharePoint or similar content sharing tools
Software Quality Assurance Plan (SWQA)	SUP.1 BP. 1	- Definition of activities to ensure the quality of the work products - Corresponding reports summarizing a list of quality activities performed and results.	- Pre-defined Volvo template exits for guidelines. - Verification and validation result together with progress reports and software trouble reports are produced during sprint development and testing phase. - The product owner ensures activities and monitoring the quality of the product. - Guideline document of the steps and team responsibilities shall be recorded in SharePoint or similar content sharing tools
Software Audit report (SWAR)	SUP.1 BP. 2-6	Purpose of the audit - Method used for the assessment - Requirements ID - Limitations - Date, attendees, and coverage - Result of the audit	Meeting record: done by the PO owner or the chosen responsible team member.
Software Verification Strategy (SWVS)	SUP.2 BP.1 SWE.4 BP.1	- Specify how verification is performed for activities, techniques, and tools of all work products. - Define verification criteria for software units including regression strategy	- Software developers agree on the necessary tools to use for their needs. Aurora uses Ceedling tool as a build system for test executions targeting Test-driven development, but any similar testing tool can be used for this purpose. - Guideline document of the steps and team responsibilities shall be recorded in SharePoint or similar content sharing tools
Change Request Management Strategy (CRMS)	SUP.10 BP. 1 ACQ.11 BP.7	Mechanism to incorporate changed or new technical requirements into the established baseline. - Change request activities - Status model for the change requests - Analysis criteria - Responsibilities for performing these activities	- Identification is done during PI planning and Sprint planning. - For reviewing and tracking the implementation of change requests: Sprint demos and daily stand-ups activities - Product Owner responsible to update and monitor activities using a SCM tool e.g JIRA for status visualization. - Guideline document of the steps and team responsibilities shall be recorded in SharePoint or similar content sharing tools
Software Release Plan (SWRP) Including: - Product Release information - Product release package - Delivery record	SWE. 5 BP.1 SPL.2 BP.1,2,6,9	Information to be included in each release with its associated elements required (hardware specification, software specification, etc) - Mapping of the requirements and their status	- Guideline document of the steps and team responsibilities shall be recorded in SharePoint or similar content sharing tools. - During the sprint release phase and PI planning as system demos and release plan demos.
Software user manual (SWUM)			Product backlog from a Software Configuration Management tool can be used for visualizing this.

For mapping which documents needed to be produced, the VDA A-SPICE 3.1 standard document [4] was used as a reference to generate such information. Once the necessary documents were identified, they were matched to their corresponding A-SPICE base practices. These base practices define the tasks and activities that are needed to be accomplished in order to fulfill the process outcomes. The description of which information needs to be produced according to those base practices is mapped in the column "Which information needs to be produced according to A-SPICE".

The column "Which SAFe practices and tools for Continuous integration tools produce the information" serves to identify the agile practices contained in the SAFe framework that produce the needed information to be contained in the identified A-SPICE documents. Field notes taken during the team's sprint planning, program increment planning, backlog refinements, daily stand-ups, retrospectives, and demos of their configuration management tools utilization were used to aid the creation of this column. Interviews performed during the Research and Data Collection phase and the planning phase of the artifact's lifecycle were also used. The information generated from the

interviews covered an explanation of the team's activities during those meetings, how they managed the development of the products, and what tools they used for this purpose.

Table 4 presents the A-SPICE work products, how the generation of the information is going to be replaced by agile methods and Continuous Integration tools, and how that information is being generated. As it shows an overview of how the information produced is being collected and which tools are used for this purpose, it partially covers parts of RQ2 as well.

Field notes taken during the Research and Data Collection and Planning phase of the artifact were used to aid the creation of Table 4. The observations were made during the team's sprint planning, program increment planning, backlog refinements, daily stand-ups, retrospectives, and demos of their configuration management tools utilization. The team's activities performed during those agile practices helped identify how the information was being produced using the correct tools and which one of their SAFe practices were producing the information that has to be covered by the A-SPICE work products.

TABLE IV. A-SPICE WORK PRODUCTS REPLACED BY AGILE METHODS AND CONTINUOUS INTEGRATION TOOLS THAT PRODUCE THEM

A-SPICE Work Products	**Tools and SAFe practices**	**How the work products are produced**
Software Requirement specification	Program increment (PI planning)	A development platform for embedded systems such as SystemWeaver has the option to generate reports when needed.
Requirement Traceability Matrix	Reports from Software Configuration Management (SCM) tools and automated testing report tools	A tool-chain integration generates the necessary information for the software configuration management (SCM) tool e.g. JIRA, to generate the reports containing the software requirements traced to their test cases
Software architectural design specification	Development platform for embedded systems reports e.g. SystemWeaver Sprint development phase	Development platform for embedded systems e.g. SystemWeaver stores the software architectural design details. A document can be generated using the tool
Product release information	Sprint retrospectives reports/documents Sprint release phase – system demos and product presentations PI planning demos/presentation and planning for product release during the upcoming sprints	The scrum master shares a document in a web-based collaborative platform, e.g. SharePoint, containing a summary of what was covered during the meeting. System and product demos information is generated by the team using PowerPoint presentations and then stored in the web-based collaborative platform. Product roadmaps are generated by the Product Owner to present the product release timeline. Generated using a SCM tool e.g. JIRA.
Configuration management record	Software Configuration Management tools e.g.JIRA and Product owner	The Product Owner is responsible for maintaining the Product Backlog and Sprint stories updated. The team generates different user stories and the PO supervises them. This can be used as a record of how the project is managed. No need for extra documents to be generated but it can be produced by a SCM tool e.g JIRA, as a document when needed.
Change request	PI planning and Sprint planning - Product owner - Software Configuration Management tools e.g. JIRA	The person responsible uses a SCM tool e.g JIRA to create a change request as an issue type. The tool allows to customize the workflow. The requests can be visualized and managed using the tool. No need to generate extra documents.
Quality record	Product owner activities managed in a Software Configuration Management tools e.g. JIRA	
Review record	Retrospectives report Backlog refinements in the middle of the sprints System demos report/slides	Retrospectives reports and demos are filled out by the team as a presentation. Uploaded by the scrum master to a web-based collaborative platform e.g. SharePoint, for storage. Backlog refinements are done using a SCM tool e.g. JIRA. Information is generated and visualized using the tool. No need to generate extra documents in this case.
Risk action request	Software Configuration Management tools e.g. JIRA or similar, for storing information about risks action requests - Adjust development if necessary, during sprint planning Product Owner responsible for tracking and closure	The person responsible for filling out an action request creates a ticket using a SCM tool e.g. JIRA. The ticket contains the necessary information to describe the risk and notify the involved parties of the action to be taken.
Validation results	Testing tools, automated documentation produced during the different testing techniques	Documents are created and populated with continuous integration tool-chain and different testing tools.
Verification results	Testing tools, automated documentation produced during the different testing techniques	
Test results	Testing tools, automated documentation produced during the different testing techniques. A universal artifact repository manager e.g. Artifactory is used for storage using stated naming convention	
Change history	Product owner Sprint planning PI planning Software Configuration Management tools e.g. JIRA	Same as Change Request
Project schedule	Software Configuration Management tool e.g. JIRA function	A SCM tool e.g JIRA has an option for generating products roadmaps according to the user stories

	Roadmaps	
Project status report	Software Configuration Management tools e.g. JIRA	A SCM tool e.g. JIRA is used for monitoring the project status. No need for extra documents to be generated
Risk analysis and status report	Software Configuration Management tools e.g. JIRA	Same as Project status report
Problem status report	Software Configuration Management tools e.g. JIRA	A SCM tool e.g. JIRA is used for monitoring the problem status tickets. No need for extra documents to be generated.
Audit report	Volvo Cars template. Audit responsible for filling in the information after the audit manually	Not generated by the tool-chain. The Software Requirements Specifications document and generated reports from the tool-chain test results have to be included in the document. A record of the result with the identified corrective actions needs to be generated manually by the involved parties. This can be shared in a web-based collaborative platform .e.g SharePoint
Analysis report	PI planning Sprint planning Software Configuration Management (SCM) tools e.g. JIRA backlog	The PI planning and Sprint planning demos and retrospectives are used to generate this information using the configuration management tools.
Meeting support record	Daily stand-ups PI Planning and Sprint Retrospectives	The information generated and issues covered in the meetings with the responsible parties are saved in a web-based collaborative platform e.g. SharePoint or similar.

C. Results concerning RQ2. How is the information produced in an agile development process going to be collected and documented to assure A-SPICE compliance?

To present how the previously mentioned information is going to be collected in an agile development process, a figure showing how the different continuous integration tools interact with each other to produce the necessary A-SPICE reports has been created. Figure 3 presents this interaction. This figure was planned in cooperation with three software developers of the Aurora team during the planning phase. The interviews covered the usage and performance of the configuration management tools and continuous integrations tool. The information produced during the interviews in the planning phase was used during the implementation phase to create the tool-chain figure.

To be able to save the A-SPICE recommended strategies and guidelines documentation, the developers can choose to write them and save them for sharing across teams either in a web-based collaborative platform e.g. SharePoint or similar and storing them in a distributed version-control system for tracking changes in the source code e.g. GIT/Gerrit.

The team tasks are created, stored, and managed using a Software Configuration Management tool (SCM) such as JIRA, where the teams can plan during the sprint planning in which tasks they are going to work. This particular SCM tool has also a function for creating product roadmaps and issues, risks, and change request reports.

Functional and non-functional requirements are stored in a development platform for embedded systems e.g. SystemWeaver. This application is used to produce requirements specification reports, software architectural reports, and software functional specification reports. SystemWeaver can be configured to support the SCM tool e.g. JIRA and therefore share the requirements. This is needed to provide requirements traceability and update them in the SCM tool. With both applications, a requirement traceability matrix can be generated using the right plugins.

Once the developer writes their source code, a program that drives continuous integration, delivery, and deployment e.g. Zuul is used for monitoring, building, and running the tests without managing cross-origin resource sharing (CORS) and one-by-one authentication. A continuous integration server tool e.g. Jenkins, is then used to run the different tests in parallel for providing continuous integration.

The continuous integration server e.g. Jenkins runs the testing tool e.g. CANoe, for functional testing and produces the software functional test specification report. This report is stored in a universal artifact repository manager e.g. Artifactory.

Running in parallel, the continuous integration server tool e.g. Jenkins also executes the tests from a test-driven development tool e.g. Ceedlings, which provide automatic test discovery, mock generation, and test execution. The reports produced from the unit testing are change log reports and release note reports.

A tool for continuous inspection of code's quality e.g. SonarQube is used for ensuring code quality and bug identification in the source code and test cases. The report produced by this application is the *Software qualification test specification report*. The bug reports and issues in the source code are also reported back to the distributed version-control system for tracking changes in the source code e.g. GIT/Gerrit.

All the previously mentioned test reports are compressed into zip files and saved in the universal artifact repository manager e.g Artifactory, where they will be labeled and tags will be created in order to sort them.

A document generation tool e.g. Doxygen will be used for the generation of documents related to the source code. This report will contain an explanation of what the code functions and variables do. It is generated from the distributed version-control system e.g. GIT and stored in the universal artifact repository manager e.g.Artifactory.

The continuous integration server tool e.g. Jenkins can also process a status report on the jobs for providing visualization of the status to the developers.

For providing further traceability, the test cases and different testing applications can be integrated into the SCM tool e.g.

JIRA using plugins. This is necessary for maintaining an updated requirement traceability matrix.

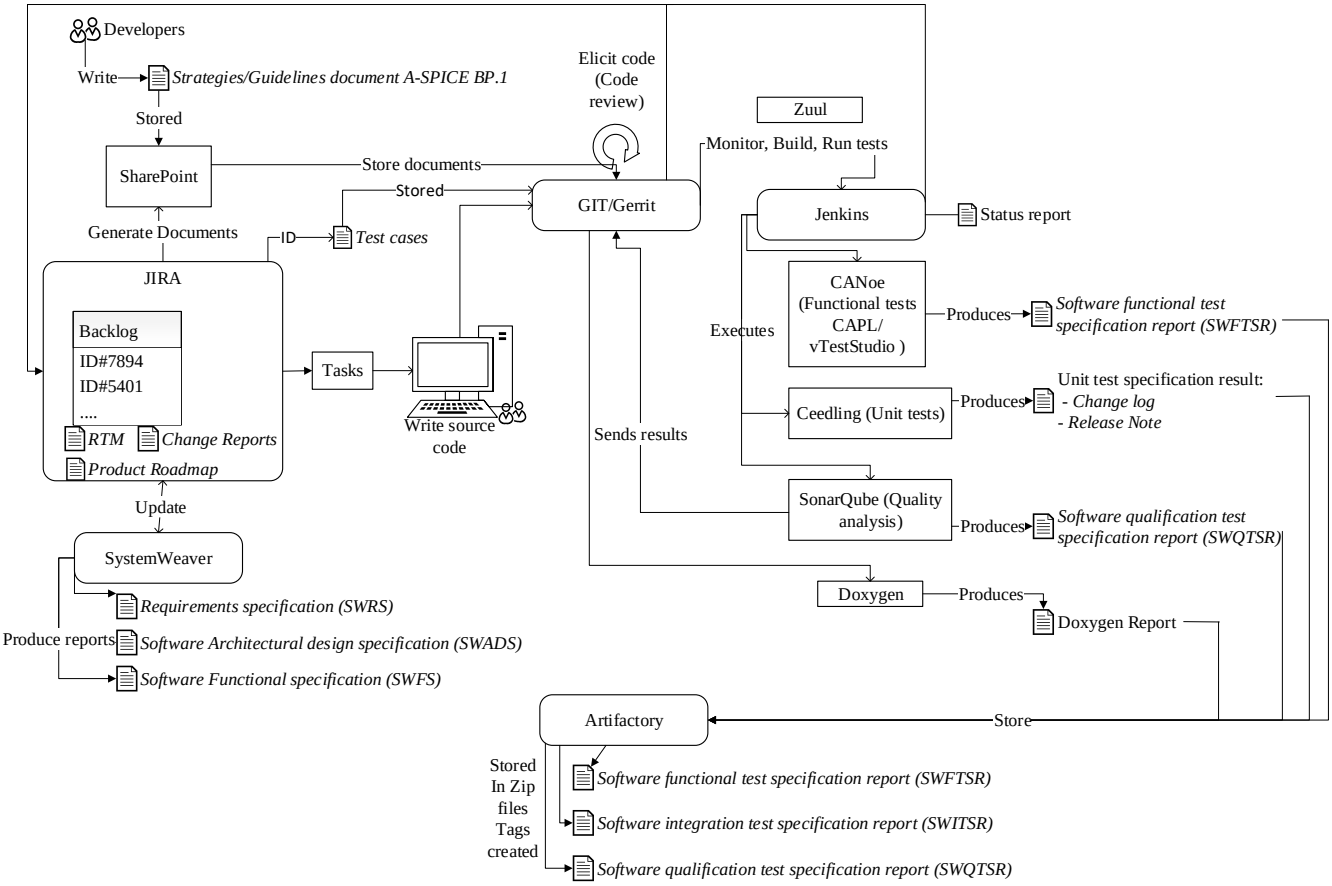


Fig. 3. ToolChain and document creation – Overview

D. Results concerning RQ3. When during the different stages of an agile development process does the information need to be produced, collected, and documented?

There are two main agile development processes in the SAFe framework in which the team participates regularly. One of them is the two-week Sprint and the activities needed to manage, develop, and release the product. The other one is the program increment planning (PI planning). This is done for planning what is going to be developed and produced during the next six sprints and to show what has been done during the previous sprints.

The following two diagrams have been created to provide an overview of when during the Sprint and PI planning documents are being created. A mapping of the different A-SPICE practices with their respective work products has been added as notes. Besides, agile activities are mapped to the corresponding A-SPICE processes. Both diagrams were created using the feedback provided by the interviews and observations of the team agile practices.

Figure 4, shows the PI planning activities with the corresponding A-SPICE processes and their work products that are being produced or updated with each activity. One of the interviews conducted during the evaluation phase revealed that the production of the different work products should be color-coded to differentiate which ones have to created and stored once or if they need to be modified during the activities “It would be good to differentiate when do we have to modify the documents or if we just have to create them once and share it with the rest of the team, such as the different strategies” (a software developer from the Aurora team, April 9th, 2020). Therefore, different colors were used to differentiate when the A-SPICE work products' needed to be updated or created during both the PI planning and sprint planning activities.

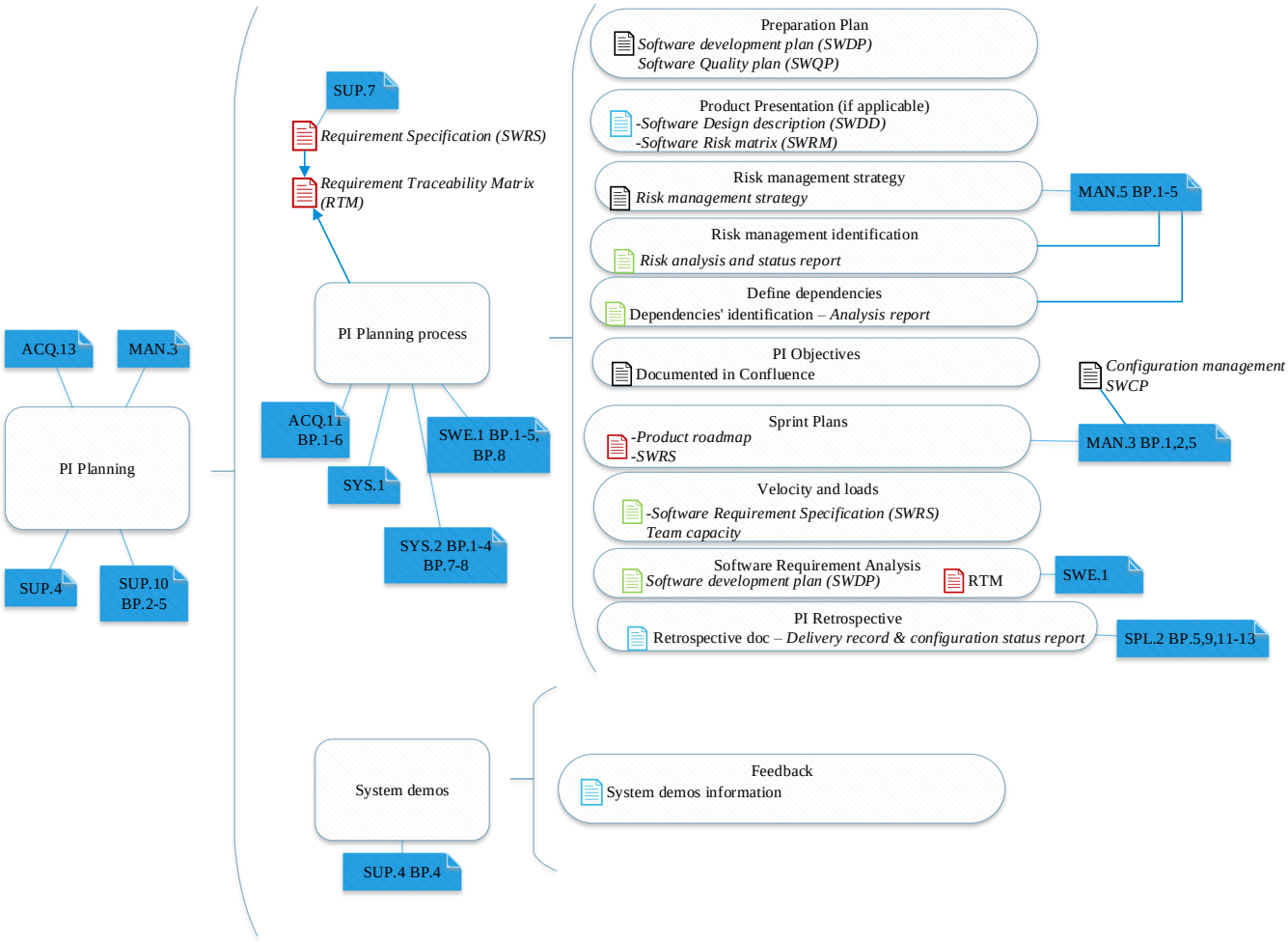
The documents in *blue* are the ones whose creation needs to be considered (if applicable). In this case, the Software design description, Software risk matrix, PI Planning retrospective report, and system demo reports need to be created only once during PI Planning.

Documents in *red* are to be updated or modified from previously existing ones, such as requirement traceability matrix, product roadmaps, the SCM tool e.g. JIRA, and requirement specification document (SWRS). These reports are usually automated using the previously mentioned tools.

Documents in *green* are the ones where the information is being updated by the team during the PI planning practices using the SCM tools such as JIRA or a development platform for embedded systems e.g. SystemWeaver, and therefore, there is no need to generate a separate written report/document.

Lastly, the documents marked in *black* are the ones that are already generated prior to the PI Planning but need to be stored in a web-based collaborative platform e.g. SharePoint or similar tools in order to comply with A-SPICE base practices. The strategies that need to be created are the software development plan, Software analysis, PI objectives document, and Software Configuration management plan.

Figure 5, shows the sprint activities with the corresponding A-SPICE processes and their work products that are being produced or updated with each activity.

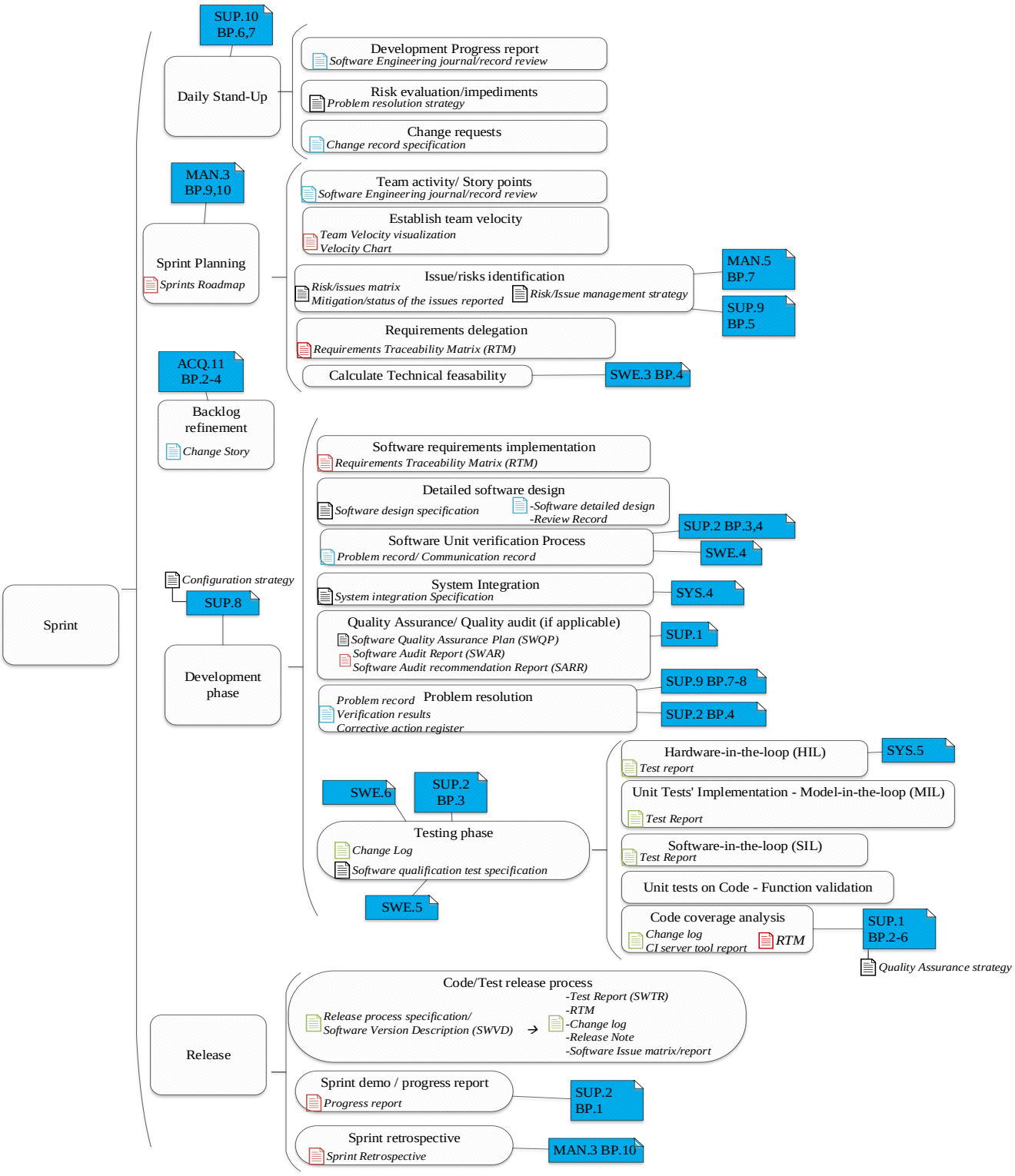


ACQ.13 – Project Requirements
ACQ.11 – Technical requirements
SUP.4 – Joint Review
SUP.7 - Documentation
SUP.8 – Configuration Management
SUP.10 – Change Request Management
SYS.1 – Requirements elicitation
SYS.2 – System Requirements Analysis
SWE.1 – Software Requirement Analysis
MAN.5 – Risk Management
MAN.3 – Project Management

Documents to generate each PI
Document to be updated/modified each PI planning
Documented information - recorded and generated using Software Configuration Management tool e.g. JIRA/ development platform for embedded systems e.g .SystemWeaver
Information recorded in a web-based collaborative platform

BP. - Best practices of each process recommended by ASPICE guideline v.3.1
- If BP are not specified on each process, all the BP applies to it.

Fig. 4. PI Planning activities with A-SPICE processes and work products - Overview



ACQ.11 – Technical Requirements
SUP.1 – Quality Assurance
SUP.2 - Verification
SUP.7 - Documentation
SUP.8 – Configuration Management
SUP.9 – Problem Resolution Management
SUP.10 – Change Request Management
MAN.3 – Project Management
MAN.5 – Risk Management
SYS.4 – System Integration and Integration Test
SYS.5 – System Qualification Test
SWE.4 – Software Unit verification
SWE.5 – Software Integration and Integration Test
SWE.6 – Software Qualification Test
BP. - Best practices of each process recommended by ASPICE guideline v.3.1
- If BP are not specified on each process, all the BP applies to it.

Documented strategy in Confluence or SharePoint
Updated/generated each Sprint
Information produced in a Software Configuration Management tool e.g. JIRA during Sprint
Automated informantion production using CI server tool e.g. Jenkins + different testing tools

Fig. 5. Sprint and A-SPICE processes with work products -Overview

This figure maps when the different documents are being generated and which ones need to be created (if applicable) and/or updated during the lifecycle of the Sprint. The interviews and field notes were used the same way as Figure 4, as both figures were presented to the team member at the same time.

To differentiate when the different work products need to be created, updated, or automatically recorded/produced, color tags have been created.

The backlog refinement activity is only held once during the middle of the second week of a two-week Sprint to keep the backlog updated and only includes the user stories that are relevant for the current sprint.

Documents marked in *black* indicate that the document only needs to be produced and stored once. If any changes are done to the strategies, regardless of the sprint phases, the responsible person in the team should update them. These documents are stored in shared locations so other teams can access them. Documents such as Quality assurance Strategy and Configuration strategy are only mapped to their corresponding A-SPICE processes and base practices. They do not belong to any Sprint activity, but they need to be produced and stored using any of the mentioned tools to comply with functional safety. Software qualification test specification is a report containing all the reported information produced by the automated tools and it is collected in a web-based collaborative platform e.g. SharePoint during the testing phase.

Red documents need to be updated and/or generated during each of the corresponding Sprint activities. Sprint roadmap document is being updated when any decision made during sprint planning affects the product roadmap. Team velocity charts report, progress report, and sprint retrospective are created during the sprint planning phase and release phase respectively. Regarding Quality audits, the necessary documents will only be created and updated if this activity is held during the development phase. Quality audits are sprint activities that are necessary to inspect or examine the development process in order to ensure compliance with the requirements. [29] Quality audit activity can be performed once during the third or fourth Sprint during a 6 week PI.

Requirement traceability matrix documents are updated whenever there have been any changes to the software and system requirements. These changes are usually performed during the Sprint planning and development phase, more specifically during software implementation practices.

Blue documents correspond to the information that is recorded, managed, and updated automatically using a Software Configuration Management tool e.g. JIRA. If needed reports can be generated using this tool.

Lastly, documents marked in *green* are produced automatically during different sprint activities. Reports are generated and stored using the necessary tools (see Fig. 5). The automated generation of the different reports happens in parallel during the testing phase and the release phase, specifically during the release process activity.

E. Evaluation

During the Evaluation phase, three interviews were conducted with three different software developers from the Aurora team in order to get feedback and to provide further adaptation of the document management strategy to their needs.

During the first evaluation interview, one of Aurora's software developers was presented with the diagrams created for mapping the sprint and PI planning with the A-SPICE processes and work products (Figures 4 & 5). The developer thought that such diagram visualized which documents need to be produced during the team's agile practices, "it will be able to help the team get a better understanding of which documents need to be produced and when we should do it" (A Aurora team software developer, April 9th, 2020).

Three interviews were conducted during the evaluation phase to assess the tool-chain diagram presented in Figure 2. Three different software developers of the Aurora team were interviewed separately to obtain their insights into the presented diagram. One of them provided feedback for identifying a missing process in the tool-chain, such as the creation of the test cases, as they were missing on the first version, "the tool-chain looks good and seems to map all the management tools that we use with the continuous integration tools. However, the only thing that I think it is missing is the creation of the test cases and how they interact with JIRA for assuring bi-directional traceability with the requirements for the requirements traceability matrix" (A software developer of the Aurora team, April 17th, 2020). Test cases were then added in the final version of the tool-chain diagram presented in Figure 2. Another one thought that some of the tools were Volvo Cars specific and therefore they shall be changed for similar ones "I don't think you should include X and X as they are Volvo Cars specific tools, you can change them for similar applications that are not company-related but serves the same purpose" (A software developer of the Aurora team, April 17th, 2020). Thus, no Volvo Cars' specific tools were used in the final version of the tool-chain diagram, and similar tools were used instead. Regardless of the feedback received, the three developers agreed that the presented tool-chain covered the team's needs, properly showing the team's management tool integration with their CI tools and correctly mapping how the documents were generated using the tools.

The last evaluation interview was conducted regarding Tables 3 and 4 to assess the information contained in the tables. One software developer of the Aurora team participated in the interview. The developer thought that the tables served their purpose in highlighting which documents need to be produced and what information shall be presented in the generated reports "These tables are helpful for showing which documents we need to fill out. It was good to present which information is needed from A-SPICE and how we are producing that information in our agile practices" (A software developer of the Aurora team, April 28th, 2020).

VII. Discussion

In this section, the major findings encountered in the results section are discussed, as well as presenting the limitations found regarding the study reproducibility.

A. Major findings

The major findings are discussed according to each research question. Besides, a description of how the document management strategy helped the team in adapting their agile practices to provide A-SPICE compliance is also provided.

1) RQ1. Which information needs to be produced to comply with automotive SPICE while following an agile development process in the automotive industry?

The field notes and interviews conducted during the creation process of the document management strategy (described in Section V) helped to provide the necessary information that was needed to later map their agile development processes to the A-SPICE processes.

Tables 3 and 4 revealed that much of the information needed for the A-SPICE work products was produced by the team's SAFe development practices. However, this information needs to be stored, maintained, and updated somehow in order to provide the previously mentioned reports to comply with functional safety. These documentation practices were not previously maintained by the team in their daily agile development practices or it happened to be in the early stages of being adapted to A-SPICE documentation requirements.

After the necessary information was identified, the team had a better understanding of which strategies need to be created for transitioning from an A-SPICE level of 1 to 3. The tables serve as a guideline for developers to be followed for adapting their agile practices to A-SPICE processes (refer to the interviews conducted during the evaluation phase).

Komiyama et al. proposed a strategy for keeping the documents to a minimum similar to the one proposed in this study. They included a table with the documentation for agile software development using A- SPICE and the rationale behind such documentation. [22] However, the identified documents in their study can only work for small companies, as they only included the A-SPICE Software engineering Process group (SWE. 1-6). [4] Their approach would have been found to be insufficient for the needs of a large automotive company such as Volvo Cars, as those documents are not enough to comply with A-SPICE. Therefore, the proposed guideline in this study (made in cooperation with the Aurora team of the ART Steer department at Volvo Cars) is considered as a more suitable adaptation for Scaled Agile software development and A-SPICE processes.

The documents identified in Tables 3 and 4 are found to be not just A-SPICE specific, as they are also being filled out by other companies outside of the automotive domain working Agile [32] as e.g Software requirement specification and Software architectural design specification documents among others. Therefore the proposed tables can also serve as a documentation guideline for software development companies outside of the automotive domain who are transitioning from a waterfall development process into an Agile oriented one.

2) RQ2. How is the information produced in an agile development process going to be collected and documented to assure A-SPICE compliance?

As covered in previous sections (see Section II and III), A-SPICE and Agile stand in opposition to each other on the issue of documentation [22].

To harmonize these opposing positions on documentation, tools were used to automate the production of as many A-SPICE work products as possible.

The team chose the tools for the presented tool-chain in Figure. 5 (see Section VI). The tool-chain integration was created to provide a suggestion for the tools that can be used to produce the necessary information without needing extra meetings or extra team activities for filling in and updating the documentation.

One of the previously mentioned A-SPICE challenges identified by other authors, as well as by the Aurora Team, is establishing bi-directional traceability of the work products. [7, 9, 21] Regarding the traceability issue, it was discovered that the tool integration chain uses the SCM tool e.g. JIRA as the main management tool to visualize the bi-directional traceability between requirements and test cases. This is due to the possibility of integrating new plugins or other applications into it. A similar approach was used by Komiyama et al. in their study. They used tools such as TestRail, Bitbucket, Bamboo, and SonarQube for providing traceability of their work products. [22] Their approach was similar to the one presented in this study. However, their toolchain is only suitable for smaller companies consisting of one or two development teams, as they manage all their requirements using the SCM tool JIRA.

The new document management strategy uses a development platform for embedded systems (e.g. SystemWeaver) integration into a software management tool e.g. JIRA to keep track of the functional and non-functional system and software requirements. This integration can provide traceability visualization of the requirements across functional teams, as well as producing the necessary digital report documents when needed (reports in doc alike format). That being said, the right applications and plugins need to be selected to provide such bi-directional traceability. As SystemWeaver is used as a development platform for embedded systems mostly used the automotive system, [35] a similar tool should be used in its place in case of this toolchain being used by a company outside the automotive domain.

The automation of the creation of the reports is a positive thing, as it helps the development teams to manage better their activities and focus them more on development than documentation. However, the need of having different applications, such as the one presented in this study, can be challenging for those companies that have established tool-chains that are hard to change. [7, 24]

Observations of the team agile practices and Continuous integration tools along with interviews helped in the creation of such toolchain for establishing and maintaining traceability of the products.

3) RQ3. When during the different stages of an agile development process does the information need to be produced, collected, and documented?

The different mapping for both PI planning and Sprint activities helped the team visualize when during the team's agile software development process the necessary documentation needs to be generated or updated.

It provided the documentation during the different SAFe team's activities. However, the mapping discovered some gaps regarding when the quality assurance strategy and configuration strategy documents need to be created or updated. A-SPICE implies that those strategies need to be created as the first step for assuring the quality of the products, but not when. [4] However, none of the team's SAFe framework activities are specific to Quality Assurance (SUP.1) or Configuration Management (SUP. 8). That being said, it does not mean that the team does not perform quality assurance activities or configuration management activities. These activities are being performed during the development phase and testing phase as development actions, such as having pre-defined responsibilities, naming conventions for their configuration items, and branching guidelines. All these different guidelines are stored using SharePoint or similar tools that allow sharing information across teams. Therefore, these team practices are considered sufficient for ensuring quality and handling configuration management, and there is no need to produce extra documentation more than once, although it shall be updated when applicable.

Regarding the usefulness of this guideline, the mappings presented in figures 4 and 5 are using the documents identified in tables 3 and 4 which are found to no just be A-SPICE specific, companies outside the automotive domain who are transitioning into a more agile development process can also benefit from these mappings [33].

B. Threats to Validity

According to Runeson & Höst [13], there are five types of validity threats that are suitable for a case study. These are internal validity, construct validity, reliability, and external validity.

1) Internal Validity

The threats refer to observer bias. The collection of qualitative data can introduce errors in the data collection by, for example, asking leading questions. This could influence the recording of the data and lead to an incorrect interpretation of the results. The solutions that can be taken to minimize the bias are: blinding the assessors, training non-blinding assessors to detect bias, or use an inter-observer. [30]. These methods were considered resource-demanding and unreliable. Another internal validity was the document management strategy which was developed by the author of the study. This could lead to research bias, where the researcher influences the results. To mitigate this threat, the document management strategy was developed in collaboration and supervision of the Aurora team. The document management strategy was also reviewed by our university supervisor.

2) External Validity

This addresses the generalizability of the case study. As this is a company-specific study, the data collected and the creation of the artifact was relative to the way of working of the Aurora team and Volvo Cars. This is why the results of the study are intended to be used as a reference point for other automotive companies looking to adapt their agile software development processes and A-SPICE practices, or for other companies outside of the automotive domain willing to adapt their practices.

3) Construct Validity

The validity of the results partly depends on the reliability of the interview transcriptions and field notes that were taken during the different phases of the document management strategy's creation and whether they were good enough to capture the correct information from the practitioners. [34] To help mitigate this issue, the interview transcripts and field notes were shown to one of the software developers to validate that the collected information was the correct one and was relevant to the research questions. [36]

Another important aspect is that the adapted document management strategy concerns the process in theory based on observations of the team's agile practices, but not in practice, as implementing the process was out of the time-frame of this study.

4) Conclusion validity

This refers to internal generalizability or the degree to which the drawn conclusions from the qualitative data collected during the interviews are reasonable. [33] Due to the limited time frame of our case study and external circumstances out of anyone's control, the interviews were not conducted with every member of the case organization. Instead, to attempt in getting representative results, the interviews were only conducted with the key team members, such as the Scrum Master, the Product Owner, and the relevant software developers in the team. The data from the interviews were completed with observations that ranged from being on the level of the Aurora team to the entire department.

5) Reliability

The reliability of the results needs to be considered regarding the mappings and the interpretations of the different A-SPICE work products. To address this, the artifact was created in collaboration with the Aurora team and its continuous feedback. This was done to ensure mutual agreement on every mapping, work product, and tool.

VIII. CONCLUSION

The goal of this study was to provide an adaptation for combining both A-SPICE and agile methodologies for large companies. This was needed to ensure the production of all the necessary documentation for complying with A-SPICE standards while keeping the development and

production of the deliverables in an agile fashion in the automotive domain.

A document management strategy was produced together with the Aurora team to serve as a guideline for such adaptation. This strategy helped to visualize how the information is being produced and collected, which information needs to be generated, and when the information needs to be managed during the different stages of an agile development process. It was found that the new document management strategy served to partially solve one of the challenges concerning bi-directional traceability of the requirements with their respective test cases. We also found that the identified issues combining both development practices were solved with the presented strategy. This was achieved by using a combination of different configuration management tools for continuous integration practices. Additionally, this study was intended as a reference point for other organizations of the automotive domain looking to adapt their agile practices with A-SPICE processes to ensure functional safety.

This study is the first attempt at adapting Scaled Agile development processes to A-SPICE functional safety practices in the automotive domain for a large company. Further studies need to be conducted to expand the proposed strategy to other teams that act as external suppliers for other large automotive companies, as well as implementing the strategy to check how it affects team productivity in the short and long term. Preferably, such implementations would uncover further gaps or issues during the implementation of the strategy.

ACKNOWLEDGMENT

Courtesy to Jan-Philipp Steghöfer for supervising and giving feedback for the case study and for teaching us about Software processes at the University of Gothenburg. And to Richard Berntsson Svensson for being the examiner and giving feedback for the thesis. And to the Aurora Team at the ART Steer department at Volvo Cars for providing a workspace at their office and for committing their resources. Special thanks to Pujun Zhu for being our company supervisor, for guiding the research and for giving feedback on the thesis.

REFERENCES

[1] X-Engineer. “Essential aspects of the V-cycle software development process”, 2020, https://x-engineer.org/graduate-engineering/modeling-simulation/model-based-design/essential-aspects-of-the-v-cycle-software-development-process/

[2] Shahbakhti, Mahdi, Jimmy Li, and J. Karl Hedrick. "Early model-based verification of automotive control system implementation." *2012 American Control Conference (ACC)*. IEEE, 2012.

[3] VersionOne. “The 13th Annual State of Agile™ Survey Report”. 2019. https://www.stateofagile.com/

[4] VDA QMC Working Group 13 / Automotive SIG. Automotive SPICE: Process reference model, process assessment model. Version 3.1. 2017. http://www.automotivespice.com/fileadmin/software-download/AutomotiveSPICE_PAM_31.pdf

[5] R. Yin, Case study research: Design and methods, 3rd ed. Beverly Hills: Sage, 2003.

[6] Bergqvist, Jacob, and Navid Gordani Shahri. *Large-scale agile transformation-A case study of Volvo Cars' transformation process*. MS thesis. 2018.

[7] Maro, Salome, Jan-Philipp Steghöfer, and Miroslaw Staron. "Software traceability in the automotive domain: Challenges and solutions." *Journal of Systems and Software* 141 (2018): 85-110.

[8] Diebold, Philipp, Thomas Zehler, and Dominik Richter. "How do agile practices support automotive SPICE compliance?." *Proceedings of the 2017 International Conference on Software and System Process*. 2017.

[9] Amalfitano, Domenico, et al. "Improving traceability management through tool integration: an experience in the automotive domain." *Proceedings of the 2017 International Conference on Software and System Process*. 2017.

[10] Komiyama, Toshihiro, et al. "Improvement of Agile Software Development Process Based on Automotive SPICE: A Case Study." European Conference on Software Process Improvement. Springer, Cham, 2019.

[11] Mueller, Markus, et al. Automotive SPICE in Practice: surviving implementation and assessment. Rocky Nook, Inc., 2008.

[12] Kähkönen, Tuomo and Abrahamsson, Pekka. “Achieving CMMI Level 2 with Enhanced Extreme Programming Approach.” *PROFES*, 2004.

[13] Runeson, P., Höst, M. Guidelines for conducting and reporting case study research in software engineering. *Empir Software Eng* **14,** 131 (2009). https://doi.org/10.1007/s10664-008-9102-8

[14] Krugler Maag Cie. “Agile in automotive, Pocket guide”. Agile and automotive SPICE v3.0. Kugler Maag Cie GmbH, Germany. 2015. p 98-101.

[15] Orecka, Anna, Sebastian Dawid, and Rafał Dzianach. "Best Practices for Achieving Automotive SPICE Capability Level 3." *International Conference on Software Process Improvement and Capability Determination*. Springer, Berlin, Heidelberg, 2012.

[16] Scaled Agile. “SAFe for Lean enterprises 5.0”. November 2019. Source: https://www.scaledagileframework.com/

[17] Petersen, Kai, and Claes Wohlin. "Context in industrial software engineering research." *2009 3rd International Symposium on Empirical Software Engineering and Measurement*. IEEE, 2009.

[18] Zelkowitz, M. “The Engineering of Large Systems”. Academic Press, 1998. Page 23-29

[19] CPrime. “What is agile? What is Scrum?”. 2020. https://www.cprime.com/resources/what-is-agile-what-is-scrum/

[20] Schweigert, T, et al. "Agile maturity model: analyzing agile maturity characteristics from the SPICE perspective." Journal of Software: Evolution and Process 26.5 (2014): 513-520.

[21] Hantke D. “An Approach for Combining SPICE and SCRUM in Software Development Projects”.2015. In: Rout T., O’Connor R., Dorling A. (eds) Software Process Improvement and Capability Determination. SPICE 2015. Communications in Computer and Information Science, vol 526. Springer, Cham.

[22] Komiyama, T., Shigeo, K., Watanabe, T., Matsui, S., Kase, M., Igarashi, I.. “Improvement of Agile Software Development Process Based on Automotive SPICE: A Case Study”. 2019. 10.1007/978-3-030-28005-5_40.

[23] A. Mulhall, “In the field: notes on observation in qualitative research”, Journal of Advanced Nursing, vol. 41, no. 3, pp. 306-313, 2003. Available: 10.1046/j.1365-2648.2003.02514.x.

[24] Steghöfer, Jan-Philipp, et al. "Challenges of Scaled Agile for Safety-Critical Systems." *International Conference on Product-Focused Software Process Improvement*. Springer, Cham, 2019.

[25] M. Broy, I. H. Kruger, A. Pretschner and C. Salzmann, "Engineering Automotive Software," in Proceedings of the IEEE, vol. 95, no. 2, pp. 356-373, Feb. 2007, doi: 10.1109/JPROC.2006.888386.

[26] Horvath, K. “Bidirectional Requirements Traceability: Why is it such a big deal?”. 2015. Intland Software.

[27] Berczuk, S. (2007). Back to basics: The role of agile principles in success with a distributed scrum team. In Agile Conference (AGILE), 2007 382-388. IEEE.

[28] Vaismoradi, Mojtaba, Hannele Turunen, and Terese Bondas. "Content analysis and thematic analysis: Implications for conducting a qualitative descriptive study." *Nursing & health sciences* 15.3 (2013): 398-405.

[29] Russell, J.P. The ASQ Auditing Handbook, Fourth Edition. 2012. ASQ Audit Division. ISBN: 978-0-87389-847-8

[30] A. Hrobjartsson et al., Observer bias in randomized clinical trials with measurement scale outcomes: a systematic review of trials with

both blinded and nonblinded assessors. Canadian Medical Association Journal, vol. 185, no. 4, pp. E201-E211, 2013. Available: 10.1503/cmaj.120744.

[31] Barriball, K. Louise, and Alison While. "Collecting data using a semi-structured interview: a discussion paper." *Journal of Advanced Nursing-Institutional Subscription* 19.2 (1994): 328-335.

[32] Altexsoft. “Technical Documentation in Software Development: Types, Best Practices, and Tools”. 2019, Mars. Altexsoft, software r&d engineering.

[33] 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, pp. 81-89

[34] J. Peter, “Construct Validity: A Review of Basic Issues and Marketing Practices”, Journal of Marketing Research, vol. 18, no. 2, p. 133, 1981. Available: 10.2307/3150948.

[35] SystemWeaver. “SystemWeaver”. 2020. https://www.systemweaver.se/

[36] Birt, Linda, et al. "Member checking: a tool to enhance trustworthiness or merely a nod to validation?." *Qualitative health research* 26.13 (2016): 1802-1811.

Appendix

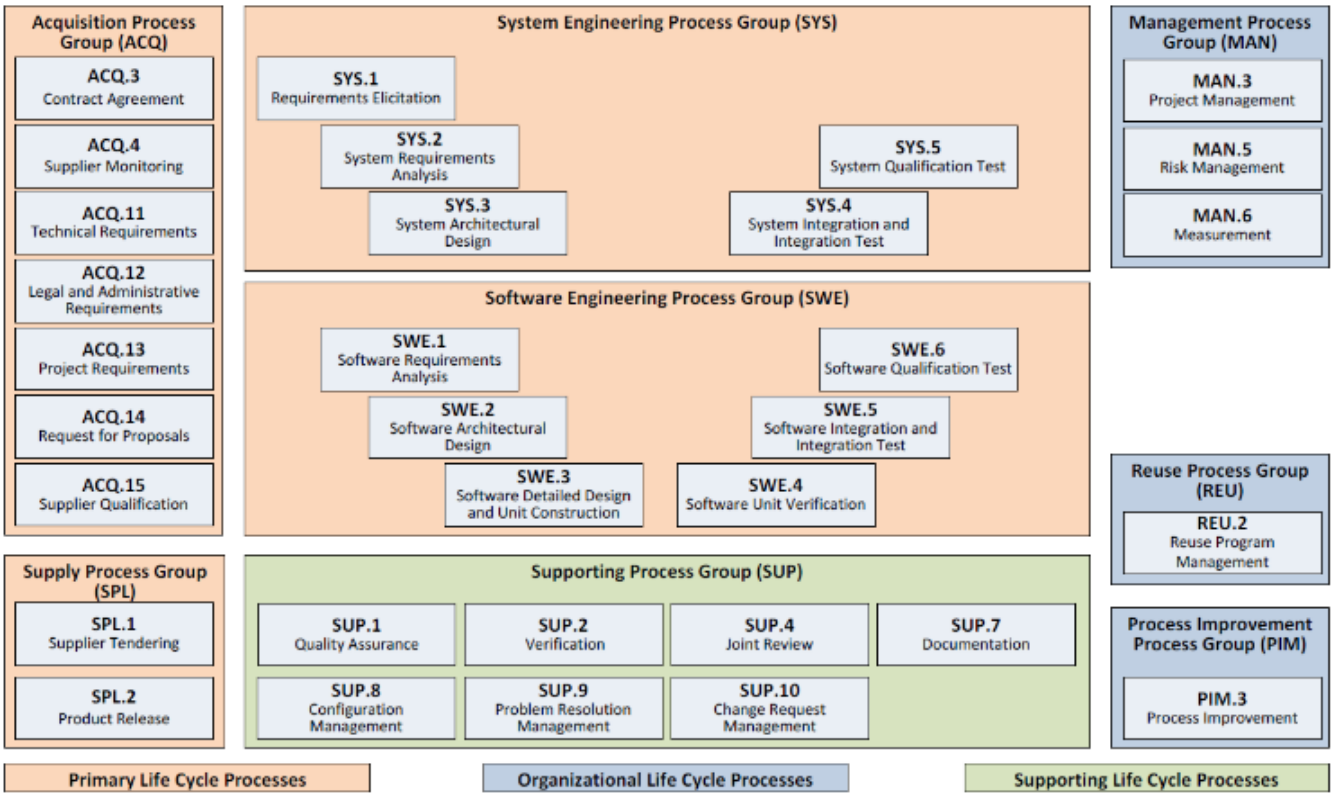


Fig. 7. Overview of A-SPICE process reference model [4]

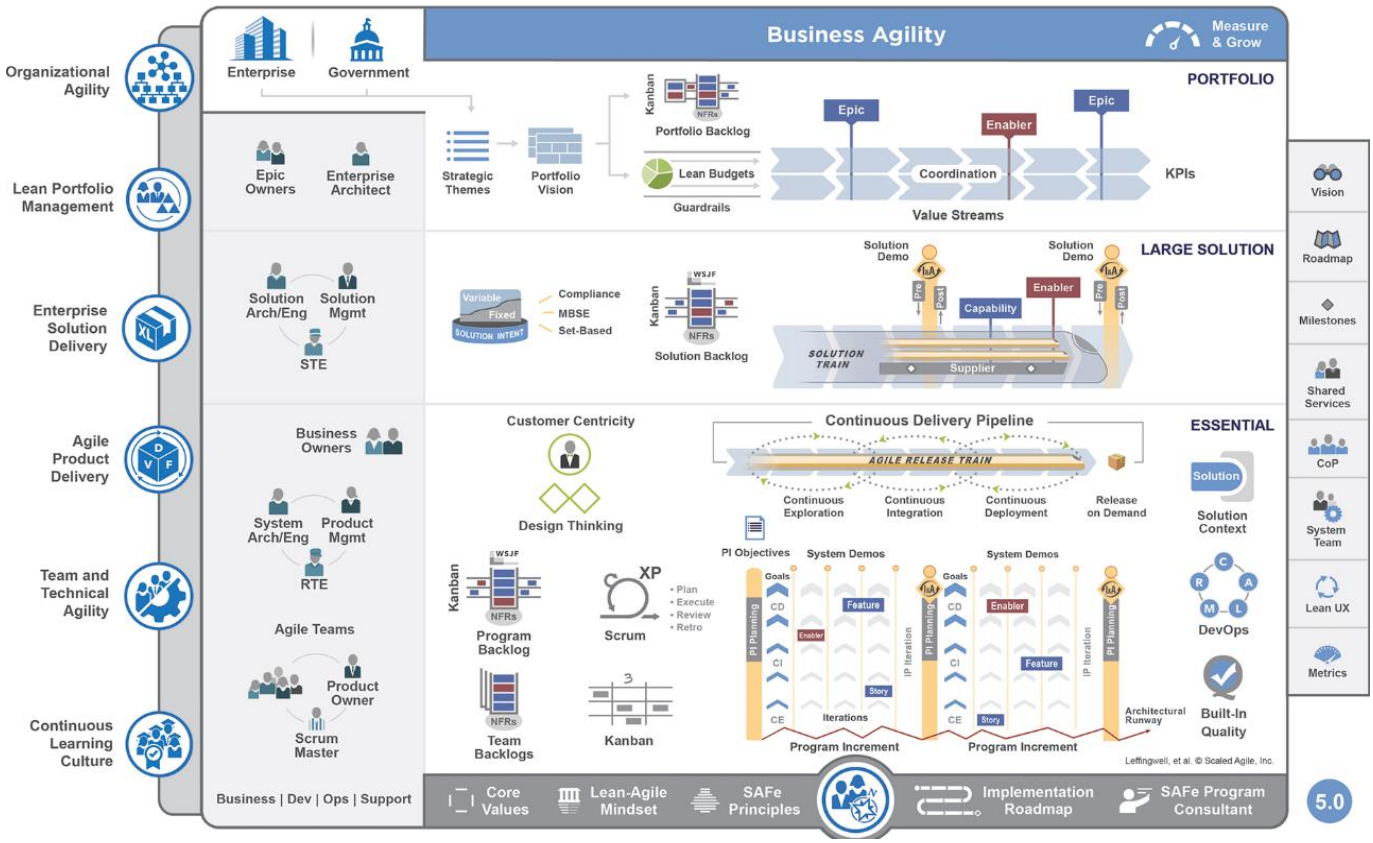


Figure 8. SAFe 5.0 agile framework – Full SAFe Overview (Scaled Agile, Inc. “Safe 5.0”)

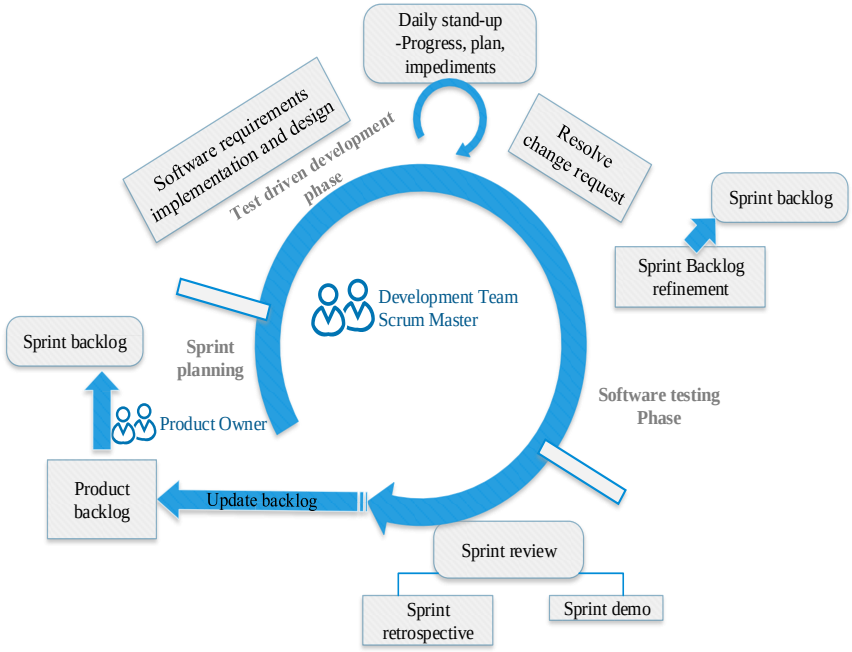


Figure 9. Agile development process – Scrum Team level Overview