IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, December 2026
Impact Factor: 7.67

A Comprehensive Comparative Study of Waterfall and Agile Development Methodologies

Osaki Miller Thom-Manuel[1] and Jethro Chinedu Ejah[2]
Lecturer, Department of Software Engineering, Ignatius Ajuru University of Education, Port Harcourt, Nigeria[1]
Lecturer, Department of Software Engineering, Ignatius Ajuru University of Education, Port Harcourt, Nigeria [2]

Abstract: *Software development methodologies play a crucial role in shaping how software systems are conceptualized, engineered, tested, and delivered. Over the years, the Waterfall model and Agile methodologies have emerged as two dominant paradigms within the field. While Waterfall follows a linear, sequential structure, Agile emphasizes iteration, customer collaboration, and adaptability. This study provides a comprehensive comparative analysis of both methodologies by examining their historical evolution, theoretical foundations, practical applications, strengths, limitations, and appropriate use cases. Through an extensive review of literature and methodological comparison, the study highlights how each paradigm influences project complexity, team collaboration, risk management, delivery timelines, and product quality. Findings reveal that Waterfall remains effective for predictable, well-defined projects, whereas Agile excels in dynamic, rapidly changing environments. The study concludes with recommendations on selecting suitable methodologies based on project scope, organizational culture, and stakeholder needs*.

Keywords*:* Waterfall model, Agile methodologies

I. INTRODUCTION

Software development methodologies are structured approaches designed to guide the planning, execution, and management of software engineering projects. As software systems grow increasingly complex, selecting an appropriate methodology has become essential for ensuring timely delivery, efficient resource utilization, and high-quality outcomes. Among the numerous methodologies that have emerged, Waterfall and Agile remain the most widely studied and applied across industries **[6].**

Waterfall, rooted in traditional engineering practices, emphasizes a strict, sequential flow of activities. Agile, on the other hand, evolved as a response to the limitations of linear models, promoting flexibility, collaboration, and iterative progress. This paper provides a comparative examination of both paradigms to determine their suitability in modern software engineering contexts.

II. BACKGROUND AND HISTORICAL EVOLUTION

2.1 Evolution of Software Development Methodologies

Early software engineering practices were heavily influenced by manufacturing and construction disciplines, where linear processes dominated. This led to the emergence of the Waterfall model in the 1970s, formalized by Winston [20] [3]. Although Royce originally described Waterfall as risky without iteration, the model gained popularity due to the fact that it is structured and follows a linear order in its executing processes. Also, its processes are easily documentable in nature [8]. Thus, it boost communication, helps in maintenance updates, aids knowledge sharing among developers and many more [26]. The basic phases of a waterfall model include Requirement Analysis, System Design, Implementation, Testing and Deployment. These steps remain a common pattern for new emerging methodologies which apply these steps in variant ways which may include iterative cycles.

By the 1990s, however, rapid technological change and increasing software complexity exposed the limitations in rigid, plan-driven models which most often resulted in late software product delivery, budget overrun, incomplete requirement specifications, production of software with wrong specifications and sometimes outright failed projects.

DOI: 10.48175/IJARSCT-31150

This led to the emergence of lightweight methodologies which are iterative. This culminated in the Agile Manifesto 2001, developed by a group of software practitioners seeking a more adaptive and human-centered approach to software development [2]. Among the 17-person team who made up the group called Agile alliance, Ken Schwaber and Jeff Sutherland officially defined the Agile Project management and were considered the founding Fathers of the Agile project management methodology [16].

III. LITERATURE REVIEW

[1] **[14] [9] [8] [11] [21] [15] [28] studies, made comparison of Waterfall method and Agile Methods of development. [1]** in their study identified benefits Agile has over Waterfall specifically in the aspects of project period records, cost efficiency, defect rates on projects' performance rates, and customer satisfaction. Microsoft was used as a case study. In [9], comparison of the two methods were made using key factors such as adaptability, stakeholder involvement, risk management, and project complexity and the strength and limitations of the methods highlighted. [8] compared the two methodologies stating their difference in the level of flexibility, documentation, and management of stakeholders' interactions.

[11] in their study investigated and compared the use of waterfall and agile methodologies in the software development process. Their result revealed that agile produces the best results compared to other techniques for big software project which is contrary to the study result of some other researchers like [30], [29] reports that Traditional plan-driven methods remain effective for large, complex infrastructure projects requiring extensive pre-planning. Depending on data collected, results can vary. [15], in their research of the evolution and integration of Agile into the United States Department of Defense (DoD), found that despite the popularity and move by many towards the shift from the use of the Traditional waterfall method of development, the DoD continued to utilize the waterfall methods.

[7] and [14] studies compared waterfall, Agile and the Spiral methods of development instead. However, this study is not intended to compare results of others but to further buttress the fact on the suitability of these two methods for particular projects so developers can make better decision on which method to use during development. Research on software methodologies consistently identify a tension between predictability and adaptability. Several studies highlight that Waterfall is effective for projects with clearly defined requirements and minimal changes [8], [18], [24], [9]. Its structured documentation is beneficial for large organizations, regulated industries, and mission-critical projects [30] . Conversely, Agile is frequently praised for its responsiveness to change, customer engagement, and iterative development cycles [10][4][9]. Studies show that Agile teams often deliver higher customer satisfaction due to continuous feedback loops and early delivery of functional components [27].

Comparative research suggests that organizations choose methodologies based on project complexity, team maturity, risk tolerance, adaptability and stakeholder expectations [9]. Hybrid approaches combining Waterfall's structure with Agile's adaptability have also gained grip in large-scale organizations [9].

IV. THEORETICAL FRAMEWORK

This study is grounded in two key theoretical perspectives:

4.1 The Plan-Driven Theory

The Plan-Driven Theory of project management was not developed by an individual but emanated from the use of several management principles of science and engineering disciplines during the development of large-scale projects, during the mid-20th century. The Plan-driven models assume that comprehensive planning, thorough documentation, and sequential execution enhance predictability and reduce risk. Waterfall is the clearest representation of this theory, emphasizing stability and upfront requirement clarity.

4.2 The Empirical Process Theory

Agile methodologies rely on empiricism [12], the belief that knowledge evolves through experience and iterative inspection. Scrum, for example, is built on transparency, inspection, and adaptation. This theory underpins the Agile assumption that requirements evolve and cannot be fully captured at the project's outset [18]. Instead, teams learn through experience, making decisions based on observed results rather than predictions. This iterative approach fosters continuous improvement, enabling organizations to deliver value effectively and respond to change [23].

V. METHODOLOGY

This study employed a qualitative comparative research design supported by an extensive literature review from academic journals, industry reports, and software engineering textbooks. The analysis focuses on Process structure, Requirements handling, Risk management approaches, Team collaboration, Flexibility, Delivery patterns and suitable environments. The objective is to highlight the distinguishing characteristics of Waterfall and Agile and evaluate their performance across different dimensions.

VI. WATERFALL DEVELOPMENT METHODOLOGY

6.1 Overview

Waterfall is a sequential model where progress flows downward through predetermined phases. Each phase must be completed before the next begins.
as shown in figure 1.

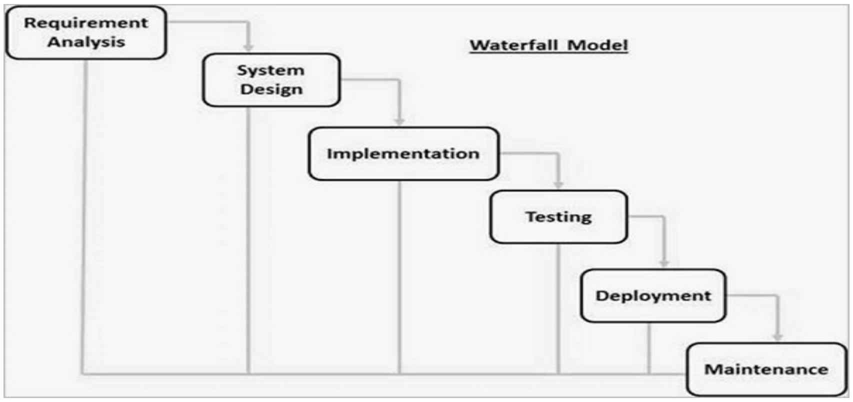

Figure 1: Waterfall Model

6.2 STRENGTHS

Clear documentation: In healthcare industries where compliance with regulatory standards is non-negotiable, extensive documentation plays a very important role. This each step of the waterfall model in product design and implementation like in the production of MRI machines is well followed and documented , meeting the stringent standards of regulatory bodies such as the FDA. These documents provide a clear development roadmap as well as serve as audit trails during compliance inspections [20].

High process visibility: The Waterfall model due to its strictly sequential, well-planned, and heavily documented nature, provides high process visibility. Consequently, project managers and stakeholders have a clear understanding of the project's progress against the initial plan and can easily track milestones and deliverable.

Suitable for stable environments: A stable environment implies clear and fixed requirements which are unlikely to change. The waterfall model by its linear approach to Software Development flows steadily downward through a series of distinct non-overlapping phases, just like a physical waterfall. Each phase must be fully completed and reviewed before the next one can begin, making it a document-driven process with clear and well defined milestones and extensive documentation.

Effective for safety-critical and regulated systems: The linear process sequence of the Waterfall model provides the structure, rigorous documentation, and clear milestones crucial for safety-critical projects in fields like aerospace or defense, thus ensuring compliance and risk control through upfront planning and testing.

6.3 Limitations

The Waterfall model is also faced with the following challenges namely: Inflexibility to changing requirements, late testing which may delay detection of defects, Long delivery cycles and delayed feedback due to late customer feedback after implementation.

VII. AGILE DEVELOPMENT METHODOLOGY

7.1 Overview

]Agile promotes iterative development through small, frequent increments [28[. Popular frameworks include Scrum, Kanban, and Extreme Programming (XP) **[14]**. Agile values Customer collaboration. Adaptive planning, Continuous delivery and Cross-functional teamwork. Figure 2 shows a generalized Agile model. It shows the most followed steps in a typical Agile model.

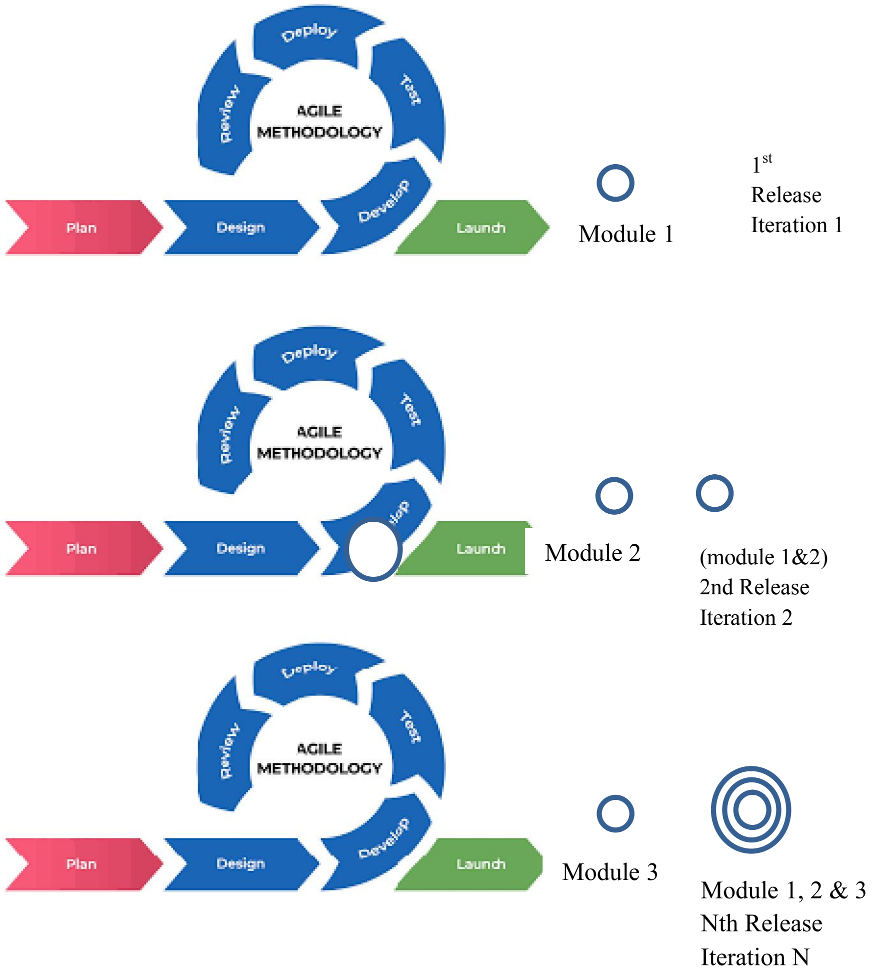

Figure 2: Genralized Agile Software Methodology Model

7.2 Strengths

- **Highly flexible and adaptive:** Agile processes welcome changes at any time of the development cycle, thus it is a flexible approach that can adapt to change.

- **Customer Collaboration:** Stakeholders and customers are involved throughout the process, providing continuous feedback.
- **Flexibility to Change:** Agile processes welcome changes even late in development to provide a competitive advantage.
- Working Software over Documentation: The primary measure of progress is functional software or a working product increment, rather than extensive documentation.
- **Self-organizing Teams**: Teams are cross-functional and empowered to manage their own work and make decisions autonomously, which enhances responsiveness. Each Team consist of members with all the necessary skills namely developers, testers, designers, analysts are in each team.
- **Early and continuous customer feedback:** Agile development delivers working software in short cycles (iterations/sprints) to get frequent input, thus ensuring that the product aligns with needs, reduces risk, and boosts satisfaction, contrasting sharply with traditional models where feedback comes too late, making changes costly.

7.3 Limitations

Documentation may be lighter: Agile emphasis on minimal documentation as a result vital project management elements like risk management are not explicitly carried out. Agile approaches' emphasis on minimal documentation, which causes them to give less attention to various elements of project management, notably risk management. Addressing the project risks is one of the key components of any project, therefore, risk management is particularly recognized as a significant project management task. Each strategy establishes its unique procedure, customs, functions, and artifacts. The implementation of technique, team efficiency and effectiveness, handling risks, and human-related difficulties are only a few of the problems and obstacles that must be overcomed in the transition to agile methodologies.

Requires experienced, self-organizing teams: This means that team members must be self-managing. They must have the freedom to decide best methods to utilize for carrying out their tasks instead of taking commands from outsiders. This also implies if the expertise is not right, mistakes will be made and in case any member leaves within short notice or dies suddenly, there is likely to be problem in the development cycle in meeting up with producing the right product at the right time.

Scope creep risk if poorly managed: Because of the evolving requirements throughout development, unending desire to make changes may result in continuous changes.

Less effective in heavily regulated environments: In other to maintain swiftness in the development process, Agile do light documentation. Thus, ordinarily, It is difficult to do comprehensive documentation in Agile Software development methodologies.

VIII. COMPARATIVE ANALYSIS

Requirements Management:

Waterfall:

In the Waterfall methodology, requirements are gathered, analyzed, and fully documented at the very beginning of the project. This phase assumes that stakeholders can clearly articulate all system needs upfront and that these requirements will remain stable throughout development. Once approved, changes to requirements are discouraged because they often require revisiting completed phases, leading to increased cost and schedule delays. This approach works well in environments where requirements are fixed, such as government or safety-critical systems [5] [24].

Agile:

Agile treats requirements as evolving rather than fixed. Instead of extensive upfront documentation, requirements are captured as user stories and refined continuously throughout the project. This allows development teams to respond quickly to new information, customer feedback, or market changes. Agile assumes that it is unrealistic to fully define requirements at the start of complex software projects, making flexibility a core strength [4]; [10].

8.2 Flexibility

Waterfall:

Flexibility in Waterfall is limited because the methodology follows a rigid, linear structure. Once a phase is completed, revisiting it is costly and time-consuming. Any change introduced late in the development cycle may require redesigning, recoding, and retesting the system. As a result, Waterfall is less suitable for projects operating in rapidly changing environments (Royce, 1970; [24].

Agile:

Agile is inherently flexible and embraces change, even late in development. Iterative cycles (sprints) allow teams to incorporate new requirements without disrupting the entire project. This adaptability enables organizations to stay competitive and align software products with evolving customer needs and technological trends [10]; [27].

8.3 Risk Management

Waterfall:

In Waterfall projects, risks—especially technical and requirement-related risks—are often discovered late because testing and user validation occur near the end of the development cycle. If critical issues emerge during testing, they can be expensive and difficult to fix. This delayed risk exposure makes Waterfall vulnerable in complex or uncertain projects [5].

Agile:

Agile mitigates risk through short development cycles and frequent testing. Each iteration delivers a working product increment, allowing teams to identify and address risks early. Continuous integration, regular reviews, and customer feedback ensure that problems are detected sooner, reducing the likelihood of project failure[4]; [27].

8.4 Customer Involvement

Waterfall:

Customer involvement in Waterfall is typically limited to the requirements-gathering phase and final acceptance testing. Once development begins, customers have minimal interaction with the project until delivery. This can result in a final product that technically meets specifications but fails to fully satisfy user expectations [23]; [24].

Agile:

Agile places strong emphasis on continuous customer collaboration. Customers or product owners actively participate throughout the project by reviewing increments, prioritizing features, and providing feedback. This ongoing involvement ensures that the product aligns closely with user needs and increases overall customer satisfaction [4]; [10].

8.5 Delivery Approach

Waterfall:

Waterfall follows a single-release delivery model where the complete system is delivered only after all development and testing phases are finished. While this ensures that the entire product is fully integrated before release, it also means that stakeholders must wait a long time before seeing any tangible results [24].

Agile:

Agile adopts incremental and continuous delivery. Functional components are released at the end of each iteration, allowing stakeholders to use, evaluate, and benefit from the software early. This approach improves time-to-market and enables faster return on investment [10]; [27].

8.6 Project Suitability

Waterfall:

Waterfall is best suited for projects with stable requirements, well-understood technologies, and strict regulatory or documentation needs. Examples include defense systems, medical software, and large government projects where compliance and predictability are critical [5]; [24].

DOI: 10.48175/IJARSCT-31150

Agile:

Agile is ideal for projects characterized by uncertainty, frequent change, and innovation. It is widely used in startups, mobile applications, web platforms, and customer-driven products. Agile's adaptability makes it effective for environments where user feedback and rapid delivery are essential [10]; [28]

IX. FINDINGS AND DISCUSSION

The analysis reveal that Waterfall and Agile are fundamentally different in structure, philosophy, and delivery approach. Waterfall excels in environments demanding predictability, thorough documentation, and compliance. Agile, however, thrives in dynamic sectors such as software startups, mobile app development, and web-based systems where rapid iteration and user feedback are crucial.

Research shows that Agile projects often achieve higher customer satisfaction and shorter delivery cycles, while Waterfall provides stronger quality assurance through documentation and structured verification. Hybrid approaches have become popular, allowing organizations to maintain structure while adopting Agile's responsiveness.

X. CONCLUSION

Waterfall and Agile remain indispensable in software engineering, each offering unique advantages depending on the project environment. Waterfall's sequential nature supports stability and rigorous documentation, making it ideal for predictable projects. Agile's iterative and collaborative approach delivers adaptability and customer satisfaction in fast-changing environments. The choice between methodologies should be guided by project requirements, organizational culture, regulatory constraints, and the need for adaptability.

XI. RECOMMENDATIONS

Consequently, it is recommended that Waterfall model be utilized for the development of Projects with stable, well-defined requirements, Regulatory or safety-critical systems projects and Organizations requiring extensive documentation. While Agile is suited for projects with incomplete requirements at the start of the development, projects that will require changes to requirements at any phase of the development, Customer-centric projects and when development Teams are self-organizing and capable of collaborating.

For the development of mega enterprise software, software for mixed-regulation environment and long term software products requiring both stability and adaptability, the Hybrid Approach consisting of a blend of Waterfall and Agile method is utilized.

REFERENCES

[1] A. A. Rana and A. Omar , Comparative Analysis of Waterfall and Agile Methodologies in Microsoft. International Journal of Academic Information Systems Research (IJAISR). Vol. 8. pp 8-13 2024.

[2] Agile Alliance, 2025, A Short History of Agile. https://agilealliance.org/a-short-history-of-agile/ 2025.12.27

[3] A. A. Nakhuda. A Comparative Analysis Of Waterfall Vs. Agile Methodologies In Software Project Management. International Research Journal of Modernization in Engineering Technology and Science, Vol. 6, 28 Jun 2024.

[4] Beck, K., & Andres, C. Extreme Programming Explained: Embrace Change (2nd ed.). Addison-Wesley. 2005.

[5] Boehm, B. W., Software Engineering Economics. Prentice Hall , 1981.

[6] C S. Maharao, A Study on Impact of Agile and Waterfall Methodologies on Projects in IT Industry. European Economic Letters 14(3). http://eelet.org.uk 2024.

[7] D.B.J. Kalpana Rao, D.B.J., Kalpana, P., Kumar, G.S.N., Atcha, N.M., A Comparative Analysis of Software Development Models: Waterfall, Agile and DevOps. In: Reddy, V.S., Prasad, V.K., Wang, J., Rao Dasari, N.M. (eds) Intelligent Systems and Sustainable Computing. ICISSC 2024. Smart Innovation, Systems and Technologies, vol 417. Springer, Singapore. https://doi.org/10.1007/978-981-97-8355-7_51, 2025

[8] A . Ebule, Agile Vs. Waterfall: A Comprehensive Analysis of Methodologies for Effective Project Management. International Journal Of Novel Research And Development (www.ijnrd.org) Vol. 10, 2025.

[9] G. Sirisha, V. Sarada, V., E. P. John, H. Bhadrappa, V. B. RamaKrishna and D. Bisen, Project Management

DOI: 10.48175/IJARSCT-31150

Methodologies: A comparative analysis of agile and waterfall approaches. SSRN Electronic Journal. 44(3), pp 17237-17246. Available on https://doi.org/10.2139/ssrn.5000321, 2024a.

[10] Highsmith, J. (2002). Agile Software Development Ecosystems. Addison-Wesley.

[11] G. Fawareh, Y. Al-Smadi, R. Saadeh, F. A. Fawareh, A. Elrashidi and H. M. Al-Shdaifat, "A Comparative Study between Agile and Waterfall Methodologies during Software Development Process," 2024 25th International Arab Conference on Information Technology (ACIT), Zarqa, JInternational Journal of Academic Information Systems Research (IJAISR) Vol. 2 , ISSN: 2643-9026www.ijeais.org/ijaisr

[12] J . Fair , Agile versus Waterfall: approach is right for my ERP project? Paper presented at PMI® Global Congress 2012—EMEA, Marsailles, France. Newtown Square, PA: Project Management Institute, 2012.

[13] J . Rohm (June 14, 2025). The 10 most popular Agile and Scrum methodologies with examples. Available on https://echometerapp.com/en/agile-methodologies-andrameworks/#:~:text=Scrum:%20A% 20popular %20agile%20framework,structured%20approach%20to%20scaling%20Agile. 14th June, 2025.

[14] K. K. Manohar , R. F. Pravin , D. R. Shilpa, A. L. Sharayu, P. D. (2022). A Comparative Study of Software Development Waterfall, Spiral and Agile Methodology. Journal of positive school of psycology 6(3).

[15] K. Taha, G. Harun, M. Jeremy, A. Christopher (2024). Comparative Analysis of Agile versus Waterfall Methodology in Government Utilities, Projects, and Services: A Case Study Exploration. International Journal of Social Relevance & Concern (IJSRC) Vol. 12, 2024.

[16] L. Quick, History of Agile Methodology: How it was Developed. Available on https://www.know ledgehut.com/ blog/agile/history-of-agile

[17] Mindmap, Understanding the Empirical Process in Agile.Available on https://mindmapai.app/mind-mapping/empiricalprocess#:~:text=The%20Empirical%20Process%2C%20fundamental%20to,effectively%20and%20respond%20to%20change., Oct 19, 2024.

[18] M, Ogunbukola, Agile vs. Waterfall Methodologies in Public Sector Projects: A Comparative Analysis. October, 2024.

[19] P. Abrahamsson, O. Salo, J. Ronkainen and J. Warsta (2017). Agile Software Development Methods: Review and Analysis. https://arxiv.org/abs/1709.08439, 2017.

[20] P M. K. Sai, Is Waterfall Project Management the Best Choice for Structured Product Development?
Available on https://www.nimblework.com/bytes/waterfall-project-management/, December 2, 2024.

[21] P. Mohamamadi, Agile vs. Waterfall: Choose the Best Development Methodology. Available on https://www.manifest.ly/use-cases/realtors/, October 07, 2024.

[22] R. P. Pawar, A Comparative study of Agile Software Development Methodology and traditional waterfall model.IOSR Journal of Computer Engineering (IOSR-JCE)

[23] Royce, W. W., Managing the development of large software systems. Proceedings of IEEE WESCON, 1–9. , 1970.

[24] SCRUMstudy (June 11, 2024).Scrum methodology empirical process control. Available on https://www.scrumstudy.com/article/scrum-methodology-empirical-process-control

[25] Sommerville, I. (2016). Software Engineering (10th ed.). Pearson.

[26] S. Shaikh, and S. Abro (2020). Comparison of Traditional & Agile Software Development Methodology: A Short Survey. International Journal of Computer Systems & Software Engineering, Vol. 5, pp 1–14.14, 2020.

[26] S. V. Chomal, Vikas S. and Jatinderkumar R. Saini. "Significance of Software Documentation in Software Development Process." S.International Journal of Engineering Innovation & Research. Vol 3, issue 4. July, pp. 410-416, 2014.

[27] VersionOne. "14th Annual State of Agile Report." VersionOne, 28 May 2020.

[28] M. A. Khan & M. Al-Shammari Exploring Agile applicability in infrastructure projects: A critical analysis. International Journal of Managing Projects in Business, 16(5), pp 1083–1103, 2023.

[29] K. O. Ogirri and I. J. Idugie, A Comparative Analysis of Traditional versus Agile Project Management Methodologies on IT Project Outcomes", Vol. 17, August, 2024.

[30] L. Abdallah, B. Boumali, G Fernandes and S. S. Boudemagh, Identifying the most used traditional project management practices in construction industry''. Procedia Computer Science.