

Article

A Case Study on Implementing Agile Techniques and Practices: Rationale, Benefits, Barriers and Business Implications for Hardware Development

Paweł Weichbroth

Department of Software Engineering, Faculty of Electronics, Telecommunications and Informatics, Gdansk University of Technology, 80-233 Gdańsk, Poland; pawel.weichbroth@pg.edu.pl; Tel.: +48-58-347-29-89

Abstract: Agile methodologies, along with the corresponding tools and practices, are claimed to facilitate teams in managing their work more effectively and conducting their work more efficiently while fostering the highest quality product within the constraints of the budget. Therefore, the rate of awareness and adoption of Agile frameworks both within and outside the software industry has increased significantly. Yet, the latest studies show that the adoption of Agile techniques and practices are not one-size-fits-all, and highlight the challenges, risks, and limitations regarding numerous domains. In this regard, the state-of-the-art literature provides comprehensive reading. However, in the case of hardware manufacturing, it seems to be sparse and fragmented. To fill this gap, the goal of this study is to analyze and present an in-depth account of the implementation of mix agile-oriented tools and practices. To tackle this goal, a single industry case study was undertaken, based on the primary data obtained through the interview protocol and the secondary data extracted from the project's documentation. The findings concern three areas. First, the rationale behind the implementation of agile for hardware development is explained. Second, the implemented agile techniques and practices are identified, as well as the supporting tools through which their adoption was successfully undertaken. Third, the areas positively impacted by their application are highlighted with the corresponding evaluation measures deployed; moreover, the barriers to adopting Agile practices encountered, and the benefits gained from particular techniques, are further discussed. The presented findings might be of great importance for both researchers and practitioners who are searching for empirical evidence regarding Agile-oriented implementations. Finally, in terms of both benefits and barriers, business implications for hardware development are formulated. Alongside this, numerous open issues and questions present interesting research avenues that concern, in particular, the effectiveness of collaboration and areas of communication through the lens of agile techniques and practices.

Citation: Weichbroth, P. A Case Study on Implementing Agile Techniques and Practices: Rationale, Benefits, Barriers and Business Implications for Hardware Development. *Appl. Sci.* **2022**, *12*, 8457. <https://doi.org/10.3390/app12178457>

Academic Editors: Przemysław Falkowski-Gilski, Tadeus Uhl, Zbigniew Lubniewski and Zhongjie Wang

Received: 13 June 2022

Accepted: 21 August 2022

Published: 24 August 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: agile; implementation; hardware development; case study; user experience

1. Introduction

The frustrations around failures of software development projects in the 1990s led to the search for more effective and cost-efficient ways of organizing and managing tasks and resources [1]. In early 2001, a group of software leaders met and discussed the possibilities of how to improve human productivity and performance, as well as how to bring value to the customers by delivering the expected software products [2]. At that meeting, the participants neither introduced nor used “agile” during discussions, instead the words “light” and “lightweight” were more common, reflecting their attitudes to the issue of bringing new software to market faster [3]. Eventually, two key opportunities were identified, namely shortening the time of delivery of the first version of the working soft-

ware, and getting feedback from the users to confirm their requirements [4]. This approach laid significant foundations for the Agile methodology as we know it today focused on speed to market, rapid feedback and continuous improvement [5].

On the other hand, the hardware businesses have also come across many obstacles and hardships in their way of updating existing and developing new products. The reasons for their failures can be divided into three categories: technical, financial, and sales/marketing [6]. It should be noted here that the second and third categories do not fall into the scope of our study, and will therefore not be further tackled. The first group of technical issues occurs during the development and early manufacturing stages often concern [6]:

- underestimating the product development;
- underestimating the complexity of scaling to mass manufacturing;
- poor quality testing;
- product overcomplexity;
- overpromising to customers.

To deal with these issues successfully, numerous methodologies, frameworks, and single methods have already been adopted by applying a different set of assumptions and settings, due to the unique needs of both the product and manufacturer. However, recent studies show that hardware development is enacted through agile methodologies [7], or just arbitrarily selected agile practices [8]. Similarly, our study falls into the scope of this modern but still insufficiently explored research domain. More specifically, the goal of this paper is to analyze and present an in-depth account of the implementation of agile techniques and practices, based on an industry case study of a project undertaken by the Tele and Radio Research Institute (TRRI) [9] and managed together with Meritus Software Systems [10].

The rest of the paper is structured as follows. In the Section 2, the research background and related work are outlined. In Section 3, the research design is specified. Next, in Section 4, the findings of the study are presented, followed by the conclusions given in Section 5.

2. Research Background and Related Work

2.1. Theoretical Background

In general, Agile is the umbrella term for a family of management practices [11]. Nowadays, many other organizations are actively adopting agile project management [12] as an iterative approach to delivering a project throughout its life cycle [13]. The reasons for implementing Agile practices concern the more flexible and adaptive working structures [14], active stakeholders and extensive user participation throughout the project [15], higher performance visibility [16], and transparency [17], to name just a few.

Considering the software industry, the Agile Alliance [18] is one of the most prominent organizations devoted to organizations and individuals that apply the values and principles of Agile. This non-profit, membership organization was founded on the Manifesto for Agile Software Development (ASD) [19]. By nature, ASD refers to a group of software development methods based on iterative and incremental development [20], where requirements and solutions evolve through collaboration between self-organizing cross-functional teams [19]. The Agile Manifesto expresses four key values, defined as follows [21]:

1. Individuals and interactions over processes and tools;
2. Working software over comprehensive documentation;
3. Customer collaboration over contract negotiation;
4. Responding to change over following a plan.

It should be noted and empathized here that, in the case of the fourth value, an agile view on the project means that responding to changes, instead of neglecting or even ignoring them, benefits the project by providing additional value.

Moreover, these four values are broken down and detailed through twelve supporting principles, formulated in the following way [22]:

1. Our highest priority is to satisfy the customer through the early and continuous delivery of valuable software;
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage;
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference for the shorter timescale;
4. Businesspeople and developers must work together daily throughout the project;
5. Build projects around motivated individuals. Provide them with the environment and support they need and trust them to get the job accomplished.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation;
7. Working software is the primary measure of progress;
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely;
9. Continuous attention to technical excellence and good design enhances agility;
10. Simplicity, the art of maximizing the amount of incomplete work, is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams;
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Based on the above principles, several agile methodologies have been built upon [23], including Scrum [24], Kanban [25], and Extreme Programming (XP) [26]. According to the 15th State of Agile Report [27], Agile adoption within software development teams increased from 37% in 2020 to 86% in 2021, where Scrum was the most popular (66%), with an additional 15% of the two following derivations, namely ScrumBan (9%) and Scrum/XP (6%); the minority concern Kanban (6%), Iterative (4%), XP (1%), Lean Startup (1%), and others (5%), whereas 2% of the respondents did not know which Agile methodology was adopted.

More specifically, over the last two decades, several agile practices have been developed that rely on the interaction of self-organizing teams with cross-functional skillsets. Globally, across a broad range of industries in the software development domain [28], the most commonly used tools by the individuals are [27]:

- Daily standups (87%), a team update meeting held each day [29], where everyone involved is asked to report on what has been completed, and commit the future work to do; moreover, the occurred impediments (blocking factors) are discussed; typically, in Scrum, the daily standup is a 15-min time-boxed event [30];
- Retrospectives (83%), a meeting that is held at the end of an iteration [31], with the aim of self-reflecting on what went well and what could be improved for the next iteration;
- Sprint/iteration planning (83%), where a sprint, only used in relation to Scrum [32], is a predefined, time-boxed period in which a team works to accomplish a set amount of work [33], and an iteration, in the context of an Agile project, is a time-box during which development takes place [34].

Considering agile planning, in particular involving workflows, tasks, and user stories, the three the most frequent practices are [27]:

- Kanban boards (77%), a visual-organization tool that tracks the workflow by depicting work items as cards at various stages of a development process, represented by labeled columns; in the simplest form, the board columns fall into three buckets: at the starts of the board is the "To Do" (or "Ready") column which contains all the cards that are next up, the "Doing" (or "In Progress") column include all the cards



that are currently being worked on, and “Done” contains all cards that have been finished [35];

- Task boards (67%), in Scrum, a task board is a visual display of the progress of the Scrum team during a sprint [36]; typically, a board is divided into four columns, namely: starting from the left, “Stories”, contains a list of all the user stories in the current sprint backlog, “To Do”, contains subtasks of the stories that work has not started on, “In progress” depicts all the tasks on that work has already begun, and “Done”, presents all the tasks that have been completed [37];
- Spreadsheets (66%), stands for a computer application used to create and manipulate spreadsheets, electronic documents in which data is arranged in tabular form and can be manipulated and used in calculations, along with formulas that relate the data. Currently, there is a plethora of spreadsheet applications on the software market, including both commercial and non-commercial, available for both desktop computers and mobile devices.

Having said that, one might ask: what are the benefits of being agile? There is no one simple answer to this question. However, the report from the 15th State of Agile survey shows that the implementation of agile has positively impacted areas within the organizations such as: managing changing priorities (70%), visibility (70%), business/IT alignment (66%), delivery speed/time to market (64%), team productivity (60%), and team morale (60%), just to quote a few. The impact was measured by various indicators, spanning from customer/user satisfaction (59%), business value (58%), business objectives achievements (50%), on-time delivery (48%), quality (48%), and productivity (41%), just to name a few.

2.2. Agile-Oriented Frameworks for Hardware Development

SAFe. The Scaled Agile Framework (SAFe), created by Dean Leffingwell, is a set of organizational and workflow patterns for implementing and scaling practices and techniques at the enterprise level in accordance with lean principles [38]. SAFe is a body of knowledge that incorporates frameworks such as Scrum, Extreme Programming, Kanban and Lean at the team level [39]. It advocates four levels: Value stream, Program, Portfolio and Team level [40]. The principal values of SAFe, which are key to SAFe’s effectiveness, concern: alignment, built-in quality, transparency, and program execution [41]. In this line of thinking, the Scaled Agile organization provides six universal principles, outlining how SAFe applies to hardware development [42]:

1. Organize Around Value;
2. Assume Variability and Preserve Options;
3. Build Incrementally, Integrate Frequently;
4. Design for Change;
5. Perform Work in Small Batches;
6. Build Continuous Integration for Hardware Development.

While SAFe is built on top of the agile methodology, harnessing its advantages, in particular, is based on ten immutable, underlying Lean-Agile principles; the above six principles have been arbitrarily selected and reformulated in the context of the hardware domain. Nevertheless, there is open space for creativity and innovation, since, as stated by the authors, “they can be used to guide hardware engineers to create and adopt their own best practices” [42].

Scrum for Hardware. Scrum for Hardware (or Scrum for HW) is a framework that combines Scrum with modular architecture and Lean/XP practices [43], customizing the agile practices and techniques for hardware and system design. The Scrum in Hardware Guide defines 8 areas to consider before, during and after implementation, namely:

1. Uncertainty in Hardware Projects;
2. Stubs and Mock-Ups;
3. Modularize;



4. Continuously Integrate;
5. Interface Design;
6. Test and Data-Driven Development;
7. Team;
8. Working Product at the End of Each Sprint.

In general, the guide can be used by any organization, small or large, regardless of its field of activity, structure, or organization.

Hybrid Stage-Gate. The Stage-Gate (SG) process is a conceptual and operational road map for moving a new-product project from idea to launch and beyond [44]. In the simplest format, SG “consists of two major components:

- (1) a series of stages, where the project team undertakes the work, obtains the needed information, and does the subsequent data integration and analysis, followed by;
- (2) gates, where go/kill decisions are made to continue to invest in the project” [44].

An overview of the general Stage-Gate system is depicted in Figure 1.

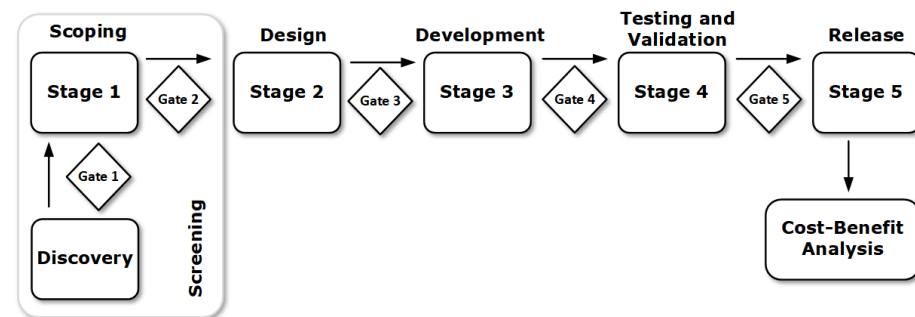


Figure 1. Overview of the generic Stage-Gate system. Source: [44].

Stage-Gate is a macro model, designed to support selecting the projects to be followed by the model, and once selected, to map out the key stages, best practices, management techniques, roles, and responsibilities. On the other hand, there is an agile approach that brings micromanagement practices and techniques to the table, which are proven to foster adaptability, encourage creativity, and improve productivity. Cooper and Sommer introduced a hybrid approach, which incorporates Agile methods for physical product development [45], by integrating them with the SG system, and then utilized them for manufactured or physical product development (hardware). In other words, they argue that “Agile is normally applied as the project management method within the stages in Stage-Gate”, while “the stages remain, and Agile is applied within some stages” [45]. Moreover, Scrum is a favored methodology since it has been indicated as “the most appropriate for hardware development” for all case studies uncovered so far in industry.

Modified Agile for Hardware Development (MAHD). Based on the Agile principles, Simpson and Hinkle [46] developed the Modified Agile for Hardware Development (MAHD) Framework, devoted to the manufacturing industry, in particular for products and portfolios that include a combination of electronics, mechanical components, as well as software elements [47]. By design, MAHD takes into account the following physical products requirements [48]:

- allow for flexibility to change, but also freeze designs;
- build quality plans and manage supply chains;
- define electrical and mechanical product attributes;
- develop documentation;
- develop flexible prototyping and validation product strategies;
- share resources and external partners.

MAHD promotes four foundation principles to follow, namely:



1. Short development cycles to drive learning and adapting to change;
2. Accountable, autonomous, and focused teams;
3. Validating work at the end of each development cycle;
4. Applying intelligent, rapid prototyping strategies.

Interestingly, while the first and the second principles are focused on the human factor, the remaining two are targeted at the product. Such a dichotomy seems to be reasonable and justified, taking into consideration the work breakdown structure in general.

2.3. Related Work

The rationale behind adopting agile (Scrum) practices to integrate the development of both hardware and software is provided by Lima et al. who claim that “the development of devices that combine hardware and software has created new challenges”, since “the new built devices have a short life cycle and frequently require upgrading” [49]. In this study, the authors introduced a development process so-called “Agile and Co-Design”, which consists of seven phases and expected outcomes: Conception (product vision), Setting-up (specifications), Design (deliverables), Definition Of Done (checklist), Testability (integration tests), Implementation (hardware and software), Verification (evaluation). The findings show the application of their approach enabled deep interaction between teams representing different areas of knowledge, and fostered knowledge exchange and innovation creation.

In a similar vein of cautionary optimism, Laanti stated that “the benefits of incremental and iterative development of hardware may be as high as with software development” [50]. However, afterward, the obstacles are articulated: “the idea of incremental design was causing problems”, since “people felt the hardware should have been created and tested as one entity”, and “it is not flexible the same way software is”. On the other hand, the application of iterations “can be used to mature the design”.

The broad spectrum of issues and interests regarding the potential, transition and applicability of agile development in the hardware sector is given by Atzberger and Kristin [51], and later by Atzberger et al. [52]. The four categories of the findings concern:

- constraints of physicality, the authors claim that “the greatest hindrance is to realize potentially shippable increments in one iteration”, and as a consequence “the technical feasibility to produce prototypes” is hardly achievable in certain industrial branches, due to “the inability to break down the product into modules” [51];
- agile mindset, an individual attitude toward understanding, collaborating, learning, adapting and growing in the spirit of trust, responsibility and ownership, is pointed out as a major challenge for not only individuals, but also for the company;
- team distribution, the inability to work co-located resulted in a higher degree of communication needs between physically dispersed project participants (teams);
- scaling, the ability to apply the principles, practices and outcomes in other projects (horizontal scaling), or at other levels of the organization (vertical scaling), in order to change the way people approach their own work and working with others.

Naturally, there are numerous other studies that have reported on the implementation of the agile approach, and its consequences on the project effectiveness and deliverables. Moreover, there are several well-known and admired industry corporations that are claimed to cultivate agile principles and practices, including SpaceX [53], Tesla [54], General Motors [55], and Toyota [56], just to name a few. Nevertheless, agile is not a silver bullet that enables everyone to become adaptive, innovative, and resilient; neither is it an excuse to stop thinking, nor a panacea for solving all issues.



3. Research Design

3.1. Case Setting

Tele and Radio Research Institute (TRRI) is located in Poland (Warsaw) and operates in the business industry. TRRI's mission is to conduct "comprehensive and interdisciplinary scientific examinations and development activities on highly advanced technologies and innovations, in particular with regard to ICT systems, electronics, electronic installation and Industry 4.0 solutions" [57]. In 2019, the Institute became a part of The Łukasiewicz Research Network Institute, which, associating 26 research institutes with 8000 employees, located in 12 cities across Poland, is the third largest research network in Europe [58]. For the sake of clarity and simplicity, henceforth, TRRI will be called the "project partner" (PP), or just a "partner".

Meritus Software Systems (Meritus) is a software producer with its headquarters in Poland (Warsaw). Established in 1990, for over 30 years the company has earned the reputation of being a trustworthy and loyal business partner among cooperating companies. Meritus is a team of enthusiastic computer specialists devoted to developing highly efficient and usable software by following the modern patterns and trends in the development of IT systems. The landmark software products are: Pinguark Warehouse Management System, Geo Train AI, merSoft ERP, and skanujfaktura.pl [10]. Henceforward, Meritus will be called the "project leader" (PL), or simply the "leader".

3.2. Project Settings

The current study is entirely based on a project, performed in cooperation between the partner and the leader, entitled "An Industry 4.0 Mobile Process Management System Supported by Artificial Intelligence" (hereafter called the "project"). The project's budget was 1.2 million euro and was conducted between January 2019 and December 2021. The aim of the project was to develop and implement a cloud-based process management system, facilitated by six AI modules, available for both web-based and mobile users. The scope of the project falls into the smart factory domain, thus following the modern trend of automation and data exchange in manufacturing technologies, including cyber-physical systems, cloud and cognitive computing, and the Internet of Things.

It should be noted here that by definition, a "smart factory" refers to the workers and machinery related to the execution of their tasks; by design, a smart factory is flexible and re-configurable, low-cost, changeable, and thereby agile [59].

3.3. Project Context and Organization

To set up the context of the study, one of the most relevant solutions is to provide the scope of the project, which serves as a measurable and exogenous variable [60]. In this line of thinking, first the requirements are briefly provided, regarding the hardware deliverables. More specifically, their features and properties, with the embedded software developed specifically for their needs, are discussed with the project schedule illustrating a summary of the physical device development process.

Remote monitoring and control of processes in technological areas, and thus the need to transmit large amounts of data while maintaining confidentiality, requires the use of efficient communication solutions that are characterized by high-performance embedded microcontrollers, while also maintaining a low price and reliable long-term operational time in specific environmental conditions that are often extreme and unfavorable for the operation of electronic systems of large-scale integration. To this end, the goal of the project was the development of a universal, cost-effective communication modem, which should be flexible enough to adapt to the changeable requirements imposed by data acquisition from different technological areas.

The guidelines for the hardware and software platforms were based on maximizing the functionality of the module solution for use in various technological areas with a



strong emphasis on aspects of Machine to Machine (M2M) communication [61] and Internet of Things (IoT) devices, especially devoted to industrial applications.

The hardware platform of the communication modem model was made in two iterations. In the first iteration, a model of the modem was made to test the functionality of the selected OSD335x solution and to check the validity of the solution in terms of the connection of the printed circuit board elements. The second, taking into account functional corrections and comments after EMC/LVD EMC tests.

First production stage [main unit]. The communication module model design is divided into two blocks. The main unit block, integrating the SOM and Ethernet communication circuits, and the second on a separate PCB with communication circuits and connectors. This division was due to the effective distribution into functional blocks and the technological requirements for the PCB for the individual functional blocks. With the proposed division, a reduction in PCB manufacturing costs was achieved, as well as the fact that in case of an error resulting from incorrect functional assumptions, it can be corrected manually or the printed circuit can be improved with only one module, which reduces the manufacturing costs.

First production stage [base unit]. The base unit includes connectors and add-on circuits required to communicate with sensors and measurement sensors. In the design of the communication modem model, it was assumed that the basic communication interface to the sensors and measurement sensors will be USB 2.0, with a high-speed, 480 MBit/s communication interface. USB 2.0 is a universal and standardized serial communication interface. One differential port and a possible 5 V power supply are required for data exchange. The USB 2.0 protocol is flexible and allows for the use of standardized data exchange protocols such as classes e.g., ACM, HID, mass storage as well as the definition of a custom class. It allows for a maximum of 256 devices on a single bus via HUB circuits.

Second production stage [main unit]. The second iteration required amendments to components of the modem, including the microprocessor and Ethernet physical layer hardware unit layout, to optimize the size of the device to be more compact. The microprocessor was changed to a version with built-in e-MMC memory. The change allowed for the design to be optimized for component availability. e-MMC memories are among the difficult to access electronic components. The change in the Ethernet physical layer layout was dictated by replacing the chip with a more modern one and adding Media-Independent Interface (MII) support, which will make it possible to use Programmable Real-Time Units (PRUs) to implement the EtherCAT (Ethernet Control Automation Technology) and Profibus (PROcess FIEld BUS) protocols. While the former protocol was chosen as it fulfills both hard and soft real-time computing requirements, since it combines the efficient and relatively high-speed message transmission of Ethernet networks [62], the latter was used because of its high speed and architecture, designed and optimized especially for communication between automation systems and decentralized devices [63].

Second production stage [base unit]. The changes that were made in the second iteration were related to the optimization of the PCB dimensions. Comments after EMC testing were also introduced. The power supply for the module requires that a noise filter be added to the input and common mode ferrites be added to the Ethernet and USB interface lines.

3.4. Research Methods

First and foremost, content analysis was used as the primary research method. By definition, according to Mackey, content analysis is “any technique for making inferences by systematically and objectively identifying special characteristics of messages” [64]. In this case, we followed the guidelines defined by Krippendorff [65], along with their practical assessment elaborated by Stemler [66]. Therefore, the following six questions have been addressed and answered.

1. **Which data are analyzed?** Due to the project complexity, considering the number of planned tasks, time frames and the total number of hardware and software deliverables, the further investigation was strictly oriented toward issues and concerns related to the project management. More specifically, the focus was on the identification, analysis and evaluation of only those processes that regarded planning, organizing, leading and controlling the efforts of the project members, engaged on the side of both the leader and the partner.
2. **How are they defined?** The data throughout the project was collected in a few digital repositories, including the data gathered by the online tools throughout the project duration, as well as the documents library available in the cloud storage.
3. **What is the population from which they are drawn?** The population is limited to one project, in particular two data repositories owned by the leader and the partner.
4. **What is the context relative to which the data are analyzed?** The context is specified by the goal of the study, which precisely indicates the scope and objectives for the researcher to follow. In particular, the data (information) must concern the implementation of agile practices throughout the project. In other words, a precedent or unrepeatable application of any agile practice was not considered for further investigation due to the limited analysis and evaluation.
5. **What are the boundaries of the analysis?** Obviously, when the study concerns a single case, the results are not meant to be generalized. Therefore, it is correct to name generalizability as a limitation of the current, and any other, qualitative research.

To sum up, the content analysis was aimed at identifying, analyzing, and evaluating the project's documentation, including documents, reports, spreadsheets, information from work communication and collaboration tools, to extract the agile practices implemented throughout the project.

The second research method, applied to validate the obtained results from the documentation study, was an in-depth interview. According to Boyce and Neale, the in-depth interviewing technique involves "conducting intensive individual interviews with a small number of respondents to explore their perspectives on a particular idea, program, or situation" [67]. Moreover, regarding the interview format, the standardized open-ended interview was adopted, since such a design allows the interviewees to "contribute as much detailed information as they desire" [68]. Moreover, the interview was supported by a questionnaire (see Appendix A). For the needs of this study, the process for conducting in-depth interviews consisted of five steps:

1. **Plan.** The identified stakeholders involved two representatives, one from the leader, and one from the partner. To reduce the bias resulting from insufficient experience and knowledge, the inclusion criteria were as follows:
 - a. at least five years of professional experience;
 - b. involvement in the project throughout its duration;
 - c. managerial role in the project.

The two respondents were both males with over 20 years of professional experience in the IT sector. In the project, the first respondent (R1) worked in the position of project manager, while the second respondent (R2) held the role of research and development manager. Therefore, they were both positively classified;

2. **Develop instruments.** The interview protocol and the questionnaire were designed around the three constructs of interest for this research, namely: (1) the rationale behind adopting the agile approach for the project, (2) the practices and tools applied, and finally (3) the performance evaluation. Appendix A shows the details regarding both data collection instruments;
3. **Collect data.** The phone interviews took place in the period from April to May 2022. Alongside safety and health issues due to the outbreak of COVID-19, this approach is now becoming widely used [69]. The data collection follow-up concerned an email request for each of the respondents to independently fill in the questionnaire;



4. **Analyze data.** To “make sense” of the collected data, a thematic analysis was conducted on the information gathered via the interview, while the survey data were compared and summarized by using the Google Sheets application;
5. **Disseminate findings.** The target audience does not only involve the project stakeholders, but also both computer science researchers and industry practitioners. To reach such a broad audience, our intention is to publish our work in open access mode, which means that our paper will be free of charge, digital and online, and free of most copyright and licensing restrictions. In addition, the paper preprint will also be posted in an electronic format and made available to the public on at least one preprint service.

4. Findings

The findings are scrutinized with regards to each question, where a strict partitioning is given to distinguish the three different areas of interest.

Part I. Agile Adoption Rationale.

Q1. What were the most important reasons for adopting Agile in the consortium?

Having in mind the experience and knowledge with respect to the reasons for the adoption of Agile in software development, there is strong agreement between the respondents regarding three of its outcomes:

- accelerate software delivery;
- improve business and IT alignment;
- reduce project risk.

Moreover, they also agree that Agile adoption enhances:

- the ability to manage changing priorities;
- software quality;
- improve engineering discipline.

However, there is no consensus on whether the agile-oriented organization (team) is able to better respond to volatile market conditions or enhance delivery predictability. Moreover, it seems that the respondents also disagree on the positive impact of agility on the team morale.

Part II. Agile Methodology, Techniques and Practices Adoption.

Q2. Which agile approach was adopted? Both respondents described the applied methodology as “iterative,” emphasizing the split of the development process into multi-phase explicit interactions, each assumed to deliver required features.

Q3: Which of the following Agile techniques and practices were used? Both respondents declared that three agile techniques were used during the project, namely:

- release planning;
- short iterations;
- sprint/iteration planning.

Moreover, they both indicated the lack of using activities for determining which products or projects to work on, in which order, and for how long, termed agile portfolio planning. Second, in the case of estimation techniques, the usage of planning poker was not confirmed either for estimating user stories or tasks. Third, daily standups were not the practice adapted to discuss a project’s progress at a high level. Fourthly, the respondents declared that it was found unnecessary to release increments frequently. Last but not least, Kanban, being a transparent and visual system for real-time communication of the workload capacity, was not implemented throughout the project.

Q4. Which Agile planning and delivery tools did you use? From the set of 22 tools possible to select, the respondents agreed on three tools, including:

- task board;
- automated build tool;
- unit test tool.

On the other hand, consensus was also reached on neglecting the majority of the planning and delivering tools, including the product road-mapping, along with the family of management tools related to requirements, product, and portfolio, as well as the customer idea.

Moreover, neither index cards nor timecards were used in the project. While the former are simply the tokens representing user stories, task breakdowns, backlog items or even reminders, and the latter are the method of recording and tracking the amount of an employee's time that is spent on each job. Second, no static analysis was performed along with story mapping to formulate new hardware features. In the case of work management, the Kanban board was not found useful.

Considering the system evaluation from the perspective of the external user, no automation test tools were deployed in the project. Interestingly, other tools that are highly appreciated from the software vendor's perspective were not found suitable in the case of a hardware manufacturer. These are:

- bug tracking;
- continuous integration tool;
- release/ deployment automation tool;
- wireframes.

Ultimately, in the case of project and knowledge management, the agile project management and Wiki tools were only declared to be used by the project leader, whereas the refactoring tool was strictly used by the project partner.

Q5. Which agile planning tools did you use? While there were 21 tools to choose from, unexpectedly, each of the respondents indicated only one different tool: Atlassian Jira, in the case of the project leader, and Microsoft Excel, in the case of the project partner. The list of the other not implemented tools can be found in Appendix A.

Part III. Agile Adoption Evaluation.

Q6. Has the implementation of agile positively impacted each of the following areas within your company? Considering the 13 areas that were positively impacted by the implementation of agile practices and supported by the corresponding tools, there was no agreement between the respondents in any of these. Keeping in mind the question stated above, the first respondent identified three areas:

- managing changing priorities;
- predictability;
- risk reduction;

which were assessed as neutral (intact) by the second respondent.

On the other hand, the second respondent declared five other areas, positively impacted by the agile implementation, namely:

- delivery speed/time to market;
- engineering discipline;
- software quality;
- team morale;
- team productivity.

However, both respondents had a neutral view on three areas, including: business/IT alignment, software maintainability, and visibility.

Q7. How has your organization measured the success of Agile delivery? Both respondents agreed on one thing only, which is that by managing the scope and schedules with an Agile-centric approach, their teams were able to deliver business value, measured by tangible and working product features, added in a fixed period of time.

Moreover, the second respondent indicated that within his organization, another ten measures were successfully implemented: budget vs. actual cost, customer retention customer and user satisfaction, defects into production, defects over time, iteration burn-down, planned vs. actual release dates, team morale, test pass/fail over time, and velocity.

Q8. What are the most significant barriers to adopting Agile practices in the consortium? There was one common barrier indicated when adopting agile practices which concerned the lack of business (customer, or product) understanding.

Nevertheless, considering the other twelve barriers, in the case of the following four, namely:

- fragmented tooling and project-related data/measurements;
- general organization resistance to change;
- pervasiveness of traditional development methods;
- unwilling to admit mistakes and learn from delivery failure;

the answers were the opposite, while their total absence concerned the project leader, their presence affected the project partner.

Moreover, the latter also reported another two barriers encountered during the adoption of Agile practices regarding inconsistent processes and practices across teams, and organizational culture at odds with agile values.

Q9. What are the benefits of adopting the release planning (TP10) technique? In total, eleven benefits were submitted under each respondent's independent evaluation, of which only the following three were found to be significant to a similar extent:

- increased efficiency by highlighting the goals or deadline for release;
- making high-quality plans;
- increased visualization potential of release planning problems.

With some doubt from the first respondent, and strong or moderate belief from the second, the other five benefits of using the release planning technique concerned:

- eliminating waste in the Planning Process;
- setting clear expectations about the objectives and product features;
- helping team members stay on track and complete their tasks on time;
- increased flexibility and decreased development lead time;
- supporting the identification and mitigation of potential issues and risks.

Ultimately, both respondents were not fully convinced of whether the release planning enabled the early identification of feature dependencies, and disagreed on the increased motivation of the developers, as well as on the mediation of multiple opinions.

Q10. What are the benefits of adopting the short iterations (TP12) technique? There was mutual agreement that the shorter the iteration, the less process overhead is needed to keep things on course. Partial agreement of using short iterations was reached for the following seven benefits:

- provide more-frequent feedback from customers than long iterations;
- afford the team more opportunities to reflect on and improve their work practices;
- deliver a rapid response to changes in priority without disrupting work in process;
- foster scope management since it is easier to move the small backlog items around than the large ones;
- eliminate internal process overheads by running short iterations since user stories are small units of work and the team can follow the iteration status in a simpler way;
- support product adaptability since it is easy to introduce new features, change existing features, and make any other functional change;
- support the team's ability to work on several features in parallel.

Finally, a different view was presented on the claim that short iterations enabled early problem detection since agile teams recognized process smells and acted on them immediately.

Q11. What are the benefits of adopting sprint/iteration planning (TP14)?

Unquestionably, sprint (iteration) planning was declared the enabler that positively influenced the product quality. Both respondents, representing the project leader and the project partner, partially agreed that using the sprint (iteration) planning technique brought the following benefits:



- helps teams to control projects;
- effective management of the product backlog;
- provides an opportunity for team members to participate in planning and hence better understand the work assignments;
- provides opportunities to build team cohesion and identification;
- helps in understanding the user stories;
- serves to validate something we do not understand, or something misinterpreted,
- enables setting precise estimates for the task;
- fosters knowledge-sharing and skill-improvement.

However, the respondents doubted whether the planning meetings helped to define the goals for each team member.

Q12. What are the benefits of using a task board (PD17)? Full agreement was reached with regards to all three benefits submitted to the evaluation in relation to this question. Thus, it was declared that using a task board definitely:

- ensures efficient diffusion of information relevant to the whole team;
- enables the team to stay focused on the work progress;
- allows the team to represent any relevant information.

On the other hand, it is a pity that the respondents did not provide any benefits other than the predefined ones.

Q13. What are the benefits of using an automated build tool (PD3)? Without any doubt, both respondents pointed out that using an automated build tool improved product quality by supporting bug and error detection. Moreover, they also agreed that using such a tool contributed to:

- saving time and money;
- by keeping a history of builds and releases, it helps in investigating bugs and errors;
- an increase in developers' productivity since build automation ensures fast feedback;
- acceleration of the product delivery by eliminating redundant tasks.

Q14. What are the benefits of using a unit test tool (PD20)? The two highest appraised benefits of using a unit test tool are concerned with fostering a higher quality of individual components, that in consequence, lead to increasing the overall system resiliency; moreover, it is also claimed to increase the testers' productivity. Additionally to these, the remaining advantages are related to:

- early detection of specification deficiencies;
- cost reduction;
- easier source code refactoring.

Both respondents agreed that the sum of the benefits achieved was greater than the sum of the costs encountered. Among the main challenges of running an Agile development project with remote teams, there was one in common: keeping the final goal in mind. In other words, team members sometimes forgot that "progress takes place when they work by this rule". Moreover, a few other issues were individually raised that concerned the differences between team members, regarding their knowledge, skills, and agile mindset, especially in the case of teams from organizations from different business sectors.

Last but not least, in the current case study, the organization and governance of the undertaken project (see Figure 1) were very similar to the Hybrid Stage-Gate approach. However, none of the respondents were familiar with its theoretical foundations, in contrast to the agile methodologies and frameworks. In general, considering the latter, the rationale behind its adoption was concerned with the aim of preventing and minimizing the risk of project failure. More specifically, the adoption of agile practices and techniques provided rewarding opportunities due to their emphasis on instant and close collaboration and communication, resulting in reducing, or even eliminating the information gap.



5. Discussion

This study contributes to the existing literature by reporting and adding an important new investigation regarding agile hardware development. The growing interest among researchers in revisiting existing agile methodologies in the new domains has opened up new frontiers. In light of the evidence derived from our case study, a better understanding of implementing agility in hardware manufacturing is provided, drawing on the reasons for and against its adoption, and indicating the practices, techniques and tools applied. Moreover, the areas positively impacted by the agile implementation are highlighted with the corresponding evaluation measures deployed, along with the barriers and obstacles encountered. These findings may be of great importance for planning and specifying possible future agile implementations.

Regarding the other related studies, Atzberger and Paetzold argue that “adaptations to the field of hardware development are inevitable, therefore, principles and practices are necessary for the companies to circumvent the hardware-related issues (...)” [51]. In this vein of argument, Sharifi and Zhang claim that “agility in manufacturing may be achieved through the implementation and integration of appropriate practices which provide the required abilities for a company to respond properly to changes” [70]. This claim has been confirmed by Montero et al., since “it is possible to use principles and practices of agile hardware development to clarify and promote the robustness” of manufacturing processes [8]. Our findings are consistent with those similar reported observations. By documenting the case of agile implementation within the hardware domain, we deliver one more piece of evidence for supporting the agile-oriented mindset.

Nevertheless, the current study suffers from two main limitations associated with the research methods applied. First, since we are aware that there is no way to generalize the results obtained from this study, there is a need to design and perform more research in similar settings. Second, the data analysis was reliant on the project documentation (secondary data) and interviews (primary data) with the selected project stakeholders, which followed the project closure. Therefore, both data sources were qualitative in nature. In the case of the former, relevant reports were carefully selected and evaluated, whereas the latter was deliberately scrutinized to validate the study’s results. Nevertheless, there is a risk of bias related to the human factor which could occur during the process of information analysis and synthesis. To sum up, more research is needed to confirm (or disprove) our findings.

6. Conclusions

In this paper, we report on selected outcomes from an agile implementation in a hardware manufacturing project. More specifically, this research is an attempt to answer the questions regarding the feasibility and validity of the implementation of agile-oriented techniques and practices, supported by the deployment of relevant tools.

The business implications, in terms of benefits resulting from the implementation of the agile techniques and practices to hardware development, concern three main areas:

- achieving an effective risk mitigation strategy;
- greater adaptability and innovation capacity;
- effective communication and collaboration.

On the other hand, the business implications in terms of challenges are related to human nature, which is manifested by:

- resistance to change;
- a lack of or inadequate management support;
- poor communication and collaboration.

Moreover, with the agile approach and the change in the way of thinking and working, an organization is able to create an environment that is open to new opportunities arising for its employees, including training, promotion, flexible working, job rotation, and job enrichment. Nevertheless, any manager should ask herself/himself: is my team



ready to become agile? Yet, the mindset shift is the hardest to make for leaders. During the transition, the key element is to change their leadership style from telling the team what to do to mentoring the team.

Several open issues and questions present interesting research avenues. In particular, there are still very few studies that have investigated domains not related to the IT industry. These poorly explored landscapes, which provide valuable opportunities for future research, can provide more lessons regarding the effectiveness of collaboration and communication through the lens of agile techniques and practices.

Funding: This research is the result of the project entitled “An Industry 4.0 Mobile Process Management System Supported by Artificial Intelligence”, co-financed by the European Union from the European Regional Development Fund under the Regional Operational Program of the Mazovia Voivode ship for 2014–2020, under grant agreement no. RPMA.01.02.00-14-b528/18.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: Special thanks are due to the two respondents for providing comprehensive and valuable answers to the questionnaire questions.

Conflicts of Interest: The author declares no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of the data; in the writing of the manuscript, or in the decision to publish the results.

Appendix A. The Questionnaire

The questionnaire began with a short introduction that briefly describes the project. The first three parts are concerned with the following areas of the project undertaken: planning, execution, and evaluation. The last part aimed to collect the respondents’ demographic data.

Part I. Agile Adoption Rationale and Selection.

Q1. What were the most important reasons for adopting Agile?

Code	Item	Answer ¹
O1	Accelerate software delivery	
O2	Better manage distributed teams	
O3	Better respond to volatile market conditions	
O4	Enhance ability to manage changing priorities	
O5	Enhance delivery predictability	
O6	Enhance software quality	
O7	Improve business and IT alignment	
O8	Improve engineering discipline	
O9	Improve project visibility	
O10	Improve team morale	
O11	Increase software maintainability	
O12	Increase team productivity	
O13	Reduce project cost	
O14	Reduce project risk	
O15	Other? Please specify:	

¹ The answers for the items (excluding O15) were predefined with the following list of five items: (1) Definitely Not, (2) Probably Not, (3) Possibly, (4) Probably Yes, (5) Definitely Yes.

Part II. Agile Methodology, Techniques and Practices Adoption.**Q2.** Which agile approach was adopted?

- (a) Scrum.
- (b) ScrumBan.
- (c) Kanban.
- (d) Iterative.
- (e) I don't know.
- (f) None.

Q3. Which of the following Agile techniques and practices did you use?

Code	Item	Answer ²
TP1	Agile Portfolio planning	
TP2	Agile/Lean UX	
TP3	Common work area	
TP4	Daily standup	
TP5	Dedicated customer/product owner	
TP6	Frequent releases	
TP7	Kanban	
TP8	Planning Poker/Team Estimation	
TP9	Product roadmapping	
TP10	Release planning	
TP11	Retrospectives	
TP12	Short iterations	
TP13	Single team	
TP14	Sprint/Iteration planning	
TP15	Sprint/iteration reviews	
TP16	Story mapping	
TP17	Other? Please specify:	

² The answers for the items (excluding TP17) were predefined with the following list of five items: (1) Definitely Not, (2) Probably Not, (3) Possibly, (4) Probably Yes, (5) Definitely Yes.

Q4. Which Agile planning and delivery tools did you use?

Code	Item	Answer ³
PD1	Agile Project Management Tool	
PD2	Automated Acceptance Test Tool	
PD3	Automated Build Tool	
PD4	Bug Tracker	
PD5	Continuous Integration Tool	
PD6	Customer Idea Management Tool	
PD7	Index Cards	
PD8	Kanban Board	
PD9	Product and Portfolio Management (PPM) Tool	
PD10	Product Roadmapping	
PD11	Refactoring Tool	
PD12	Release/Deployment Automation Tool	
PD13	Requirements Management Tool	
PD14	Spreadsheet	
PD15	Static Analysis	
PD16	Story Mapping Tool	
PD17	Task board	
PD18	Timecards	



PD19	Traditional Product Management Tool
PD20	Unit Test Tool
PD21	Wiki tool
PD22	Wireframes
PD23	Other? Please specify:

³ The answers for the items (excluding PD23) were predefined with a list of two items: (1) No, and (2) Yes.

Q5. Which agile planning tools did you use?

Code	Item	Answer ⁴
PT1	Atlassian Jira	
PT2	Atlassian Jira Align	
PT3	Axosoft	
PT4	Azure DevOps	
PT5	Azure DevOps Server (Microsoft TFS)	
PT6	Broadcom Rally (CA Agile Central)	
PT7	Bugzilla	
PT8	Digital.ai Agility (formerly VersionOne)	
PT9	Digital.ai TeamForge (CollabNet TeamForge)	
PT10	Google Docs	
PT11	Hansoft	
PT12	HP Agile Manager	
PT13	HP Quality Center/ALM	
PT14	IBM Rational Team Concert	
PT15	In-house / self-made solution	
PT16	LeanKit	
PT17	Microsoft Excel	
PT18	Microsoft Project	
PT19	Pivotal Tracker	
PT20	Targetprocess	
PT21	Trello	
PT22	Other? Please specify:	

⁴ The answers for the items (excluding PT22) were predefined with a list of two items:(1) No, and (2) Yes.

Part III. Agile Adoption Evaluation.

Q6. Has the implementation of agile positively impacted each of the following areas within your company?

Code	Item	Answer ⁵
IP1	Business/IT alignment	
IP2	Cost reduction	
IP3	Delivery speed/time to market	
IP4	Engineering discipline	
IP5	Managing changing priorities	
IP6	Managing distributed teams	
IP7	Predictability	
IP8	Risk reduction	
IP9	Software maintainability	
IP10	Software quality	
IP11	Team morale	
IP12	Team productivity	



IP13 Visibility

IP14 Other? Please specify:

⁵ The answers for the items (excluding IP14) were predefined with the following list of five items: (1) Strongly disagree, (2) Disagree, (3) Neither agree nor disagree, (4) Agree, and (5) Strongly agree.

Q7. How has your organization measured the success of Agile delivery?

Code	Item	Answer ⁶
ME1	Budget vs. actual cost	
ME2	Burn-up chart	
ME3	Business value delivered	
ME4	Cumulative flow chart	
ME5	Customer retention	
ME6	Customer/user satisfaction	
ME7	Cycle time	
ME8	Defect resolution	
ME9	Defects into production	
ME10	Defects over time	
ME11	Earned value	
ME12	Estimation accuracy	
ME13	Individual hours per iteration/week	
ME14	Iteration burndown	
ME15	Planned vs. actual release dates	
ME16	Planned vs. actual stories per iteration	
ME17	Product utilization	
ME18	Release burndown	
ME19	Revenue sales impact	
ME20	Scope change in a release	
ME21	Team morale	
ME22	Test pass/fail over time	
ME23	Velocity	
ME24	WIP, Work in Progress	
ME25	Other? Please specify:	

⁶ The answers for the items (excluding ME25) were predefined with a list of two items:(1) No, and (2) Yes.

Q8. What are the most significant barriers to adopting Agile practices?

Code	Item	Answer ⁷
BA1	Fragmented tooling and project-related data/measurements	
BA2	General organization resistance to change	
BA3	Inadequate management support and sponsorship	
BA4	Inconsistent processes and practices across teams	
BA5	Insufficient training and education	
BA6	Lack of business/customer/product understanding	
BA7	Lack of skills/experience with agile methods	
BA8	Minimal collaboration and knowledge pairing	
BA9	Not enough leadership participation	
BA10	Organizational culture at odds with agile values	
BA11	Pervasiveness of traditional development methods	
BA12	Regulatory compliance or government issue	
BA13	Unwilling to admit mistakes and learn from delivery failure	
BA14	I don't know	

BA15 Other? Please specify:

⁷ The answers for the items (excluding BA15) were predefined with the following list of five items: (1) Strongly disagree, (2) Disagree, (3) Neither agree nor disagree, (4) Agree, and (5) Strongly agree.

Q9. What are the benefits of adopting the release planning (TP10) technique?

Code	Item	Answer ⁸
B1-TP10	Eliminating Waste in the Planning Process	
B2-TP10	Increased Flexibility and Decreased Development Lead Time	
B3-TP10	Increased Motivation of the Developers	
B4-TP10	Mediation of multiple opinions	
B5-TP10	Increased efficiency by highlighting the goals or deadline for release	
B6-TP10	Making high-quality plans	
B7-TP10	Increased visualization potential of release planning problems	
B8-TP10	Setting clear expectations about the objectives and product features	
B9-TP10	Early identification of feature dependencies	
B10-TP10	Helping team members stay on track and complete their tasks on time	
B11-TP10	Supporting the identification and mitigation of potential issues and risks	
B12-TP10	Other? Please specify: ...	

⁸ The answers for the items (excluding B12) were predefined with the following list of five items: (1) Definitely Not, (2) Probably Not, (3) Possibly, (4) Probably Yes, (5) Definitely Yes.

Q10. What are the benefits of adopting the short iterations (TP12) technique?

Code	Item	Answer ⁹
B1-TP12	Provide more-frequent feedback from customers than long iterations	
B2-TP12	Afford the team more opportunities to reflect on and improve their work practices	
B3-TP12	Deliver a rapid response to changes in priority without disrupting work in process	
B4-TP12	Enable early problem detection since agile teams recognize process smells and act on them immediately	
B5-TP12	Foster scope management since it is easier to move the small backlog items around than the large ones	
B6-TP12	Eliminate internal process overheads by running short iterations since user stories are small units of work and the team can follow the iteration status in a simpler way.	
B7-TP12	The shorter the iteration, the less process overhead is needed to keep things on course.	
B8-TP12	Support product adaptability since it is easy to introduce new features, change existing features, and make any other functional change	
B9-TP12	Support the team's ability to work on several features in parallel	
B10-TP12	Other? Please specify: ...	

⁹ The answers for the items (excluding B10) were predefined with the following list of five items: (1) Definitely Not, (2) Probably Not, (3) Possibly, (4) Probably Yes, (5) Definitely Yes.

Q11. What are the benefits of adopting sprint/iteration planning (TP14)?

Code	Item	Answer ¹⁰
B1-TP14	Helps teams control projects	
B2-TP14	Effective management of the product backlog	
B3-TP14	Positively influences product quality	
B4-TP14	Provides an opportunity for team members to participate in planning and hence better understand the work assignments	
B5-TP14	Provides opportunities to build team cohesion and identification	
B6-TP14	Helps in understanding the user stories	



- B7-TP14 Serves to validate something we do not understand, or something mis-interpreted
 B8-TP14 Planning meetings help to define the goals for each team member
 B9-TP14 Enables setting precise estimates for the task
 B10-TP14 Fosters knowledge-sharing and skill-improvement
 B11-TP14 Other? Please specify: ...

¹⁰ The answers for the items (excluding B11) were predefined with the following list of five items: (1) Definitely Not, (2) Probably Not, (3) Possibly, (4) Probably Yes, (5) Definitely Yes.

Q12. What are the benefits of using a task board (PD17)?

Code	Item	Answer ¹¹
B1-PD17	Ensures efficient diffusion of information relevant to the whole team	
B2-PD17	Enables the team to stay focused on the work progress	
B3-PD17	Allows the team to represent any relevant information	
B4-PD17	Other? Please specify: ...	

¹¹ The answers for the items (excluding B4) were predefined with the following list of five items: (1) Definitely Not, (2) Probably Not, (3) Possibly, (4) Probably Yes, (5) Definitely Yes.

Q13. What are the benefits of using an automated build tool (PD3)?

Code	Item	Answer ¹²
B1-PD3	Saving time and money	
B2-PD3	By keeping a history of builds and releases, it helps in investigating bugs and errors	
B3-PD3	Increases developers' productivity since build automation ensures fast feedback	
B4-PD3	Accelerates product delivery by eliminating redundant tasks	
B5-PD3	Improves product quality by supporting bug and error detection	
B6-PD3	Other? Please specify: ...	

¹² The answers for the items (excluding B6) were predefined with the following list of five items: (1) Definitely Not, (2) Probably Not, (3) Possibly, (4) Probably Yes, (5) Definitely Yes.

Q14. What are the benefits of using a unit test tool (PD20)?

Code	Item	Answer ¹³
B1-PD20	Early detection of the specification deficiencies	
B2-PD20	Fostering higher quality of individual components, and thus increase overall system resiliency	
B3-PD20	Increases testers' productivity	
B4-PD20	Cost reduction	
B5-PD20	Easier refactoring the source code	
B6-PD20	Other? Please specify: ...	

¹³ The answers for the items (excluding B6) were predefined with the following list of five items: (1) Definitely Not, (2) Probably Not, (3) Possibly, (4) Probably Yes, (5) Definitely Yes.

Q15. In your opinion, considering the implementation of agile techniques and practices, is the sum of the benefits achieved greater than the sum of the costs encountered?

(1) Definitely Not, (2) Probably Not, (3) Possibly, (4) Probably Yes, (5) Definitely Yes.

Q16. In your opinion, what are the main challenges of running an Agile development project with remote teams?

Q17. What have been the most valuable lesson(s) learned in easing the adoption of agile practices and techniques?

Part IV. The Respondent's Data.

Q18. What is your age?

Q19. What is your sex?

Q20. How many years of experience in the IT sector do you have?



- Q21.** What was your role in the project?
Q22. Were you involved in the project throughout its duration?

References

- Denning, S. How to Make the Whole Organization “Agile.” *Strategy Leadersh.* **2016**, *44*, 10–17. <https://doi.org/10.1108/SL-06-2016-0043>.
- Hohl, P.; Klünder, J.; van Bennekum, A.; Lockard, R.; Gifford, J.; Münch, J.; Stupperich, M.; Schneider, K. Back to the Future: Origins and Directions of the “Agile Manifesto”—Views of the Originators. *J. Softw. Eng. Res. Dev.* **2018**, *6*, 15. <https://doi.org/10.1186/s40411-018-0059-z>.
- Kannan, V.; Basit, M.A.; Bajaj, P.; Carrington, A.R.; Donahue, I.B.; Flahaven, E.L.; Medford, R.; Melaku, T.; Moran, B.A.; Saldana, L.E.; et al. User Stories as Lightweight Requirements for Agile Clinical Decision Support Development. *J. Am. Med. Inform. Assoc.* **2019**, *26*, 1344–1354. <https://doi.org/10.1093/jamia/ocz123>.
- Misra, S.; Kumar, V.; Kumar, U.; Fantazy, K.; Akhter, M. Agile Software Development Practices: Evolution, Principles, and Criticisms. *Int. J. Qual. Reliab. Manag.* **2012**, *29*, 972–980. <https://doi.org/10.1108/02656711211272863>.
- Pikkarainen, M.; Salo, O.; Still, J. Deploying Agile Practices in Organizations: A Case Study. In *Software Process Improvement, Proceedings of the 12th European Conference, EuroSPI 2005, Budapest, Hungary, 9–11 November 2005*; Richardson, I., Abrahamsson, P., Messnarz, R., Eds.; Springer: Berlin/Heidelberg, Germany, 2005; pp. 16–27.
- Why New Hardware Businesses Fail and How to Make Sure Yours Doesn’t. Available online: <https://predictabledesigns.com/13-reasons-why-hardware-startups-fail-and-how-to-make-sure-yours-doesnt/> (accessed on 12 April 2022).
- Huang, P.M.; Darrin, A.G.; Knuth, A.A. Agile Hardware and Software System Engineering for Innovation. In Proceedings of the 2012 IEEE Aerospace Conference, Big Sky, MT, USA, 3–10 March 2012; pp. 1–10.
- Joaquin, M.; Alexander, A.; Matthias, B.; Jens, H.; Kristin, P. Enhancing the Additive Manufacturing Process for Spare Parts by Applying Agile Hardware Development Principles. In Proceedings of the 2019 IEEE 10th International Conference on Mechanical and Intelligent Manufacturing Technologies (ICMIMT), Cape Town, South Africa, 15–17 February 2019; pp. 109–116.
- The Tele and Radio Research Institute—PPTF. Available online: <https://pptf.pl/en/pptf-members/the-tele-and-radio-research-institute/> (accessed on 14 May 2022).
- We Are a Software Producer Using Artificial Intelligence—Meritus Sp. z o.o. Available online: <https://meritus.pl/en/> (accessed on 16 May 2022).
- Serova, E.; Kalmykov, O. On the Issue of Implementation of Agile and Strategy as a Practice Mixed-Method in Strategic Planning. In *Eurasian Business Perspectives, Proceedings of the 35nd Eurasia Business and Economics Society Conference, Rome, Italy, 7–9 April 2021*; Bilgin, M.H., Danis, H., Demir, E., Vale, S., Eds.; Springer: Cham, Switzerland, 2021; pp. 127–139.
- Mkoba, E.; Marnewick, C. Conceptual Framework for Auditing Agile Projects. *IEEE Access* **2020**, *8*, 126460–126476. <https://doi.org/10.1109/ACCESS.2020.3007874>.
- Jiménez, V.; Afonso, P.; Fernandes, G. Using Agile Project Management in the Design and Implementation of Activity-Based Costing Systems. *Sustainability* **2020**, *12*, 10352. <https://doi.org/10.3390/su122410352>.
- Masood, Z.; Farooq, S. The Benefits and Key Challenges of Agile Project Management under Recent Research Opportunities. *Int. Res. J. Manag. Sci.* **2017**, *5*, 20–28.
- Cram, W.A.; Marabelli, M. Have Your Cake and Eat It Too? Simultaneously Pursuing the Knowledge-Sharing Benefits of Agile and Traditional Development Approaches. *Inf. Manag.* **2018**, *55*, 322–339.
- Shameem, M.; Kumar, C.; Chandra, B.; Khan, A.A. Systematic Review of Success Factors for Scaling Agile Methods in Global Software Development Environment: A Client-Vendor Perspective. In Proceedings of the 2017 24th Asia-Pacific Software Engineering Conference Workshops (APSECW), Nanjing, China, 4–8 December 2017; pp. 17–24.
- Benefits of Agile Project Management | Kanbanize. Available online: <https://kanbanize.com/agile/project-management/benefits-of-agile> (accessed on 23 April 2022).
- Agile Alliance. Agile Alliance | 2015. Available online: <https://www.agilealliance.org/> (accessed on 02 May 2022).
- Paulk, M.C. Agile Methodologies and Process Discipline. *J. Contrib.* **2002**. <https://doi.org/10.1184/R1/6620972.v1>.
- de la Barra, C.L.; Crawford, B.; Soto, R.; Misra, S.; Monfroy, E. Agile Software Development: It Is about Knowledge Management and Creativity. In *Computational Science and Its Applications—ICCSA 2013, Proceedings of the 13th International Conference, ICCSA 2013, Ho Chi Minh City, Vietnam, 24–27 June 2013*; Murgante, B., Misra, S., Carlini, M., Torre, C.M., Nguyen, H.-Q., Taniar, D., Apduhan, B.O., Gervasi, O., Eds.; Springer: Berlin/Heidelberg, Germany, 2013; pp. 98–113.
- Manifesto for Agile Software Development. Available online: <https://agilemanifesto.org/> (accessed on 23 April 2022).
- 12 Principles behind the Agile Manifesto | Agile Alliance. Agile Alliance | 2015. Available online: <https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/> (accessed on 1 May 2022).
- Flewelling, P. *The Agile Developer’s Handbook: Get More Value from Your Software Development: Get the Best out of the Agile Methodology*; Packt Publishing Ltd.: Birmingham, UK, 2018; ISBN 978-1-78728-073-1.
- Schwaber, K.; Sutherland, J. The Scrum Guide. In *The Definitive Guide to Scrum: The Rules of the Game*; Creative Commons: Mountain View, CA, USA, 2020.

25. Ahmad, M.O.; Markkula, J.; Oivo, M. Kanban in Software Development: A Systematic Literature Review. In Proceedings of the 2013 39th Euromicro Conference on Software Engineering and Advanced Applications, Santander, Spain, 4–6 September 2013; pp. 9–16.
26. Beck, K.; Hendrickson, M.; Fowler, M. *Planning Extreme Programming*; Addison-Wesley Professional: Boston, MA, USA, 2001; ISBN 978-0-201-71091-5.
27. 15th Annual State of Agile Report | Digital.Ai. Available online: <https://digital.ai/resource-center/analyst-reports/state-of-agile-report> (accessed on 23 April 2022).
28. Khan, A.A.; Shameem, M.; Kumar, R.R.; Hussain, S.; Yan, X. Fuzzy AHP Based Prioritization and Taxonomy of Software Process Improvement Success Factors in Global Software Development. *Appl. Soft Comput.* **2019**, *83*, 105648.
29. Hallett, C. Best Practices in Industry and CSE Senior Design. Diploma Thesis, University of Nebraska-Lincoln, Lincoln, NE, USA, 19 April 2021.
30. What Is a Daily Scrum? Available online: <https://www.scrum.org/resources/what-is-a-daily-scrum> (accessed on 16 May 2022).
31. Lagerberg, L.; Skude, T. The Impact of Agile Principles and Practices on Large-Scale Software Development Projects: A Multiple-Case Study of Two Software Development Projects at Ericsson. In Proceedings of the 2013 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, Baltimore, MD, USA, 10–11 October 2013.
32. What Is an Agile Iteration? | Wrike Agile Guide. Available online: <https://www.wrike.com/agile-guide/faq/what-is-agile-iteration/> (accessed on 16 May 2022).
33. What Is a Sprint? Available online: <https://airfocus.com/glossary/what-is-a-sprint/> (accessed on 16 May 2022).
34. What Is an Iteration? Agile Alliance | 2015. Available online: [https://www.agilealliance.org/glossary/iteration/#q=~\(infinite-false~filters-\(postType-\(~page~post~aa_book~aa_event_session~aa_experience_report~aa_glossary~aa_research_paper~aa_video\)-tags-\(~iteration\)\)~searchTerm~sort-false~sortDirection~asc~page-1](https://www.agilealliance.org/glossary/iteration/#q=~(infinite-false~filters-(postType-(~page~post~aa_book~aa_event_session~aa_experience_report~aa_glossary~aa_research_paper~aa_video)-tags-(~iteration))~searchTerm~sort-false~sortDirection~asc~page-1) (accessed on 3 May 2022).
35. Dalton, J. Kanban Board. In *Great Big Agile: An OS for Agile Leaders*; Dalton, J., Ed.; Apress: Berkeley, CA, USA, 2019; pp. 187–188, ISBN 978-1-4842-4206-3.
36. Digital, M. Agile Concepts: The Scrum Task Board. Available online: <https://manifesto.co.uk/agile-concepts-scrum-task-board/> (accessed on 16 May 2022).
37. Wirdemann, R. *Scrum Mit User Stories*; Carl Hanser Verlag: Munich, Germany, 2009; ISBN 978-3-446-41656-7.
38. Sreenivasan, S.; Kothandaraman, K. Improving Processes by Aligning Capability Maturity Model Integration and the Scaled Agile Framework®. *Glob. Bus. Organ. Excell.* **2019**, *38*, 42–51.
39. Putta, A.; Paasivaara, M.; Lassenius, C. Adopting Scaled Agile Framework (SAFe) a Multivocal Literature Review. In Proceedings of the 19th International Conference on Agile Software Development: Companion, New York, NY, USA, 21–25 May 2018; pp. 1–4.
40. Alqudah, M.; Razali, R. A Review of Scaling Agile Methods in Large Software Development. *Int. J. Adv. Sci. Eng. Inf. Technol.* **2016**, *6*, 828–837.
41. Core Values. Scaled Agile Framework. Available online: <https://www.scaledagileframework.com/safe-core-values/#:~:text=The%20four%20Core%20Values%20of,participates%20in%20a%20SAFe%20portfolio> (accessed on 4 May 2022).
42. Applying SAFe to Hardware Development. “Scaled Agile Framework”. Available online: <https://www.scaledagileframework.com/applying-safe-to-hardware-development/> (accessed on 4 May 2022).
43. Justice, J.; Sammiceli, P. *Scrum for Hardware*; Independently Published: Chicago, IL, USA, 2018; ISBN 978-1-983373-31-2.
44. Cooper, R.G. Perspective: The Stage-gate® Idea-to-launch Process—Update, What’s New, and Nexgen Systems. *J. Prod. Innov. Manag.* **2008**, *25*, 213–232.
45. Cooper, R.G.; Sommer, A.F. The Agile–Stage-gate Hybrid Model: A Promising New Approach and a New Research Opportunity. *J. Prod. Innov. Manag.* **2016**, *33*, 513–526.
46. MAHD About Us: MAHD: The Definitive Agile for Hardware Development Resource. Available online: <https://agileforhardware.org/introduction-to-the-mahd-framework/> (accessed on 14 April 2022).
47. Agile for Hardware Home > MAHD: The Definitive Agile for Hardware Development Resource. Available online: <https://agileforhardware.org/> (accessed on 20 July 2022).
48. An Introduction to the MAHD Framework > MAHD: The Definitive Agile for Hardware Development Resource. Available online: <https://agileforhardware.org/complete-agile-for-hardware-framework/> (accessed on 14 April 2022).
49. Lima, G.L.B.; Ferreira, G.A.L.; Saotome, O.; Da Cunha, A.M.; Dias, L.A.V. Hardware Development: Agile and Co-Design. In Proceedings of the 2015 12th International Conference on Information Technology—New Generations, Las Vegas, NV, USA, 13–15 April 2015; pp. 784–787.
50. Laanti, M. Piloting Lean-Agile Hardware Development. In *Proceedings of the Scientific Workshop Proceedings of XP2016*; Association for Computing Machinery: New York, NY, USA, 2016; pp. 1–6.
51. Atzberger, A.; Paetzold, K. Current Challenges of Agile Hardware Development: What Are Still the Pain Points Nowadays? *Proc. Des. Soc. Int. Conf. Eng. Des.* **2019**, *1*, 2209–2218. <https://doi.org/10.1017/dsi.2019.227>.
52. Atzberger, A.; Gerling, C.; Schrof, J.; Schmidt, T.S.; Weiss, S.; Paetzold, K. Evolution of the Hype around Agile Hardware Development. In Proceedings of the 2019 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC), Valbonne Sophia-Antipolis, France, 17–19 June 2019; pp. 1–8.
53. Does SpaceX Use Agile? Available online: <https://www.EclipseAviation.com> (accessed on 4 April 2022).

54. Tesla Agile Development: Product Management at Its Best | 280 Group. Available online: <https://280group.com/product-management-blog/tesla-agile-development/> (accessed on 21 May 2022).
55. General Motors Accelerates Transformation. Available online: <https://media.gm.com/media/us/en/gm/home.detail.html/content/Pages/news/us/en/2018/nov/1126-gm.html> (accessed on 21 May 2022).
56. Agile Transformation Spotlight: The Success Story of Scrum & Toyota. Available online: <https://www.scrum.org/resources/agile-transformation-spotlight-success-story-scrum-toyota> (accessed on 21 May 2022).
57. The Tele and Radio Research Institute. Available online: <https://itr.lukasiewicz.gov.pl/en/> (accessed on 13 April 2022).
58. About Us. Available online: <https://lukasiewicz.gov.pl/en/about-us/> (accessed on 21 May 2022).
59. Hozdic, E. Smart Factory for Industry 4.0: A Review. *IJMMT* **2015**, *7*, 28–35.
60. Poulis, K.; Poulis, E.; Plakoyiannaki, E. The Role of Context in Case Study Selection: An International Business Perspective. *Int. Bus. Rev.* **2013**, *22*, 304–314.
61. Yang, S.; MR, A.R.; Kaminski, J.; Pepin, H. Opportunities for Industry 4.0 to Support Remanufacturing. *Appl. Sci.* **2018**, *8*, 1177.
62. Seno, L.; Zunino, C. A Simulation Approach to a Real-Time Ethernet Protocol: EtherCAT. In Proceedings of the 2008 IEEE International Conference on Emerging Technologies and Factory Automation, IEEE, Hamburg, Germany, 15–18 September 2008; pp. 440–443.
63. Abhish, K.; Rakesh, V. Real-Time Analysis of a Multi-Client Multi-Server Architecture for Networked Control Systems. *J. Eng. Appl. Sci.* **2015**, *10*, 7522–7526.
64. Mackey, D.A. Content Analysis. In *The Encyclopedia of Criminology and Criminal Justice*; John Wiley & Sons, Ltd.: Hoboken, NJ, USA, 2014; pp. 1–5, ISBN 978-1-118-51738-3.
65. Krippendorff, K. *Content Analysis: An Introduction to Its Methodology*; SAGE Publications: Thousand Oaks, CA, USA, 2018; ISBN 978-1-5063-9567-8.
66. Stemler, S. An Overview of Content Analysis. *Pract. Assess. Res. Eval.* **2019**, *7*, 17. <https://doi.org/10.7275/z6fm-2e34>.
67. Boyce, C.; Neale, P. *Conducting in-Depth Interviews: A Guide for Designing and Conducting in-Depth Interviews for Evaluation Input*; Pathfinder International: Watertown, MA, USA, 2006.
68. Turner, D.W., III. Qualitative Interview Design: A Practical Guide for Novice Investigators. *Qual. Rep.* **2010**, *15*, 754–760.
69. Mousavi, M.S.; Fararouei, M.; Afsar-Kazerooni, P.; Nasirian, M.; Ghaem, H. Evaluation of Conducting Phone Interviews on Sexual Behavior: An Iranian Population-Based Study. *Shiraz E-Med. J.* **2022**, *23*. <https://doi.org/10.5812/semj-118708>.
70. Sharifi, H.; Zhang, Z. Agile Manufacturing in Practice—Application of a Methodology. *Int. J. Oper. Prod. Manag.* **2001**, *21*, 772–794. <https://doi.org/10.1108/01443570110390462>.

