

This is the post-print version of the manuscript

Agile Usage in Embedded Software Development in Safety Critical Domain—a Systematic Review

Surafel Demissie, Frank Keenan, Özden Özcan-Top, Fergal McCaffery

Regulated Software Research Center, Dundalk Institute of Technology and Lero
Dundalk, Ireland

{surafel.demissie, frank.keenan, ozden.ozcantop, fergal.mccaffery}@dkit.ie

Abstract. Safety critical embedded software is a specific type of embedded software that needs to provide correct functionality to avoid loss of human life. Embedded software controls much of the functionalities in Medical, Automotive, Aerospace and Cyber-Physical-Systems. The development of embedded software is different from ordinary software development as such development needs to be coordinated with the hardware development. Additionally, regulation processes and audits are also in place before placing the products to market. The objectives of this study are to understand the challenges of embedded safety critical software development, to investigate agile practices which have been in use in the domain, the factors affecting agile implementation in embedded safety critical software development. We have performed a systematic review and a mapping study to achieve these objectives. This paper outlines the result of the systematic review and mapping study.

Keywords: Agile Software Development, Embedded, Medical Domain, Safety Critical, Software Development Challenges.

1 Introduction

Nowadays embedded systems (ES) are everywhere from home appliances, wearable devices and electric cars to control systems in complex plants. By 2020, there will be 50 to 100 billion devices that will be connected through the advancement of internet of things (IOT) and embedded systems [1].

ES are composed of two basic components: hardware and software. The hardware component contains microprocessor or microcontroller, memory, input output (I/O) interfaces as well as the user interface. The software in ES is ‘embedded’ inside the hardware and provides control functionalities. Unlike commercial software that focus on algorithm and data processing, embedded software is often written for the specific hardware.

This is the post-print version of the manuscript

Having hardware and software components to constitute the overall ES, the development of ES is characterized by simultaneous development of hardware and software. This is known as *co-design* [2]. A typical ES design life-cycle, as defined by Berger [3], has hardware and software development processes in parallel. Such development processes are dependent on one another and testing of one unit will require stubs of the other and this can be challenging [4].

ES can be simple control units as in printers and cameras or safety critical systems like automobiles and medical devices. Given their criticality, evidence through highly-regulated process is required. For example, in the medical domain, depending on their geographical location companies need to provide evidence that they went through the desired process to get the approval by the regulatory bodies. In the European Union medical devices must have the CE mark [15]. This process includes satisfying standards such as medical device quality management standard (EN ISO 13485:2003) [16], medical device risk management standard (EN ISO 14971:2009) [17] and the medical device product level standard (IEC 60601-1 [18]).

Modern ES functionalities are getting complex and most of these functionalities are relying on the embedded software. For example, infusion pumps today contain tens of thousands of lines of code [5] and this number will go higher for recent premium class automobile which contains close to 100 million lines of software code [6]. With the increasing of complexities, safety critical domains are calling for a better software development practice. For example, in the medical domain, [7] report that complexity is exceeding software maturity and the industry is not taking full advantage of well-known techniques for engineering software.

The development of safety critical ES must deal with challenges at high level concerning certification and regulation and technical challenges associated with ES at a lower level.

One approach that may offer assistance is the agile software development (ASD) [8] which has been a hot topic in safety critical domains in recent times. Agile methods recommend a high degree of expert customer involvement, ability to incorporate changing requirements and short development cycles producing working software. There are numerous agile methods including Scrum [9], eXtreme Programming (XP) [10], DSDM [20] and DevOps [12]. Previous studies of agile implementation in safety critical domains report both benefits and challenges. ES has also been reported to benefit from ASD [21],[22],[23] . But as in safety critical domains, agile implementation in embedded systems also reported to have challenges particularly due to the hardware and software dependency.

The purposes of the study are to reveal the challenges of embedded safety critical software development in practice, to investigate the agile practices which have been in use and the factors affecting agile implementation in the embedded safety critical software development. We performed a systematic literature review to achieve these purposes. The review included 30 studies from Automotive, Medical, Aircraft, Aerospace, Mechatronics

This is the post-print version of the manuscript

and Safety critical domains. The existing literature covers agile usage and challenges in safety critical domain and ES themes separately, the review we performed focuses on agile usage on the safety critical ES. The rest of the paper is structured as follows: In Section 2, we provide the followed review protocol. In Section 3, the results of the review is given. Then we discuss the results for each research question in Section 4. Finally, in Section 5 we conclude the review.

2 The Review Protocol

The systematic literature review has been performed following a review protocol, which defines the research questions, selected digital libraries, search strings, inclusion and exclusion criteria and data extraction procedure. The review protocol was defined in the guidance of [13] and [14].

2.1 Research Questions

In this research, the following research questions have been defined:

- RQ1: What are the challenges related to agile implementation specifically in embedded safety critical domains?
- RQ2: What agile practices have been used in embedded safety critical domains?
- RQ2.1 How are the agile practices extended to ensure regulatory requirements of the safety critical domains?

2.2 Search Strings

The following search strings have been selected and arranged to address the research questions above. In some cases, the search strings have been adapted to suit some of the specific requirements of the digital libraries that were selected in this review.

*(“agile” OR “scrum” OR “XP” OR “extreme programming” OR “test driven development” OR “TDD” OR “lean” OR “DevOps” OR “feature driven development”) AND
 (“embedded” OR “embedded system” OR “embedded software”) AND
 (“Safety critical”)*

This is the post-print version of the manuscript

Additionally, we have applied the snowballing technique to avoid missing any relevant studies [19]. We used the following the digital libraries for the search process:

- IEEE Xplore
- ACM Digital library
- Google scholar
- ScienceDirect
- SpringerLink

2.3 Inclusion/ Exclusion Criteria

The following inclusion and exclusion criteria were defined for the review:

Inclusion Criteria

- Studies on agile implementation for embedded software and embedded system development.
- Studies on agile implementation for embedded safety critical systems.

Exclusion criteria

This is the post-print version of the manuscript

- Studies discussing general agile software development practices (non-embedded).
- Studies that are not in the safety critical domain.

2.4 Data Extraction

After defining the inclusion and exclusion criteria, a data extraction template has been defined on tabulated format on spreadsheet with contents of year, author/title, agile practices, domain, challenges, implementation detail and summary.

3 Results

The search process has been performed applying the keywords on each digital library. All of the search results from each database have been recorded on a spreadsheet. The initial search resulted in a total of 292 studies. In addition to the spreadsheet, we have used the Mendeley¹ tool to manage the organization of the studies. The first screening results (292 studies) have been imported on Mendeley and each study has been analyzed based on the inclusion and exclusion criteria.

After analyzing the individual studies and removing the duplicates, the final screening resulted in a total of 30 studies. The stages of the review process have been discussed and analyzed with/by the senior researchers in this study.

The numbers of the studies found from each digital library after the first and second screening are shown on Figure 1 below.

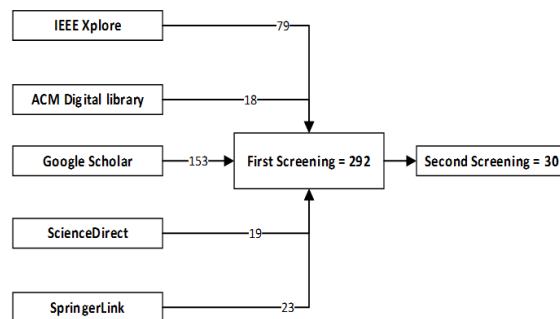


Figure 1 Screening Results

¹ <https://www.mendeley.com/>

This is the post-print version of the manuscript

The studies which passed the inclusion criteria have been categorized in two groups:

- Studies that report agile implementation for embedded at a general level without specifying safety critical domain. These studies include S12, S13, S14, S16, S17, S18, S20, S21, S22, S23, S24, S26, S27, S28 and S29.
- Studies that specifically address the safety critical domain with the embedded software characteristic. These studies includes S1- S11, S15, S19, S25 and S30.

In the first group most of the studies are the case studies and experience reports. Previous systematic reviews such as S22, S26, and S27 have also been identified. S26, which also addresses the previous review S22, report the result of a review that includes agile

implementation with respect to embedded software, hardware and integrated circuit. This review concludes that most of the previous reports are case studies and experience reports and there is lack of rigorous empirical research on the actual benefits of agile methods in embedded domain. A review by S22 addressed the implementation of agile methods in embedded software development. Study S28 performs a mapping of the principles of the agile manifesto to embedded system development. Table 1 summarizes studies that have been categorized in the first group.

Table 1: Studies on ES

Studies	Type
S29	Empirical study
S14, S17, S18, S21, S23	Case study
S12, S13, S16, S20	Experience report
S22, S24, S26, S27, S28	Systematic Review

In the second group a total of 15 studies have been identified to be in safety critical domain. Some of the domains are cyber-physical systems, automotive, medical, aircraft/avionics, mechatronics and general safety critical. Table 2 shows the volume of studies that has been identified from each domain. The majority of these studies are case studies and experience reports. Some of the studies address agile implementation for safety critical systems without addressing embedded systems characteristics. On the other hand, studies such as S2, S3, S4, S6, S7, S10 and S30 address safety critical embedded systems and software.

Table 2: Domain Specific Publications

Domain	Publications	Embedded
Cyber-Physical sy	S11	x
Automotive	S3, S30	x

This is the post-print version of the manuscript

	S1	
Medical	S7, S5	
	S4	x
Aircraft, Avionics,	S30	
	S6	x
Mechatronics	S8	
	S2, S7	x
Safety critical	S9	
	S10	x

3.1 Challenges of Agile Implementation

The collected studies have been analyzed based on challenges and factors affecting agile implementation in embedded safety critical domain. The reported challenges have been categorized in four major areas:

- Hardware development
- Team based communication
- Tools and automation support
- Standards and complex regulation process

3.1.1 Hardware Development

These group of articles discuss challenges of implementing agile where there is hardware development in parallel. The main challenge that has been discussed by the studies was long feedback loop of hardware. Study S2 reports that companies within the embedded systems domain struggle with the alignment of hardware and software development cycles and practices. S10 states the report of practitioners on the difficulty of hardware development to break down in iterations. Another study, S11 also address that the general barrier for using agile methods for hardware development is the higher difficulty in modifying hardware. S12 and S13 state the difficulty of managing hardware iterations which results in different paces of software and hardware development. In some domains, such as mechatronics, mechanical parts also cause long feedback loops S8.

3.1.2 Team Based Communication

This is due to the diversified disciplines occurred in hardware and software development teams. Study S1 reports the challenge of 'individualism and lack of complete knowledge, as well as long communication chains and low cross-function mind set. Study S6 addresses the challenge of team based communication when complex distributed teams are working on different

This is the post-print version of the manuscript

interfaces of a product and results in problems and delays during the products hardware-software integration. S14 reports challenges related to geo separated and multidisciplinary teams in collaboration and integration. The study reports challenges related to the implementation of practices such as pair programming as a result of one engineer having domain specific knowledge and calls for a simple and tangible way to engage hardware engineers in pair programming and increase knowledge sharing.

In S15, S16 and S17 the challenge of team members' specialized domain knowledge and limited knowledge have been addressed.

3.1.3 Tools and Automation Support

This challenge is related to the availability and support of tools to automate and support hardware functionalities. Studies S18 and S19 reports lack of tools to support specific embedded requirements to be challenging. In S19 the challenges of DevOps adoption in the embedded systems domain have been reported. The study categorizes the challenges in four groups, hardware dependency, limited visibility of customer environments with regard to configuring test environment, scarcity of tools and absence of feature usage data in system performance data.

3.1.4 Standards and Complex Regulation Process

The availability of regulation process and standards are reported to bring difficulty in implementing agile in regulated domains. Studies such as S5, S10, and S19 reported this challenge. S10 stated that standards set obstacles for continuous delivery and integration and require special attention. S5 analyzes two standards, IEC 62304 and IEC 82304-1 and reports on adopting DevOps methods in tightly regulated software development of medical devices. The study states that such standards require special attention from continuous integration and prevent using continuous deployment after the deployment to the customer. Therefore, new tools and methods should be specifically developed for using DevOps in regulated software development. Table 3 summarizes the studies with the challenges that have been categorized

Table 3: Challenges Identified

Studies	Challenges
S2, S5, S10, S11,	Hardware development
S1, S6, S7, S14,	Communication
S18, S19	Tools support
S5, S10, S19	Regulation process

3.2 Agile Practices and Implementation

The majority of studies implement a combination of agile practices. In reports such as S4, S13, S20, S21, S22 Scrum is used with combination of XP. In S4 a combination of practices such as *unit tests*, *adaptive planning*, *iterative and*

This is the post-print version of the manuscript

incremental approach have been used. S20 implements the combination of Scrum and XP practices. Practices such as *sprint planning meeting*, *Daily Scrum*, *Sprint review (retrospective)*, *Unit Test*, *Test First* and *Pair Programming*. S14 discusses the implementation of Scrum with *acceptance criteria*. This study stated that practice such as acceptance criteria can be used to define the aim of each of the stakeholders to manage geo separated teams and collaboration.

In some reports Scrum framework has been extended. For example, S11 extend Scrum for cyber physical systems (CPS) known as Scrum-CPS. This report proposes two sprints, design sprint and hardware sprint that synchronizes using the concept of Agile Release Train (ART) Additionally, Scrum has been combined with model driven software development S6 and platform-based design approach S21.

Some studies improve the co-design processes using XP practices such as *system metaphor*, *planning game*, *small release*, *testing*, *refactoring* and *pair programming* S23.

Studies such as S18, S24, and S25 have reported the implementation of TDD. Some of these reports also addressed the challenge of tools support to effectively implement agile practices in the context of embedded systems.

Another study S12 reports the implementation Lean and proposes a framework based on *cadence meetings*. The proposed meeting is a way of reviewing that will be held every 6 weeks at the decision points known as *synchronization*.

4 Discussions

In this review we have investigated the challenges related to agile implementation specifically in embedded safety critical domains (RQ1). The results of our review showed that there are challenges related to agile implementation in safety critical embedded software development.

One of the challenge that has been investigated in our review is hardware development which mostly cause long feedback loops. The long lead time of the hardware development affects agile implementation as hardware loops will be longer than the software development loops. As observed in some of the case studies in mechatronics domain, mechanical development also causes long feedback loops.

In addition to hardware development, the other challenge that has been reported to affect agile implementation is team based communication where diversified team members with domain specific knowledge exists. Agile software development encourages team based communication through practices such as cross-functional teams, pair programming and daily stand-up meeting. The implementation of such team based practices in embedded safety critical domain has been reported to be difficult as a result of diversified team members with domain specific knowledge.

A third challenge that has been identified in our review is the availability of tool and automation support. This challenge can also be related to the previous challenge, hardware development. As hardware development affect the corresponding software development, the availability of tools that can automate the hardware has been reported to be beneficial for effective agile implementation.

This is the post-print version of the manuscript

Standards and regulation process have been reported to affect agile implementation in safety critical embedded domain. Some of the reports in our review states that the implementation of some agile practices such as continuous integration and continuous deployment needs to consider standards and regulation process.

Agile practices that have been implemented in embedded and safety critical domains have also been investigated in this review (RQ2). Most of the studies report the implementation of more than one practices. The combination of Scrum and XP practices have been reported in most of the studies. Additionally, practices such as test-driven development, acceptance testing have been used with the combination of Scrum and XP. Some studies have also investigated the challenges of Lean and DevOps for embedded domain and calls for tools support.

We have also investigated how agile practices have been extended (RQ2 -1). Our review has shown agile practices that have been extended, combined with other development technologies such as platform based design approach and model driven development. Scrum has been extended to address hardware-software designs (Scrum-CPS). A Lean approach to address hardware-software development through practices *cadence meeting*, *synchronization* and *two-level planning* is has also been investigated.

5 Conclusions

In this paper, we have discussed the result of the systematic review on agile usage in embedded safety critical domain. Our review has identified challenges and factors affecting agile implementation in such domains. Challenges related to hardware development, tools support, team based communication and regulation process have been reported to affect agile implementation.

The majority of previous studies on agile implementation in safety critical embedded domain are case studies and experience reports. Several combinations of practices from Scrum and XP have been used in different variations and in combination with other development technologies. Studies on the implementation of Lean and DevOps have also reported to be affected by the challenges.

The studies call for tools support to automate the hardware development and effectively use agile practices, a better way to manage communication between diversified members with specialized domain knowledge.

Acknowledgments. This work was supported with the financial support of the Science Foundation Ireland 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).

This is the post-print version of the manuscript

References

1. Perera, C., Liu, C.H., Jayawardena, S., Chen, M.: A Survey on Internet of Things from Industrial Market Perspective. *IEEE Access*. 2, 1660–1679 (2015).
2. Wolf, W., Wolf, W.: Chapter 7 – Hardware and Software Co-design. In: *High-Performance Embedded Computing*. pp. 383–432 (2007).
3. Berger, A.: *Embedded Systems Design—An Introduction to Processes, Tools, and Techniques* s. CMP Books, CMP Media LLC, Lawrence, Kansas 66046 (2002).
4. Teich, J.: Hardware/software codesign: The past, the present, and predicting the future. *Proc. IEEE*. 100, 1. Perera, C., Liu, C.H., Jayawardena, S., Chen, M.: A Survey on Internet of Things from Industrial Market Perspective. *IEEE Access*. 2, 1660–1679 (2015).
2. Wolf, W., Wolf, W.: Chapter 7 – Hardware and Software Co-design. In: *High-Performance Embedded Computing*. pp. 383–432 (2007).
3. Berger, A.: *Embedded Systems Design—An Introduction to Processes, Tools, and Techniques* s. CMP Books, CMP Media LLC, Lawrence, Kansas 66046 (2002).
4. Teich, J.: Hardware/software codesign: The past, the present, and predicting the future. *Proc. IEEE*. 100, 1411–1430 (2012).
5. Jiang, Z., Mangharam, R.: High-Confidence Medical Device Software Development. *Found. Trends® Electron. Des. Autom.* 9, 309–391 (2015).
6. Infographic: How Many Millions of Lines of Code Does It Take?, <http://www.visualcapitalist.com/millions-lines-of-code/>.
7. Fu, K.: Trustworthy medical device software. *Inst. Med. Work. Public Heal. Eff. FDA* 510. 510, 1–20 (2011).
8. Highsmith, J., Cockburn, A.: *Agile software development: the business of innovation*. Computer (Long. Beach. Calif). 34, 120–122 (2001).
9. Schwaber, K., Beedle, M.: *Agile Software Development with Scrum*. (2001).
10. Beck, K.: Embracing Change with Extreme Programming. *Arch. Dis. Child. Fetal Neonatal Ed.* 92, F83–F88 (2007).
11. Mary Poppendieck, T.P.: *Lean software development : an agile toolkit*. Addison-Wesley (2003).
12. Jabbari, R., bin Ali, N., Petersen, K., Tanveer, B.: What is DevOps? A Systematic Mapping Study on Definitions and Practices. *Proc. Sci. Work. Proc. XP2016 - XP '16 Work.* 1–11 (2016).

This is the post-print version of the manuscript

13. Petersen, K., Vakkalanka, S., Kuzniarz, L.: Guidelines for conducting systematic mapping studies in software engineering: An update. In: Information and Software Technology. pp. 1–18 (2015).
14. Kitchenham, B., Charters, S.: Guidelines for performing Systematic Literature Reviews in Software Engineering. *Engineering*, 2, 1051 (2007).
15. British Standards Online (America), <http://www.bsiamerica.com/en-us/Sectors and-Services/Industry-sectors/Healthcare-and-medical-devices/CE-marking-for-medicaldevices/>.
16. EN ISO 13485:2003 Medical Device: Quality Management Systems. Requirements for the Regulatory Process, 24-July-2003
17. EN ISO 14971:2009 Medical Devices. Application of Risk management to medical devices 31-July-2009
18. EN 60601-1 Medical Electrical Equipment. General requirements for basic safety and essential performance. Collateral standard. Usability 31-May2010.
19. C. Wohlin, “Guidelines for snowballing in systematic literature studies and a replication in software engineering,” in *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering - EASE '14*, 2014, pp. 1–10
20. J. Stapleton, *DSDM, Dynamic Systems Development Method: The Method in Practice*. 1997.
21. P. Manhart and K. Schneider, “Breaking the ice for agile development of embedded software: An industry experience report,” *Proc. - Int. Conf. Softw. Eng.*, vol. 26, pp. 378–386, 2004.
22. M. Kaisti et al., “Agile methods for embedded systems development - a literature review and a mapping study,” *EURASIP J. Embed. Syst.*, vol. 2013, no. 1, p. 15, 2013.
23. O. Salo and P. Abrahamsson, “Agile methods in European embedded software development organisations: a survey on the actual use and usefulness of Extreme Programming and Scrum,” *Software, IET*, vol. 2, no. 1, pp. 58–64, 2008.

Appendix: Selected Studies

- S1. Katumba, B., Knauss, E.: Agile Development in Automotive Software Development: Challenges and Opportunities. *Prod. Softw. Process Improv. Profes* 2014. 8892, 33–47 (2014).
- S2. Eklund, U., Holmström Olsson, H., Strøm, N.J.: Industrial Challenges of Scaling Agile in Mass-Produced Embedded Systems. In: *AGILE METHODS: LARGE-SCALE DEVELOPMENT, REFACTORING, TESTING, AND ESTIMATION*. pp. 30–42 (2014).
- S3. Manhart, P., Schneider, K.: Breaking the ice for agile development of embedded software: An industry experience report. *Proc. - Int. Conf. Softw. Eng.* 26, 378–386 (2004).
- S4. Cordeiro, L., Barreto, R., Barcelos, R., Oliveira, M., Lucena, V., Maciel, P.: TXM: An Agile HW/SW Development Methodology for Building Medical Devices. *ACM SIGSOFT Softw. Eng. Notes*. 32, 4 (2007).
- S5. Laukkarinen, T., Kuusinen, K., Mikkonen, T.: DevOps in Regulated Software Development: Case Medical Devices. *2017 IEEE/ACM 39th Int. Conf. Softw. Eng. New Ideas Emerg. Technol. Results Track*. 15–18 (2017).
- S6. Mirachi, S., da Costa Guerra, V., da Cunha, A.M., Dias, L.A.V., Villani, E.: Applying agile methods to aircraft embedded software: an experimental analysis. *Softw. Pract. Exp.* 47, 1465–1484 (2017).
- S7. Shatil, A., Hazzan, O., Dubinsky, Y.: Agility in a large-scale system engineering project: A case-study of an advanced communication system project. *SwSTE2010 IEEE Int. Conf. Softw. Sci. Technol. Eng.* 47–54 (2010).
- S8. Eklund, U., Berger, C.: Scaling Agile Development in Mechatronic Organizations - A Comparative Case Study. In: *2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP)*. pp. 173–182. IEEE (2017).
- S9. Duffau, C., Grabiec, B., Blay-Fornarino, M.: Towards Embedded System Agile Development Challenging Verification, Validation and Accreditation: Application in a Healthcare Company. *2017 IEEE Int. Symp. Softw. Reliab. Eng. Work.* 82–85 (2017).
- S10. Axelsson, J., Papatheocharous, E., Nyfjord, J., Törngren, M.: Notes On Agile and Safety-Critical Development. *ACM SIGSOFT Softw. Eng. Notes*. 41, 23–26 (2016).
- S11 Wagner, S.: Scrum for cyber-physical systems: a process proposal. *Proc. 1st Int. Work. Rapid*

This is the post-print version of the manuscript

Contin. Softw. Eng. - RCoSE 2014. 51–56 (2014).

S12. Laanti, M.: Piloting Lean-Agile Hardware Development. Proc. Sci. Work. Proc. XP2016 - XP '16 Work. 1–6 (2016).

S13. Salo, O., Abrahamsson, P.: Agile methods in European embedded software development organisations: a survey on the actual use and usefulness of Extreme Programming and Scrum. Software, IET. 2, 58–64 (2008).

S14. Jie, J.L.H.: Industrial Case Study of Transition from V-Model into Agile SCRUM in Embedded Software Testing Industries. SIGSOFT Softw. Eng. Notes. 41, 1–3 (2016).

S15. Könnölä, K., Suomi, S., Mäkilä, T., Jokela, T., Rantala, V., Lehtonen, T.: Agile methods in embedded system development: Multiple-case study of three industrial cases. J. Syst. Softw. 118, 134–150 (2016).

S16. Punkka, T.: Agile Hardware and Co-Design Agile development – a brief intro. 1–8 (2012).

S17. Van Schooenderwoert, N.: Embedded agile project by the numbers with newbies. Proc. - Agil. Conf. 2006. 2006, 351–363 (2006).

S18. Smith, M., Miller, J., Huang, L., Tran, A.: A More Agile Approach to Embedded System Development. IEEE Softw. 26, 50–57 (2009).

S19. Lwakatare, L.E., Karvonen, T., Sauvola, T., Kuvaja, P., Olsson, H.H., Bosch, J., Oivo, M.: Towards DevOps in the embedded systems domain: Why is it so hard? Proc. Annu. Hawaii Int. Conf. Syst. Sci. 2016–March, 5437–5446 (2016).

S20. Greene, B.: Agile methods applied to embedded firmware development. Proc. Agil. Dev. Conf. ADC 2004. 71–77 (2004).

S21. Cordeiro, L., Mar, C., Valentin, E., Cruz, F., Patrick, D., Barreto, R., Lucena, V.: A Platform-Based Software Design Methodology for Embedded Control Systems: An Agile Toolkit. In: 15th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ecbs 2008). pp. 408–417. IEEE (2008).

S22. Mingjuan Xie, M.S. and G.R.: Empirical Studies of Embedded Software Development Using Agile Methods : a Systematic Review. J. Inf. Syst. 2, 21–26 (2012).

S23. Chae, H., Lee, D., Park, J., In, H.: The Partitioning Methodology in Hardware/Software Co-Design Using Extreme Programming: Evaluation through the Lego Robot Project. In: The Sixth IEEE International Conference on Computer and Information Technology (CIT'06). pp. 187–187. IEEE (2006).

S24. Srinivasan, J., Dobrin, R., Lundqvist, K.: “State of the Art” in Using Agile Methods for Embedded Systems Development. 2009 33rd Annu. IEEE Int. Comput. Softw. Appl. Conf. 2, 522–527 (2009).

This is the post-print version of the manuscript

S25. Cordemans, P., Van Landschoot, S., Boydens, J., Steegmans, E.: Test-Driven Development as a Reliable Embedded Software Engineering Practice. In: *Embedded and Real Time System Development: A Software Engineering Perspective*. pp. 91–130 (2014).

S26. Kaisti, M., Rantala, V., Mujunen, T., Hyrynsalmi, S., Könnölä, K., Mäkilä, T., Lehtonen, T.: Agile methods for embedded systems development - a literature review and a mapping study. *EURASIP J. Embed. Syst.* 2013, 15 (2013).

S27. Bozheva, T., Hulkko, H., Ihme, T., Jartti, J., Salo, O., Van Baelen, S., Wils, A., consortium, I.-A.: Agile in embedded software development:

S28. Kaisti, M., Mujunen, T., Mäkilä, T., Rantala, V., Lehtonen, T.: Agile principles in the embedded system development. In: *Lecture Notes in Business Information Processing*. pp. 16–31 (2014).

S29. Ronkainen, J. and Abrahamsson, P. (2003). Software development under stringent hardware constraints: do agile methods have a chance? *Extreme Programming and Agile Processes in Software Engineering, 2003*, pp.1012–1012.

S30. Bosch, J. (2012). *Applying Agile Development in Mass-Produced Embedded Systems* Ulrik, , 111(January 2016). Available from: <http://link.springer.com/10.1007/978-3-642-30350-0>.