



Structured software development versus agile software development: a comparative analysis

Alok Mishra¹ · Yehia Ibrahim Alzoubi²

Received: 2 January 2023 / Revised: 28 April 2023 / Accepted: 24 May 2023
© The Author(s) 2023

Abstract The use of agile methodology has become widespread in organizations that previously relied on traditional or structured software development methods, such as the waterfall approach. Successful completion rates for agile initiatives are at 40%, whereas only 15% of waterfall projects were successfully completed. To take advantage of the benefits of each method, many organizations are using a hybrid approach that combines Agile with Waterfall and structured software development methodologies. Rather than asking which methodology is superior, it is more productive to determine which one is most effective for a particular project. Agile projects have a failure rate of 10%, while waterfall projects fail 30% of the time. This study aims to compare and contrast agile and waterfall methodologies in order to create a decision tree for selecting the most suitable method for a software development project. Several cases and scenarios are examined using the hybrid development approach. Despite many advantages of Agile methodologies, a combination approach may be necessary, as Waterfall may be required for certain projects or development phases. This study explores the feasibility of combining Waterfall and Agile methodologies in software development management.

Keywords Agile methods · Waterfall · Structured software development · Comparative, analysis · Decision tree

✉ Alok Mishra
alok.mishra@ntnu.no

Yehia Ibrahim Alzoubi
yehia.alzoubi@aum.edu.kw

¹ Faculty of Engineering, Norwegian University of Science and Technology (NTNU), Gjøvik, Norway

² College of Business Administration, American University of the Middle East, Egaila 15453, Kuwait

1 Introduction

The two most common methodologies of software development to date are agile and waterfall. According to surveys, 71% of businesses prefer agile, whereas 51% of businesses utilize waterfall (Serebryantseva 2022). When developing a software project, specific practices and standards are described as being used in a software development methodology. There are many different methodologies available to developers, each with its own set of quirks. But there isn't a standard methodology that fits all projects, so it's crucial to pick the method that will benefit the project the most. The overall development process is more effective and seamless when a set methodology is followed since everyone on the team is aware of their roles and due dates. Additionally, it makes it easier for the team leader to move through the project and divide the work efficiently to avoid disruption and delays. A poor software development methodology, however, might seriously impede the project's progress or even cause it to fail. The list of potential issues includes anything from missed deadlines and subpar goods to high-risk misunderstandings with the customers and within the team (Mishra et al. 2021).

Understanding the flow of several development methodologies is crucial, especially agile and waterfall as the two most used methodologies. Due to its organized and structured phases, the waterfall methodology is still commonly employed. When the requirements are fixed, functioning software is one of the advantages of waterfall (Kuusinen et al. 2016). Waterfall, however, suffers from several drawbacks, including an inability to react to requirements changes and a lack of client collaboration. Agile methodology, on the other hand, is exceptional since they place a strong emphasis on clients' interaction and

treat them like team members. Also, in agile development, the development process may be made faster by timely addressing the customer's feedback (Almeida and Simões 2019).

The software firms realized that certain project aspects, for instance, purpose, requirements, resources, architecture, and scale will determine which methodology best suits them—either agile or waterfall or even a hybrid of the two (Bishop and Deokar 2014). Anecdotal evidence and instances of overcoming hardship by well-known specialists over the past few years show that using suitable methodology is sometimes more effective and sensible (Chandran and Das Aundhe 2022; Fagarasan et al. 2021). Important issues are evolving, and so are people and their perspectives (Bishop and Deokar 2014). Although waterfall development is still necessary, utilizing the agile methodology often leads to higher progress. Accordingly, in some circumstances, like a large project with scattered teams, combining both methodologies may be the best choice (Alam et al. 2022).

Consequently, employing a combination structure is the best course of action. This highlights the reality that the issue is frequently not just the variety of methodologies that are now available but also the proper use of the existing methodology (Thesing et al. 2021). In light of this, the novelty of this study lies in its examination of this viewpoint to present an insightful perspective on when agile is more suitable than waterfall, and vice versa. There has been a scant comparative study published despite an ongoing argument about whether the method is more successful (Włodarski et al. 2020). It also evaluates whether software development companies should adopt a hybrid approach that incorporates both agile and waterfall methodologies. This study will particularly answer the following questions:

When is the agile methodology preferable over the waterfall methodology, and vice versa?

When is it best to combine the waterfall and agile methodologies?

By addressing the aforementioned research questions and completing a comprehensive analysis of both waterfall and agile approaches, the findings of this study contribute to the corpus of research as follows.

- In light of this, this paper discusses a broad overview of the traits of both waterfall and agile methodologies. Data were gathered from currently available literature (e.g., conferences, journals, white papers, blogs, and professional and industrial websites like “Standish group report” (The-Standish-Group 2020), “Pixel Plex” (Serebryantseva 2022), and “Vitality Chicago” (Mersino 2022)). This was essential to investigate the two methodologies as well as the different forms of combination frameworks applied in real-world contexts.

- This paper suggested a decision tree that incorporates several scenarios of whether using agile or waterfall approaches is the most effective.
- The paper detail situations in which businesses would need to use hybrid methodologies—combining agile and waterfall—for certain projects. If a hybrid methodology is taken into consideration, the model will offer trustworthy insight into which components of the different strategies should be blended or which model is the best option.

The remaining sections of the paper are organized as follows. Section 2 discusses the research background and related work. The properties of agile and waterfall are covered in Sect. 3. Section 4 compares agile and waterfall methodologies. The possibilities of a hybrid methodology of agile and waterfall are covered in Sect. 5. Section 6 discusses the findings and limitations of this paper. Section 7 concludes this paper and indicates future research directions.

2 Background and related work

Software development methodologies have been through several modifications; agile and waterfall methodologies are continually improving to fit new patterns and, most importantly, maintain firms' competitiveness. Today's software firms are looking for methods that succeed, which has led to the development of several combination frameworks. The literature that compares agile and waterfall methodologies or advocates merging the two methodologies in the development life cycle is scanned in this section. Despite the enormous number of studies that have been reported for this aim, this section reviews a few of them. Because there have been many studies published on this topic, we examined the literature at a high level in this section. Even though other studies have compared the agile and waterfall approaches, this article stands out because it presents the most recent research on the subject and offers a decision tree that software project organizations may use to determine when and how the two methodologies may be merged. Moreover, for businesses to consistently execute successful projects, this article makes the point and provides evidence for the need for some form of a hybrid methodology that combines agile and waterfall methodologies. According to our investigation of the literature, three main categories were identified in the context of agile and waterfall methodologies: comparison between the two methodologies, the transition from waterfall to agile, and the hybrid combination between the two methodologies. This section briefly investigates the literature about each category.

2.1 Comparisons between agile and waterfall methodologies

Agile methodology and communication that improve software development have been the subject of several studies (e.g., Alzoubi et al. 2016; Friess 2019; Ammad et al. 2019; Alzoubi and Gill 2021, 2020; Shameem et al. 2020; Hummel et al. 2015). Numerous studies (e.g., Kuusinen et al. 2016; Friess 2019; Al-Saqqa et al. 2020; Dingsøy et al. 2018; Alzoubi et al. 2015) have also examined other elements of agile development, such as customer interaction, validation, verification, cost reduction, incentives, and increased performance. All of these studies found that adopting agile concepts, such as customer involvement, rapid iteration development, and informal communication, boosted project success. Many previous studies have compared waterfall and agile methodologies and asserted that agile increased the likelihood that the project would be successful, monitored all processes during each iteration to provide the client with a complete understanding of the software product, and could rapidly repair defects. Table 1 summarizes a few of these studies.

According to Kumar and Bhatia (2014), agile methodology is responsive to changes, has less paperwork, and has rapid development, which may lower the cost and time of development. However, the waterfall is a reusable component that may be used to shorten the project's duration and provide it with greater order (Kumar and Bhatia 2014). Furthermore, Yadav et al. (2015) stated that using agile methodology may boost a project's success rate since customers are more involved and issues may be fixed more quickly. The waterfall, however, lengthens the development process (Yadav et al. 2015). According to the results of a study done by Khoza and Marnewick (2020), agile methodologies are 60% more effective in process development than waterfall, however, there is no distinction between projects that use waterfall or agile methodologies in terms of the success of the business. The findings showed that success in project management is higher for projects that employ a waterfall methodology (Khoza and Marnewick 2020). Regarding the deliverable's success, there is a distinct difference. Although this was not true for all initiatives, projects that used agile methodologies were more successful than waterfall projects (Khoza and Marnewick 2020).

Based on an analysis of three projects that used the waterfall methodology and three projects that used the agile methodology, Thummadi and Lyytinen (2020) looked at how much the two methodologies cause diversity in software design practices. The analysis revealed that, in both situations, a method-induced variance using the waterfall and agile methodologies accounts for around 40% of all

activities, with the other 60% being justified by an individual designer's habits, the project's circumstances, and ambient noise. The impact of the methodology on software design activities is typically less significant than anticipated, and the impact of the designers and project's circumstances on process and outcome must therefore not be minimized (Thummadi and Lyytinen 2020).

More recently, Alam et al. (2022) examined the relationship between the size of the project and the best methodology (i.e., agile or waterfall) since they believed that the methodology chosen is mostly responsible for software quality. The six-pointed star model's factors were used to conduct the comparison. The findings indicated that for small projects, practically all methodological criteria are in favor of agile methodologies. Both methodologies are largely similar for medium-sized projects (Chandran and Das Aundhe 2022). The waterfall is significantly more suitable for all factors of the six-pointed star model, according to the results for large projects. Since agile is a paradigm that prioritizes the needs of the client, the respondents favor it (Alam et al. 2022; Dursun and Goker 2022). Thesing et al. (2021) conducted a review of the literature to determine when agile or waterfall methodologies were better suited for various circumstances. They then verified their responses through interviews with subject-matter specialists. Their findings indicated that which approach is best depends on several criteria, including time, cost, scope, organizational structure, and team qualities.

In order to evaluate the effects of a certain development strategy on the achievement of student computer endeavors, Wodarski et al. (2020) described an experiment that was conducted at two engineering institutions in Europe. In order to identify trends related to the use of waterfall, agile, and no-process methods on the aspects of quality of product and teamwork and team productivity, a systematic, metric-based evaluation scheme was used. The results showed that, in comparison to the waterfall team, an agile strategy resulted in stronger team cohesiveness and superior production. However, there was a decline in functional sufficiency. Similar to this, agile has lower quality metrics scores than waterfall. It appears that agile teams prioritize delivering a demonstrable piece of software on schedule while somehow ignoring its quality (Włodarski et al. 2020).

Sinha and Das (2021) investigated the implementation of the software quality assurance or testing phase and how it will function with the agile approaches. The results showed that while the waterfall technique is rigid and ineffective at communicating, it offers greater structure and quality assurance. The agile approach, on the other hand, is particularly effective if the sprints are completely dedicated to bringing about new functional enhancements as opposed to resolving quality problems early in the SDLC process. Additionally, using agile methodologies, serious performance flaws aren't

Table 1 Summary of the recent waterfall versus agile literature comparison

Study	Methodology	Pros	Cons	Project domain	Project characteristics	Development patterns
Kumar and Bhatia (2014)	Agile and traditional methodologies comparison	<ul style="list-style-type: none"> Agile is a light-weight documentation Agile is cost reduction, flexible, and rapid The waterfall is a reusable component that may be utilized to reduce time 	<ul style="list-style-type: none"> The waterfall is less adaptable to change and less effective at handling rapid change There must be a sizable number of developers for the waterfall 	<ul style="list-style-type: none"> The waterfall is better suited for mission-critical and governmental projects Agile is better suited to large projects 	<ul style="list-style-type: none"> Management should choose the methodology that is most appropriate for the project 	<ul style="list-style-type: none"> Agile reduces cost and time of development, and is flexible to changes Waterfall provides more organization in predefined time and cost projects
Yadav et al. (2015)	Agile and traditional methodologies comparison	<ul style="list-style-type: none"> Agile increases the success rate, customers are more involved, and faster to fix bugs The waterfall results in longer development time 	<ul style="list-style-type: none"> Lack of documentation and decreased management control in agile 	<ul style="list-style-type: none"> Industrial organizational projects 	<ul style="list-style-type: none"> Continuous delivery and fixing bugs in agile 	<ul style="list-style-type: none"> Customer involvement in agile Documentation in waterfall
Khoza and Marnewick (2020)	Agile and traditional methodologies comparison	<ul style="list-style-type: none"> Agile is 60% more effective in process development than waterfall 	<ul style="list-style-type: none"> No difference between a waterfall or agile projects in terms of the success of the business 	<ul style="list-style-type: none"> Success in project management is higher for projects that employ a waterfall 	<ul style="list-style-type: none"> Projects that used agile were more successful than waterfall projects 	<ul style="list-style-type: none"> Agile practices enhance the development
Thummadi and Lyytinen (2020)	Agile and traditional methodologies comparison	<ul style="list-style-type: none"> The waterfall process was less repetitive than the agile methodology The application and technical elements were more strongly influenced by both agile and waterfall 	<ul style="list-style-type: none"> With regard to iterations, waterfall and agile methodologies closely resemble one another and show how particular requirements are handled locally 	<ul style="list-style-type: none"> Three projects in a big IT organization 	<ul style="list-style-type: none"> About 40% of all activities are affected by method-induced variation utilizing the waterfall and agile 	<ul style="list-style-type: none"> 60% of the difference can be accounted for by a designer's habits, the project's fitness requirements, and ambient noise
Włodarski et al. (2020)	Agile and traditional methodologies comparison	<ul style="list-style-type: none"> Agile development is stronger team cohesiveness and superior production 	<ul style="list-style-type: none"> Agile functional sufficiency and lower quality metrics scores than waterfall 	<ul style="list-style-type: none"> Engineering students' developers 	<ul style="list-style-type: none"> Two engineering institutions in Europe 	<ul style="list-style-type: none"> One group of students used agile and the other used waterfall
Sinha and Das (2021)	Agile and traditional methodologies comparison	<ul style="list-style-type: none"> Waterfall offers greater structure and quality assurance 	<ul style="list-style-type: none"> Waterfall technique is rigid and ineffective at communicating Agile has more performance flaws 	<ul style="list-style-type: none"> General 	<ul style="list-style-type: none"> General 	<ul style="list-style-type: none"> Software quality assurance

Table 1 (continued)

Study	Methodology	Pros	Cons	Project domain	Project characteristics	Development patterns
Thesing et al. (2021)	Agile and traditional methodologies comparison	<ul style="list-style-type: none"> Waterfall has comprehensive planning in advance, a clear scope, stability, and a long-term outlook Planning for agile projects is gradual, adaptable, and short-term 	<ul style="list-style-type: none"> Initial requirements in a waterfall are ambiguous and vague Agile might not be appropriate for the corporate culture where the structure is hierarchical 	<ul style="list-style-type: none"> Agile is not appropriate if the project is vital, cannot be broken down, or is a step-by-step process 	<ul style="list-style-type: none"> The best option is waterfall if the leadership rejects the agile principle or the firm cannot handle frequent deliveries 	<ul style="list-style-type: none"> Stability, systematic planning, and documented results in a waterfall Rapid recognition of altered requirements in agile
Alam et al. (2022)	Agile and traditional methodologies comparison	<ul style="list-style-type: none"> Flexibility in agile and well-organization in waterfall 	<ul style="list-style-type: none"> Both methodologies are similar for medium-size projects 	<ul style="list-style-type: none"> Agile suits more small projects Waterfall suits more large projects 	<ul style="list-style-type: none"> Better understanding of client requirements in agile 	<ul style="list-style-type: none"> Methodology chosen is mostly responsible for software quality
Raza (2023), Hamilton (2023)	Agile and traditional methodologies comparison	<ul style="list-style-type: none"> Agile enhances speed and team coherence 	<ul style="list-style-type: none"> Waterfall results in more quality and higher performance projects 	<ul style="list-style-type: none"> General domains 	<ul style="list-style-type: none"> General characteristics 	<ul style="list-style-type: none"> DevOps SDLC methodologies are adopted among several firms

found until a late stage of project development (Sinha and Das 2021).

Raza (2023) outlined the differences between the agile and waterfall methodologies from an industry perspective. Waterfall projects have a sequential structure, a high emphasis on documentation, and little interaction between engineers and clients. Agile, on the other side, emphasizes better collaboration between developers and consumers through reduced paperwork. Although agile methodologies are popular among software companies, they lack a consistent system architecture, an agile mentality isn't completely adopted throughout the business, and they are of poor quality (Hamilton 2023).

2.2 Transition from waterfall to agile

Many previous studies have reported the transition from waterfall to agile. Table 2 summarizes a few of these studies. Kuusinen et al. (2016) developed a mixed approach centered on performing decomposition tasks or functions in a waterfall while adhering to agile concepts. For agile to fit a traditional waterfall company, Richter et al. (2016) explored applying agile as a semi-structured framework that applies bottom-up change. Dima and Maassen (2018) used the Delphi technique with surveys and interviews to identify the most common software development methods in the IT industry and their effects on management. The findings showed a trend in favor of the agile software development methodology as it quickly meets changing consumer demands (Alzoubi and Gill 2022). According to this study, software development methods vary as technology dynamics do at the business level (Dima and Maassen 2018). Even though the agile methodology has several advantages, including regular customer contact, teamwork, and greater assessments of the advancement achieved toward goals, a major problem is the pressure on team members to handle growing pressure from the outside (Dima and Maassen 2018; Mishra et al. 2018).

In their paper, Kasauli et al. (2020) summarized the findings of an exploratory case study that concentrated on two divisions of a sizable systems engineering firm that was adopting agile. They described the difficulties that teams encountered while transitioning to agile, and they suggested potential solutions to the problems. The problems were due to the lack of developers' awareness of the changes of requirements, managing these requirements, and the lack of basic agile knowledge like user stories (Kasauli et al. 2020). Fagarasan et al. (2021) and Almeida and Simões (2019) assessed the agile methodology capabilities inside waterfall software. This assessment took into account variables including predictability, practicability, and intricacy. The findings demonstrated that Scrum is the most approved agile methodology since it can handle complicated issues

Table 2 Summary of the recent literature on transition from waterfall to agile

Study	Methodology	Pros	Cons	Project domain	Project characteristics	Development patterns
Soundararajan and Arthur (2009)	Transition to agile methodology	<ul style="list-style-type: none"> Agile increased developers and customer satisfaction, the team received higher priority, the focus is on core requirements, and it is flexible to changes Effective and consistent communication increases productivity and top-level management support 	<ul style="list-style-type: none"> Functions should be listed and prioritized 	<ul style="list-style-type: none"> Short-delivery, small, and multiple projects 	<ul style="list-style-type: none"> Early delivery Feedback and integration are continuous 	<ul style="list-style-type: none"> Customer satisfaction increased using agile Core requirements are better handled using agile
Richter et al. (2016)	Transition to agile methodology	<ul style="list-style-type: none"> Agile is more client-oriented Waterfall is less time-consuming and more organized compared to agile Agile is more client involvement Flexibility to requirements changes 	<ul style="list-style-type: none"> The Issues in understanding agile principles Challenging expectations in various agile projects Low consumer expectations 	<ul style="list-style-type: none"> Structured projects that demand quick change are problematic Extreme Programming (XP) projects showed a high failure rate 	<ul style="list-style-type: none"> Agile practices Communication effectiveness 	<ul style="list-style-type: none"> Communication effectiveness and flexibility increased using agile
Dima and Maassen (2018)	Transition to agile methodology	<ul style="list-style-type: none"> Agile is more client-oriented Waterfall is less time-consuming and more organized compared to agile Agile is more client involvement Flexibility to requirements changes 	<ul style="list-style-type: none"> Agile creates more pressure on the developers Conflicts and employee demotivation as a result of using agile Lack of basic knowledge about agile 	<ul style="list-style-type: none"> Waterfall projects used some of the agile concepts like frequent client meetings 	<ul style="list-style-type: none"> Better understanding of client requirements 	<ul style="list-style-type: none"> Flexible deadlines and/or deliveries when obstacles happened
Kasauli et al. (2020)	Transition to agile methodology	<ul style="list-style-type: none"> Agile can enhance the product performance Agile suits all project sizes Agile is essential to the successful completion of software projects Agile is more flexible and client-oriented 	<ul style="list-style-type: none"> Waterfall suits more small and medium projects Agile was adopted by 16% of the respondents Agile approaches lack the quality aspects 	<ul style="list-style-type: none"> Automotive projects 	<ul style="list-style-type: none"> Transition to agile may face many challenges due to the lack of awareness of agile concepts 	<ul style="list-style-type: none"> Training developers
Almeida and Simões (2019), Fagarasan et al. (2021)	Transition to agile methodology	<ul style="list-style-type: none"> Agile is more client-oriented Waterfall is less time-consuming and more organized compared to agile Agile is more client involvement Flexibility to requirements changes 	<ul style="list-style-type: none"> Waterfall suits more small and medium projects Agile was adopted by 16% of the respondents Agile approaches lack the quality aspects 	<ul style="list-style-type: none"> Waterfall is a well-proved methodology 	<ul style="list-style-type: none"> In hybrid methodology, the waterfall is used in the planning phase, while agile is used for iterative delivery 	<ul style="list-style-type: none"> Project can be delivered as parts instead of one delivery at the end of the waterfall project
Wafa et al. (2022)	Transition to agile methodology	<ul style="list-style-type: none"> Agile is more flexible and client-oriented 	<ul style="list-style-type: none"> Agile was adopted by 16% of the respondents Agile approaches lack the quality aspects 	<ul style="list-style-type: none"> Different projects' domains 	<ul style="list-style-type: none"> Different levels of individual experience, geographic location, and employment in Pakistan's IT sector 	<ul style="list-style-type: none"> Quality, client satisfaction, cost, and time depend on agile methods
Saarikallio and Tyrväinen (2022)	Transition to agile methodology	<ul style="list-style-type: none"> Agile is more flexible and client-oriented 	<ul style="list-style-type: none"> Agile approaches lack the quality aspects 	<ul style="list-style-type: none"> Large-scaled software development organization 	<ul style="list-style-type: none"> Software development teams are isolated from other administrative teams 	<ul style="list-style-type: none"> Formal quality aspects and quality-based culture

Table 2 (continued)

Study	Methodology	Pros	Cons	Project domain	Project characteristics	Development patterns
KnowledgeHut (2023), ProductPlan (2023)	Transition to agile methodology	<ul style="list-style-type: none"> Agile is more responsive and flexible 	<ul style="list-style-type: none"> Failure in transformation due to improper application of an agile mindset and a lack of management support 	<ul style="list-style-type: none"> Large-scaled software development organization 	<ul style="list-style-type: none"> Large software firms, in general 	<ul style="list-style-type: none"> Awareness of the agile approach and not hurrying in the transformation process

and quickly adjust to changing market conditions (Fagarasan et al. 2021).

The total effect of agile methods on the software development, developers, and management of project dimensions was examined by Wafa et al. (2022). To investigate the effect of agile methodologies on Pakistan’s IT business, they polled several software engineering experts from various backgrounds. The same data were then quantitatively analyzed. The results imply that applying Agile methods and efficient software development are strongly correlated. The investigation demonstrates that the achievement of projects for software development in terms of quality, client satisfaction, cost, and time depends critically on agile methods (Wafa et al. 2022).

To respond to the inquiry, "What effect does increasing aspects of quality have?", (Saarikallio and Tyrväinen 2022) studied this inquiry by putting into practice treatments based on earlier research and information obtained from semi-structured interviews, and the concerns were addressed. The setting was an established company with a complicated structure of revenue streams, so the combination of the project, service, and product-based work made the adoption of agile methods problematic. As evaluated by reported flaws, interventions significantly improved quality. In order to move the software industry forward, it is therefore argued that agile methodologies are insufficient on their own without a quality-focused environment (Saarikallio and Tyrväinen 2022).

Agile transformations, according to KnowledgeHut (2023) company, may contribute to all agile advantages when done well including responsiveness and flexibility into an organization’s basic foundation, and its effects are seen at every level of the company. However, as the numbers demonstrate, firms frequently embark on an aggressive full-scale change only to realize midway through that it is not succeeding (ProductPlan 2023). Agile transformation failures can occur for several reasons, including the improper application of prominent agile techniques, a lack of management support, a lack of awareness of the approach that is best for your firm, and hurrying the transformation process (KnowledgeHut 2023).

2.3 Hybrid methodology combining agile and waterfall methodologies

The third category of the literature identified focused on the hybrid combination development of agile and waterfall in order to enhance the development process of the traditional methodologies. Table 3 summarizes a few of these studies. Käpyaho and Kauppinen (2015) used a mixed approach and presented a framework to tackle waterfall’s prototype documentation concerns and improve communication by deploying agile concepts in the development process. According to

Table 3 Summary of the recent literature on combining waterfall and agile

Study	Methodology	Pros	Cons	Project domain	Project characteristics	Development patterns
Käpyaho and Kauppinen (2015)	Waterfall prototyping and agile	<ul style="list-style-type: none"> Agile is flexible and faster time-to-market, increases business value and team autonomy, more effective communication, and multi-skilled teams can emerge 	<ul style="list-style-type: none"> Top management support to shift to scrum Dealing with feedback-related obstacles 	<ul style="list-style-type: none"> Large projects Applications with high severity in healthcare 	<ul style="list-style-type: none"> Agile practices Communication 	<ul style="list-style-type: none"> Communication effectiveness and flexibility are increased using agile
Stoica et al. (2013)	Combination Of agile and waterfall	<ul style="list-style-type: none"> Increase the success rate, delivery on time and within budget Accept ongoing adaptation for less cost Increased customer involvement 	<ul style="list-style-type: none"> Hand-over projects issues The project team might not adhere to agile properly 	<ul style="list-style-type: none"> Complex commercial contexts Architectural applications 	<ul style="list-style-type: none"> Teamwork, self-organizing, and ongoing integration When new team members join, waterfall documentation and structure are advantageous 	<ul style="list-style-type: none"> A waterfall method is a good option for small projects
Kuusinen et al. (2016)	Combination of agile and waterfall	<ul style="list-style-type: none"> Applying bottom-up management in agile Agile enables decomposition tasks implementation Combination can enhance productivity 	<ul style="list-style-type: none"> Firm incentive programs are inappropriate for agile The project team might not adhere to the agile mindset properly Issues in understanding agile principles by management and development team 	<ul style="list-style-type: none"> Scrum of Scrums: large-scale distribution projects R-Scrum: projects with rapid change in large enterprises, regulated settings, and particular organizational settings 	<ul style="list-style-type: none"> Agile practices Waterfall organization 	<ul style="list-style-type: none"> Management effectiveness and rapid development for agile
Signoretta et al. (2020)	Combination of agile and waterfall	<ul style="list-style-type: none"> Team autonomy, communication, trust, and team involvement were enhanced 	<ul style="list-style-type: none"> Business policy, budget allocation, and team autonomy at stake represent challenges 	<ul style="list-style-type: none"> A sizable software development company with scattered offices in three different nations and regions 	<ul style="list-style-type: none"> Two software firms migrating to a hybrid method 	<ul style="list-style-type: none"> Product focus rather than project focus as a remedy
Kirpitsas and Pachidis (2022)	Combination of agile and waterfall	<ul style="list-style-type: none"> Emphasizes control, deadlines, and short project cycles 	<ul style="list-style-type: none"> Lacks the precise instructions that teams need 	<ul style="list-style-type: none"> General projects 	<ul style="list-style-type: none"> Certain project management needs 	<ul style="list-style-type: none"> Faster and more adaptable solutions
DevFacto (2023), ProjectManager (2023)	Combination of agile and waterfall	<ul style="list-style-type: none"> Structure and consistency of the waterfall and versatility and adaptability of agile 	<ul style="list-style-type: none"> Cautious project management of deadlines and schedules 	<ul style="list-style-type: none"> General domains 	<ul style="list-style-type: none"> General characteristics 	<ul style="list-style-type: none"> Disagreement amongst members who are accustomed to working in various methods

Soundararajan and Arthur (2009), Kumar and Bhatia (2014), and Stoica et al. (2013), combining agile and waterfall might be advantageous while concentrating on the needs of each methodology.

Richter et al. (2016) discussed a framework that supports a dual application model (combining the finest elements of traditional and agile methodologies). Following the agile methodology, the combination framework is more helpful since customer participation is increased (Nerur et al. 2005). The fundamental goals of utilizing agile methodologies and how they may be effective throughout SDLC were discussed in Soundararajan and Arthur (2009), Stoica et al. (2013), Srivastava et al. (2020). They addressed how agile methodologies improve software quality, self-management, and efficient communication between team members and stakeholders (Soundararajan and Arthur 2009; Stoica et al. 2013).

Signoretti et al. (2020) conducted a case study involving two software development organizations from a major business that are migrating to the hybrid approach in order to determine the failure and success aspects of a hybrid approach deployment. Focus groups and semi-structured interviews were employed to gather information. The results revealed a number of success elements, including team autonomy, communication, trust, and team involvement. The failure reasons, on the other hand, include a business policy, budget allocation, and team autonomy at stake. The authors also discussed potential remedies for the issue of "product focus rather than project focus," which teams perceived as being of the highest significance for the change (Signoretti et al. 2020).

While companies are currently under continual pressure to create software more quickly owing to new demands arising around the world, this ongoing endeavour results in unique development methods that are ostensibly motivated by practice (Kirpitsas and Pachidis 2022). Modern approaches are neither fully linear nor fully agile, so choosing the right mix of strategies to achieve objectives and ensure value generation for enterprises presents a problem. Technology companies still need to ensure value generation and conformity to industry norms, directives, and laws. Similarly, it is anticipated that IT auditing will continue to evolve in line with changes in company strategies, sociocultural trends, and the requirement for IT governance in the coming years (Kirpitsas and Pachidis 2022).

In recent years, the usage of hybrid models, which include aspects of both waterfall and agile methods, has grown in favour since it enables businesses to benefit from the advantages of both methods (DevFacto 2023). The hybrid method is largely supported by the industry as a way to enhance their development practices. The

hybrid model combines the structure and consistency of the waterfall technique with increased versatility and adaptability to change. The hybrid approach does raise certain issues, though, such as the necessity for cautious project management of deadlines and schedules, as well as the possibility for confusion and disagreement amongst members of the team who are accustomed to working in various methods (ProjectManager 2023).

3 Agile and waterfall developments mechanisms

This section discusses an overall view of the characteristics of the waterfall and agile methodologies. Different agile framework concepts are introduced here including Extreme Programming (XP), Scrum, and Kanban to underline the characteristics of agile frameworks.

3.1 Agile development

When changes to the software project are needed, the waterfall methodology's drawbacks and rigidity become apparent. Several software development experts convened and decided to introduce a new methodology that can handle these changes. After the agile alliance group gathered in 2001, the term "agile" first appeared (Poppendieck and Cusumano 2012). According to Kuusinen et al. (2016), agile methodologies place a greater emphasis on people, working software, collaboration, customer cooperation, and change than they do on waterfall sequential, procedures, contracts, extensive documentation, and plans. Since then, agile methodology has grown in popularity throughout the business (Käpyaho and Kauppinen 2015). Agile development frameworks and phases are discussed in the following sections.

3.1.1 Agile development frameworks

Several frameworks are available for the agile methodology, including Scrum, Kanban, XP, Behavior-driven development, Feature-driven software development, Dynamic Systems Delivery, and Crystal, among others (Gustavsson et al. 2022). We won't go into detail about all of the agile frameworks in this paper; instead, we'll concentrate on the three most popular ones: Scrum, Kanban, and XP. The two pillars of scrum development are management and speed. Approximately two weeks pass during a scrum sprint, and a client-tested version of the finished product is produced. In turn, XP seeks to quickly supply the solution (Ahmed et al. 2023). Test-driven development, in which automated unit tests are written before the actual code, is the main tenet of XP. Kanban, on the other hand, makes extensive use of visualization and a unique tool called a

Kanban board. There are various columns in this board, which might be digital or physical, including backlog, in process, testing, and done. The jobs shift across columns, making it obvious how the project is progressing.

3.1.2 Agile development phases

Agile methodology emerged in reaction to the increasing need for a rapid and more flexible method of development (Halani and Jhaharia 2022). Sprints of one to four weeks are used in the agile workflow (e.g., the Scrum method), and there are weekly sprint reviews when members of the team report their progress and set plans for the upcoming sprint. In order to be prepared to make any appropriate changes and guarantee that everything satisfies the customers' requirements, the team also processes feedback from customers. Agile places a greater emphasis on software quality than documentation, which makes documentation less crucial. The Scrum Master, the Product Owner, testers, developers, and data engineers make up a typical agile team. A special position that acts as a liaison between the customer and the team is that of the Product Owner. The project's success is the responsibility of the Product Owner. To do this, the Product Owner must ensure that there is effective communication with customers and that everyone knows what is required (Kavlakoglu 2022). Typically, as shown in Fig. 1, the following phases dictate how software is developed using the agile methodology (Haag and Cummings 2009).

- **Initiation:** The major project requirements, including functionality and intended outcomes, are delivered to the team by the product owner. At this phase, the timelines and costs are also projected.

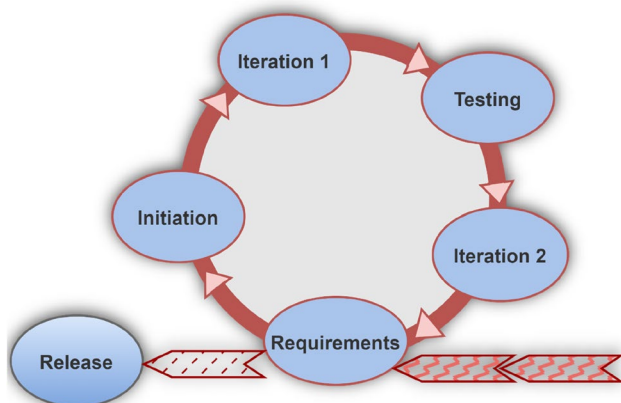


Fig. 1 Agile methodology phases

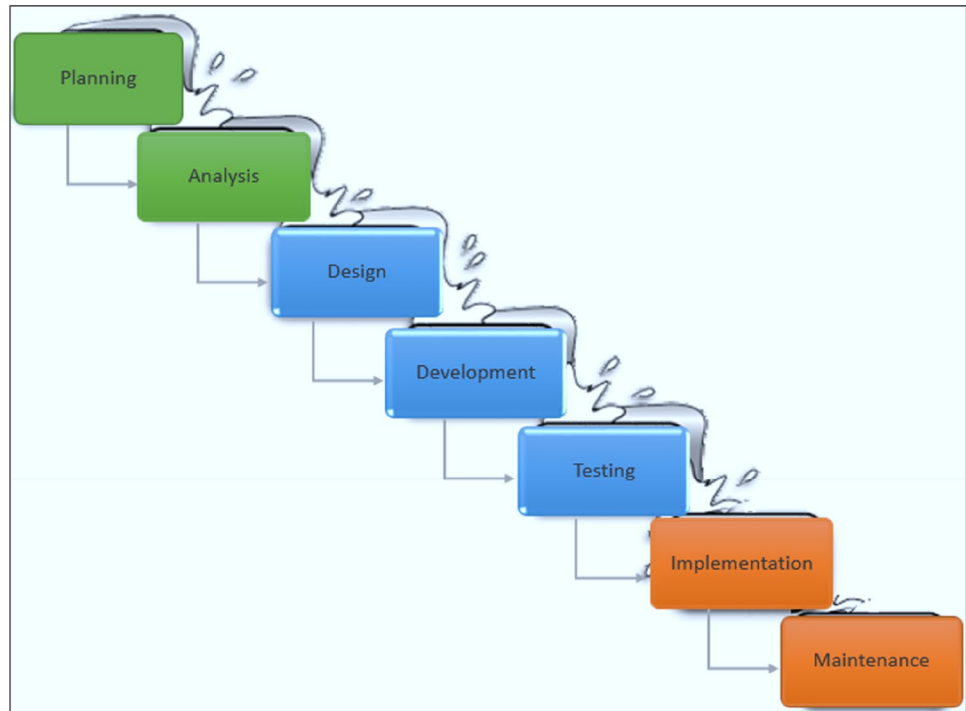
- **Iteration 1:** Development starts after the team has all the information. The user interface and the software architecture are first designed as a prototype.
- **Testing and review:** In this phase, the customer evaluates the feature that has been created and makes changes or additions. Additionally, testers look for any problems.
- **Iteration 2:** Bug fixes and changes are handled during this stage. The following product components are also produced. Testing and review processes are repeated numerous times until the solution matures into a finished product and the customer is happy with the results.
- **Release:** The product is ready for release once the quality assurance team certifies that the solution functions without a hitch and poses no security risks.
- **Maintenance:** The team offers post-release assistance to make sure the solution operates perfectly. They further go through iterations to improve the current product.

3.2 Waterfall development

The software development process, in the waterfall, starts with extensive planning and analysis. Each subsequent step can start only after the preceding one is finished (Yu and Mishra 2010). This methodology is quite rigid and unresponsive to changes. Each component of the final product is considered and documented before the process starts, which is one of the waterfall's main pillars. Progress can be easily monitored and quantified because the entire scope of the project is outlined in advance (Yu and Mishra 2010). A waterfall team consists of a project manager, a business analyst, testers, and developers, among other positions. Customers and other stakeholders are only really involved in the initial planning phase and final evaluations. The typical waterfall software development lifecycle phases are depicted in Fig. 2 (Haag and Cummings 2009).

- **Planning:** The design, functions, problems, and goal of the project are first defined in this phase.
- **Analysis:** At this phase, all business requirements, costs, and schedules are documented.
- **Design:** The project's architecture, models, and design principles are put into production once all requirements have been reviewed and decided.
- **Development:** All project is broken down into components during the development phase and assembled in a linked sequence until the entire final product is assembled.
- **Testing:** The quality crew enters the game, in this phase, once the developers have done developing their code. Their responsibility is to look for errors, security holes, or bugs. When testers discover an issue, developers are notified to fix the code.

Fig. 2 Waterfall software development lifecycle (Source Haag and Cummings 2009)



- **Implementation:** The finished version is ready for public distribution after development and testing are completed.
- **Maintenance:** After installation, the system can be adjusted and changed by the customer's specifications and feedback, or if the system or component does not function as planned.

4 Comparison between agile and waterfall developments

For more than 20 years, software project management has been dominated by waterfall and agile methodologies (Sinha and Das 2021). Each has advantages and disadvantages. In order to construct the decision tree in the following section, this section discusses the pros and cons of each of the two methodologies (Table 4). This table is independent of above citation.

4.1 Agile development pros and cons

If agile didn't have a lot of benefits, it wouldn't have emerged as one of the most successful methodologies (Govil and Sharma 2022). Let's examine the advantages agile methodology offers. Agile is a communication- and customer-oriented methodology that improves customer requirements understanding, feedback, and sharing among team members through informal communication and

face-to-face interaction (Kodmelwar et al. 2022). Moreover, agile methodology is fast in detecting bugs and delivering the product (Saarikallio and Tyrväinen 2022). Also, it is flexible with changing the focus of the project based on the customer's feedback (Hauck and Vieira 2021). On the other hand, agile has some drawbacks, such as unclear outcomes, time and cost constraints (due to this, it may be challenging to guarantee that everyone is on the same page or work may be lost or miscommunicated between team members, especially when people leave and join the group in the middle of projects) (Aitken 2014). Furthermore, if the relationship between management and the technical team weakens, especially when new members join the team and old members depart, which happens frequently in agile development, it could result in time and money waste (Mishra et al. 2021).

4.2 Waterfall development pros and cons

The waterfall is a well-established development methodology that has been approved over many years. Additionally, it comes with thorough instructions and user manuals. Also, it is built on clearly identified requirements, which make the timeframe and cost frame predicted and expected as well as the performance aspects (Kodmelwar et al. 2022). The waterfall methodology also gives the team members more and a progress flow structure so that even if some team members quit or join, the plan won't be significantly affected. However, there are drawbacks to everything. The waterfall methodology lacks the adaptability to accept

Table 4 Comparison between agile and waterfall methodologies

	Agile	Waterfall
Timeline	Compared to the waterfall, agile is far more flexible and allows for experimentation with multiple approaches. Instead of having a set deadline, the schedule changes as the project goes along. The emphasis is on delivering functional software on a regular basis, ideally within a few weeks to a few months	There is a set schedule for the waterfall. The concept is that the project's inception and end have already been planned out from the outset
Flexibility	The agile methodology is designed to be flexible. Even at a later point of the project, the methodology welcomes adjusting to multiple possibilities, integrating new knowledge, or changing customer requirements	Since each step must be finished in its entirety prior to proceeding to the next, the waterfall methodology is less adaptable than agile. The waterfall is appropriate for companies with a solid picture of where they are aiming from beginning to end because the project is planned out in advance
Customer involvement	Customer participation in project development at each stage is a crucial component of agile. Consequently, as the project moves through its many phases, customers are required to participate and provide input to the development team	The customer is not involved in the waterfall method once the end goal has been determined, with the exception of some verify or deliveries. Since the project's route is predetermined from the beginning, taking into account customer feedback is not a continuous step in the procedure
Budget (cost)	Agile is flexible and welcomes flexibility and changes in the course—even in the project's final stages. The cost is typically more adaptable as a result	The cost is typically established. Since the project is planned out from beginning to end, it is less possible to alter the cost in the middle of it
Quality	Project quality is high because consumers are continuously involved in the development process	The level of quality can vary; if the requirements were established correctly from the start, the quality level will be high; otherwise, if the requirements were changed from what was first agreed upon, the level of quality may not satisfy the requirements
Documentation	Low to medium level of user support and documentation guidelines	High-level user support guidelines and documentation
Level of expertise	Developers can have intermediate to high expertise	Developers should be highly skilled
Organizing	Self-organizing team	Hierarchical organizing team

changes in customer requirements, the rate of development, and the ability to patch defects or security flaws until that phase is complete (Wafa et al. 2022). In addition, the waterfall methodology mandates that customers first outline the project from beginning to end. This may become a problem if the customer is divided about its goals and doesn't become aware of this until the project's execution or a later stage. Moreover, testing occurs at the completion of a waterfall project, the final quality check step requires a lot of time, and there is no review throughout the project lifespan, which could lead to a failure to meet the requirements of the customer.

4.3 Agile and waterfall comparison

Agile adoption is primarily driven by quick development cycles, transparency, productivity, and efficiency (Singer 2022). Based on the Standish group report (The-Standish-Group 2020), Fig. 3 demonstrates that in 2020, less than 15% of waterfall projects and more than 40% of agile initiatives were successfully completed. Moreover, compared to 10% of agile projects, about 30% of waterfall programs failed. There are many causes of the failure of the project, but they can all be traced back to poor project management. Project failures can be caused by a lack of time, misunderstood requirements, or inadequate communication. Software development project failure statistics support the claim that 55% of projects fail due to a lack of time. Furthermore, compared to about 50% of agile initiatives, almost 60% of waterfall projects experienced difficulties throughout the project lifecycle. These challenges were caused by limited time and changing customer requirements (Mersino 2022).

The recent studies confirm part of the Standish group report, while conflict with others. A 2019 TrustRadius.com study revealed that 42% of respondents believed that an agile project would succeed, compared to 26% for a waterfall project (PremierAgile 2023). According to a study by Khoza

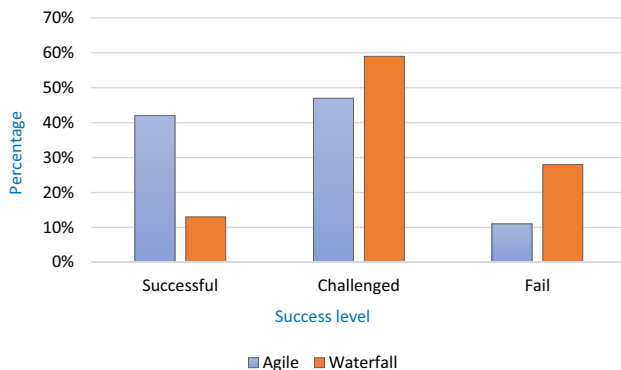


Fig. 3 The successful rate among agile and waterfall methodologies (source The-Standish-Group 2020)

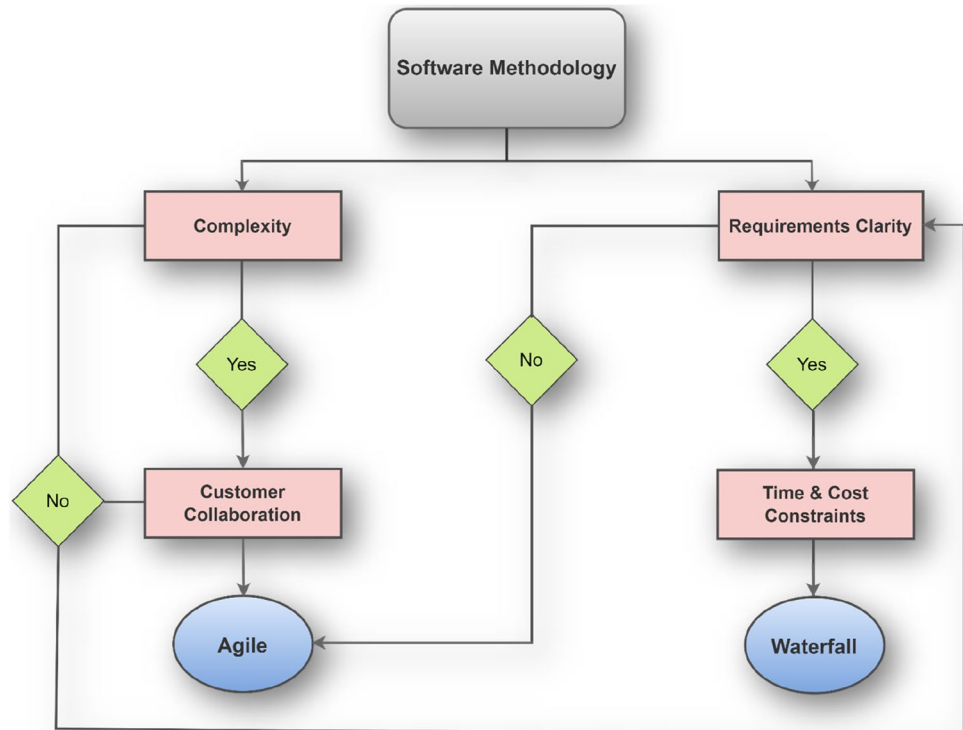
and Marnewick (2020), waterfall initiatives only have an average success rate of 47% while agile projects have an average success rate of 88.2%. The success percentage for agile initiatives is greater than 50%, according to 78% of the respondents (Wafa et al. 2022). Agile methodologies have a success ratio that is two times greater than waterfall-based solutions (Leong et al. 2023).

Agile software development projects take 22.4% longer than anticipated compared to 13.65% longer than the estimated time for the waterfall method (Khoza and Marnewick 2020). However, according to a more recent study, agile-based solutions generate products 37% more quickly than waterfall programs (Leong et al. 2023). It is noteworthy that agile software development initiatives cost 42.62% more than the first estimate. Contrarily, waterfall projects come in at 10.16% less than planned (Khoza and Marnewick 2020). Wafa et al. (2022) reported that 68% of the participants in the survey believed agile practices have a greater impact on their productivity. Compared to 53% who found working with the waterfall challenging, 21% felt working with agile was challenging (PremierAgile 2023). According to 68% of respondents, agile has a greater impact on improving product quality than waterfall, 16% believe it has no impact, and 15% believe it has a smaller impact (Wafa et al. 2022).

Based on the pros and cons of waterfall methodology and agile methodologies, discussed in Sects. 3, 4.1, and 4.2, and summarized in Table 1 (Mersino 2022; Ana 2022; Boersma 2022). In summary, the agile approach involves breaking down projects into smaller tasks, allowing for more feedback and adjustments throughout the process. It is flexible in that it allows for changes to be made based on the project's requirements and priorities. Collaboration is also a key aspect of agile, emphasizing teamwork and interaction between team members and clients. However, the agile methodology can present challenges when dealing with large and complex projects and may require extensive stakeholder involvement that can be difficult to maintain over time.

In contrast, the waterfall method follows a sequential process where each project stage must be completed before moving on to the next one. It is a systematic approach that relies on a well-defined project strategy and scope. Documentation is also a key component of the waterfall methodology and is required for each project stage. However, there are several drawbacks to this approach, including its inflexibility compared to agile, making it challenging to adapt to changes in scope or requirements. Managing large and complex projects can also be problematic. Additionally, the waterfall methodology places less emphasis on collaboration and teamwork, which can result in communication breakdowns and project failures.

Fig. 4 Decision tree to select if agile or waterfall methodology is more suitable



4.4 Decision tree

The waterfall project management is more established compared to an agile project. Waterfall is a method of project management where all of the elements of the product are laid out in advance. The project moves along on a predetermined schedule. On the other hand, agile uses an iterative process rather than outlining the full project from the beginning. Customers are given immediate access to small components of the product by developers, who then use their comments to refine and create the following component. The project may work better with the waterfall or agile methodologies, depending on its unique characteristics. Figure 4 provides a summary of some considerations that can help to choose the suitable methodology. These recommendations were extracted from several research and professional websites such as (Mersino 2022; Halani and Jhaharia 2022; Kavlakoglu 2022; Boersma 2022; AdobeTeam 2022; Hoory and Bottorff 2022).

- Project complexity* For complicated projects, the decision between waterfall and agile methodologies depends on several variables, including the nature of the project, the amount of flexibility necessary, and any legal needs. The ideal strategy, in the end, is thoroughly analyzing the project's unique requirements and selecting the technique that best suits them. In general, the waterfall methodology works well for simple projects and for making improvements to current
- products, while agile is a better option for complicated projects. Agile emphasizes communication and collaboration, which may be especially beneficial when executing large initiatives which call for input from several stakeholders. Conversely, a waterfall could be more appropriate for tasks that are well-defined and have a set route to completion. When engaged in initiatives that have stringent regulation standards, this technique can be quite beneficial.
- Clarity of requirements* The waterfall methodology is an excellent method for managing projects when the customer is aware of every requirement from the outset. If the project's requirements are not known from the beginning, the agile methodology works well. The tight regulations and deliverables for each phase ensure that they are met, making waterfall projects more suitable for projects with restrictions or criteria. As an illustration, since safety is a concern, the Department of Defense and the aerospace sector are two industries that would probably choose to employ waterfall over agile.
- Time constraints* The waterfall is all about set timelines, but if they're not too strict, agile would be a good choice. Agile has advantages in that it produces software faster, but it suffers to deliver predictable completion dates and costs for projects.
- Budget constraints* Similar to the timetable, the waterfall is best suited when money is limited, while agile works better when money is infinite or large.

- *Customer involvement level* Agile works best when the customer demonstrates true engagement in the development process. Contrarily, waterfall involves little customer engagement in the development process.
- *The team collaboration level* The degree of group cooperation agile places a strong emphasis on the value of teamwork and communication. Developers can work more autonomously while using the waterfall methodology because they are each in charge of their components of the project.
- *No methodology is a "magic solution"* for all types of projects. Each is best suited for specific project categories that meet predetermined criteria. The strategy is likely to fail if these project conditions are not met, or are only partially achieved.

5 The hybrid software development approach

It is evident from the discussion in Sects. 2, 3, and 4, particularly the decision tree section, that neither agile nor waterfall can be used for all kinds of software projects or situations. As a result, a plethora of authors and industry professionals have recently advocated using the hybrid software development methodology (Signoretti et al. 2020). To put it another way, depending on the project, either agile or waterfall should be used, and in certain situations, both methodologies may be used at various stages of the project (Włodarski et al. 2020). The adoption of both agile and waterfall methodologies in the same software development firm is explored in this section using various suggestions and recommendations.

According to Fagarasan et al. (2021), a typical hybrid model observed in the information technology business is that when the starting and planning stages are carried out using the waterfall methodology. Nevertheless, the agile methodology, which emphasizes iterative processes, is used for implementation operations. In order to use and embrace hybrid methodology within a software firm, there are two strategies to enhance the delivery processes (Fagarasan et al. 2021).

- In the case of young firms that are fresh to software development, it may be easier to utilize the agile methodology at the organizational level to create the delivery model and adhere to the methodology's precepts.
- A more hybrid methodology is advised for established firms that already have a delivery methodology to combine the conventional methods that have made the firm successful up to this point with the agile methodology.

According to Küpper et al. (2018), the evolution of hybrid software development methods involves three common levels. The first level consists of procedures, which refer to the range of tasks carried out by the development team during the software development life cycle (SDLC) and the guidelines that govern how these tasks should be executed. The second level is frameworks, which encompass one or more techniques and approaches that define the management of the SDLC that the team must adhere to. Finally, the third level is context, which pertains to the overall success criteria and is derived from the project goals.

Leong et al. (2023) suggested that in order to improve the chances of success in agile development, it is necessary to integrate three essential components: people, procedures, and organizational elements, into the project management paradigm. This has led to the development of a new project management methodology called the hybrid methodology. Kirpitsas and Pachidis (2022) stated that in the hybrid method, the creation of a prioritized product backlog of user stories follows the design phase to support the agile process. The team then engages in iterative development, primarily following agile principles and gradually expanding on those requirements. During the sprints, the development team conducts unit testing, while systems integration testing and user acceptance testing are carried out once the iterative development is complete. The tested and developed code can then be delivered to customers or put into production, along with the necessary documentation.

In most businesses, there is no clear winner when it comes to methodologies; therefore, agile and waterfall must coexist because both have a lot to contribute. Some projects require both the responsiveness of an agile methodology and the discipline of a waterfall methodology. Taking the best of the two could be preferable, but it is unclear how to do so, particularly when it comes to handling criteria in a large project (Kasauli et al. 2020). Positive results are possible when combining the two thanks to expectations and interaction (AdobeTeam 2022). For instance, unlike a waterfall, a hybrid methodology does not require engineers to wait for one phase to end before moving on to the next (Leong et al. 2023). Instead, as quickly as feasible, the subsequent phase starts (Khan and Akbar 2022). As a result, the project's components and modules are developed independently (one of the key agile features). The documentation may still be kept up to date and deadlines can still be reached because the distinctive waterfall characteristics are also present (Kassab et al. 2018). Hybrid methodology benefits from lower project costs, faster development, and well-managed documentation (Morgan 2018).

Management of projects is changing. Teams that were formerly compartmentalized are now expected to collaborate to reduce reporting hassles and improve visibility across

all task kinds. To increase team productivity, complete faster production, provide precise metrics to all project stakeholders, and foster a climate that is conducive to all forms of work, 44% of project directors embrace a hybrid methodology, according to AdobeTeam (2022). Adapting the principles to match the team is one of the basic principles of agile project management. Agile teams that rigidly follow the rules they've established lose the objective of the sense of agility altogether (Kirpitsas and Pachidis 2022). This demonstrates that hybrid methodology is feasible and appropriate for many projects. A hybrid methodology might combine the waterfall's organized up-front project plan with an agile emphasis on delivering small software components (Boersma 2022). This hybrid methodology accomplishes some of the favorite parts of waterfall methodology while completing tiny software components faster. Customers prefer receiving software more rapidly, while management loves the more disciplined approach to software development (Dima and Maassen 2018). Many developers value being free from the numerous sessions that agile requires of their members (Hoory and Bottorff 2022).

Some software development experts blend a structured, waterfall methodology with elements of the agile mindset that succeed for them. The team members may find genuine benefit in conducting frequent retrospectives to pinpoint opportunities for improvement. Whatever the cause, the best way to ensure success is to address the plan for project management by selecting a methodology that works for team members or the specific project (Boersma 2022). By employing a solution that supports mixed methodologies, the hybrid methodology can help all teams and the company as a whole by enhancing productivity and visibility (Prenner et al. 2022). The hybrid methodology may enable total insight into what has been done, what is being done, and what to do in the future (Gill et al. 2018). This will not be possible without increased visibility and adequate planning, training, and benchmarks. While project management may select waterfall to ensure dependencies and deadlines, developers may select agile to improve code productivity. Providing teams with the freedom to pick their work manner boosts productivity (Özkan and Mishra 2019).

6 Discussion

This review investigated the perennial debate over whether agile or waterfall software management methodologies are better. This study addressed the following two research questions in order to resolve this conflict: When is the waterfall methodology preferred over agile, and vice versa? (RQ1) And when should waterfall and agile approaches be combined? (RQ2). Many firms are still using the waterfall methodology since it simply works and has a successful

track record, even though it has begun to be replaced by agile methodology as the industry norm for software project delivery. The managers who use a waterfall are at ease because everything is thoroughly recorded, and a centralized decision-making procedure minimizes human mistakes.

Waterfall and agile are two distinct management methodologies that work best for particular project types. Waterfall could be the greatest option when the project goals are known from the beginning. When a project must adhere to rigorous standards, the waterfall methodology is preferable because it calls for deliverables for each step before moving on to the next (Pai et al. 2019). Agile, on the other hand, is especially suitable for teams that intend to move quickly and don't know exactly how the project will turn out before they begin. Agile demands a dynamic team that is cooperative and self-organized, as well as regular progress updates from customers and enterprise owners. Agile philosophy has impacted software development practices overall. Despite this, there is growing concern regarding software design.

Achieving high quality may be challenged with the adaptability of any improvement strategy that neglects to adequately take the design issue into account (Arcos-Medina and Mauricio 2019; Schmalz et al. 2014). The failure rate was identified, in the Standish Group report, based on project size. Smaller projects are more likely to be successful than larger ones since they include shorter feedback cycles and lower risk (The-Standish-Group 2020). Small projects have a success rate that is three times higher than large ones for agile projects, and for waterfall projects, small initiatives have a 6X higher success rate than large ones (Mersino 2022). Additionally, the feasibility of combining the waterfall and agile methodologies in software development management is investigated in this paper. To take the finest elements of each methodology and use them in various project contexts, many experts and authors suggested creating a hybrid combination of the two methodologies (Alam et al. 2022).

7 Conclusions and future directions

Agile software development has been advocated for more than 20 years as a solution to the rigidity issue with structured/traditional methodologies such as the waterfall. However, current evidence indicates that many software firms continue to use the waterfall methodology. Although there are many alternative methodologies available for software development, the waterfall and agile methodologies are generally regarded as the most efficient. The flexibility of agile and the stability of the waterfall are the two most substantial benefits. There are ongoing discussions regarding whether Agile Methods or structured methodology is

better suited for software firms, which never seem to end. Therefore, it is important to thoroughly grasp the project scope before selecting a development methodology to ensure that the project is on the right track.

The goal of this study was to investigate this claim by comparing the waterfall and agile methodologies in order to determine why both approaches are so popular. The article also included a decision tree that could be used to decide whether the agile or waterfall methodology is better suited for a certain project setting. The study came to the conclusion that no single silver belt methodology is appropriate for all kinds of software development projects. The software firms may therefore need to use a hybrid framework that combines agile and waterfall. Utilizing the framework's possible beneficial usage of both agile and waterfall methodology's features is its core goal. Because it's popular and hip, some people strive to incorporate agile into every project they work on. However, the waterfall methodology is a far superior choice in other sectors, such as marketing and construction. It is crucial to remember that agile is a very flexible methodology that can be tailored to the specific requirements of the firm. While the waterfall methodology is quite rigid and it is very formal, it too may be altered as needed. Therefore, while it is advised to adhere to the fundamental tenets of each methodology to ensure that it functions as intended, firms could also adapt their own methodology and adjust it as necessary.

Future research might concentrate on examining an actual company that uses a hybrid software development methodology. This is essential to learn how and when they employ agile or waterfall, and what kinds of projects they successfully accomplished utilizing the hybrid methodology. All of that will assist in validating the findings of this paper. To sum up, successful firms must be able to deliver software on a massive scale. It is advised that firms utilizing the current waterfall software delivery paradigm implement agile concepts as the digital world is already at a pivotal time.

Funding Open access funding provided by NTNU Norwegian University of Science and Technology (incl St. Olavs Hospital - Trondheim University Hospital). This research received no external funding.

Declarations

Conflict of interest Authors doesn't have any conflict of interest.

Human and animals' participants Human Participants and/or Animals are not involved in this research.

Informed consent There was no informed consent in this research article.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- AdobeTeam (2022) Project management—agile versus waterfall. Adobe Experience Cloud Blog. <https://business.adobe.com/blog/basics/agile-vs-waterfall>, viewed 23 October 2022 [Online]
- Ahmed M, Khan SUR, Alam KA (2023) An NLP-based quality attributes extraction and prioritization framework in agile-driven software development. *Autom Softw Eng* 30(1):7
- Aitken A (2014) Dual application model for agile software engineering. In: 47th Hawaii international conference on system sciences, Waikoloa, USA. IEEE, pp 4789–4798
- Alam I, Sarwar N, Noreen I (2022) Statistical analysis of software development models by six-pointed star framework. *PLoS ONE* 17(4):e0264420
- Almeida F, Simões J (2019) Moving from waterfall to agile: perspectives from IT Portuguese companies. *Int J Serv Sci Manag Eng Technol (IJSSMET)* 10(1):30–43
- Al-Saqqā S, Sawalha S, AbdelNabi H (2020) Agile software development: methodologies and trends. *Int J Interact Mob Technol* 14(11):246–270
- Alzoubi YI, Gill AQ (2020) An empirical investigation of geographically distributed agile development: the agile enterprise architecture is a communication enabler. *IEEE Access* 8:80269–80289
- Alzoubi Y, Gill A (2021) The critical communication challenges between geographically distributed agile development teams: empirical findings. *IEEE Trans Prof Commun* 64(4):322–337
- Alzoubi YI, Gill AQ (2022) Can agile enterprise architecture be implemented successfully in distributed agile development? Empirical findings. *Glob J Flex Syst Manag* 23(2):221–235
- Alzoubi YI, Gill AQ, Al-Ani A (2015) Distributed agile development communication: an agile architecture driven framework. *J Softw* 10(6):681–694
- Alzoubi YI, Gill AQ, Al-Ani A (2016) Empirical studies of geographically distributed agile development communication challenges: a systematic review. *Inf Manag* 53(1):22–37
- Ammad G, Janjua UI, Madni TM, Cheema MF, Shahid AR (2019) An empirical study to investigate the impact of communication issues in GSD in Pakistan's IT industry. *IEEE Access* 7:171648–171672
- Ana (2022) Software development statistics. True List. <https://truelist.co/blog/software-development-statistics/>, viewed 24 October 2022 [Online]
- Arcos-Medina G, Mauricio D (2019) Aspects of software quality applied to the process of agile software development: a systematic literature review. *Int J Syst Assur Eng Manag* 10:867–897
- Bishop D, Deokar A (2014) Toward an understanding of preference for agile software development methods from a personality theory

- perspective. In: 47th Hawaii international conference on system sciences, Waikoloa, USA. IEEE, pp 4749–4758
- Boersma E (2022) Is there a place for the waterfall methodology in 2021? Plutora.com. <https://www.plutora.com/blog/waterfall-methodology>, viewed 26 October 2022 [Online]
- Chandran K, Das Aundhe M (2022) Agile or waterfall development: the Clementon company dilemma. *J Inf Technol Teach Cases* 12(1):8–15
- DevFacto (2023) A hybrid software development method. <https://www.devfacto.com/blog/a-hybrid-software-development-method>, viewed 25 April 2023 [Online]
- Dima AM, Maassen MA (2018) From waterfall to agile software: development models in the IT sector, 2006 to 2018. Impacts on company management. *J Int Stud* 11(2):315–326
- Dingsøyir T, Moe NB, Seim EA (2018) Coordinating knowledge work in multiteam programs: findings from a large-scale agile development program. *Proj Manag J* 49(6):64–77
- Dursun M, Goker N (2022) Evaluation of project management methodologies success factors using fuzzy cognitive map method: waterfall, agile, and lean six sigma cases. *Int J Intell Syst Appl Eng* 10(1):35–43
- Fagarasan C, Popa O, Pisla A, Cristea C (2021) Agile, waterfall and iterative approach in information technology projects. In: IOP conference series: materials science and engineering, Oradea, Romania. IOP Publishing, vol 1169, no 1, p 012025
- Friess E (2019) Scrum language use in a software engineering firm: an exploratory study. *IEEE Trans Prof Commun* 62(2):130–147
- Gill AQ, Henderson-Sellers B, Niazi M (2018) Scaling for agility: a reference model for hybrid traditional-agile software development methodologies. *Inf Syst Front* 20(2):315–341
- Govil N, Sharma A (2022) Validation of agile methodology as ideal software development process using fuzzy-TOPSIS method. *Adv Eng Softw* 168:103125
- Gustavsson T, Berntzen M, Stray V (2022) Changes to team autonomy in large-scale software development: a multiple case study of scaled agile framework (SAFe) implementations. *Int J Inf Syst Proj Manag* 10(1):29–46
- Haag S, Cummings M (2009) *Management information systems for the information age*, 9th edn. McGraw-Hill Inc., New York
- Halani KR, Jhahharia K (2022) A quantitative study of waterfall and agile methodologies with the perspective of project management. In: *Contemporary challenges for agile project management: IGI Global*, pp 111–133
- Hamilton T (2023) Agile vs waterfall—difference between methodologies. <https://www.guru99.com/waterfall-vs-agile.html>, viewed 25 April 2023 [Online]
- Hauck JCR, Vieira M (2021) Towards a guide for risk management integration in agile software projects. In: Yilmaz M, Clarke P, Messnarz R, Reiner M (eds) *Systems, software and services process improvement. EuroSPI 2021. Communications in computer and information science*. Springer, Cham, pp 73–87
- Hoory L, Bottorff C (2022) Agile versus waterfall: which project management methodology is best for you? *Forbes Advisor*. <https://www.forbes.com/advisor/business/agile-vs-waterfall-methodology/>, viewed 23 October 2022 [Online]
- Hummel M, Rosenkranz C, Holten R (2015) The role of social agile practices for direct and indirect communication in information systems development teams. *Commun Assoc Inf Syst* 36(1):15
- Käpyaho M, Kauppinen M (2015) Agile requirements engineering with prototyping: a case study. In: 23rd International requirements engineering conference (RE), Ottawa, Canada. IEEE, pp 334–343
- Kasauli R, Knauss E, Nakatumba-Nabende J, Kanagwa B (2020) Agile islands in a waterfall environment: challenges and strategies in automotive. In: *Evaluation and assessment in software engineering, Trondheim, Norway*. ACM, pp 31–40
- Kassab M, DeFranco J, Neto VG (2018) An empirical investigation on the satisfaction levels with the requirements engineering practices: agile versus waterfall. In: *International professional communication conference (ProComm)*, Toronto, Canada. IEEE, pp 118–124
- Kavlakoglu E (2022) Which project management methodology should you use to manage your next project? IBM. <https://www.ibm.com/cloud/blog/agile-vs-waterfall>, viewed 26 October 2022 [Online]
- Khan AR, Akbar R (2022) Software development process evolution and paradigm shift: a case study of Malaysian companies. *Mehran Univ Res J Eng Technol* 41(2):155–171
- Khoza LT, Marnewick C (2020) Waterfall and agile information system project success rates: a South African perspective. *S Afr Comput J* 32(1):43–73
- Kirpitsas IK, Pachidis TP (2022) Evolution towards hybrid software development methods and information systems audit challenges. *Software* 1(3):316–363
- KnowledgeHut (2023) Agile transformation and its challenges. <https://www.knowledgehut.com/blog/agile/agile-transformation-and-challenges>, viewed 24 April 2023 [Online]
- Kodmelwar MK, Futane PR, Pawar SD, Lokhande SA, Dhanure SP (2022) A comparative study of software development waterfall, spiral and agile methodology. *J Posit Sch Psychol* 6(3):7013–7017
- Kumar G, Bhatia PK (2014) Comparative analysis of software engineering models from traditional to modern methodologies. In: 4th international conference on advanced computing & communication technologies, Rohtak, India. IEEE, pp 189–196
- Küpper S, Rausch A, Andelfinger U (2018) Towards the systematic development of hybrid software development processes. In: *International Conference on Software and System Process*. ACM, Melbourne, Australia, pp 157–161
- Kuusinen K, Gregory P, Sharp H, Barroca L (2016) Strategies for doing agile in a non-agile environment. In: 10th ACM/IEEE international symposium on empirical software engineering and measurement, Ciudad Real, Spain. ACM, pp 1–6
- Leong J, May Yee K, Baitsegi O, Palanisamy L, Ramasamy RK (2023) Hybrid project management between traditional software development lifecycle and agile based product development for future sustainability. *Sustainability* 15(2):1121
- Mersino A (2022) Why agile is better than waterfall. *Vitality Chicago*. <https://vitalitychicago.com/blog/agile-projects-are-more-successful-traditional-projects/>, viewed 24 October 2022 [Online]
- Mishra D, Aydin S, Mishra A, Ostrovska S (2018) Knowledge management in requirement elicitation: situational methods view. *Comput Stand Interfaces* 56:49–61
- Mishra A, Abdalhamid S, Mishra D, Ostrovska S (2021) Organizational issues in embracing agile methods: an empirical assessment. *Int J Syst Assur Eng Manag* 12(6):1420–1433
- Morgan JD (2018) Applying 1970 waterfall lessons learned within today's agile development process. *PM World J* 7:1–19
- Nerur S, Mahapatra R, Mangalaraj G (2005) Challenges of migrating to agile methodologies. *Commun ACM* 48(5):72–78
- Özkan D, Mishra A (2019) Agile project management tools: a brief comparative view. *Cybern Inf Technol* 19(4):17–25
- Pai A, Joshi G, Rane S (2019) Integration of agile software development and robust design methodology in optimization of software defect parameters. *Int J Syst Assur Eng Manag* 10:1043–1051
- Poppendieck M, Cusumano MA (2012) Lean software development: a tutorial. *IEEE Softw* 29(5):26–32
- PremierAgile (2023) Agile vs waterfall (traditional) project management. <https://premieragile.com/differences-between-agile-and-waterfall/>, viewed 26 April 2023 [Online]
- Prenner N, Klünder J, Schneider K (2022) Defining frames to structure agile development in hybrid settings: a multi-case interview study. In: *International conference on software and system processes and*

- international conference on global software engineering (ICSSP 2022), Virtual. ACM, pp 34–44
- ProductPlan (2023) The waterfall to agile transition: how roadmaps can keep you on track. <https://www.productplan.com/learn/waterfall-to-agile-transition/>, viewed 25 April 2023 [Online]
- ProjectManager (2023) Agile vs waterfall and the rise of hybrid projects. <https://www.projectmanager.com/guides/agile-vs-waterfall-hybrid-projects>, viewed 25 April 2023 [Online]
- Raza M (2023) Agile vs waterfall SDLCs: what's the difference? <https://www.bmc.com/blogs/agile-vs-waterfall/>, viewed 24 April 2023 [Online]
- Richter I, Raith F, Weber M (2016) Problems in agile global software engineering projects especially within traditionally organised corporations: an exploratory semi-structured interview study. In: 9th international C* conference on computer science & software engineering, Porto, Portugal. ACM, pp 33–43
- Saarikallio M, Tyrväinen P (2022) Quality culture boosts agile transformation—action research in a business-to-business software business. *J Softw Evol Process* 35(1):e2504
- Schmalz M, Finn A, Taylor H (2014) Risk management in video game development projects. In: 47th Hawaii international conference on system sciences, Waikoloa, USA. IEEE, pp 4325–4334
- Serebryantseva V (2022) Agile vs waterfall: which approach should you choose for your project. *Pixel Plex*. <https://pixelplex.io/blog/agile-waterfall-software-development/>, viewed 23 October 2022 [Online]
- Shameem M, Kumar RR, Nadeem M, Khan AA (2020) Taxonomical classification of barriers for scaling agile methods in global software development environment using fuzzy analytic hierarchy process. *Appl Soft Comput* 90:106122
- Signoretti I et al (2020) Success and failure factors for adopting a combined approach: a case study of two software development teams. In: Morisio M, Torchiano M, Jedlitschka A (eds) Product-focused software process improvement. PROFES 2020. Lecture notes in computer science, vol 12562. Springer, Cham, pp 125–141
- Singer M (2022) 15+ surprising agile statistics: everything you need to know about agile management. *Enterprise Apps Today*. <https://www.enterpriseappstoday.com/>, viewed 24 October 2022 [Online]
- Sinha A, Das P (2021) Agile methodology versus traditional waterfall SDLC: a case study on quality assurance process in software industry. In: 5th international conference on electronics, materials engineering & nano-technology (IEMENTech), Kolkata, India. IEEE, pp 1–4
- Soundararajan S, Arthur JD (2009) A soft-structured agile framework for larger scale systems development. In: 16th annual IEEE international conference and workshop on the engineering of computer based systems, San Francisco, USA. IEEE, pp 187–195
- Srivastava A, Mehrotra D, Kapur P, Aggarwal AG (2020) Analytical evaluation of agile success factors influencing quality in software industry. *Int J Syst Assur Eng Manag* 11:247–257
- Stoica M, Mircea M, Ghilic-Micu B (2013) Software development: agile versus traditional. *Inform Econ* 17(4):64–76
- Thesing T, Feldmann C, Burchardt M (2021) Agile versus waterfall project management: decision model for selecting the appropriate approach to a project. *Procedia Comput Sci* 181:746–756
- The-Standish-Group (2020) Standish group chaos report 2020. The Standish Group. <https://www.standishgroup.com/news/45>, viewed 24 October 2022 [Online]
- Thummadi BV, Lyytinen K (2020) How much method-in-use matters? A case study of agile and waterfall software projects and their design routine variation. *J Assoc Inf Syst* 21(4):864–900
- Wafa R, Khan MQ, Malik F, Abdusalomov AB, Cho YI, Odarchenko R (2022) The impact of agile methodology on project success, with a moderating role of Person's job fit in the IT industry of Pakistan. *Appl Sci* 12(21):10698
- Włodarski R, Poniszewska-Marañda A, Falleri J-R (2020) Comparative case study of plan-driven and agile approaches in student computing projects. In: 2020 international conference on software, telecommunications and computer networks (SoftCOM), Split, Croatia. IEEE, pp 1–6
- Yadav M, Goyal N, Yadav J (2015) Agile methodology over iterative approach of software development: a review. In: 2nd international conference on computing for sustainable global development (INDIACom), New Delhi, India. IEEE, pp 542–547
- Yu L, Mishra A (2010) Risk analysis of global software development and proposed solutions. *Automatika* 51(1):89–98

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.