Improving Multi-domain Stakeholder Communication of Embedded Safety-critical Development using Agile Practices: Expert Review

Surafel Demissie[a], Frank Keenan, Róisín Loughran[b] and Fergal McCaffery[c]
Regulated Software Research Centre, Dundalk Institute of Technology and Lero, Dundalk, Ireland
{*surafel.demissie, frank.keenan, roisin.loughran, fergal.mccaffery*}*@dkit.ie*

Keywords: Agile Software Development, Embedded Medical Software, Safety-critical, Communication Challenges, Software Process Improvement.

Abstract: The development of embedded safety critical software is different from ordinary software development as such development needs to be coordinated with the hardware development. A typical embedded system project involves multi-domain experts such as business unit, software developers, hardware engineers and firmware developers. Agile methods have been successfully adopted in software engineering in general, and more recently in embedded safety critical development. A previous systematic literature review (SLR), conducted as part of this research, reported that one of the challenges of embedded safety-critical software development is multi-domain stakeholder communication. Additionally, suitable agile practices which have been used in embedded safety critical domains have been investigated. This earlier work proposed a process using a combination of suitable agile practices to support multi-domain stakeholder communication. In order to validate this proposed process, an expert review has been conducted. This paper outlines the proposed process and the findings of the expert validation.

1 INTRODUCTION

An embedded system is a special-purpose computer that is designed to perform a specific task. Today such systems are everywhere in our day to day life from household items such as a digital camera, refrigerator and TVs, to complex and critical devices like pacemakers and smart grid control units (Vahid and Givargis, 2000). A typical embedded system consists of hardware and software components. The hardware includes a microprocessor or microcontroller, memory, input-output (I/O) and additional components such as sensors and actuators which enable interaction with the environment. The embedded software is application specific as it is dedicated to performing predesigned specific tasks repeatedly, thereby controlling the functionality of the hardware device.

Due to its proximity to the hardware, the development of embedded software depends on the corresponding hardware development process as it must provide correct functionality. Such a parallel development process of hardware and software is known as Co-design (Wolf, 1994) (Berger, 2002) (Teich, 2012). The hardware and software development activities require diverse stakeholders such as hardware, firmware and software developers that must have close interaction and knowledge sharing (Rong et al., 2014).

Agile Methods (AMs) (Highsmith and Cockburn, 2001) are an umbrella of software engineering methods that are based on iterative, incremental and evolutionary software development process. Numerous AMs exist with Scrum (Schwaber and Beedle, 2001) and eXtreme Programming (XP) (Beck, 2007) being the most popular. AMs recommend a high degree of expert customer involvement, ability to incorporate changing requirements and short development cycles producing working software. Previous studies of agile implementation in safety-critical embedded domains report both benefits and challenges due to the hardware and software dependency.

The SLR conducted as part of this research investigated the *challenges* related to agile implementation and *suitable agile practices* in embedded safety-critical domains (Demissie et al., 2018). The review was conducted following a review protocol that defined research questions, selected digital libraries, search strings, inclusion and exclusion criteria and data extraction procedure. The result of the study re-

[a] https://orcid.org/0000-0001-6475-2402
[b] https://orcid.org/0000-0002-0974-7106
[c] https://orcid.org/0000-0002-0839-8362

vealed that one of the challenges reported in most of the studies was multi-domain stakeholder communication that occurred due to the diversity between participating members.

Most of agile practices identified are from Scrum and XP methods. The latest report from (VersionOne.Inc, 2018), the largest and longest-running survey on agile, states that Scrum and XP are continuing to be the most common AMs that constitute about 70% of agile usage. Agile software development (ASD) encourages team-based communication through practices such as Cross-Functional Teams, Pair Programming (PP) and Daily Stand-up Meetings.

The present paper attempts to answer the following research question:

> *How can we improve Multi-Domain stakeholder communication within embedded safety critical domains using a combination of suitable agile practices?*

In order to address this research question, a process has been proposed. Table 1 shows the list of agile practices selected for the proposed process.

Table 1: Suitable Agile Practices.

Method	**Agile Practice**
Scrum	Daily Scrum (Stand-up) Sprint Retrospective Scrum
XP	User Stories On-Site Customer The Planning Game Pair Programming (PP) Continuous Integration Refactoring Test-Driven Development (TDD)
Acceptance Testing (AT) & Acceptance-Test Driven Development (ATDD)	

To validate the proposed process, expert reviews have been conducted. The purpose of this paper is to present the proposed process and the expert reviews conducted. The next section presents the description of the proposed process. Next, the expert review is presented and the results discussed. Finally, some conclusions and possibilities for future work are outlined.

2 PROPOSED PROCESS

The Sync-Up process has been proposed to support multi-domain stakeholder communication within embedded safety-critical software development. The initial version of the proposed process has been validated through expert review. In the following sub-sections, we present the latest version of the proposed process.

2.1 The Sync-Up Process

The Sync-Up process is developed by combining the agile practices identified as most suitable through the SLR.The overall approach is proposed based on the foundation of Acceptance-Test Driven Development (ATDD).In standard ATDD, a group consisting of a software developer, a customer and a tester, will gather to discuss and distill on user stories and develop acceptance tests (ATs) and examples for each user stories (Pugh, 2010; Gärtner, 2012). Additionally, they will also collaborate on the development and demo phases.In order to support multiple stakeholder communication in an embedded safety critical domain, the standard ATDD will be extended to involve hardware, firmware and embedded developers. We called the combined team the Embedded System Design (ESD) team.The ESD team will sync up through the whole project.Figure 1 shows the latest version of the proposed process. The detailed steps of the process are presented below.

Step:1: Define User Stories, Prioritise, Estimate User Stories & Develop Acceptance Tests (ATs)

In Step 1, user stories will be written for the feature that is under consideration. When writing User Stories, having a combination of engineers from each domain, such as the ESD team, will help the team write the User Stories with a complete understanding of the system from embedded hardware and software perspective. For each User Stories, the team also writes acceptance tests (ATs). ATs are conditions of satisfaction that are conducted to "determine whether a system satisfies its acceptance criteria (i.e., initial requirements and current needs of its user) and to enable the customer to determine whether to accept the system." (IEEE Standard 610, 1998).

After defining the User Stories and ATs, the team will prioritise the User Stories based on features with more business values and dependencies. After prioritisation, the team will estimate the User Stories. Estimation is a process of deciding the amount of time it takes to complete the implementation of the User Stories. Having diverse members to collaborate on estimation helps the team to achieve better estimates. The final activity of Step 1 is the release plan. The release plan describes which feature will be delivered in the upcoming release. In release planning, a list

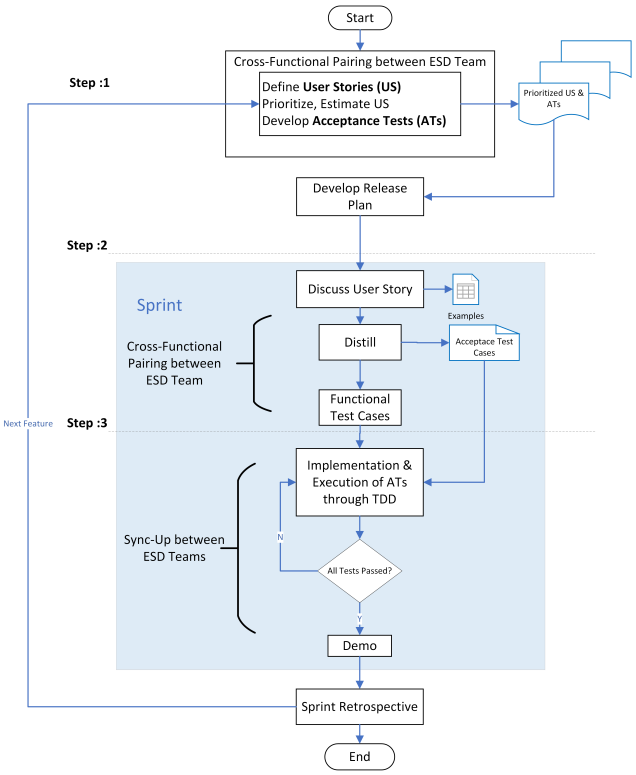

Figure 1: The Sync-Up Process.

of User Stories being considered in the coming Sprint will be decided and the team will decide to commit on the completion of the selected user stories.

Step: 2: Sprint, Discuss & Distill User Stories

Step 2 is composed of sprint activities such as Sprint Planning, Daily Scrums and Sprint Review. Before the implementation of the User Stories, the ESD team will implement the ATDD steps, discuss and distill. During the discuss stage, the ESD team members will collaborate and develop examples. The examples developed will help to clarify technical constraints and feasibility from the embedded hardware and software aspects of the feature under consideration. After defining the examples, members of the ESD team will implement the distill stage. In this stage, the team will capture the examples in a format that works with the test automation framework. These test cases will be written using cross functional pairing involving technical experts of the ESD team. As a result of pairing between the technical experts, the test cases written will have a better structure and constraints.

After defining acceptance test cases, the ESD team will develop functional test cases. These test cases are written from user perspective to ensure that each part of the system is tested against the functional specifications.

Step 3: Implementation and Execution of ATs

In this step, members of the ESD team will execute the ATs collaboratively. Using a Cross-functional Pairing, the engineers will collaborate during the execution of the test cases. This stage is executed until all ATs for the current iteration are passed. The implementation of ATs is conducted following the principles of TDD. In TDD, developers first write a failing test. This is followed by writing the minimum amount of code that is required to get the ATs passed (Green). Once the ATs are green, members of the ESD team will conduct Exploratory Testing through cross-functional pairing. For example, the tester and developer can pair to test the edge computations. Additionally, the embedded engineer and the software developer can pair to test the behavioural and performance-related parameters from the embedded perspective.

The ESD team will fix any defects throughout the development phase through the Sync-Up process. Once all tests are passed, the User Stories will be marked as done and the ESD team will move to the demo stage to demo the feature to the product owner (PO).

The final activity of Step 3 is the implementation of Sprint retrospective. It is a continuous improvement activity that is held prior to the next Sprint. The objective of this activity is to inspect the previous Sprint with regards to people, relationships, process, and tools and plan for improvements on the next Sprint. In order to validate the proposed process, we have conducted an expert review. The result of the review will be presented on the next section.

2.2 Expected Benefit of the Sync-Up Process

The proposed process will enable User Stories and ATs to be written syncing with the technical experts of the ESD team. The involvement of multi-domain stakeholders earlier and capturing requirements in the form of ATs can reduce the communication gap of such diverse members. One of the main benefit of ATs is to have clear requirement and improve collaboration. The ATs written in collaboration with experts of the ESD team should have clear technical specifications that satisfy all stakeholders.

3 EXPERT VALIDATION

The expert reviews have been conducted to validate the proposed process. Experts have been selected based on their knowledge and experience on agile practices and their implementation within embedded safety critical domains. The main criteria that have been used for selecting the experts were:

- Having many years of experience implementing agile software development either as Product Owner, Scrum Master, Developer or Consultant;
- Having experience in embedded safety critical domains;
- Having experience in implementing or coaching the implementation of specific agile practices such as ATDD.

The search for experts was conducted through LinkedIn, a social network for professionals. The researcher contacted candidate experts that satisfied the criteria and to date three experts that were willing to evaluate the proposed process were selected as expert reviewers.

Expert 1 is a speaker, consultant and author of a dozen books on agile, Lean and managing high-technology product development. The expert has helped managers, teams and companies to move to an agile approach.

Expert 2 has fourteen years of experience in software testing of embedded Medical devices such as injection devices and infusion pumps, smart lighting embedded software and various embedded test automation tools projects. At the time of the review, the expert was working on two projects. On one of the projects, he is a Scrum Master that involves eight people composed of software developers, software testing and product owner. On the other project, he is working as delivery manager that involves hardware engineers, firmware developers, mobile app developers and quality assurance (QA).

Expert 3 has over twenty years of experience in different industries where he worked in many different roles including developer, tester, analyst, product manager, test manager, and agile/lean coach. The expert had a well-known case study on the implementation of ATDD.

For each expert, the meetings were conducted using video conferencing tools Skype and Zoom. For Expert 1 and Expert 2, two meetings have been conducted. For Expert 3, one meeting was conducted as a result of the availability of the expert and a walk-through of the process was presented and the expert was asked for his advice. For the first two experts, the first meeting was an introductory session where the researcher and the experts would get to know each other. In this meeting, the experts were asked questions related to their experience in agile software development, embedded software development and challenges faced by the experts. The researcher then presented a walk-through of the proposed process. Expert 2 was willing to see the proposed process with a practical example, so the researcher presented an ideal example to demonstrate the expected outcomes of the process.

On the second meeting, the experts were asked to provide their advice on the proposed process. We asked the experts to point out the deficiency observed, benefits, improvement and other suggestion they would like to add. The list of presented questions is given in the accompanying appendix. With the consent of the experts, the video conferences were recorded and transcribed for reference. The summary of the review conducted with each expert is presented in the following sub-sections.

3.1 Challenges and Experience

Expert 1

This expert was initially asked about the experience of implementing agile and the challenge of multi-domain stakeholder communication. The expert re-

sponded that from her experience working with embedded clients, hardware teams have always been "almost separated from firmware teams". The software team, on the other hand, would have often been in a third location. The expert also stated that even if the teams were in the same location, they would have been on "different forces". The separation of such diverse teams was creating a number of long integration groups. The expert noted that for embedded software, most problems show up after the product is in the field and this would make it "really difficult to tell where the problem arises". When asked to comment on the benefits of the syncing up, the expert stated that to have a coherent User Stories that has important components from the architecture, we need to sync up various layers such as the application program interface (API), middle-ware and the platform which encompasses both firmware and hardware. The expert stated that user stories should impact on the architecture, not just the software side. Additionally, the expert noted that ATs should also include the hardware acceptance, the firmware and the mechanical aspects of the system. The expert used an example of a client that implemented User Stories for software and firmware and design by contact for the hardware. The client used separate Kanban for software and firmware User Stories, so the firmware was always verified in advance of the software. The expert advised the client that the firmware and software teams have to work together otherwise they would run into problems because the cycle time of the firmware Kanban and the software will be different. The expert went on to state that the client didn't like the proposed advice because the client was not looking at the story cycle time, merely the software and firmware. The expert stated that we might end up with big teams involving software, firmware and hardware developers, but all stakeholders should come together and create the User Stories. Additionally, having the firmware and mechanical experts will help the team to consider the implications of the firmware and mechanical components on the user stories.

Expert 2

The expert stated that he has been working with diverse teams such as hardware, firmware, application software and scientist teams. The expert noted that the hardware development team were not following agile and they will work at "their own pace". The firmware team on the other hand, would work based on the initial available hardware. Additionally, the application software team will be completely relying on the embedded software. The expert noted that such diverse teams haven't been integrating their tasks properly.

The expert has given an example of an endoscope project from his previous experience. He stated that on the project, the teams initially agreed to use an Android operating system for displaying the real-time image of the endoscope camera. The selection of this operating system would have "some delays in microsecond" on the processing of the image. During the requirement analysis and development phases, the stakeholders had all agreed on the delay and the team delivered first build and went for a trial with doctors and scientists. Once it went into the doctors and scientists "they found that this delay was unacceptable".

The expert stated that this change cost the team around four months of delay because they needed to completely change the operating system from Android to Linux. The change in the operating system required additional changes to the video connect and the communication protocols. The expert stressed that the involvement of some of stakeholders, such as the scientist team, at the later stage created "huge delay" and a clear example of the communication gap and miss-collaboration of diverse members.

When asked to comment on the benefits of the syncing up, the expert stated that most of the time failure occurs because stakeholders were not reviewing and analysing requirements and test cases. The expert referred to a project on which he was working. In this project, the teams initially agreed on a test case and the developers started based on this test. The expert stated that when about 40% of the development was completed a reviewer team of stakeholders started reviewing the test cases and found that the test cases were "not something which they were expecting". The expert went on to state that this change has ended up wasting 40% of the time of the software developer, firmware engineers and test protocol designers because detailed test plan development was already started. The expert stated that all members of the stakeholders should sync up early and agree on the user stories and test cases before development started, to avoid the cost of rework

Expert 3

When asked to comment on the benefits of the Sync-Up process, the expert stated that he believes the process can work and help the ESD team deliver faster. He also noted that we will have problems in convincing organisations to change to this process. According to the expert, problems will come mainly from management who will not understand that "pairing people doesn't mean we do less work". The expert went on to state that the main reason we pair people is because "we want to inject quality" at the beginning rather than detecting problems at the end. He noted that dur-

ing the development phase, if we “find issues late”, there will be a lot of rework. By injecting some quality at the beginning of this phase as a result of pairing, we can remove the rework. The expert noted that the impression he got from leaders is, if people are paired, then they will only get one thing done instead of two things that they could work and “this could be a problem”. To overcome this the expert emphasised the importance of “coaching managers”and explaining the concepts from Lean on limiting work in progress (WIP). He stated that by limiting WIP we can actually get more things done. According to the expert, occupying the members of the team only for about 70% of their time is more likely produce more work than if we occupy them for 100%.

3.2 Comments on a Process

In the first part of the review conducted with each expert, we have explored the experience of the experts and the challenge of multi-domain stakeholder communication.The experts have reported examples projects where the communication gap between multi-domain stakeholders created project delays and time-wasting of developers and engineers. The experts also highlighted the importance of syncing up between multi-domain stakeholders earlier before development is started to avoid the communication gap.The second section of the review includes the researcher presenting the walk-through of the proposed process and each expert was asked to give their comments on the proposed process.The summary of the comments given by each expert will be presented in the following subsections.

3.2.1 Expert 1

A walk-through of the proposed process was presented to the expert and the expert was asked to provide the deficiency and recommendations observed. She stated that if the hardware, software and firmware teams in their own silos, separately sync up their activities after some period of time, “that’s not an agile”.The expert suggested that the activities in Step 1,define User Stories and ATs should include all members of stakeholders in one cross-functional team.The expert stated that in this stage syncing up is “non-existence” as we’re part of a cross functional team with representative of all layers of architecture.The expert also noted that the discussions about the features such as the discuss and distill stages in Step 2, should also include the entire team.The expert went on to state that in the implementation phase, Step 3, we might need “little sync ups” frequently.In her experience the hardware and firmware teams iterate on the design and create tooling for simulations.The expert stated that as long as simulations are available, the hardware and firmware teams can have something to share with other members such software developers and syncing up possible.

3.2.2 Expert 2

The expert stated that Step 1 can be implemented in two ways.The first way suggested by the expert was that members of stakeholders all comes together and discuss and create their requirements.The expert went on to state that the best scenario he could suggest would be if the product owner (PO), Scrum Master, technical expert from each domain like the hardware lead, the firmware lead, application lead and testing member all involved and sync up to create User Stories and ATs.The expert went on to state that hardware teams usually work in multiple activities and depends on tools, labs and also their work is usually depend on vendors.The expert stated that to incorporate the changes in line with other members such as firmware and software teams, sync up is important.

The second way suggested by the expert was if multi-domain stakeholders create a cross-functional team and write the User Stories and ATs. The expert noted that from his experience, creating a cross-functional team in embedded domain is difficult. The expert went on to state that the hardware team dynamics are different than software and their working pattern is also different. According to the expert, the hardware team generally work in a pure waterfall model. Software team on the other hand follows the design of hardware team and adopt the changes proposed by hardware team. Despite the difficulty of creating a cross functional team, the expert suggested that the activities in Step 1, define User Stories and ATs, can be implemented using cross-functional teams composed of each domain.

For Step 2, the expert stated that functional test cases should be included on the proposed process. The expert stated that after the discuss and distill on the User Stories, the functional test case will be created and at the same time the developer would be developing the code. The expert stated that functional test cases are elaborated test cases that are used to test detailed functional requirement.

3.3 Expert 3

This expert was asked to give his advice on the stages of ATDD. In his experience, he preferred to do the examples with the discuss and distill practices after starting the Sprint. The expert noted that the discuss

and distill practices are very focused and demanding activities and we cannot expect people to do them for more than an hour or more. Based upon his experience, he found that it's "more effective" to discuss and distill on the User Story bases. The expert went on to state that when we start our Sprint, we can take a User Story and get the stakeholders together and design the examples. Once each of the stakeholders are happy with the examples, developers will start writing the functional tests cases and implement the code for those specific User Story.

The expert also suggested that the discuss and distill activities could result in the splitting of the User Stories. The expert went on to state that when we define User Stories, and get people to write examples, we will find out very soon weather the User Stories is too big and needs to be spilt. According to the expert, if there are three or more examples in a User Story, the expert suggested that the User Story is "already too big". The expert suggested at that point, the team can identify a set of examples and by looking at the examples, the team can find a way of splitting the User Story in two or three. And then we will add the User Story to the current Sprint or to the next Sprint depending on how priorities are stored.

All three experts were presented with the walkthrough of the proposed process and the experts were asked to give their advice on any deficiency they observed, and any improvements they may suggest. The improvements and suggestions made by each expert were taken into consideration and applied to the initial proposed process. The improvements suggested by the experts is presented in the following sub-sections.

4 DISCUSSIONS

The expert review was conducted with three experts to validate the proposed process. Initially, the experts were asked about their experiences and the challenge of multi-domain stakeholder communication. The experts stated that multi-domain stakeholder communication challenges have affected their projects. The experts discussed some example projects where the involvement of some members of the stakeholders late in development led to significant delays. Additionally, from the experience of the experts, the diverse teams involved in embedded projects had several integration points. All three experts stated that multi-domain stakeholders have to sync up earlier when analysing requirements and should have a common understanding.

Improvement 1

According to Expert 1, the way diverse teams sync up in different stages has to be clarified. She suggested that the initial stages of defining the User Story and ATs should be conducted using a cross-functional team. On the other hand, the expert stated that during the development stages, Step 3, we will need little sync ups as separate silos with simulations on the hardware and continuous integration with the software. Expert 2 on the other hand stated that Step 2 activities can be implemented using cross functional team. This expert noted the difficulty of creating cross functional teams in embedded projects. Expert 2 suggested that he prefers if the hardware lead, the firmware lead, application lead, the PO and SM all comes together and create the User Story and ATs. This expert highlighted the importance of syncing up during the development phase.

- On the initial version of the proposed process, the Sync-Up process was designed to involve only the embedded engineer to sync up with development teams in all phases. The changes to the initial process made as part of feedback from the two experts, attempt to resolve this issue i.e. The ESD team constitute a cross functional team and write the User Story and ATs using cross-functional pairing.
- Both experts stated that syncing up is important in development phase, i.e. Step 3. For this phase we have kept syncing up between ESD teams.

Improvement 2

Expert 2 stated that functional test cases should be discussed after starting the Sprint. The expert stated that these test cases are different from acceptance test cases and focus on if the system is functioning as expected by the users.

> On the initial version of the proposed process, functional test cases were implemented when developing ATs. On the latest version of the proposed process, the implementation of functional test cases, have been placed after the discuss, distill activities.

All three experts stated that multi-domain stakeholders need to sync up during development frequently in their separate silos. The experts stated that the frequency of syncing up during development will depend on the availability of simulations and the cost of prototyping. The experts stated that when the hardware is not ready, the simulation and prototyping of hardware can be used by software and firmware teams

in advance and development can be started. As stated by Expert 1, for a big machine, prototyping can be very expensive. On the other hand, for small devices such as wearable items, prototyping can be done using board prototypes.

5 CONCLUSIONS

In this paper, we discussed the challenges within multi-domain stakeholder communication in embedded safety critical domains. In order to deal with these challenges, this paper has proposed a Sync-Up process using a combination of suitable agile practices. The validation of the Sync-Up process was conducted through expert reviews. The experts involved have shared their experiences on the challenges of multi-domain stakeholder communication and the importance of involving all stakeholders when we define the user story and ATs. The improvements and suggestions made by each expert have been taken into account and changes have applied to the initial process. In the future, we plan to conduct the second validation of the updated Sync-Up process through case studies.

ACKNOWLEDGEMENTS

This work was supported with the financial support of the Science Foundation Ire-land grant 13/RC/2094 and co-funded under the European Regional Development Fund through the Southern & Eastern Regional Operational Programme to Lero - the Irish Software Research Centre (www.lero.ie).

REFERENCES

Beck, K. (2007). Embracing Change with Extreme Programming. *Archives of disease in childhood. Fetal and neonatal edition*, 92(October 2008):F83–F88.

Berger, A. (2002). *Embedded Systems Design—An Introduction to Processes, Tools, and Techniques s*, volume 2002. CMP Books, CMP Media LLC, Lawrence, Kansas 66046.

Demissie, S., Keenan, F., Özcan-Top, Ö., and McCaffery, F. (2018). Agile Usage in Embedded Software Development in Safety Critical Domain–A Systematic Review. In *International Conference on Software Process Improvement and Capability Determination*, pages 316–326. Springer.

Gärtner, M. (2012). *ATDD by example: a practical guide to acceptance test-driven development*. Addison-Wesley.

Highsmith, J. and Cockburn, A. (2001). Agile software development: the business of innovation. *Computer*, 34(9):120–122.

IEEE Standard 610 (1998). IEEE Standard Glossary of Software Engineering Terminology. Standard, Institute of Electronical and Electronics Engineers, Washington, DC.

Pugh, K. (2010). *Lean-Agile Acceptance Test-Driven-Development: Better Software Through Collaboration*. Pearson Education.

Rong, G., Liu, T., Xie, M., Chen, J., Ma, C., and Shao, D. (2014). Processes for Embedded Systems Development: Preliminary Results from a Systematic Review. *Proceedings of the 2014 International Conference on Software and System Process*, pages 94–98.

Schwaber, K. and Beedle, M. (2001). *Agile Software Development with Scrum*, volume 18.

Teich, J. (2012). Hardware/software codesign: The past, the present, and predicting the future. *Proceedings of the IEEE*, 100(SPL CONTENT):1411–1430.

Vahid, F. and Givargis, T. (2000). *Embedded Systems Design: A Unified Hardware/Software Introduction*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition.

VersionOne.Inc (2018). 12th annual state of agile report. Technical report.

Wolf, W. H. (1994). Hardware-software co-design of embedded systems. *Proceedings of the IEEE*, 82(7):967–989.

APPENDIX

Background and Experience

1. What's your experience with agile software development?
2. Tell me about your experience with embedded system development?
3. What's your experience with agile in embedded system development (where you have the hardware, software development teams working in parallel)?
4. From your experience, have you experienced with communication challenge between diverse members?

The Sync-Up Process

1. Can you name and explain the major benefits you have observed in the Sync-Up Process?
2. Can you name and explain briefly any deficiency you have observed in the Sync-Up Process?
3. Do you have any suggestions to improve the Sync-Up Process?
4. Is there anything else you would like to mention about this process?