Adoptability Challenges in Work Environment of Organizations using Agile Software Development Methods

M Subhan Dar[1], Shahra Asif[2], Saleem Zubair[3]
[1]Department of Software Engineering, Superior University Lahore, Pakistan, msse-f20-005@superior.edu.pk
[2] Department of Software Engineering, Superior University Lahore, Pakistan, msse-f20-008@superior.edu.pk
[3]Department of Software Engineering, Superior University Lahore, Pakistan, saleem.zubair@superior.edu.pk

Abstract
Agile project management is an ongoing method to project completion that actually took place all across length of the project's life cycle. Because loop allows you to adapt as you go rather than maintaining a linear route, iterative methods are mainly applied in software development to ensure velocity and agility. Many pitfalls abound in agile software development adaptations that organizations fail to sidestep. New competitive challenges, fluctuating market dynamics, technological disruption, and the ever-changing demands of empowered customers confront organizations around the world. Organizations are all under tremendous pressure to adapt to change and deliver good products and services to customers more swiftly. Research measured at the challenges that could be encountered and offered advice for how agile development might flourish as it becomes a component of a company's family. Our paper gives a comprehensive review of the most significant obstacles that companies encounter while adopting agile techniques. Adaptability. The agile approach encompasses a variety of techniques, which each have different usage in various sectors. Because certain other standards existing today clash with agile methodology, the adaptation of any of the agile techniques in work environments posed a problem. In this paper, we will cover some of the challenges that firms face in adopting the agile software development life cycle.

Keywords:
Adobtability challenges, agile development, development methods, private organizations, public organizations, software development; work environments.

1. Introduction

The technological era seems to have had a major effect on how organizations are now managed. Organizations today live in an industry that is forever evolving. They're adapting to developing companies and opportunity, as well as shifting economic circumstances or the development of alternative products and services. The ability to develop quality software and put goods to market rapidly is a competitive edge in the business sector. Software design is a complicated process that demands the combined action of a multi-skilled team of organizational and functional experts. The shortage of strategic management and effective application development methodology adds complexity to the system, which might also rapidly get to be a roadblock for company. In software development, there's several think positives, with its own set of strengths and weaknesses. In software development, the waterfall and agile methodologies are now the almost always used [1].

The competitive climate allows for the adoption of newer and better project development methodologies so order to offer greater quality in less time. Organizations experience a range of problems and obstacles in innovating different equipment and methods. The significance of application development cleared the door for the separation of various techniques for distinct information systems. Agile software development [2] is one of the most notable techniques. Current methodology was becoming ubiquitous in both small and big software firms throughout the world. The techniques were initially invented for small, co-located projects [3]. “Groups must deliberately develop toward the optimal mix of agile and scheme approaches that matches their situation,” according to Boehm.

Clearly, most enterprises cannot disregard the agile wave, but implementation of green approaches will probably give numerous issues for enterprises entrenched in classical systems development methodologies, even as two software engineering approaches are based on opposing concepts. Software development process adjustments, so according past research, are complex organizational occurrences that could be performed simply by replacing current methods and software with fresh ones. Such adjustments may have an effect on the organization, culture, and management techniques, among several other things. As a corollary, knowing the implications of a change phenomenon all throughout system is an essential first step in organizing and executing such changes [4].

Furthermore, the research reports on agile practices that exist in the both scientific and non-scientific settings appear to predominantly describe favorable outcomes of agile adoption. Adoption of agile approaches, on the other

Manuscript received August 5, 2024
Manuscript revised August 20, 2024
https://doi.org/10.22937/IJCSNS.2024.24.8.15

extreme, has proved to be a challenging process. Agile adopters, it has been stated, are typically ignorant of what agile adoption truly implies and how substantial a shift is expected. While the positives of agile techniques have been universally acknowledged, proxy indicator is normally regarded to be weak. The fact that companies usually have relevant data to demonstrate the effectiveness of development methods before and after adoption is one cause for this [5]. For several years, software vendors have employed the waterfall model and its successor approaches, known as regulated or present lifestyle, as their primary software development life cycle.

Throughout the last 40 years, these techniques were well, and software firms have truly responded to them. Conventional methods, but at the other hand, include fundamental disadvantages, such as unnecessary and excessive data, delivery delays, no change in specifications, customer frustration, and so on. To fix the challenges of traditional approaches, agile techniques have been suggested. The Agile manifesto's concepts mostly have filled with new principles that encourage software firms to adopt agile methods instead of just traditional approaches. One of the main features of the Agile software development methodology is that it values people, their roles, and interaction over methods and regulations [6].

Agile players are getting changing their creative initiatives in leading to new in their settings. Their product development processes are less rigid as either a result of this habit, which lowers cycle times and enables them to adapt faster to changing customer needs. Scrum is amongst the most well-known and deployed paradigms under the hood of agile methodology when it comes to receiving such maximum versatility in the design process using agile techniques.

It's extremely well-suited to quick and responsive product planning and maintenance, and also some complicated and create successful [7]. Companies have the potential to define their competencies in order to build abilities to respond to various contexts and technical progress as a result of the global competition and rapid change in needs of consumers. The most essential issue for plan is to control what their company's core capabilities are and how such competencies be linked to core products and finished goods [8]. Agile methods procurement now is growing rapidly in public sector computer procurement projects. Agile methodologies were primarily done in small teams, projects, and organizations, but in past months, they have also been scaled up for its use in big system development and dispersed systems engineering.

With exception of a few increased research companies, state entities and public entities have been sluggish to adopt agile techniques. Agile methodologies were used by organizations in order to increase the outcomes of their IT projects. The positives of agile methodologies to companies include the flexibility to accommodate shifting priority, greater coordination between IT and business objectives, increased system performance, and increased customer, user, and shareholder value with both the software product. However, there's certain barriers to applying agile methodology in the public sector. Both highly hierarchical systems are inconsistent with agile methodology. It is important to translate to a physical prototype in which the process are formal and determined by the development team [9].

Many companies, especially the big ones, have spent many years transforming their cultures such that processes were codified and followed. Conversely, agile methodologies can achieve better results for the public sector than proposal approaches or big bang deployments [10]. The applicability of any agile methods in a corporate setting proved problematic owing to existing standards that disagreed with agile. In this paper, we will address some of the difficulties that major corporations have in adopting an innovative software development life cycle [1]. Furthermore, we analyze our observations and finish with such a summary of findings as well as suggestions for future research [11].

2. Background

Agile approaches were officially recognized through the development of the agile manifesto around with a generation later. These methodologies emphasize many ideologies in the software development sector, such as quick delivery, quality improvements, change acceptability, customer happiness, and light documenting, amongst other. Scrum, Extreme Programming (XP), Feature Driven Development (FDD), Crystal family, Dynamic Systems Development Methodology (DSDM), lean, and other techniques are all included in Agile umbrella.

They've founded a special business strategy is focused on managing projects or the system developer [12]. Some many research papers about implementing software development in traditional software development projects already have been compiled in IEEE, ACM, and software engineering journals, and they identify the challenges, issues, and influence of using scrum on software development productivity and quality. Agile adoption always presents special problems, demanding fundamental organizational reforms for a satisfactory completion.

As a basis, previous research by other scientists, it would be a wonderful chance to study on the present position of agile adoption challenges and the effects of agile on software quality. Small to medium businesses that use fast in their organization got selected for this study by taking surveys and compilation of data on Locating Agile, the Influence of Agile, Resistor and Acceptor Factors with in Implementation of Agile Methods, or the Influence of Agile on Quality Management [2].

3. Agil Software Development Adaptability

Agile is a series of tools that put a set of principles and ideals described in the agile manifesto in to the effect. We must examine the implementation of the key agile processes such as Scrum, Kanban, Extreme Programming (XP), Lean Development, Crystal, Dynamic System Development Method (DSDM), and Feature Driven Development while addressing and defining the challenges of agile in large corporations [1].

3.1 Main Agile methodologies

The below is the list of main agile methodologies with their definitions and area of focus during applicability:

3.1.1 Scrum

Agile Scrum is a lightweight, iterative approach that provides processes for controlling and managing the program and process improvement process. Because the updates are sequential and incrementally in terms of the outcomes to construct object-oriented software, Sprint is a blend of the Iterative development models. Scrum was developed to improve development pace and harmonize people and organizations. Scrum offers an interactive way of sorting on a variety of work with a variety of effects, with perks such as dynamic sprint requirement selection and thus no particular measures to be taken [13].

3.1.2 Kanban

Technique is a fairly essential topic that is used in various production facilities. The Kanban Methodology has subsequently been used multitude of manufacturing, disassembly, and demand planning with huge achievement. It's a study area strategy that allows a team be even more successful by eliminating idle time. Idle time may emerge in any operation, workflow, or procedure, and it's linked closely to opportunity on the inside of the process. Overproduction, superfluous motion, mistakes, spent waiting, and wait are all indicators of loss in the Kanban [14].

3.1.2 Extreme Programming XP

XP Is a lean technique that simplifies most of the traditional design and development process, particularly long criteria formulation and substantial documentation. It stresses the need for more small development teams and simple codes [15]. The four main operations of the XP life span are coding, testing, observing, and planning [4]. Transparency, cleanliness, feedback, and boldness are four ways that XP benefits a software project. It encourages constant contact with the consumer and teams, and also some retaining simplicity, delivering frequent feedback through monitoring, and aggressively dealing with problems [16].

3.1.3 Lean Development Methodology

Lean development is an agile method that aims to reduce construction money and effort, reducing defects, and providing really the functions that the function is defined. The Minimum Viable Product (MVP) methodology is a Lean technique in which a team delivers a bare-bones edition of its new product, learns from clients want they want, don't like, or want to be added, but then serializes relying on that input. Lean development advocates for generating as few samples as possible, but developing the brand line on customer feedback or users [1].

3.1.4 Crystal

Crystal Methods is an agile SDM that is predicated on the premise that individuals have a greater impact on innovation development than systems or techniques [12]. Crystal Methods is a collection of methodological components that businesses may come together to create methods designed for specific projects. Large or stability projects need a huge amount of method components than shorter, non-critical operations. Organizations could only build and apply as much strategy as their organization requires utilizing Crystal Methods[17].

3.1.5 Dynamic System Development Method (DSDM)

The Dynamic Systems Development Method is a framework that combines a lot of current project management expertise. The DSDM structure has expanded from its beginning in the software engineering community to become a serval dimensions for sophisticated major issue assignments as application development, instructional design, and as such corporate development activities have fused [18].

3.1.6 Feature Driven Development (FDD)

FDD is a highly iterative, agile, and highly adaptable software development approach. At all phases, quality is paramount. At all steps, delivers periodic, measurable operational solutions. Allows developers to create with interesting and precise performance and relevant data with the smallest quantity of energy and interruption. Clients, administrators, and technicians all like him [19].

Table 1: Different Agile Methodologies

#	Agile Methodology	Focus Areas
1	**Scrum**	1. Incremental Development 2. Product deliverables in short time (Sprint) 3. Working product and features
2	**Kanban**	1. Development, Operation & Security 2.Work Visualization and Transparency 3. Work In progress Limit
3	**Extreme Programming (XP)**	1. High Quality Software 2. High quality life for the development team 3. Frequent releases 4. Engineering Best practices
4	**Lean Development**	1. Time and resources optimization 2. Release Minimum Viable product
5	**Crystal**	1. People empowerment not processes 2. Communication, transparency and accountability 3. Project uniqueness
6	**Dynamic System Development Method (DSDM)**	1. Project life cycle with governance 2. Projects with 4 phases
7	**Feature Driven Development (FDD)**	1. Customer centric 2. Feature based development

The above summarizes main agile methodologies commonly used in different setup. However, in next section this document we will focus on Scrum Agile methodology in analysis

4. Using Earlier research identifying challenges in adopting agile Methodologies

Agile methodologies have indeed been extremely prevalent in the last fifteen years, and the types of companies that are using them have changed immensely. This shift is evident in studies about the use of agile methods. Early adopters, single team research, and application in small businesses were the subject of early research; later, the high voltage to multi-team implementation and diffusion in investment cost [11]. Upgrades to software development process are a complicated organizational transformation that can't be achieved done by removing instruments and processes. In this respect, adopting agile development is analogous to other organizational development events, and number of writers have noted major obstacles based with their own encounters.

Linval et al. claim the much more difficult aspect of implementing agile methodologies is mixing with the existing environment, but Cohn and Ford indicate that neglecting to encourage any stakeholder group to utilize the new approach could have a negligible effect on the project's outcome. Both highlight the importance of grasping the larger business culture, as well as the systems and systems that support it[15]. Time-boxed small updates of functional software are still at the foundation of agile methods. It necessitates different amounts of interaction (which results in even less documentation) but also solid technology discipline from personality teams that really are adaptable to change [20].

4.1 Challenges for small teams and early adopters

Several researches have been carried on the problems that a development team may encounter while using agile techniques. In software development, these techniques stress the human aspect, whilst agile development focuses on individual skills and abilities. Customers, developers, and other stakeholders must all take a leading role in an agile process map before it could be successfully implemented. Because the agility setup is still very much self-driven and personality, individuals must also be focused on working according to the agile specification of various roles.

There could also be psychological impediments to performance with agile methodologies [11]. Leadership styles, software process techniques, and software development practices are all instances of agile practices that could be used to enable a method accomplish its agile objectives. Compared to standard techniques, agile processes specialize in resolving unstable objectives, delivering integrated advertising in small distances, with top standard, and on money.

Agile techniques are attractive and feasible solutions for improving performance, project budget controls, connecting with the organization's business plan, and satisfying customers often and continually, based on the current constraints of software development. Adoption is a vital part in enabling the company to enjoy the benefits of speed and solving the typical issues raised all throughout adoption process [21]. A number of studies on team in agile software development have now been conducted on even a variety of issues. Some may have focused on themes such as knowledge and personal traits, which have been critical to team composition. Someone else has concentrated on establishing task-effective conventions in groups, team member enthusiasm, and the relevance of a team vision.

Others have emphasized on how teams connect during daily stand-up meetings, how teams take decisions, and how to establish self-management. Some have offered frameworks to aid cooperation improvement. Another field of study has become the performance of agile software development projects as a collective. Evaluations of the consequences of teamwork are referred to as team performance. The quality of the active contribution, the team's capability to achieve strategic goals and objectives, and the willingness of team members to work together during the future are example of such consequences [22].

Team adaptability was shown to be an essential antecedent for solid performance in ports connected in highly dynamic situations along with teams encountering creativity responsibilities. The potential of a team to modify team performance activities in response to indications from the environment. it is important that resulted in effective group performance was termed as team adaptability [23].

4.2 Challenges for large organizations

Agile techniques' success in small, co-located teams has prompted their application in other domains: Agile approaches are frequently being used in large-scale projects. Consequently, attaining the same cost advantages in these fields as on the "home turf" of agile techniques is problematic. Agile techniques are predicated on the premise that high-quality software may also be created by small teams employing continuous design improvement and testing based on prompt response and change.

New issues develop as agile development methodologies have been used to major projects. Prefund mental principles in agile methodology are heavily skewed when these approaches have been used in large-scale projects, per the “top burning research question” poll. Self-management is a central tenet of agile techniques, although studies from sectors besides software development suggests that it might impede the power to interact well across teams. In sizable agile, group members also must have an appropriate organizational infrastructure and connect extensively with specialists from outside team, while the organizations must identity.

When several teams are working in parallel, having a developing architecture might hamper project progress, and even some techniques, including the scrum of scrum, have been proved to be unproductive in big projects [24]. Vast companies regularly have huge and dispersed development organizations working on large projects, demanding the expansion of change management. According to Leffingwell 2007, scaling comprises a multitude of challenges, notably coordination between many agile projects, a shortage of up-front design, a paucity of requirement specification, along with all the issues of remote projects, although most major corporations are dispersed.

Remains a major problem, many large corporations have opted to embrace agile techniques, given the existence of research here about how to scale scrum methodology to complex projects and how to begin implementing agile changes in large organizations [25]. When adopting continuous delivery approaches in large-scale projects, influencing factors in agile development are vastly improved. Self-management is a central tenet of agile techniques, although research from disciplines other than software development demonstrates that this could impede the potential to coordinate properly across teams. In sizable agile, group members also must have an appropriate organizational network and engage closely with expertise outside team, while the teams need inner [24].

4.3 Challenges for public organizations

The public sector is the component of the economy that is responsible for implementing various public services. With the expansion and development of technological advances, state agencies have increasingly begun to include software products throughout their development processes. As a response, the incentive to implement different and improved techniques to Software Engineering (SE) was becoming essential for the future of software packages in government. In addition to enhancing the performance of their IT projects, listed companies have been employing agile methods.

The merits of DevOps to businesses include the opportunity to handle fluctuating priorities, improved coordination between this and corporate objectives, increased software quality, and high voltage gain satisfaction with the final output. The introduction of change management in the public sector, but at the other hand, is extremely problematic. Institutional highly hierarchical institutions are unsuitable with iterative development. It is important to translate to a physical prototype in which the operations are informal and determined by the project team after many public entities, especially for large, have spent years transforming their mindset whereby the processes were codified and followed.

However, current methodology can provide better outcomes for the public sector than plan-driven ones. Delivery with a boom there's very few investigations on use of the agile method in government enterprises. However, a tiny majority of scholars do touch on the matter [26] It was observed using these agile techniques when working for the state was challenging for only certain firms, as agile methods were not implemented there and government entities were inexperienced with agile methodologies. If there is a proposal to apply agile methodologies, it should be reflected in the public organization's acquisition announcements. When the project starts, the project team members should settle on the approach to employ and how the duties will be defined and allotted [11]. Eventually, the stakeholder schedule in the public sector encompasses a wide variety of parties with diverse histories, duties, expectations, political aspirations, and ambitions from a company value viewpoint.

The clarity of project objectives and targets is hardly ever developed by incorporating and regulating this. As a sense, ensuring project "effectiveness" from either the start is pretty difficult. Emerald has the real challenges of just this journal including a full transcript archive. Many projects have failed every time related to budget, time, and/or scope management problems, commonly termed as venture "efficiency" difficulties. Have used a public software project as an example, Fryberg and Buzzier (2011) reported on high numbers of out-of-control technology projects.

A lack of appropriate project governance structures, especially multi-stakeholder involvement during the early phases of the project when confusion as to what should be done or by whom, occasioned a strong negative inventory levels that usually lasts all throughout project life-cycle, was identified as among the most compelling reasons for the clearly evident failures. Similarly, such programs need communication and collaboration, flexibility, and agility in development process, organizational, and performance practices in order to leverage change or incorporate different stakeholders in creating value early and frequently. Previous project governance research has mostly focused on static organizational functions, leaving minimal space for investigation on micro-level instrumental in fostering that emerge across the life cycle [25].

5. Agile Adoptability Challenges in Organizations

The preceding usefulness of Agile Software development in software project is based on my own experience and even the methodology's methodologies. The scalability of Scrum guiding principles in large corporations was necessary to support us to take techniques from several other agile methodology and even integrate several best approaches. This part will describe some of the problems of agile adapting in companies and organizations based on personal experience as well as facts from other networks and academics [11].

5.1 Overall

This section will provide readers with an understanding of many of the major concerns and challenges that business organizations face while employing Agile method to create software:

- Lack of document or Incompleteness
- Gaps in team's communication
- Lack of technical Skills
- Stakeholder communication and involvement
- Education, experience and commitment

TABLE 2 SOURCES OF DIFFERENT CHALLENGES

The source of the challenge	Challenge description
Lack of Documentation	Working software versus thorough documentation has been one of the agile principles, which is largely misunderstood as "no documentation at all." Furthermore, agile methodologies were meant to be used and performed in small, body software projects at beginning. When evaluating a small system in a challenging way to a complex region in a complex world, there are different types of documentation required. In a large public organization, exchange of information in a broader meaning is expected than in a small private company [11]. Scrum, like other agile software development processes, lowers the volume of documentation required. In essence, agile techniques urge for the code to be a document in of itself. As a result, agile-methods-trained developers use more annotations in their code. Several developers indicated when they included extra descriptions to any challenging code and any adjustments they made. Many developers agree, however, that without documentation, it is exceedingly hard to conduct jobs for developers working on sections of the system they never have worked on it before, but also for new developers with hardly any history with the project [25].
Gaps in team's communication	Because the squad was founded to communicate effectively, another of the obstacles in agile methodology is

	establishing communication process within the squad along with external communicate directly with those other stakeholders in possible to lessen inefficiencies [1]. When team is working on various projects and the plan alters often, as well as when principles established are absent, communication may well be tough. Such issues come up, for example, because when latest update is not prioritized or susceptible to approaches to measuring, then when knowledge for practical is not conducted in a timely way [27].
Lack of technical Skills	The agile approach gave the assembled teams (Squads) a sense of flexibility and authority. To prevent generating substandard software, there is a demand for experienced employees to communicate on the original document. Employee turnover has become a huge problem towards some extent since this is a pertinent factor [1]. Fear among developers as a consequence of the awareness of competence shortage was noted in seventeen of the firms examined. Stand-up meetings, an onsite consumer, and the employment of storyboards and whiteboards, so according interviewees, made development defects extremely apparent to the rest of the team, because all these techniques demand direct and continual contact and collaboration among team members. Storyboards, for example, keep track of sprint backlog and highlight a developer's lack of interest [28].
Stakeholder communication and involvement	It's harder included stakeholders in the strategy since agile planning and specification work is completed gradually across the project. Also, because to the (incremental) product launch pattern, integrating customers and end users in pilot testing is a complex process. Especially since its Agency has a feedback from the end user groups that are geographically separated [11]. Feedback from customers is a priceless step in ensuring that the product being produced and released continually adds benefit to the customer. It mandates that company officials offer input on the functionalities that've been established. The majority of the players struggle to acquire key business stakeholders to participate sprint mass protests. Some people in the company comes to our website every now and then to develop relationships. Others may not feel the same way, and they want to meet regularly for only them [29].
Education, experience and commitment	This shortage of collaboration strategies in inexperienced companies clearly indicates that there is indeed a significant demand for highly-educated software engineers in the agile arena who would have the needed cooperation and skills in communication, and also the flexibility to embrace the agile principles. As a corollary, it's important to not only offer students with a solid awareness of what to expect while functioning in personality, agile software development teams, but to also discuss and feel the agile concepts with them throughout an actual student project [30].

6. Conclusions

We shared our results on the challenges of implementing agile methodologies in a public sector in this article. Documentation, people education, experience, and dedication, stakeholder communication and engagement, responsibilities in an agile setup, location of agile teams, legislation, and complexity of SW architecture and interoperability have all been acknowledged as challenges. There're a few caveats to this study that just might lead to more exploration. For small to bigger and public organizations, the implementation plan for software development is linked with the progressive software development process, with the top priorities being lowered costs, productivity improvements, and delighted customers.

The use of agile in so many major markets in order to attain the primary benefits of agility agile adaptation has a range of problems, most of which have been discussed previously but are not limited to them. The connection of sections two and three for the application of agile processes and constraints in companies improves the understanding for major organizations' future usage of agile. There is a need to work on new methodologies that will let companies to adapt to the problems outlined above while sustaining agility and flexibility.

References

[1] E. Mutabaruka, "Agile Methodology Software Development Adaptability Challenges in Corporate Organization," *Available at SSRN 3851349*, 2021.

[2] S. A. Ruk, M. F. Khan, S. G. Khan, and S. M. Zia, "A survey on Adopting Agile Software Development: Issues & Its impact on Software Quality," 2019, pp. 1–5.

[3] M. Paasivaara, "Adopting SAFe to scale agile in a globally distributed organization," 2017, pp. 36–40.

[4] S. Nerur, R. Mahapatra, and G. Mangalaraj, "Challenges of migrating to agile methodologies," *Communications of the ACM*, vol. 48, no. 5, pp. 72–78, 2005.

[5] M. Laanti, O. Salo, and P. Abrahamsson, "Agile methods rapidly replacing traditional methods at Nokia: A survey of opinions on agile transformation," *Information and Software Technology*, vol. 53, no. 3, pp. 276–290, 2011.

[6] T. J. Gandomani and M. Z. Nafchi, "Agile transition and adoption human-related challenges and issues: A Grounded Theory approach," *Computers in Human Behavior*, vol. 62, pp. 257–266, 2016.

[7] A. Grass, J. Backmann, and M. Hoegl, "From empowerment dynamics to team adaptability: Exploring and conceptualizing the continuous agile team innovation process," *Journal of Product Innovation Management*, vol. 37, no. 4, pp. 324–351, 2020.

[8] M. Danilovic and P. Leisner, "Analyzing core competence and core products for developing agile and adaptable corporation," 2007, pp. 49–60.

[9] M. Uhl-Bien and M. Arena, "Leadership for organizational adaptability: A theoretical synthesis and integrative framework," *The Leadership Quarterly*, vol. 29, no. 1, pp. 89–104, 2018.

[10] I. Vacari and R. Prikladnicki, "Adopting agile methods in the public sector: a systematic literature review.," presented at the Embrapa Informática Agropecuária-Artigo em anais de congresso (ALICE), 2015.

[11] J. Nuottila, K. Aaltonen, and J. Kujala, "Challenges of adopting agile methods in a public organization," *International Journal of Information Systems and Project Management*, vol. 4, no. 3, pp. 65–85, 2016.

[12] Y. Sharif, "Exploring facilitators of transition and adoption to agile methods: a grounded theory study," *Journal of Software*, vol. 9, no. 7, 2014.

[13] A. Srivastava, S. Bhardwaj, and S. Saraswat, "SCRUM model for agile methodology," 2017, pp. 864–869.

[14] R. B. Wakode, L. P. Raut, and P. Talmale, "Overview on kanban methodology and its implementation," *IJSRD-International Journal for Scientific Research & Development*, vol. 3, no. 02, pp. 2321–0613, 2015.

[15] J. Abdelnour-Nocera and H. Sharp, "Adopting agile in a large organization: balancing the old with the new," Technical report 2007/12, Milton Keynes, UK, 2007.

[16] L. Cao, K. Mohan, P. Xu, and B. Ramesh, "How extreme does extreme programming have to be? Adapting XP practices to large-scale projects," in *37th Annual Hawaii International Conference on System Sciences, 2004. Proceedings of the*, 2004, pp. 10-pp.

[17] J. A. Livermore, "Factors that impact implementing an agile software development methodology," 2007, pp. 82–86.

[18] B. J. Voigt, M. Glinz, and D.-I. C. Seybold, "Dynamic system development method," *no. January*, 2004.

[19] S. Goyal, "Major seminar on feature driven development," *Jennifer Schiller Chair of Applied Software Engineering*, 2008.

[20] K. Sureshchandra and J. Shrinivasavadhani, "Adopting agile in distributed development," 2008, pp. 217–221.

[21] A. S. Campanelli and F. S. Parreiras, "Agile methods tailoring–A systematic literature review," *Journal of Systems and Software*, vol. 110, pp. 85–100, 2015.

[22] T. Dingsøyr and Y. Lindsjørn, "Team performance in agile development teams: Findings from 18 focus groups," 2013, pp. 46–60.

[23] C. Schmidt, T. Kude, J. Tripp, A. Heinzl, and K. Spohrer, "Team adaptability in agile information systems development," 2013.

[24] T. Dingsøyr and N. B. Moe, "Towards principles of large-scale agile development," 2014, pp. 1–8.

[25] T. Lappi and K. Aaltonen, "Project governance in public sector agile software projects," *International Journal of Managing Projects in Business*, 2017.

[26] R. Phalnikar, V. Deshpande, and S. Joshi, "Applying agile principles for distributed software development," 2009, pp. 535–539.

[27] M. Hummel, C. Rosenkranz, and R. Holten, "The role of communication in agile systems development," *Business & Information Systems Engineering*, vol. 5, no. 5, pp. 343–355, 2013.

[28] K. Conboy, S. Coyle, and X. Wang, "People over Process: Key Challenges in Agile Development,(99), 48–57," 2010.

[29] G. Van Waardenburg and H. Van Vliet, "When agile meets the enterprise," *Information and software technology*, vol. 55, no. 12, pp. 2154–2171, 2013.

[30] A. Meier, M. Kropp, and G. Perellano, "Experience report of teaching agile collaboration and values: agile software development in large student teams," 2016, pp. 76–80.