On the Continuous Delivery in IoT Systems

Igor Muzetti Pereira

[1]Federal University of Ouro Preto (UFOP)
Ouro Preto – MG – Brazil

igormuzetti@ufop.edu.br

Abstract. *The development of IoT systems brings the adequacy of software engineering to current paradigms, such as IoT, industry 4.0, smart cities and environments, and wearable devices. IoT systems require the integration of different technologies such as sensors, actuators, edge devices, cloud computing, big data, artificial intelligence, and IT operations. The successful delivery of products depends directly on the continuous cooperation of professionals with different skills. However, we noticed a specific resistance in carrying out continuous and automated software deliveries to hardware devices by professionals. This research investigates the practices and technologies for the continuous software delivery pipeline on hardware devices that make up an IoT system.*

1. Introduction

Companies developing Internet of Things (IoT) systems rely on software components as one of the main assets in their products. As systems become more complex, companies need to adapt to ensure continuous development processes [Luz et al. 2019]. In their development processes, these companies have teams that develop embedded systems that traditionally use plan-oriented methods [Lindgren and Münch 2016]. Moreover, they have software development and operations teams that are familiar with agile methodologies. However, an IoT environment requires ongoing cooperation between hardware and software development and operations teams to ensure quality attributes are met. Although there are studies on using agile methods in hardware teams, these teams do not accept these practices very easily [Mattos et al. 2018].

IoT leads to an era where, instead of just developing software, we need to design software systems that encompass multidisciplinarity and integrate different areas to realize successful products according to their purposes. This means that software is just one facet of IoT, which, along with others, is necessary for IoT solutions [Motta et al. 2018]. Therefore, software engineering is adapting to the construction and maintenance of these complex IoT systems, seeking to reduce development costs and increase the quality of the final product [Beyer et al. 2016].

IoT software systems involve recent technologies that question the traditional way of developing. Software engineering, as a discipline, has undergone constant changes since its conception. Several concepts, methods, tools, and patterns support software development [Bourque et al. 2014]. It is then possible to consider the need to evolve existing development practices to keep the construction of IoT software systems.

Continuous Delivery (CD) can handle this adaptation, such as automating the build, test, and delivery process achieving better time to market, better customer satisfaction, and good product quality [Chen 2015a]. Applying CD in the hardware and firmware development implies various challenges as a resistance of the professionals in using agile practices [Greene 2004, Srinivasan et al. 2009]. This research aims to propose and evaluate a conceptual model capable of assisting in planning continuous delivery pipelines for IoT systems focused on the hardware and firmware development.

The remainder of this document is organized as follows: Section 2 presents the related work necessary to understand this research. In turn, Section 3 presents the details of the different steps used to execute and collect the results obtained so far and how we plan the future steps. Section 4 describes the main and most relevant preliminary and expected results acquired by this research. Finally, Section 5 reports a final remarks and addresses the next steps of this research.

2. Related Work

Usually, the embedded software systems runs continually and does not stop. Its starts when the hardware is started and must run until the hardware has been turned off. Moreover, physical constraints, the need for direct interaction with the hardware, and security and reliability issues can affect the software project [Sommerville 2015].

There was difficulty in getting papers about continuous delivery in the IoT systems. The selected studies approach mainly the use of Extreme Programming (XP) and Scrum on embedded software projects. We consider that the concept of Internet of Things (IoT) systems evolved from the embedded systems and so, we see a correlation between them. Continuous delivery (CD) is an outcome of the agile methods used mainly after continuous integration.

CD is a discipline where teams build valuable software to be released into production at any time [Chen 2015b]. IoT is a computational paradigm that considers objects that communicate and cooperate with software systems to achieve common goals. These objects can be sensors, actuators, radio frequency identification (RFID) tags, gateways, smartphones, wearable devices, etc [Atzori et al. 2010].

Ronkainem and Abrahamsson [Ronkainen and Abrahamsson 2003] explore the possibilities of using agile development techniques in an embedded domain where the hardware sets tight requirements for the software. An empirical study was conducted in the development of low-level telecommunications software. The main challenge identified was a co-design, which means that the development of embedded systems is characterized by the need to develop software and hardware simultaneously. Therefore, this study approaches the software and hardware development of digital signal processors (DSPs) and application-specific integrated circuits (ASICs). The focus of the future of this research is the pipeline development of the hardware and firmware.

Manhart and Schneider [Manhart and Schneider 2004] report a case study of a software development department for buses. This organization is used to traditional software process improvement models like SPICE. However, research shows that the organization has combined the practices of this traditional model with agile elements. According to the authors, this combination fostered an acceptance of agile ideas and helped them break the ice to an extension of the adoption of agile processes. This application of agile practices in the embedded software demonstrated that it is possible to change the culture of the teams. This cultural change is a challenge to this research that focuses on incentive the application of the continuous delivery pipeline in hardware and firmware teams.

Greene [Greene 2004] cited several unique challenges in the embedded software development as many firmware developers are not classically trained software engineers but often have electrical engineering backgrounds and pick up programming as needed to solve reluctant to adopt the latest software tools and methodologies. They strive to develop code as expediently as possible. So, the paper describes the experience of implementing Scrum and XP to build firmware for the Intel Itanium processor family. We expected to find these challenges in IoT systems development teams.

Srinivasan et al. [Srinivasan et al. 2009] describe others challenges about the use of the agile practices in the development of embedded systems as meeting hard real-time requirements, supporting safe experimentation, generating sufficient documentation and, supporting test driven development.

Salo and Abrahamsson [Salo and Abrahamsson 2008] surveyed agile methods in European embedded software development organizations and showed the actual use and usefulness of XP and Scrum. Continuous integration was a fourth practice more used; 44% said that used systematically, mostly, or sometimes. But curiously, TDD was the practice that never applied with 41%. Moreover, all the Scrum practices received a high percentage of never and do not know responses. The paper shows that agile practices are not dominated by the 13 European industrial organizations studied.

The studies [Xie et al. 2012, Shen et al. 2012, Rong et al. 2014, Blašković et al. 2021] conducted a systematic review of the application of agile methods to embedded software development. They identified that XP and Scrum are the methods more used by these projects. But does not exist a depth discussion about the practices as CI are used. The papers too appoint that exists an immature development environment to apply TDD. As CI makes part of the continuous delivery, begin CI a phase important to this, our research could explore it better.

3. Experimental Procedures

This section describes the stages planned for this doctoral research. Figure 1 shows the overview of the four stages. Until this moment, two stages were concluded: the systematic mapping study (SMS) and the exploratory study involving interviews and a questionnaire. The third stage is building a conceptual model of the delivery continuous to hardware and firmware development in the IoT systems projects. Finally, the fourth stage aims to perform a experiment human-oriented.

IoT systems induce an articulation of studies in software engineering to prepare the development and operation of software systems materialized in physical objects and structures interconnected with embedded software and hosted in clouds. At the start of this research, to understand the boundaries between development and operation in the different architectural layers of IoT systems, we performed a systematic literature mapping (SLM) to search for evidence in the scientific literature about DevOps principles applied to IoT systems.

The SLM was divided into two parts and each one was discussed and published in two international conferences. After discussing with the two communities of these conferences, we decided to change DevOps to Continuous Delivery (CD). The main reason was that we realize this traditional practice involved inside the DevOps concept that can

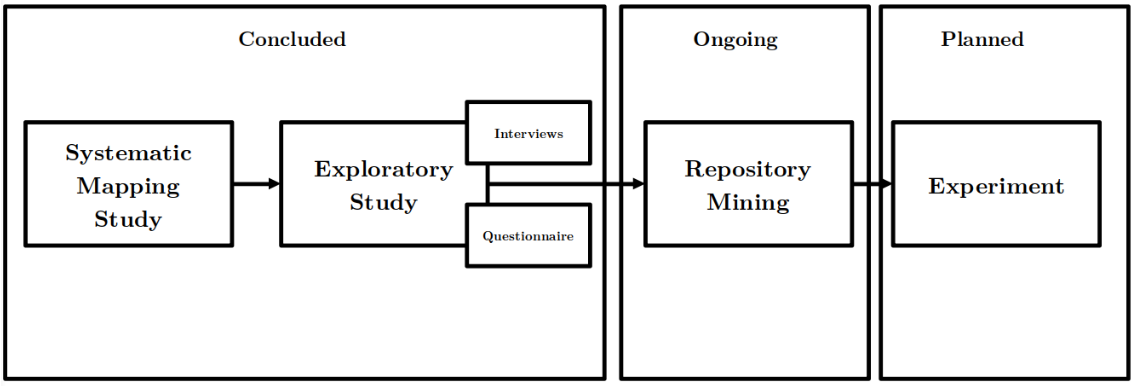

Figure 1. Steps of research experimental procedures

bring more focus on the future of this research. The conduction of this study followed guidelines [Kitchenham and Charters 2007].

We consider the second step as an exploratory study as described by Wohlin et al. [Wohlin et al. 2012]. We use a survey as an empirical strategy to collect information from professionals working with continuous delivery and IoT systems in the industry. In the first phase of this study, we used interviews to collect information and a questionnaire to verify this information in the second phase. To understand IoT system projects seeking continuous delivery, we first interviewed thirty-one experienced professionals. In the second phase, we applied a questionnaire to another group of thirty professionals from these areas to cross-validate our understanding of the collected information.

The third step of this study will focus on quantitatively analyzing a dataset extracted from a platform hosting structured or semi-structured text. We will explore these data in a way continuous and immediate. The data collection and analysis will be simultaneous, and we get a theoretical sampling. This mining is essential to identify evidence of using a continuous delivery pipeline. Most and less popular repositories will study the GitHub platform. We will not consider projects considered "toys" or documentation only. Only IoT software development projects will be valid. We will take into account ethical aspects for this type of study, as discussed in the work of Gold and Krinke [Gold and Krinke 2022]. Finally, we will inspire our process on the lessons learned, and the method proposed in the results by Barros et al. [Barros et al. 2021] and Vidoni [Vidoni 2022], respectively.

An experiment is a study in which an intervention is deliberately introduced to observe its effects on some aspects of reality under controlled conditions. We plan to do a human-oriented experiment to confirm conventional wisdom, i.e., to test people's conceptions and evaluate the conceptual model about continuous delivery pipelines on IoT systems. In the next section, we discuss the obtained and expected results of each experiment.

4. Results

The steps presented in the 1 shows two preliminary results of this research project and two other expected results. SLM enabled our understanding of the challenges and benefits of approaching DevOps in IoT systems and how an automated pipeline is necessary for continuous delivery. The exploratory study addressed the state of practice on this topic from perspectives of the professionals. The conceptual model from repository mining will be based on concepts found in the literature and evidence from open-source IoT system projects. Finally, the human-oriented experiment will allow us to evaluate the proposed conceptual model.

The SMS expanded our previous studies [Pereira et al. 2021a, Pereira et al. 2021d] by adding other recent studies, including the first semester of 2021. There are thirty-three studies now, and we analyzed them more horizontally as a systematic mapping study. A list of the practices and technologies DevOps covered in the studies was presented and discussed. Five categories of benefits and six of the challenges perceived in the studies were analyzed.

A preliminary result about the main differences of DevOps on IoT systems was published in SBES 2021 [Pereira et al. 2021c]. In favor the feedback received from the community, we expanded the research and added two other research questions. Our new focus started with the pursuit of continuous delivery in developing IoT systems. The additional research questions involved understanding what quality practices and the challenges perceived faced for continuous delivery in IoT systems projects by professionals. A follow up paper that describes this research was published in SBQS 2021 [Pereira et al. 2021b].

Participants presented the differences perceived when seeking continuous delivery as benefits for their processes. For instance, when teams use the automated pipelines, adhering to agility even across hardware teams, professionals gain greater possibilities to deliver more often and better meet customer expectations. The participants in this study made some use of quality practices. Of course, they can be applied to other domains. However, we see an effort by hardware and firmware professionals to implement them, either to keep up with what professionals in different areas are already doing or to meet customer requirements. The explored challenges mainly involve bringing professionals specialized in hardware closer to the practices used by development and operations professionals who are already more familiar with the agile way of working. The correct selection of components in the solution in the face of technological advances is a constant pressure that helps in the competitiveness of the market. The support of top management in the transformation is essential to keep business expectations aligned with the teams' technical capacity.

Our research will aim to generate a conceptual model for pipelines of continuous delivery in IoT systems projects. The model will be derived from publicly accessible records of processes and practices in popular open-source IoT software systems. We start our derivation process on continuous delivery practices in IoT systems by collecting and analyzing the publicly available documentation for each project, community mailing lists, websites documenting each project, open-source code repositories, blog posts, web articles, tutorials, and manuals.

With the conceptual model extracted from the repository mining study, a hypothesis is planned to be formalized, and the data collected during the experiment will be used to, if possible, reject the hypothesis. If the hypothesis can be ruled out, conclusions are drawn based on testing the hypothesis under certain risks. We will analyze the density of failures reported by participants when using pipelines for continuous delivery in an IoT system. A possible hypothesis that we are thinking about today would be that: "Contributors of software-to-hardware projects in IoT systems are not very comfortable building a continuous delivery pipeline."

5. Final Remarks

To summarize the doctoral project of the author completed, ongoing, and future activities, this section presents the schedule of activities involved.

Table 1. General Doctoral Schedule

#	Activities	2020/1	2020/2	2021/1	2021/2	2022/1	2022/2	2023/1	2023/2
0	Obtain PPGCC/DECOM/UFOP credits	X	X						
1	Participate in area communities and events	X	X	X	X	X	X	X	X
2	Conduct a literature review	X	X	X					
3	Conduct interviews		X	X					
4	Apply questionnaire			X					
5	Write and submit papers resulting from the studies carried out		X	X	X				
6	Present accepted studies			X	X				
7	Participate in the doctoral qualification exam.					X			
8	Development of a concept model about continuous delivery pipelines in IoT systems				X	X			
9	Evaluation of pipelines composed by the conceptual model.						X	X	
10	Write and submit papers for the second half of the doctorate					X	X	X	
11	Write the thesis							X	X
12	Defend the doctoral thesis								X

In this moment, the author is cursing the 2022/1 semester. This text is part of the Ph.D. project and explains the completed, ongoing and next steps to complete the doctoral research. This paper documents half of this doctorate, and the next steps will be added to these and will be part of a new final document that will become the final thesis.

References

Atzori, L., Iera, A., and Morabito, G. (2010). The internet of things: A survey. *Computer Networks*, 54(15):2787–2805.

Barros, D., Horita, F., Wiese, I., and Silva, K. (2021). *A Mining Software Repository Extended Cookbook: Lessons Learned from a Literature Review*, page 1–10. Association for Computing Machinery, New York, NY, USA.

Beyer, B., Jones, C., Petoff, J., and Murphy, N. R. (2016). *Site reliability engineering: How Google runs production systems*. O'Reilly Media, Inc.

Blaškovic, K., Candrlic, S., and Jakupovic, A. (2021). Systematic review of methodologies for the development of embedded systems. *International Journal of Advanced Computer Science and Applications*, 12(1).

Bourque, P., Fairley, R. E., and Society, I. C. (2014). *Guide to the Software Engineering Body of Knowledge (SWEBOK)*. IEEE Computer Society Press, Washington, DC, USA, 3rd edition.

Chen, L. (2015a). Continuous delivery: Huge benefits, but challenges too. *IEEE Software*, 32(2):50–54.

Chen, L. (2015b). Continuous delivery: Huge benefits, but challenges too. *IEEE Software*, 32(2):50–54.

Gold, N. E. and Krinke, J. (2022). Ethics in the mining of software repositories. *Empirical Software Engineering*, 27(1).

Greene, B. (2004). Agile methods applied to embedded firmware development. In *Agile Development Conference*, pages 71–77.

Kitchenham, B. and Charters, S. M. (2007). Guidelines for performing systematic literature reviews in software engineering.

Lindgren, E. and Münch, J. (2016). Raising the odds of success: the current state of experimentation in product development. *Information and Software Technology*, 77:80–91.

Luz, W. P., Pinto, G., and Bonifácio, R. (2019). Adopting devops in the real world: A theory, a model, and a case study. *Journal of Systems and Software*, 157:110384.

Manhart, P. and Schneider, K. (2004). Breaking the ice for agile development of embedded software: an industry experience report. In *Proceedings. 26th International Conference on Software Engineering*, pages 378–386.

Mattos, D. I., Bosch, J., and Olsson, H. H. (2018). Challenges and strategies for undertaking continuous experimentation to embedded systems: Industry and research perspectives. In *Agile Processes in Software Engineering and Extreme Programming*, pages 277–292, Cham. Springer International Publishing.

Motta, R. C., de Oliveira, K. M., and Travassos, G. H. (2018). On challenges in engineering iot software systems. In *Proceedings of the XXXII Brazilian Symposium on Software Engineering (SBES)*, page 42–51, New York, NY, USA. Association for Computing Machinery.

Pereira, I. M., Carneiro, T., and Figueiredo, E. (2021a). A systematic review on the use of devops in internet of things software systems. In *Proceedings of the 36th Annual ACM Symposium on Applied Computing (ACM/SAC)*, page 1569–1571. Association for Computing Machinery.

Pereira, I. M., de Senna Carneiro, T. G., and Figueiredo, E. (2021b). Investigating continuous delivery on iot systems. In *XX Brazilian Symposium on Software Quality (SBQS)*. Association for Computing Machinery.

Pereira, I. M., de Senna Carneiro, T. G., and Figueiredo, E. (2021c). Main differences of devops on iot systems. In *Brazilian Symposium on Software Engineering (SBES)*, page 315–319. Association for Computing Machinery.

Pereira, I. M., de Senna Carneiro, T. G., and Figueiredo, E. (2021d). Understanding the context of iot software systems in devops. In *3th International Workshop on Software Engineering Research Practices for the Internet of Things (SERP4IoT)*.

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. In *Proceedings of the 2014 International Conference on Software and System Process (ICSSP)*, page 94–98, New York, NY, USA. Association for Computing Machinery.

Ronkainen, J. and Abrahamsson, P. (2003). Software development under stringent hardware constraints: Do agile methods have a chance? In *Extreme Programming and Agile Processes in Software Engineering*, pages 73–79, Berlin, Heidelberg. Springer Berlin Heidelberg.

Salo, O. and Abrahamsson, P. (2008). 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.

Shen, M., Yang, W., Rong, G., and Shao, D. (2012). Applying agile methods to embedded software development: A systematic review. In *2012 Second International Workshop on Software Engineering for Embedded Systems (SEES)*, pages 30–36.

Sommerville, I. (2015). *Software Engineering*. Pearson, 10th edition.

Srinivasan, J., Dobrin, R., and Lundqvist, K. (2009). State of the art in using agile methods for embedded systems development. In *Industrial Experience in Embedded Systems Design (IEESD)*, pages 1195–1200.

Vidoni, M. (2022). A systematic process for mining software repositories: Results from a systematic literature review. *Information and Software Technology*, 144:106791.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., and Wesslén, A. (2012). *Experimentation in Software Engineering*, volume Ed. 1ª. Springer Berlin Heidelberg.

Xie, M., Shen, M., Rong, G., and Shao, D. (2012). Empirical studies of embedded software development using agile methods: A systematic review. In *Proceedings of the 2nd International Workshop on Evidential Assessment of Software Technologies (EAST)*, page 21–26, New York, NY, USA. Association for Computing Machinery.