

AGILE: ADVANTAGES, DISADVANTAGES, ENABLERS, AND BARRIERS

P. Lutwama¹, M. Dlulane¹, T. Pillay¹, F.S. Hassan^{1*} & S. Grobbelaar¹

ARTICLE INFO

Article details

Submitted by authors 3 Jul 2024
Accepted for publication 28 Oct 2024
Available online 13 Dec 2024

Contact details

* Corresponding author
ferdinand.hassan@tuks.co.za

Author affiliations

1 Department of Engineering and
Technology Management,
University of Pretoria, Pretoria,
South Africa

ORCID® identifiers

P. Lutwama
<https://orcid.org/0009-0007-1042-0024>

M. Dlulane
<https://orcid.org/0009-0002-0435-5768>

T. Pillay
<https://orcid.org/0009-0004-9475-5689>

F.S. Hassan
<https://orcid.org/0009-0005-2171-1332>

S. Grobbelaar
<https://orcid.org/0000-0001-8148-2440>

DOI

<http://dx.doi.org/10.7166/35-4-3058>

ABSTRACT

This scoping review provides an expanded view of the Agile methodology in the software industry - an iterative, incremental project management framework that is renowned for its use in software development. Research questions covering the advantages, disadvantages, enablers, and barriers to Agile were formulated to analyse accepted research papers from 2019 to 2023. Notable overlapping findings were found in the 25 studies that were included. The results revealed that human factors such as communication, conflict resolution, and understanding Agile are essential to the successful implementation of Agile software development.

OPSOMMING

Hierdie literatuur omvangsoorsig bied 'n uitgebreide siening van die Agile-metodologie in die sagteware-industrie - 'n iteratiewe, inkrementele projekbestuurraamwerk wat bekend is vir die gebruik daarvan in sagteware-ontwikkeling. Navorsingsvrae wat die voordele, nadele, instaatstellers en struikelblokke tot Agile dek, is geformuleer om aanvaarde navorsingsvraestelle van 2019 tot 2023 te ontleed. Opmerklike oorvleuelende bevindings is gevind in die 25 studies wat ingesluit is. Die resultate het aan die lig gebring dat menslike faktore soos kommunikasie, konflikoplossing en begrip van Agile noodsaaklik is vir die suksesvolle implementering van Agile-sagteware-ontwikkeling.

1. INTRODUCTION

Software development involves a series of non-routine, difficult, and constantly evolving tasks [1]. Similarly, managing software projects can be difficult, requiring the coordination of individual, team, and organisational resources [2]. Traditional project management approaches employ a waterfall methodology in which the client's expectations are clearly defined at the project's start, with the project thoroughly planned from beginning to end [3]. However, this approach is more suited for industries such as mining and construction, in which changes to a plan or process are rare after the project starts and impossible after certain milestones have been met - unlike software engineering, in which a lot of change is expected during the development process [4]. The inability of traditional project management techniques to handle the constant change that software development projects require has led software engineers to seek alternative solutions to manage their projects [4].

In 2001, the publication of the Agile manifesto served as a watershed moment for the software development industry [5]. It combined a series of ideas about and principles of iterative methods of planning and development [6]. 'Agile' is a broad term encompassing iterative and incremental software development methodologies to deliver functional software within relatively short timeframes [7]. Agile prioritises people over processes, and maintains a commitment to flexibility and adaptability amid uncertainty and complexity. Agile approaches prioritise a generative mindset, using only necessary processes, tools, procedures, and documentation [8]. In addition, Agile recognises that different situations require tailored solutions, methodologies, or approaches [4].

The focus of the Agile manifesto is its values - a series of statements that form the core of the Agile framework. These include putting individuals and interactions ahead of processes and tools, valuing working software over comprehensive documentation, collaborating with the customer rather than extensive contract negotiations, and the ability to be flexible and to respond to changes [9]. Since its release, Agile has become the preferred project management approach for software development organisations worldwide [5]. Many organisations adopt Agile methodologies in their software development projects to empower employees, promote self-management, and respond effectively to customer feedback [10].

Agile has undergone several developments since the publication of the Agile manifesto. A set of explanations of the manifesto called the Agile principles has since been published and reworked for various industries [11], [12]. According to Bustard [5], Agile methodologies are not adopted in a standard form; instead, teams customise and blend various methodologies to align with their unique organisational requirements. This approach suggests a sustainable continuous improvement model as organisations experiment with and refine Agile practices in order to discover the most effective combination for their contexts. Scrum, Kanban, eXtreme Programming (XP), lean development, and others are just some of the techniques and methodologies that have been developed for implementing Agile [13], [14].

Therefore, this study aims to explore the adoption of Agile in software development by focusing on four aspects: advantages, disadvantages, enablers, and barriers. Based on the study's aim, the following research questions were defined:

1. What are the potential advantages of Agile processes in a software development project?
2. What are the potential disadvantages of Agile processes in a software development project?
3. What are the enablers of Agile processes for a software development project in an organisation?
4. What are the barriers to Agile processes for a software development project in an organisation ?

2. LITERATURE STUDY

The history of Agile began before the publication of the Agile manifesto. Whitley *et al.* [15] conducted a comprehensive review of the origins of Agile, and traced the earliest related approaches to the 1930s. Walter Shewhart, a physicist and statistician, proposed a plan-do-study-act (PDSA) cycle to improve processes and products at Bell Labs [16]. This incremental and iterative methodology is regarded as the foundation of Agile, and has been iterated and refined over the years [17], none more so than by William Edwards Deming, a mentee of Shewhart [15]. The development and subsequent popularity of the Toyota production system (TPS) in Japan further influenced the growth of iterative approaches such as Agile [15].

The success of TPS and of Japanese manufacturing influenced the development of similar methods in the United States (US), with the manufacturing and aerospace industries leading the development of the agility paradigm and the ‘theory of agile manufacturing’ [15], although Abbas *et al.* [18] stated that Agile and iterative methods were already practised at the US National Aeronautics and Space Administration (NASA) as early as the 1960s. These approaches were transplanted into the software development industry and into the military and project management contexts [15]. However, the widespread popularity of the Agile approach is often attributed to the 2001 manifesto for Agile software development [14].

Since the publication of the manifesto and the subsequent increased uptake of Agile methods, there has been a lot of research into Agile [19]. Initial research focused on problems with adoption as the methods became mainstream [13]. For instance, closed-mindedness and a preference for traditional methods were some of the early barriers to adoption in some organisations [20]; conservative managers were unwilling to allow their young programmers to use what was considered a method that seemed less plan-driven [21]. In addition, the lack of facilities to situate collocated teams and inadequate training in Agile limited adoption [22]. On the other hand, personal interest and the influence of leading Agile proponents were key factors behind early adopters’ decision to use Agile [22].

Other literature has focused on post-adoption reviews [13]. Critical success factors, including the delivery strategy, the specific agile method chosen, and the team’s capability, have been identified as influential in Agile adoption [23]. Customer-centric factors, including commitment, collaboration, and satisfaction, have also been identified as critical success factors [24]. To sustain an Agile methodology, companies must commit to being completely ‘Agile’, focus on delivering business value, and maintain consistency [7]. Similarly, the relative advantage of the Agile methodology and the presence of an Agile coach in a company were factors that [25] identified as sustaining the continued use of Agile in organisations.

Some recent studies have explored the adoption of Agile in distributed settings [13], driven mainly by software development becoming a more global collaboration rather than a collocated one [26]. A global focus allows software development companies to access the best talent worldwide at a lower cost, to have easier access to customers, and to be able to maintain 24-hour operations [26]. However, cultural and time zone differences and the dispersion of teams affect communication and cooperation [27]. A four-step process for implementing a distributed Agile system was suggested by [28]. COVID-19 further entrenched the presence of distributed software development teams [29], and there is a raft of research on the effects of remote work on Agile [30], [31], [32].

This literature review demonstrates the depth and breadth of research on Agile methodology, and several literature review studies have been performed on various aspects of Agile in software development [33], [34], [35], [36]. This review adds to the body of review studies on Agile while providing a more recent and relevant outlook, especially in the wake of the increased adoption of remote work globally.

3. RESEARCH METHODOLOGY

3.1. Research questions and search strategy

A scoping review was conducted using a systematic search of digital research databases to determine and analyse the current research landscape in order to answer our proposed research questions. A scoping review is a systematic approach to reviewing the literature that has gained popularity owing to its ability to provide an overview of a broad research area, identify key characteristics of the existing body of literature, and map out the available evidence on a particular topic [37]. Scoping reviews are particularly useful in fields where the existing evidence is diverse or rapidly evolving, making it difficult to conduct a traditional systematic review [38].

The methodology of a scoping review typically involves several key stages: identifying the research question, searching for relevant studies, selecting studies based on predefined criteria, charting the data, and collating, summarising, and reporting the results [39]. The research questions were defined through search strings that included relevant information for studies, with exclusion strings and/or in-house filters to ensure that the captured research aligned with the research questions being presented.

Given the broad nature of Agile methodology research, particular criteria were shared throughout the research questions:

- The industry of note must be the software industry.
- Papers must have been published within the period 2019 to 2023, as papers published more recently may still be subject to peer review/analysis.
- Papers must not be opinion pieces that lack empirical evidence to support their conclusions.

The search strings were then applied to Scopus and Web of Science - which are digital research citation databases - to provide access to the publications required to answer the questions posed in this study.

The first research question, “*What are the potential advantages of Agile processes in a software development project?*”, was chosen to highlight the importance and value that an Agile approach could provide to a software development team. It would provide meaningful objectives for organisations to align their strategies and guidance on conducting an Agile approach to achieve success. The second research question, “*What are the potential disadvantages of Agile processes in a software development project?*”, was chosen to help organisations to check what they needed to avoid in order to achieve their goal when using Agile software. It would also assist in planning and budgeting for the project.

The third research question, “*What are the enablers of Agile processes for a software development project in an organisation?*”, was chosen to provide a basic checklist for those wishing to adopt Agile methodologies in the workplace, particularly for those in the software industry. This question assists with taking the previous research questions to the next level, thus providing a rounded view of Agile. The fourth research question, “*What are the barriers to Agile processes for a software development project in an organisation?*”, was chosen to underline major issues that software development projects face while implementing Agile processes.

These strings were then refined over multiple iterations to ensure that the research obtained was adequate, with further refinement done through the physical reading of the papers for potential further exclusions.

Table 1 provides the inclusion and exclusion criteria of the review.

Table 1: Research acquisition: Inclusion and exclusion criteria

Research question	Inclusion criteria	Exclusion criteria
RQ1: What are the potential advantages of Agile processes in a software development project?	Date range: 2019-2023 Language: English Relating to Agile methodologies and enablers Industry: Software	1. Opinion pieces 2. Theoretical frameworks without empirical evidence 3. Studies outside of the software industry 4. Studies pending peer review (if stated)
RQ2: What are the potential disadvantages of Agile processes in a software development project?	Date range: 2019-2023 Language: English Relating to Agile methodologies and enablers Industry: Software	
RQ3: What are the enablers of Agile processes for a software development project in an organisation?	Date range: 2019-2023 Language: English Relating to Agile methodologies and enablers Industry: Software	
RQ4: What are the barriers to Agile processes for a software development project in an organisation?	Date range: 2019-2023 Language: English Relating to Agile methodologies and enablers Industry: Software	

Table 2 provides an overview of the research papers, based on the databases chosen for analysis.

Table 2: Research acquisition: Overview of papers from databases

Research question	Source	Research string(s)	Number of relevant papers
RQ1: What are the potential advantages of Agile processes in a software development project?	Scopus	("agile advantages") OR ("agile benefits") AND ("software development") OR ("success rate") OR ("cost benefit")	Original Articles: 14 Excluded: 4 Final count: 10
RQ2: What are the potential disadvantages of Agile processes in a software development project?	Web of Science	("agile disadvantages") OR ("agile challenges") AND ("software development") OR ("software engineering")	Original Articles: 12 Excluded: 8 Final count: 4
RQ3: What are the enablers of Agile processes for a software development project in an organisation?	Web of Science	((AK=(agile development) OR AK=(agile team) OR AK=(agile methodology*) OR AK=(scrum) OR AK=(agile)) AND (AK=(enabler*) OR AK=(criteria) OR AK=(common)))	Original Articles: 34 Excluded: 28 Final count: 6
RQ4: What are the barriers to Agile processes for a software development project in an organisation?	Web of Science	("Agile" OR "Agile processes" OR "Agile methodology") AND ("software development" OR "software engineering") AND ("barriers" OR "challenges" OR "obstacles") AND ("organization" OR "organizational")	Original Articles: 23 Excluded: 18 Final count: 5

Using these parameters, a total of **25** papers were retrieved for analysis, each sectioned into the research questions through their research strings. These papers are detailed further in the references section for future review.

3.2. Data extraction and analysis

Once the relevant papers for the scoping review had been identified, the studies were imported into ATLAS.ti - a qualitative data tool for sorting and analysing text data - for further data extraction and analysis. The researchers used the tool to highlight relevant details and to gather consensus across relevant papers, thus allowing for the proposed research questions to be answered in full.

4. RESULTS AND DISCUSSIONS

4.1. What are the potential advantages of Agile processes in a software development project?

The capacity to provide high-quality software products efficiently and swiftly in response to changing requirements has become critical for enterprises that look to keep up in today's fast-paced and ever-changing software industry [40]. Agile processes place a strong emphasis on adaptability, teamwork, and iterative development [41]; this guides teams to respond swiftly to shifting consumer needs and dynamic markets [42]. Agile processes provide several benefits that could greatly improve project outcomes and customer satisfaction by encouraging close collaboration among cross-functional teams, accepting change as a normal part of the development process, and encouraging continuous improvement.

Essentially, the Agile methodology recognises that requirements can emerge, evolve, or change over time. The benefit is that it welcomes these changes rather than resists them; and as a result the software is customised to the end-user's needs [43]. The Agile manifesto emphasises regular reflection and adjusting team behaviour accordingly [11]. This fosters ongoing learning and enhancement through transparency, trust, and accountability, and results in improved software development processes by identifying areas for improvement and enhancing team performance [44]. The Agile approach also enables complex software

development projects to be broken down into simpler components. This leads to better knowledge sharing and communication with teams, increased productivity, and an overall higher success rate in the project [10], [43].

Another benefit of the iterative Agile software development process is that it allows for faster product delivery and early feedback [45]. Improved quality is achieved through refactoring, which refines the architectural blueprint and performance of the software. By embracing the principle of continuous attention to technical excellence and good design, teams benefit by developing valuable, reliable, and resilient software for future growth and change [43], [46]. Furthermore, although risk management is not alluded to in the Agile manifesto or principles [47], fault detection is possible, as it can be found earlier thanks to the frequent tests that are done on each iteration.

Moreover, the continuous feedback loop ensures that new features do not lead to potential risks or defects [43], [48]. Agile software development aims to increase stakeholder satisfaction by delivering value to customers in a timely manner. This is achieved through continuous collaboration, communication, and trust-building [8]. The core value of Agile lies in people and interaction ahead of processes [9]; so Agile approaches usually lead to motivated project development teams, fostering productivity and motivation through supportive leadership and clear goals [49], [50].

In addition, adopting Agile principles could optimise cost control by maximising the return on investment in software development projects. By introducing solutions to the market earlier, waste is reduced, and design simplicity is achieved, maximising the amount of work not done [51]. Overall, Agile methodology provides benefits in delivering value for the customer. Ultimately, by mastering the twelve principles of the Agile manifesto, software development projects could produce high-quality results efficiently and effectively [52].

4.2. What are the potential disadvantages of Agile processes in a software development project?

Although Agile software development approaches have many benefits, certain drawbacks have been identified by practitioners and researchers in the field. One of those disadvantages is that as teams become bigger, the effectiveness of Agile reduces. of Agile in large teams. Agile development works better in small teams. For large-scale projects, the expenses are normally too high when using Agile approaches, thus outweighing the benefits [53]. For projects to be successful, all the specifications must be clear, sufficient, and documented to avoid losing track of the project and help it to stick to schedule and budget [54].

Throughout the development lifecycle, emerging and changing specifications give software developers the flexibility to adjust specifications to meet the project's outcomes. Nevertheless, this tenet of Agile increases the possibility of never-ending projects and significantly reduces predictability in the project lifecycle's ultimate product delivery requirement [54]. Agile approaches are resisted by organisations with a strong waterfall development philosophy [55]. The Agile software development methodology places more of an emphasis on development than on design. It ignores product design, which can sometimes be difficult, and concentrates on procedures for obtaining requirements and writing code. Because Agile systems have less documentation, it is also unsuitable for maintenance [55].

Finally, there are some presumptions that must be met to benefit from the use of Agile approaches in development. Among these are the development team's and the clients' direct communication and collaboration, the project's constantly changing requirements, and the developers' strong personal backgrounds and skill sets [56]. It is best to look for other approaches to the development process if these presumptions do not apply to a software development project, in order to achieve better outcomes [56].

4.3. What are the enablers of Agile processes for a software development project in an organisation?

Software development projects have succeeded because of the adequate use of their processes, tools, and mindset [57]. However, consensus about the importance of understanding when Agile could be implemented ahead of other project management methodologies is still lacking [58]. In contributing to this debate, we identified several criteria that provide insights into typical aspects of enabling effective Agile software development.

Communication is one of the identified enablers of Agile software development. Good communication is required throughout the software development cycle [59], from ensuring that teams are adequately introduced to the concepts of Agile to ensuring that the correct communication streams are updated and documented according to the project scale [60], [61], [62]. The level of customer input significantly influences the enabling of Agile software development in the industry. Companies adopt Agile to meet the software complexities required by customers; and owing to the rapidly changing landscape, the user's needs should be able to evolve and not affect the delivered product [60]. The attainment of the product remains a foundational tenet of a customer's need, and thus projects that require a great deal of customer input and review benefit from the use of Agile [63].

As with any project, development costs affect the attractiveness of the project team to external stakeholders. Preserving the cost-benefit ratio of each project can be especially important. Thus Agile software development is typically used for projects that do not shy away from potentially higher costs, as the incremental software review may require customers to inject more capital into the development [60], [63]. A company's decision to implement Agile typically includes the potential to save time, as costs and time typically complement each other in efficiency [61]. Because Agile implements an iterative process to manage and meet customer requirements, projects that can maintain and manage what could become a complicated development structure could use Agile. This could be done by efficiently analysing iteration times [60], [61] or refining project-suggested sub-criteria in the planning phase: speed, sustainability, responsiveness, learning, flexibility, and leanness [62].

In adopting Agile, changing the way of thinking in teams becomes a deliverable that needs attention, with adequate attention to team training being a suggested procedure. The higher the number of teams/individuals that need training, the more difficult the process becomes [60], [61]. Similarly, a team's mindset depends on various factors of human behaviour that may not respond rationally to quantitative data. As a result, Agile should be done with teams that show willingness or in company cultures that adopt change as a core value [64].

Overall, Agile is used to be flexible and to shorten the development process while allowing for amendments as customers feed back additional information and require changes. The flexibility of Agile sub-methodologies provides ample room to implement various types (or hybrids) to suit team needs, such as in Scrum and Kanban: the former is suited to a structured framework with defined roles and ceremonies, while the latter suits projects requiring continuous delivery [65].

4.4. What are the barriers to Agile processes for a software development project in an organisation?

Agile methodologies have revolutionised software development by offering flexibility, adaptability, and responsiveness to change. However, implementing Agile processes in an organisation comes with its own set of problems. Software development projects can involve actors from different organisations, and therefore, each actor follows their own business strategy, resulting in different hard-to-align interests, which could cause conflict [66]. Handling the trade-off between pleasing partners and maintaining flexibility may also become an issue. Moreover, weak assumptions about how to handle organisations give rise to complexity and uncertainty when adopted in large organisations. Lack of training before Agile adoption results in a lack of clarity about the new Agile working method and about how it would be implemented [67].

Effective communication between all stakeholders to ensure that they align on aspects such as roadmaps, prioritisation, and handling customer feedback and data can be difficult [66]. It is also hard to ensure that appropriate communication and information are preserved and altered. Part of this is that there is usually poor documentation of the work done [48]. It also takes work to deliver when there is continuous and potentially overwhelming customer feedback. Customers could also have a broad range of requirements, and the challenge is to balance the implementation of these requirements by prioritising and communicating when the requirements will be delivered [66]. Ineffective communication also contributes to increased technical debt - a group of short-term-focused design or implementation features that create a technical environment that may make modifications in the future more expensive or even impossible [68].

The ability to balance the need for teams to be autonomous and the need for alignment, quality, and process control remains difficult. Organisational alignment hinders the team's ability to work autonomously. There are many variations of Agile, and teams usually have the autonomy to choose what works for them. However, this may make collaboration with other teams difficult and create problems in delivery [10]. While teams work independently, their work is related, and it is difficult to predict the exact delivery time of other teams in the organisation [48]. This is also reflected in problems with testing because teams work on small parts of complex projects and lack a holistic view of the system [48].

Saeeda *et al.* [48] listed a number of problems with testing that may arise, including a lack of acceptance testing owing to the unavailability of customers to ensure that products meet their requirements; a lack of regression testing because of the complexity of the projects, and the pace at which they change; the lack of skills in the team to perform regression testing; and the lack of integration testing, as it is difficult for independent teams to test the system as a whole and how individual elements will work together [48]. Given the pace and pressure in these organisations and the lack of skills, these problems with testing are usually not prioritised [48].

5. CONCLUSION

This research aimed to answer long-standing questions about Agile software development by using a more modern outlook. We discovered that there was some overlap in the four research questions about what was considered an advantage of Agile and a typical enabler of Agile, with the opposing questions having overlaps as well. The success of implementing the methodology is intrinsically related to how well the implementation occurs; a sufficient awareness of the methodology and of its requirements allows teams to reap the benefits, while a lack of established processes may present pain points that impede the results of the system.

Similarly, the type of organisation/team may either benefit or experience drawbacks from implementing Agile methodologies purely because of human factors. Interpersonal conflicts and stakeholder engagements test the foundations of Agile's iterative nature, while personal preferences in managing teams can impede the full implementation of Agile. Communication and an understanding of Agile are essential for successful implementation, especially when organisations implement different frameworks (e.g., Scrum, Kanban, hybrids). Owing to the broad sub-categories of Agile frameworks, companies should focus on confirming team understanding throughout the implementation and utilisation processes.

This research was limited by its self-imposed review window. In addition, the review results may not apply to all contexts. One potential for future studies, however, would be to investigate the impact of company culture on the adoption of Agile methodologies. Another could be to evaluate whether the prioritisation of customer involvement in Agile should be emphasised. In summary, organisations need to recognise and address the potential advantages, disadvantages, enablers, and barriers when implementing Agile processes. By leveraging Agile's strengths while considering and proactively mitigating the challenges that Agile methodologies may introduce, organisations could maximise the benefits of adopting Agile and foster a culture of innovation, responsiveness, and continuous improvement in their software development projects.

REFERENCES

- [1] **Brodbeck FC.** Communication and performance in software development projects. *European Journal of Work and Organizational Psychology*. 2001;10(1):73-94. doi:10.1080/13594320042000043
- [2] **Colomo-Palacios R, Casado-Lumbreras C, Soto-Acosta P, García-Crespo Á.** Decisions in software development projects management: An exploratory study. *Behaviour & Information Technology*. 2013;32(11):1077-1085. doi:10.1080/0144929X.2011.630414
- [3] **Thesing T, Feldmann C, Burchardt M.** Agile versus waterfall project management: Decision model for selecting the appropriate approach to a project. *Procedia Computer Science*. 2021;181:746-756. doi:10.1016/j.procs.2021.01.227
- [4] **Munteanu VP, Dragos P.** The case for Agile methodologies against traditional ones in financial software projects. *European Journal of Business and Management Research*. 2021;6(1):134-141. doi:10.24018/ejbmr.2021.6.1.741
- [5] **Bustard D.** Beyond mainstream adoption: From Agile software development to Agile organizational change. *2012 IEEE 19th International Conference and Workshops on Engineering of Computer-Based Systems*. 2012. pp. 90-97. doi:10.1109/ECBS.2012.18

- [6] **Pope-Ruark R.** Introducing Agile project management strategies in technical and professional communication courses. *Journal of Business and Technical Communication*. 2015;29(1):112-133. doi:10.1177/1050651914548456
- [7] **Russo NL, Fitzgerald G, Shams S.** Exploring adoption and use of Agile methods: A comparative case study. *Proceedings of the Nineteenth Americas Conference on Information Systems*, 2013. <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=29b5e24ff84c6569fc417b5d7f56268c105c3833>.
- [8] **Fernandez DJ, Fernandez JD.** Agile project management – Agilism versus traditional approaches. *Journal of Computer Information Systems*. 2008;49(2):10-17. <https://www.tandfonline.com/doi/abs/10.1080/08874417.2009.11646044>. doi:10.1080/08874417.2009.11646044
- [9] **Beck K, Beedle M, Van Bennekum A, Cockburn A, Cunningham W, Fowler M, Martin RC, Mellor S, Thomas D, Grenning J, et al.** Agile manifesto. *Agile Alliance*. 2001 [accessed 2024 Jun 22]. <https://www.agilealliance.org/wp-content/uploads/2019/09/agile-manifesto-download-2019.pdf>
- [10] **Moe NB, Šmite D, Paasivaara M, Lassenius C.** Finding the sweet spot for organizational control and team autonomy in large-scale Agile software development. *Empirical Software Engineering*. 2021;26(5):101. doi:10.1007/s10664-021-09967-3
- [11] **Marbach P, Rosser L, Osvalds G, Lempia D.** Principles for Agile development. *INCOSE International Symposium*. 2015;25(1):524-537. doi:10.1002/j.2334-5837.2015.00079.x
- [12] **Leffingwell D, Martens R, Zamora M.** *Principles of Agile architecture: Intentional architecture in enterprise-class systems*. Leffingwell, & Rally Software Development Corp. 2008. https://scalingsoftwareagility.wordpress.com/wp-content/uploads/2008/08/principles_agile_architecture.pdf
- [13] **Dingsøyr T, Nerur S, Balijepally V, Moe NB.** A decade of Agile methodologies: Towards explaining Agile software development. *Journal of Systems and Software*. 2012;85(6):1213-1221. doi:10.1016/j.jss.2012.02.033
- [14] **Gorgadze N.** Agile project management approach, methods and its application in Georgia. *Globalization and Business*. 2021;6(12):191-196.
- [15] **Whiteley A, Pollack J, Matous P.** The origins of Agile and iterative methods. *The Journal of Modern Project Management*. 2021;8(3).
- [16] **Victor R.** Iterative and incremental development: A brief history. *IEEE Computer Society*. 2003;36(6):47-56.
- [17] **Best M, Neuhauser D.** Walter A Shewhart, 1924, and the Hawthorne factory. *BMJ Quality & Safety*. 2006;15(2):142-143.
- [18] **Abbas N, Gravell AM, Wills GB.** Historical roots of Agile methods: Where did “Agile thinking” come from? *Agile Processes and eXtreme programming in Software Engineering*. 2008. pp. 94-103. doi:10.1007/978-3-540-68255-4_10
- [19] **Cooper RG, Sommer AF.** Agile-stage-gate for manufacturers. *Research-Technology Management*. 2018;61(2):17-26. doi:10.1080/08956308.2018.1421380
- [20] **Mahanti A.** Challenges in enterprise adoption of Agile methods – A survey. *Journal of Computing and Information Technology*. 2006;14(3):197. doi:10.2498/cit.2006.03.03
- [21] **Boehm B.** Get ready for Agile methods, with care. *Computer*. 2002;35(1):64-69. doi:10.1109/2.976920
- [22] **Vijayarathy L, Turk D.** Agile software development: A survey of early adopters. *Journal of Information Technology Management*. 2008;19(2):1-8.
- [23] **Chow T, Cao D-B.** A survey study of critical success factors in Agile software projects. *Journal of Systems and Software*. 2008;81(6):961-971. doi:10.1016/j.jss.2007.08.020
- [24] **Misra SC, Kumar V, Kumar U.** Identifying some important success factors in adopting Agile software development practices. *Journal of Systems and Software*. 2009;82(11):1869-1890. doi:10.1016/j.jss.2009.05.052
- [25] **Senapathi M, Srinivasan A.** An empirical investigation of the factors affecting Agile usage. *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*. 2014. pp. 1-10. doi:10.1145/2601248.2601253
- [26] **Chadli SY, Idri A, Ros JN, Fernández-Alemán JL, De Gea JMC, Toval A.** Software project management tools in global software development: A systematic mapping study. *SpringerPlus*. 2016;5(1):2006. doi:10.1186/s40064-016-3670-7
- [27] **Shrivastava SV, Date M.** Distributed Agile software development: A review. *Journal of Computer Science and Engineering*. arXiv preprint arXiv:1006.1955. 2010;1(1) pp. 10-17.
- [28] **Sureshchandra K, Shrinivasavadhani J.** Adopting Agile in distributed development. *2008 IEEE International Conference on Global Software Engineering*. 2008. pp. 217-221.
- [29] **NicCanna C, Razzak MA, Noll J, Beecham S.** Globally distributed development during COVID-19. *2021 IEEE/ACM 8th International Workshop on Software Engineering Research and Industrial Practice (SER&IP)*. 2021. pp. 18-25. doi:10.1109/SER-IP52554.2021.00010

- [30] Ozkan N, Erdil O, Gök MŞ. Agile teams working from home during the Covid-19 pandemic: A literature review on new advantages and challenges. *Lecture Notes in Business Information Processing Lean and Agile Software Development*. 2022. pp. 38-60. doi:10.1007/978-3-030-94238-0_3
- [31] Mancl D, Fraser SD. COVID-19's influence on the future of Agile. *Lecture Notes in Business Information Processing*. 2020. 396, pp. 309-316. doi:10.1007/978-3-030-58858-8_32
- [32] Neumann M, Bogdanov Y, Sager S. The Covid 19 pandemic and its effects on Agile software development. *The 5th International Conference on Software Engineering and Information Management (ICSIM)*. 2022. pp. 51-60. doi:10.1145/3520084.3520093
- [33] Perkusich M, Chaves e Silva L, Costa A, Ramos F, Saraiva R, Freire A, et al. Intelligent software engineering in the context of Agile software development: A systematic literature review. *Information and Software Technology*. 2020;119:106241. doi:10.1016/j.infsof.2019.106241
- [34] Fitriani WR, Rahayu P, Sensuse DI. Challenges in Agile software development: A systematic literature review. *2016 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*. 2016. pp. 155-164. doi:10.1109/ICACSIS.2016.7872736
- [35] Satria D, Sensuse DI, Noprison H. A systematic literature review of the improved Agile software development. *2017 International Conference on Information Technology Systems and Innovation (ICITSI)*. 2017. pp. 94-99. doi:10.1109/ICITSI.2017.8267925
- [36] Vallon R, Da Silva Estácio BJ, Prikladnicki R, Grechenig T. Systematic literature review on Agile practices in global software development. *Information and Software Technology*. 2018;96:161-180. doi:10.1016/j.infsof.2017.12.004
- [37] Pham MT, Rajić A, Greig JD, Sargeant JM, Papadopoulos A, McEwen SA. A scoping review of scoping reviews: Advancing the approach and enhancing the consistency. *Research Synthesis Methods*. 2014;5(4):371-385. doi:10.1002/jrsm.1123
- [38] Peters MDJ, Godfrey C, McInerney P, Khalil H, Larsen P, Marnie C, et al. Best practice guidance and reporting items for the development of scoping review protocols. *JBI Evidence Synthesis*. 2022;20(4):953-968. doi:10.11124/JBIES-21-00242
- [39] Peters MDJ, Marnie C, Tricco AC, Pollock D, Munn Z, Alexander L, et al. Updated methodological guidance for the conduct of scoping reviews. *JBI Evidence Synthesis*. 2020;18(10):2119-2126.
- [40] Mohammed IA. Critical analysis on the impact of software engineering in the technological industry. *International Journal of Research and Analytical Reviews*. 2020;7(4):182-186.
- [41] Meso P, Jain R. Agile software development: Adaptive systems principles and best practices. *Information Systems Management*. 2006;23(3):19-30. doi:10.1201/1078.10580530/46108.23.3.20060601/93704.3
- [42] Gunasekaran A, Yusuf YY. Agile manufacturing: A taxonomy of strategic and technological imperatives. *International Journal of Production Research*. 2002;40(6):1357-1385. doi:10.1080/00207540110118370
- [43] Kumar G, Bhatia PK. Impact of Agile methodology on software development process. *International Journal of Computer Technology and Electronics Engineering (IJCTEE)*. 2012;2(4):46-50.
- [44] Mishra A, Alzoubi YI. Structured software development versus Agile software development: A comparative analysis. *International Journal of System Assurance Engineering and Management*. 2023;14(4):1504-1522. doi:10.1007/s13198-023-01958-5
- [45] De Vicente Mohino J, Bermejo Higuera J, Bermejo Higuera JR, Sicilia Montalvo JA. The application of a new secure software development life cycle (S-SDLC) with Agile methodologies. *Electronics*. 2019;8(11):1218. doi:10.3390/electronics8111218
- [46] Azanha A, Argoud ARTT, Camargo JB de, Antonioli PD. Agile project management with Scrum. *International Journal of Managing Projects in Business*. 2017;10(1):121-142. doi:10.1108/IJMPB-06-2016-0054
- [47] Tavares BG, Da Silva CES, De Souza AD. Risk management analysis in Scrum software projects. *International Transactions in Operational Research*. 2019;26(5):1884-1905. doi:10.1111/itor.12401
- [48] Saeeda H, Ahmad MO, Gustavsson T. Challenges in large-scale Agile software development projects. *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing*. 2023. pp. 1030-1037. doi:10.1145/3555776.3577662
- [49] Natarajan T, Pichai S. Behaviour-driven development and metrics framework for enhanced Agile practices in scrum teams. *Information and Software Technology*. 2024;170:107435. doi:10.1016/j.infsof.2024.107435
- [50] Beek K, Wagenaar G, Kester L, Overbeek S, De Rooij E. Measuring team effectiveness in Scrum. *Quality of Information and Communications Technology. QUATIC 2023*. 2023. pp. 233-247. doi:10.1007/978-3-031-43703-8_17
- [51] Alshammari FH. Cost estimate in scrum project with the decision-based effort estimation technique. *Soft Computing*. 2022;26(20):10993-11005. doi:10.1007/s00500-022-07352-w

- [52] Haj Hamad RM, Al Fayoumi M. Scalable Agile transformation process (SATP) to convert waterfall project management office into Agile project management office. *2018 International Arab Conference on Information Technology (ACIT)*. 2018. pp. 1-8. doi:10.1109/ACIT.2018.8672701
- [53] Santos P de O, De Carvalho MM. Exploring the challenges and benefits for scaling Agile project management to large projects: A review. *Requirements Engineering*. 2022;27(1):117-134. doi:10.1007/s00766-021-00363-3
- [54] Behutiye W, Seppänen P, Rodríguez P, Oivo M. Documentation of quality requirements in Agile software development. *Proceedings of the 24th International Conference on Evaluation and Assessment in Software Engineering*. 2020. pp. 250-259. doi:10.1145/3383219.3383245
- [55] Rajagopal P. *Agile marketing strategies*. Springer; 2022.
- [56] Šmite D, Moe NB, Gonzalez-Huerta J. Overcoming cultural barriers to being agile in distributed teams. *Information and Software Technology*. 2021;138:106612. doi:10.1016/j.infsof.2021.106612
- [57] Dikert K, Paasivaara M, Lassenius C. Challenges and success factors for large-scale Agile transformations: A systematic literature review. *Journal of Systems and Software*. 2016;119:87-108. doi:10.1016/j.jss.2016.06.013
- [58] Coram M, Bohner S. The impact of Agile methods on software project management. *12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'05)*. pp. 363-370. doi:10.1109/ECBS.2005.68
- [59] El-Najar T, Ahmad I, Alkandari M. Easycomm: A framework and tool to solve client communication problem in Agile development. *IAENG International Journal of Computer Science*. 2019;46(1).
- [60] Barraood SO, Mohd H, Baharom F. An initial investigation of the effect of quality factors on Agile test case quality through experts' review. *Cogent Engineering*. 2022;9(1). doi:10.1080/23311916.2022.2082121
- [61] Maier NM, Emmerich P. Agile or traditional project organization: A quantitative assessment of decision criteria among firms in the DACH region. *Central European Business Review*. 2022;11(5):67-83. doi:10.18267/j.cebr.308
- [62] Schramm VB, Damasceno AC, Schramm F. Supporting the choice of the best-fit Agile model using Fitradeoff. *Pesquisa Operacional*. 2023;43(spe1). doi:10.1590/0101-7438.2023.043spe1.00264750
- [63] Radnejad M, Gandomani TJ. Structure and criteria defining business value in Agile software development based on hierarchical analysis. *2022 Second International Conference on Distributed Computing and High Performance Computing (DCHPC)*. 2022. pp. 117-122. doi:10.1109/DCHPC55044.2022.9732113
- [64] Humpert L, Mundt E, Bretz L, Anacker H. Analysis of the criteria for comparing systems engineering and Agile methods. *2022 IEEE International Symposium on Systems Engineering (ISSE)*. 2022. pp. 1-6. doi:10.1109/ISSE54508.2022.10005556
- [65] Raharjo T, Purwandari B, Budiardjo EK, Yuniarti R. The essence of software engineering framework-based model for an Agile software development method. *International Journal of Advanced Computer Science and Applications*. 2023;14(7). pp 802-811. doi:10.14569/IJACSA.2023.0140788
- [66] Figalist I, Elsner C, Bosch J, Olsson HH. Scaling Agile beyond organizational boundaries: Coordination challenges in software ecosystems. *Agile Processes in Software Engineering and Extreme Programming: 20th International Conference, XP 2019*, 2019. pp. 189-206.
- [67] Rolland K-H, Conboy K. Operationalizing Agile methods: Examining coherence in large-scale Agile transformations. *Agile Processes in Software Engineering and Extreme Programming-Workshops. XP 2020*, 2020. pp. 75-83.
- [68] Martini A, Stray V, Moe NB. Technical-, social- and process debt in large-scale Agile: An exploratory case-study. *Agile Processes in Software Engineering and Extreme Programming - Workshops. XP 2019*. 2019. pp. 112-119. doi:10.1007/978-3-030-30126-2_14