iMedPub Journals http://www.imedpub.com

American Journal of Computer Science and Information Technology 2021

Vol. 9 No. 10: 115

Software Project Management: Evaluation of Agile Software Processes a Short Comparison with Waterfall Model

Abstract

Worldwide software companies are increasingly adopting agile project management concepts, tools, and procedures. Despite the fact that agile software project management techniques appear to be a more flexible and sound alternative to traditional project management approaches, additional research into how to use agile project management is required. This article presents the findings of a study on the practical effects of applying agile project management in software project management.

Change is inevitable as an organization strives to become nimbler. It has an impact on old culture and introduces new project management methods; therefore not everyone will be convinced! What does it take to make that change a success? Since its inception in the 1990s in IT, agile project management has come a long way.

Many people are still skeptical of the concept, particularly if it involves a cultural shift or a change in the way things have traditionally been done you know where you are with tried and true methods, tools, and approaches.

This study explains the art of project management, project management phases, and short analysis on traditional project management methodology. The study further expresses the need for agile methodology, principles of agile methodology, its benefits and various frameworks in agile methodology.

Agile allows customers to double-check that the product is precisely what they expected by testing it frequently throughout the process. This also improves the project's chances of being finished on schedule and under budget. The customer is considered a member of the implementation team by Agile, and they are involved at every stage of the process. This paper analyses the art of project management, traditional methodology, agile development cycle, benefits of agile methodology, the agile frameworks, and methodologies.

Choosing the proper agile methodology from the several forms of agile methodology necessitates a dynamic approach. To entice talent and deliver outstanding digital experiences in this fiercely competitive industry, the advantages and downsides of agile methodology must always be weighed while choosing the framework for one's firm.

This research, which established the agile software development process, focused on the iterative approach and important features of agile methodology over Waterfall in software project management.

Problem statement: What method of organizing software development work to choose? Is one of the most difficult and timeless problems in project management. This is a discussion about development methods. The traditional technique in use is the Waterfall model; nevertheless, this paper explains why agile project management methods should be preferred over Waterfall in the modern software

Wumi Ajayi^{1*}, Roqib Akintunde Akinyemi², Elias Patience Elewechi², Michael Oluwagbenga Taiwo² and Abraham Olanrewaju²

¹Computer Science Department, Babcock University, Ikenne, Nigeria ²Lead City University, Department of Computer Science, Toll Gate, Ibadan, Nigeria

Corresponding author:

Wumi Ajayi, Computer Science Department, Babcock University, Ikenne, Nigeria

ajayiw@babcock.edu.ng

Citation: Ajayi W, Akinyemi RA, Elewechi EP, Taiwo MO, Olanrewaju A (2021) Software Project Management: Evaluation of Agile Software Processes a Short Comparison with Waterfall Model. Am J Compt Sci Inform Technol Vol.9 No.10: 115.

development business. Customers are less involved in the old process. As the project progresses, certain clients will want to be more involved. The Agile

technique overcomes this gap in the client process. It ensures that the client is kept informed throughout the process. Customers get the ability to see the work in progress and make decisions and modifications as it progresses.

In a changing world, where clients value speed and agility, the agile revolution became necessary. This paper will discuss why Agile software development is preferable than the waterfall methodology.

Customers can no longer afford to be trapped into long IT projects that cannot be updated or adjusted using waterfall once they've started. Iterative approaches are widely used in software development projects with agile methodology to enhance velocity and adaptability since iteration allows you to alter as you go rather than following a linear path.

Keywords: Software project; Agile frameworks; Traditional project; Waterfall model; Scrum

Received: October 26, 2021; Accepted: November 09, 2021; Published: November 16, 2021

Introduction

Going by the definition of Software Project, it is the entire process of software development, from requirement collecting to testing and maintenance that is carried out in a specific time frame to produce the desired software product, according to the execution techniques [1]. This is divided into types, the traditional and the modern. The traditional involves the Waterfall Methodology while an example of the Modern Software Project Management technique is AGILE which would be our main focus in this Study.

Literature Review

To create successful software, software teams are required. Most software businesses use agile software development to create high-quality software in less time and at a lower cost. The contribution of successful software initiatives is influenced by several factors. Software team productivity is another essential idea in agile software development that contributes to project success.

Our main or primary focus is to look at the Agile Software Management technique in comparison to Waterfall model. Both methodologies can help developers produce high-quality project management. Counting on the precise project requirement, knowing the difference between agile and waterfall can better equip a development team to settle on the proper process and methods in delivering a successful software project. We can as well introduce us to the edge that the Agile Management Technique has over the Waterfall Model which includes agile separates a project into sprints; and Waterfall dividing hers into phases. Agile allows requirement changes at any point in time; Waterfall on the contrary does avoid scope changes once the project starts. Also agile enables the project to work without a fanatical project manager, while Waterfall does not [2]. impacts total project success; consequently, studying team member productivity is of great interest. Agile team members should be taught to understand and manage productivity factors on a regular basis because they are self-managed. The primary aim, according to the agile manifesto, is to satisfy the client by delivering valuable software early and frequently. Working software should be delivered between a couple of weeks to a couple of months, which thus demonstrates a project's progress. The agile manifesto defined the criteria of software productivity and customer satisfaction by short iterations and deliverables. As a result, examining agile team productivity is critical for success in agile software development; nevertheless, there are various other elements at play, and their impact on agile teams must still be extensively and efficiently investigated. As a result, examining agile team productivity is critical for success in agile software development; nevertheless, there are various other elements at play, and their impact on agile teams must still be extensively and efficiently investigated.

Hence, there therefore exist and provide for us a reasonable reason for us to carefully and meticulously look into what the Agile can do such as Fatter Software development lifecycle, predictable schedule in sprints, empowering teams to manage projects as well as promoting efficient communications among team members. We would be taking us through the beautiful pieces of agile by using two of her framework SCRUM and KANBAN.

I strongly believe that this paper will do first a constructive criticism to the study of Software Methodological application as well as making a very strong case for the use and acceptability even recommendatory use for the Agile Software Management Techniques across board.

The art of project management

Project management, as defined by some authors, is "the

application of knowledge, skills, tools, and techniques to project activities in order to meet project requirements" [3]. As an integral part of software engineering processes, project management, along with business analysis and requirement specification, design, programming, and testing, has been a source of considerable debate for years. According to survey results from the Project Management Institute (PMI), even as business project management procedures evolve, only approximately half of them (54 percent) are fully aware of the necessity and usefulness of these practices.

Project management has proven to be a critical component of a company's efficiency and eventual success, regardless of industry. Firms that use proven project management strategies squander 28 percent less money and have 2.5 times more successful projects. According to project management experts, a successful project is one that is not only completed on time and under budget, but also provides the anticipated advantages.

Project management phases

Any project, regardless of its scope, should follow a set of steps that can be controlled and managed. A typical project management process, according to the Project Management Institute, contains the following phases:

- 1. Initiation
- 2. Planning
- 3. Execution
- 4. Performance/Monitoring
- 5. The project has come to an end.

These phases describe the project management lifecycle and serve as a roadmap for completing certain tasks. However, this structure is far too broad. Within each phase of a project, there are generally several internal stages. Depending on the scope of work, the team, the industry, and the project itself, they can vary significantly. Humanity has produced a large variety of project management approaches and methodologies to discover a universal approach to project management [4].

Traditional project management methodologies

Traditional techniques follow the above-described classic structure and adopt a step-by-step approach to project execution. As a result, the project proceeds in stages from start to finish, including planning, implementation, and monitoring. This strategy, which is also known as linear, consists of a series of internal phases that are consecutive and carried out in a chronological order. Traditional project management, which is most typically used in the construction or manufacturing industries when little or no adjustments are necessary at every stage, has also found applicability in software engineering. The waterfall model, often known as the waterfall process, has been a popular software development paradigm since the early 1970s [5].

The waterfall model

The waterfall paradigm places a heavy emphasis on planning and

© Under License of Creative Commons Attribution 3.0 License

specification development, which can account for up to 40% of the project's time and budget. A precise sequencing of project phases is another fundamental principle of this approach. A new stage of a project does not begin until the preceding one has been completed. The strategy works effectively for projects with a clear scope, a single deliverable, and a set timeframe (Figure 1).



The waterfall method necessitates meticulous planning, substantial project documentation, and close monitoring of the development process. This should, in principle, produce in ontime, on-budget delivery, reduced project risks, and predictable final outcomes. However, due to the multiple constraints, the waterfall method is sluggish, costly, and inflexible when applied to the actual software engineering process. In many circumstances, the company's inability to adapt the product to changing market demands leads to a massive waste of resources and eventual project failure.

Why do we need agile methodology?

We had the Waterfall model of software development before Agile came around. The waterfall model is a top-down approach to the creation of a system or software that follows a sequential process. This was an easy-to-understand and linear model. Requirements gathering, software design, implementation, testing, deployment, and maintenance were all aspects of the waterfall methodology [6].

This methodology, however, had a few flaws, including the following:

• It was too time-consuming. You cannot move on to the next step unless you finish the previous one.

• This methodology was only appropriate for projects with consistent requirements.

- The working software is supplied only after the model's final stage has been completed.
- It is tough to go back to earlier phases and make modifications that you had not considered during the initial phase.

Because of the Waterfall model's shortcomings, the leaders of

several development approaches decided to gather to find a solution to these issues. These leaders came to an agreement on a lightweight development method and were able to put it into words in the form of a manifesto. It was titled "The Agile Software Development Manifesto." There are four values and twelve principles in this declaration.

Agile project management methodology

In contrast to established techniques, the agile methodology was developed to make software engineering more flexible and efficient. Agile project management has become a standard in project management, with 94 percent of firms using it in 2015. The history of agile may be traced back to 1957, when IBM and Motorola were developing software using incremental development approaches (now known as Agile). Despite not knowing how to categorize the technique they were using, they all agreed that it was distinct from the Waterfall in several ways [7].

However, when a group of 17 software development professionals met in 2001 to examine alternate project management approaches, the modern-day agile methodology was officially established. They set out the flexible, lightweight, and teamoriented software development strategy in the Manifesto for Agile Software Development, which they had a clear vision of. The Manifesto clearly explains the core ideas of the new approach, with the goal of "discovering better ways of producing software." The philosophy has become a ubiquitous and efficient new way to manage projects, especially when combined with the Twelve Principles of Agile Software.

When it comes to software development, agile techniques adopt an iterative approach. Agile projects, unlike a traditional linear waterfall model, are made up of a series of smaller cycles called sprints. Each one is a mini project with a backlog and stages for design, development, testing, and deployment within a predetermined scope of work.

Agile development cycle

A potentially shippable product increment is given at the end of each Sprint. As a result, new features are added to the product with iteration resulting in progressive project expansion. The risks of producing a possibly failing product are considerably reduced when features are evaluated early in the development process (Figure 2).



The main agile aspects:

• Flexibility: As new requirements arise; the scope of work may alter.

• Work breakdown: The project is broken down into small cycles (known as Sprints in Scrum).

• Teamwork is important: Team members collaborate closely and have a clear understanding of their roles.

• Iterative improvements: The work done during a cycle is frequently reassessed to improve the final output.

• Collaboration with a client: A client is actively involved in the development process and can amend requirements or accept the team's recommendations. The Agile Manifesto's four values are as follows:

o Individuals and Interactions vs. Tools and Processes

o Working software trumps thorough documentation

o Customer collaboration trumps contract negotiations

o Adapting to Change trumps sticking to a plan

The 12 principles that are mentioned in the agile manifesto are as follows [8]:

1. Customer satisfaction through timely and consistent supply of useful software.

2. Be opened to altering needs, especially late in the development process.

3. Consistently provide working software (weeks rather than months)

4. Businesspeople and developers work together daily.

5. Projects are created on the backs of motivated people who can be trusted.

6. The best method of communication is a face-to-face interaction (co-location)

7. The ability to maintain a continuous pace of development is the most important indicator of progress.

8. Consistent development, i.e., development that can keep up with the pace of the rest of the world.

9. Maintaining a constant focus on technical excellence and good design

10. Simplicity is crucial—the art of minimizing the amount of effort that is not done.

11. Self-organizing teams produce the best architectures, needs, and designs.

12. The team considers ways to improve its effectiveness on a regular basis and adjusts as needed.

2021

Vol. 9 No. 10: 115

Agile methodology benefits: The following are some of the benefits of AGILE methodology.

• The software delivery is persistent in AGILE.

• Working feature(s) are given to the customer at the end of each Sprint. They will be more satisfied because of this.

• Customers can examine the produced features to see if they meet their requirements.

• If customers have any input or request adjustments to the features, these can be addressed in the current or possibly next product version.

• Even in the last stages of product development, changes might be made.

• In AGILE, product development and businesspeople connect daily.

• The product's design receives a lot of consideration.

Agile frameworks

Agile refers to a wide range of frameworks and methodologies that share the above-mentioned ideas and ideals. Each one has its own set of applications and distinguishing characteristics. Scrum, Kanban, Hybrid, Lean, Bimodal, and XP are the most prominent frameworks.

The agile methodologies listed below are some of the most popular agile methodologies to choose from:

Scrum: It is the most popular agile framework. It is utilized entirely by 58 percent of businesses, while another 18 percent use it in conjunction with other methods. The New Product Development Game was first described in 1986 and formulated over a decade later [9]. The presentation was based on what they learned while using the strategy over the preceding few years. While Scrum predates the Agile Manifesto, it is based on agile concepts and adheres to the values set forth in that declaration.

Scrum aims to maintain good collaboration between individuals working on complicated projects, and it is constantly changing and adding details. It is founded on the methodical interactions of three key roles: Scrum Master, Product Owner, and Team.

A project's scrum master: Who pivotal player. His primary role is to remove any hurdles that may obstruct the team's ability to function effectively.

The product owner: who is usually a customer or other stakeholder, is actively involved throughout the project, communicating the product's overall vision, and providing timely feedback on the job done after each Sprint.

The scrum team: It is a cross-functional and self-organizing group of people who oversee product implementation. To remain flexible and productive, it should include up to seven team members **(Figure 3)**.



Sprints and artifacts: A sprint, or fundamental unit of work in scrum, is a short development cycle required to produce shippable product iteration. A sprint lasts anything from one to four weeks: Longer iterations lack the consistency and flexibility that are key to Scrum's success. Even though there is no standard time (if it is less than four weeks), all sprints within a project should be the same length. This makes planning and tracking progress much easy. The Product Backlog, the Sprint Backlog, and the Sprint Burndown Chart are the three key artifacts used by Scrum to manage requirements and track progress. A variety of recurrent meetings, such as the Daily Scrum (Standup), Sprint Planning, Review, and Retrospective sessions, help to codify the process.

The product backlog: It is a prioritized list of feature requests for the project's final product. It is a single point of contact for all requirements. As new needs, fixes, features, and details are amended or added, the product Backlog is updated.

The sprint backlog: It is a list of tasks that must be completed by the team to deliver a working increment of software at the end of each Sprint. To put it another way, team members agree on which product items to deliver and how they will do it.

The sprint burn down chart: It depicts the amount of work that remains in a Sprint. It benefits both the team and the Scrum Master because it indicates daily progress and may predict whether the Sprint target will be met on time.

Scrum meetings: A variety of recurrent meetings, such as the Daily Scrum (Standup), Sprint Planning, Review, and Retrospective sessions, help to codify the process (the Sprint Retrospective).

The daily scrum: It is a timed meeting in which a Development Team coordinates its work and establishes a plan for the following 24 hours. The event should last 15 minutes and be held every day at the same time and place.

At sprint planning, the work that needs to be done is planned. This event is attended by everyone involved in the Sprint (a Product Owner, a Scrum Master, and a Development Team). They provide answers to two important questions: what work can be done and how it will be done. For a one-month Sprint, Sprint Planning takes

no more than eight hours. The meeting usually takes less time for shorter Sprints.

At Sprint Planning, the work that needs to be done is planned. This event is attended by everyone involved in the Sprint (a Product Owner, a Scrum Master, and a Development Team). They provide answers to two important questions: what work can be done and how it will be done. For a one-month Sprint, Sprint Planning takes no more than eight hours. The meeting usually takes less time for shorter Sprints.

The team and the product owner meet at the Sprint Review at the end of each Sprint. The team displays completed work and answers questions regarding the product increment during this casual discussion. All the participants work together to figure out what they should do next to improve the product's value. For one-month Sprints, the Sprint Review is a four-hour time-boxed meeting.

Retrospective meetings are held for the entire team to reflect on their work during the Sprint. Participants examine what went well and what went wrong, identify ways to improve, and devise a strategy for putting these good changes into action. After the review and before the following Sprint Planning, the Sprint Retrospective is held. For one-month Sprints, the event lasts three hours.

When to use scrum: Scrum is a good fit for long-term, complicated projects that require stakeholder feedback, which can have a significant impact on project requirements. Scrum may be the best option when the exact amount of work cannot be determined, and the release date is not set.

Scrum has earned the trust of 89 percent of agile users by prioritizing customer needs and on-time/on-budget delivery. As a result, the number of businesses that have adopted this strategy is astonishing. Microsoft, IBM, Yahoo, and Google are among the companies that have contributed to a public spreadsheet. According to the Scrum Alliance's newest research, Scrum has applications outside of IT. Companies in the banking, consulting, and entertainment industries use this technique to manage their work processes and improve customer collaboration. Most respondents to the 2016 State of Scrum Report (98 percent) said they would use this framework to move forward.

Kanban: Comprehensive solution to handling work

Kanban is another popular project management framework. Kanbanas is one of the project management frameworks used by 43% of firms. Kanban is a simple, yet powerful, technique to producing software products that evolved from a visual system of cards used in Toyota manufacturing as a production control method. Kanban, which is Japanese for "visible signal," focuses on the visualization of workflow and prioritizes Work In Progress (WIP), restricting its scope to properly match it to the team's capacity [10]. When a task is finished, the team can move on to the next item in the pipeline. As a result, the development process allows for more planning flexibility, quicker turnaround, defined objectives, and openness.

Kanban board: In contrast to Scrum, Kanban does not require established procedures inside the process or fixed iterations. The workflow is visualized using a Kanban board, which is often represented by sticky notes and whiteboards, or online platforms such as trello (Figure 4).

BACKLO	BACKLOG		IN PROGRESS		COMPLETED	
Figure 4	Agile metho	project odologies.	management:	Best	practices	and

Trello makes Kanban more automated and digital. Because each Kanban board contains concise information about a work item, everyone on the team understands who is accountable for the item, what each person's responsibility is, when it is scheduled to be completed, and so on. To give more information, team members can post comments, send screenshots, documents, or links. Kanban teams collaborate in a cooperative manner. The ability to track progress allows coworkers to better comprehend each other's contributions to the common goal, resulting in a greater concentration on finishing the assignment well and on time.

Lean development: The notion of "just in time production" underpins this software development strategy. The goal of Lean software development is to speed up the process while lowering costs. The following are the seven steps that summarize Lean development:

1. Remove unneeded items (anything that does not add value to the customer's project is removed)

2. Development of high quality (producing high quality in development necessitates discipline and management of the quantity of residuals produced)

3. Create Knowledge (the team is encouraged to document the entire infrastructure to preserve its value in the future)

4. Differentiated commitments (this point encourages the team to focus less on planning and anticipating ideas without first having a thorough understanding of the business's requirements)

5. Fast delivery (as soon as feasible, deliver value to the consumer)

6. Respecting the team (two key criteria are communicating and managing disagreements)

7. Optimize the entire process (the development sequence must be perfected enough to be able to delete errors in the code, to create a flow of true value).

Extreme programming: Extreme Programming (XP) is a software development process that prioritizes collaboration,

communication, and feedback. It emphasizes continuous improvement and client happiness. This strategy, like scrum, makes use of sprints, or short development cycles. This is created by a group of people to produce a productive and efficient workplace.

In a circumstance where clients have frequent and fluctuating expectations, the Extreme Programming technique comes in handy. It encourages developers to accept modifications in customer requirements, even if they arise late in the development process.

In Extreme Programming, the project is thoroughly tested from the start by obtaining feedback that enhances the system's output. This also serves as a spot check to ensure that any customer demands are met quickly.

Crystal: The Crystal methodology is a group of minor agile approaches that includes Crystal Clear, Crystal Yellow, Crystal Red, and others. Every approach has its own set of guidelines. Alistair Cockburn, who was involved in the creation of the Agile manifesto for software development, introduced it.

Team size, system seriousness, and project preferences are three aspects that characterize project traits. Projects are classified into four stages based on system criticality, which are: Comfort (C), Discretionary money (D), Essential money (E), and Life (L). The largest number of individuals can be involved in a project depending on its scale; the larger the project, the more people participating. Many jobs can be accounted for if the project is huge, and vice versa. Furthermore, interaction, people, expertise, communication, talents, and community are all united in crystal approach.

Dynamic Systems Development Method (DSDM): The DSDM is an iterative and incremental strategy based on the Rapid Application Development (RAD) methodology. The technique provides a fourphase framework:

- Feasibility and business study
- Functional model / prototype iteration
- Design and build iteration
- Implementation

Direct and co-located cooperation between developers and users is the greatest way for projects to evolve.

Teams that are self-managed and empowered must be able to make time-sensitive and essential project-level choices. The process of design and development is iterative and gradual, and it is mostly driven by constant, iterative user feedback. Working software deliverables are systems that fulfill vital present business needs against less critical future needs. The release of workable software on a regular basis is appreciated over the delivery of perfect software on a rare basis. Throughout the project lifecycle, continuous integration and quality assurance testing is performed in real time. Regular communication and collaboration among all project stakeholders promote visibility and transparency. All modifications made during the development process must be reversible.

Discussion

With this concise study, we have been able to highlight the following challenges and recommendations that best sort this work out. To note that People are more important than processes, and mindset is more important than skillset: In contrast to waterfall implementations, where processes are more crucial, the relevance of an individual contributor or a team member grows multi-fold throughout testing cycles for agile projects. This necessitates the careful recruitment of individuals who possess not just the necessary talents, but also the necessary mindset. The key for success is 'motivated individuals who are proactive, have great communication skills, and the capacity to think beyond the box'.

Make documentation important

The fast-paced nature of testing in an agile organization benefits from strong communication channels at all levels. Focusing on adequate documentation, on the other hand, may assist speed up the testing process, efficiently handle quality concerns through change management, decrease the time spent on resource onboarding, and assess the whole project life cycle. Multiple contributors cooperate to handle the demand for documentation, including dedicated technical writers, functional specialists from testing teams, project managers, and analysts. Agile teams should be given standards and criteria for the amount of documentation and coding comments they must include [11].

Conclusion

Because software engineering is such a fast-paced industry, it necessitates flexibility and reactivity in all aspects of project development. Agile frameworks enable the delivery of cuttingedge products and the cultivation of creative experiences while maintaining product alignment with market trends and customer requirements. The efficiency of agile software development teams is critical to the success of any software project. An agile team is made up of numerous components, all of which are linked to human input when it comes to software development. These elements of agile teams are garnering greater attention in the context of productivity, and hence warrant additional investigation.

References

- 1 Selvaraja JN (2015) Online Coworking Spaces Management & Booking System. Doctoral Dissertation.
- Santos JM (2021) Agile vs. Waterfall: Differences in Software Development Methodologies. Aust Inst Proj Mngmt 10: 24-29.
- 3 Buric A (2013) Agile modeling. Inform It: the Trusted Technology Source for it Pros and Developers. Harv Bus Rev 12: 54-55.
- 4 Eby K (2019) Demystifying the 5 phases of project management. Luettavissa Tech Netw Bos. 2: 18-20.

American Journal of Computer Science and Information Technology

Vol. 9 No. 10: 115

- 5 Melo C, Kon F, Conradi R, Cruzes D (2014) Productivity of agile teams: an empirical evaluation of factors and monitoring processes. In Anais do XXVII Concurso de Teses e Dissertações SBC Open Lib 28: 25-30.
- 6 Singh V (2021) "What is sd-wan, how does it work and why do you need it?" 15: 777-779.
- 7 Fatema I, Sakib K (2018) Using qualitative system dynamics in the development of an agile teamwork productivity model. Int J Adv Softw Semantic Schol 11: 170-185.
- 8 Murali M. (2021) 4 Values and 12 Principles of the Agile Manifesto. Agile Alliance 12: 89-95.

- 9 Asproni G (2004) Motivation, Teamwork, and Agile Development. Agile Times. Springer Book 4: 8-15.
- 10 Osterwalder ,Pajares J, Poza D, Villafañez F, Paredes AL (2017) Project management methodologies in the fourth technological revolution. In Advances in Management Engineering Springer Professional Book 17: 121-144.
- 11 Kassab M, Laplante P, Defranco J, Neto VV, Destefanis G (2021) Exploring the Profiles of Software Testing Jobs in the United States. Int Busin Mach. 9: 905-916.