iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** XI **Month of publication:** November 2024

DOI: https://doi.org/10.22214/ijraset.2024.65371

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue XI Nov 2024- Available at www.ijraset.com

Mastering Change: The Journey from Waterfall to Agile and Hybrid Project Management

Manju Raman
Synergy Technologies LLC, Jacksonville, Florida

Abstract: ***The author explores the project management methodologies transitioning from Waterfall to Agile to Hybrid in this article. The author analyses each methodology to document the suitability, advantages, and disadvantages of Waterfall and Agile methodologies. Waterfall has a very constricted approach, with the activities happening in a sequence and with no deviation allowed from them. The stakeholders play a vital role in the early phases when providing the requirements, and they are only involved during the testing phase. This methodology demands extensive documentation of the business case, project plan, requirements, product design, etc., and at the same time, lacks flexibility in adapting to changes in requirements during later phases of the project. Agile, on the other hand, advocates an iterative approach to development. With the close collaboration of the stakeholders or the customers, the requirements are documented by the Product Owner, and any questions or clarifications the developers need are immediately available from the customers/stakeholders. Stakeholders working in close contact with the team result in high demands and changing requirements from them and can tax the team. Agile does not favour extensive documentation, and thus, it can adversely affect the team if they are working on a long-running project with new team members joining the team. Both methodologies have their advantages, which were incorporated into the Hybrid methods and are expected to overcome the challenges faced by the teams when using either.***

Keywords: ***Project Management, Agile methodology, Scrum, Agile, Software, Hybrid Methodology***

I. INTRODUCTION

Before the advent of Agile, traditional project management followed a linear process where activities needed to be completed sequentially. Each predecessor activity had to be finished before any dependent tasks could begin. According to the Project Management Body Of Knowledge PMBOK [16], the project life cycle comprises several phases: Feasibility (validating the business case and assessing organizational capability), Design (planning and analysis leading to deliverable design), Build (constructing the deliverable), Test (quality assurance and remediation), Deploy (putting the deliverable to use), and Close (formally concluding the project and updating knowledge bases).

This traditional Project Management approach mandated that a review be conducted after the completion of each phase. A strict timeline was created, and a strict scope was set in traditional methodology before the development began; this allowed for a very restricted change in requirements or flexibility on the part of the development team. In contrast, the Agile Software Development Manifesto emerged as a response to the need for more adaptable, lightweight frameworks. Representatives from methodologies like Extreme Programming, Scrum, and Crystal convened to advocate for an approach less focused on documentation and more on collaboration and responsiveness to change.

Agile project management was influenced by a 1986 article by Hirotaka Takeuchi and Ikujiro Nonaka in the Harvard Business Review titled "The New Product Development Game." This work used rugby as a metaphor to highlight the benefits of self-organizing teams in innovative product development. Agile adoption by organizations requires a cultural transformation, emphasizing teamwork, increased collaboration, and iterative progress tracking rather than a one-size-fits-all management strategy.

In a project management landscape where strict rules dictate requirements management, Agile's iterative and incremental approach allows for adaptability—enabling teams to respond to customer requests throughout the development cycle.

II. OBJECTIVES

The objective of this review is to:

1) Review and document areas where both Agile and traditional methodologies succeed in delivering value.
2) Review and identify challenges associated with both Agile and traditional methods.
3) Analyse how hybrid methodology blends both traditional and Agile approaches.

©IJRASET: All Rights are Reserved

4) Review and document how the hybrid approach helps enhance project success rates.

III. LITERATURE REVIEW

This article aims to review publications documenting the pros and cons of traditional Waterfall and Agile methodologies. The Hybrid approach proposes to overcome these challenges by taking the pros of both methods and applying them to project or product management.

A. *Waterfall Methodology*

Dr. Winston W. Royce, in his paper "Managing the development of large software systems: concepts and techniques," talks about the phased approach to computer program development [3]. He states that regardless of size, every program would have two phases – Analysis and Coding. He proceeds to elaborate that on large-scale development projects, additional phases may be required – System Requirements, Software Requirements, Analysis, Program Design, Coding, Testing, and Operations. He also emphasizes the need for extensive documentation: "The first rule of managing software development is ruthless enforcement of documentation requirements." [3]

Project Management methodology, as documented in PMBOK, further elaborates on a project's phases. It insists on a linear approach, where the preceding phase is completed before starting the next phase. Traditional project management methodology continues to emphasize extensive documentation.

1) *Phases in Traditional Methodology*

a) Feasibility: This phase involves analysis or studies to determine whether the planned project undertaking is viable and will deliver value to the organization and its customers.
b) Design: This is the phase where planning and analysis are conducted. Stakeholders are identified, and requirements are elicited from them. The analysis results are utilized to generate a design for the product.
c) Build/Development: The deliverable is worked on in this phase. If there are multiple deliverables, they may overlap with the Test and Deploy phases.
d) Test: This is the phase where the deliverables from the build/development phase are evaluated thoroughly against the requirements and design.
e) Deploy: This phase follows the testing phase and deploys the output from the testing phase to Production for customer use.
f) Close: This phase follows the deployment phase. When the final deliverable is received, lessons learned are documented, and resource releases are conducted in this phase.

2) *Roles in Traditional Methodology*

Below are the Roles in Traditional Methodology -

a) Project Manager (PM): The PM is responsible for the overall planning, execution, and delivery of the project. This role involves managing resources, developing schedules, ensuring that the team adheres to the schedule, tracking the project budget, and meeting project goals. The PM also monitors risks, helps resolve any issues the team may face, communicates project progress to the stakeholders, and manages cross-functional activities.[11]
b) Business Analyst: The Business Analyst is a liaison between the stakeholders and the Project team. The business analyst elicits requirements from stakeholders and translates them into project specifications.
c) Team Lead: The team lead directs the work of a specific team (such as development or testing) and ensures team alignment with project goals. They oversee the daily operations within the team and ensure that the team receives any help or guidance it needs, whether technical or logistical. The team lead communicates project progress, risks, etc., to the Project Manager.
d) Developers: The Developers are responsible for the actual coding and technical implementation of the project goal. They are accountable for conducting unit testing and debugging the code developed. They work with the Quality Assurance team to ensure that the final product meets the quality standards.[12]
e) Testers/Quality Assurance (QA): The QA team creates the test plan, conducts different types of testing, and identifies any defects within the system. They focus on validating and verifying that the product meets the required standards and specifications.

f) Stakeholders: Stakeholders are individuals or groups of individuals from the business who have a vested interest in the project's outcome and provide valuable input and feedback.[14] They play a significant role in critical project decisions and approvals, especially during the planning and review stages, to ensure that the project aligns with business goals and provides the desired outcomes.

B. Agile Methodology

You cannot speak of Agile without discussing the Agile manifesto and the 12 principles of Agile. These values and principles are the building blocks of agile methodology. Below are the Agile Manifesto and Principles as defined in the Scrum Guide [17]:

Agile Manifesto 4 Values:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

The principle behind the Agile Manifesto:

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference for the shorter timescale.
- Businesspeople and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity--the art of maximizing the amount of work not done--is essential.
- The best architecture, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.

1) Roles in Agile Methodology: Not all Agile methodologies define specific roles.

Among those that do, Scrum, Extreme Programming (XP), and Feature-Driven Development (FDD) have clearly outlined roles. In contrast, Kanban and Lean Programming do not propose specific roles.

The Scaled Agile Framework (SAFe) takes a different approach [18]. It incorporates Agile, Lean, and DevOps principles to extend Agile practices across large organizations. SAFe also includes defined roles for team members to support scalability and alignment.

Agile roles were first formally introduced within the Scrum framework, developed by Jeff Sutherland and Ken Schwaber in the early 1990s. The structure and responsibilities of these roles were detailed in the Scrum Guide, initially published in 2010, and later updated in 2020. According to the Scrum Guide, the primary Scrum roles and their responsibilities include:

a) Development Team: The people in the Scrum Team are committed to creating any aspect of a usable increment for each sprint. They are accountable for planning the activities for the Sprint, adhering to the Definition of Done, working towards the print goal, adapting the plan when necessary, and maintaining professional accountability to each other. [2]

b) Product Owner: The product owner is accountable for maximizing the value of the product resulting from the work being conducted by the Scrum Team. The product owner is responsible for establishing and conveying the Product Goal, creating, and organizing Product Backlog items, prioritizing them, and ensuring the Product Backlog is transparent and easily understood. [13]

c) Scrum master: Is accountable for the effectiveness of the Scrum Team, and they do this by enabling the Scrum Team to improve its practices within the Scrum framework.

They support the Scrum Team by coaching members to self-manage and develop cross-functionality, helping them focus on delivering high-value increments, removing impediments as and when they arise, and ensuring Scrum events are practical and time-boxed. They assist the Product Owner in defining the Product Goal, managing the Product Backlog, and facilitating stakeholder collaboration. Additionally, the Scrum Master supports the organization by guiding Scrum adoption, advising implementations, encouraging an empirical approach to complex work, and facilitating communication between stakeholders and Scrum Teams.

2) *Events and Artifacts in Agile Methodology:* In addition to defining roles, Agile specifies specific ways of working, referred to as "Ceremonies." The most crucial of these ceremonies is the **Sprint**, which encompasses all the other ceremonies. Sprint is an iteration during which the development team completes the work, evaluates it, and creates a **Minimum Viable Product (MVP)** for stakeholders to review. Sprints typically last between 2 to 4 weeks. all Agile methodologies define specific roles.

Sprint Planning is the ceremony where the development team selects the work to be completed during the Sprint to deliver a functioning component of the final product, which is then added to the Sprint Backlog. The work is drawn from the Product Backlog, a prioritized list of user stories and requirements maintained by the Product Owner. The Product Owner is responsible for ensuring that the Product Backlog contains the necessary user stories for the development team to work on. It is also advised that the Product backlog must have at least 2-3 Sprints worth of user stories, if not more [2].

Once the Sprint Backlog is defined, the Sprint begins. The development team, along with the Scrum Master, meets daily for a Daily Scrum. This is a time-boxed, 15-minute event where each team member shares progress updates, identifies any blockers, and outlines the work they plan to complete that day. The Daily Scrum is strictly for sharing brief updates and is not intended to resolve issues.

Throughout the Sprint, the Quality Assurance (QA) team or testers validate the completed work against the Definition of done. The definition of done is nothing but the acceptance criteria outlined in the user stories, which need to be met for work to be considered done.

At the end of the Sprint, the Sprint Review takes place, during which the development team displays the completed work to stakeholders. The Sprint Review allows the stakeholders to assess progress toward the project goal and raise concerns about the component or MVP delivered.

Finally, the Sprint Retrospective is one of the most vital Agile events. This ceremony allows the development team to reflect on how the Sprint was conducted, identify areas for improvement, and discuss changes to implement in the next Sprint. It provides a crucial opportunity for continuous improvement and improved processes.

C. Highly Effective Features Of Traditional Methodology

The traditional waterfall methodology has been used for a long time for project management because of its advantages.

a) Structured and Predictable: The Waterfall methodology follows a sequential and linear approach, where each phase must be completed before moving on to the next.[15] This structured nature makes the development process predictable, enabling precise planning, well-defined milestones, and a straightforward project timeline.

b) Extensive Documentation: Waterfall emphasizes detailed documentation at each stage of development. This documentation ensures a thorough understanding of project requirements, design specifications, and implementation guidelines. It also serves as a valuable reference for future maintenance and updates to the application/system.

c) Easy Management: With fixed requirements, planned timelines, and well-defined phases, Project Managers can efficiently allocate resources, assign tasks, and track the project's progress.

d) Clear Deliverables: The Waterfall methodology emphasizes the production of tangible and well-defined deliverables at the end of each phase. For example, at the end of the planning phase, there would be clearly defined requirements, a project plan, a system design, a project schedule, etc. This enables stakeholders and clients to understand what to expect when each phase ends clearly, fostering transparency and effective communication.

Suitable for Small Projects: The literature reviewed noted that waterfall suits smaller projects with well-defined requirements. Its straightforward approach can be beneficial in cases where changes are less likely to occur during the development process.

D. Highly Effective Components And Practices Of Agile Methodology

Agile prioritizes customer satisfaction by welcoming changes throughout development to maintain high satisfaction levels. It focuses on delivering functional components frequently, allowing customers to see incremental progress with each sprint.

Agile promotes close collaboration between the development and business teams or customers, fostering a supportive and trusting environment that motivates the development team. It values face-to-face communication as the most effective method for collaboration.

Additionally, Agile encourages sustainable development with self-organizing teams committed to excellence and quality. Teams are prompted to reflect on their work after each iteration, enabling them to adjust their practices for improved effectiveness.

1) Iterative Development and Incremental Delivery: Agile teams develop products in short cycles (iterations). This allows for incremental delivery of functional software. Each iteration builds upon the last, delivering small, usable pieces of the product that stakeholders can review and provide feedback on.
2) Continuous Feedback and Adaptation: Continuous feedback from stakeholders, product owners, and end-users is vital to Agile methodology. This feedback helps teams adjust priorities, refine requirements, and improve iteratively. This also reduces the risk of misalignment with user needs.
3) Cross-Functional Teams and Collaboration: Agile teams consist of cross-functional team members with varying skill sets (e.g., developers, testers, product owners, and designers). Cross-functional teams improve problem-solving capabilities, enhance knowledge sharing, and foster a shared understanding of project goals.
4) Customer Involvement: Agile methodologies promote the direct involvement of customers and stakeholders in the development process. Regular feedback from customers and stakeholders ensures that the product meets evolving expectations and thus maximizes customer satisfaction.
5) User Stories and Prioritization (Backlog Management): Agile teams work on user stories (from the backlog), which are concise, user-centric descriptions of the requirements. Backlog refinement and prioritization allow teams to focus on delivering high-value features first, aligning development efforts with business priorities.
6) Daily Stand-Ups: These are short, daily meetings that help team members communicate progress, challenges, and next steps. These promote transparency, facilitate quick problem-solving, and keep everyone aligned.
7) Retrospectives and Continuous Improvement: At the end of each iteration/sprint, teams conduct retrospectives to reflect on what went well and identify areas for improvement. This practice promotes a culture of continuous improvement and helps the team refine its process over time.
8) Test-Driven Development (TDD) and Continuous Testing: Agile encourages early and frequent testing, with practices like TDD helping ensure that features work as expected. Continuous testing helps identify issues early and supports the quality and reliability of each increment.
9) Automation and Continuous Integration (CI): Automated testing, CI, and sometimes Continuous Deployment (CD) help Agile teams integrate changes frequently, ensuring the software is always deployable. This reduces integration risk and speeds up delivery by identifying issues early in development.
10) Empowered and Self-Organizing Teams: Agile teams are encouraged to self-organize and take ownership of their work. This fosters creativity, motivation, and accountability because the team members have greater control and responsibility.
11) Agile Frameworks and Ceremonies (e.g., Scrum, Kanban): Frameworks like Scrum and Kanban provide structure to Agile practices, including ceremonies like Sprint Planning, Sprint Review, and Sprint Retrospective. These frameworks help teams manage workloads, track progress, and communicate consistently.
12) Focus on Value and Minimum Viable Product (MVP): Agile emphasizes delivering value early and often. It encourages teams to create a minimally viable product that addresses users' core needs at the end of each iteration. This approach allows teams to evaluate ideas in the real world, gather feedback, and iterate quickly to incorporate changes or fix defects.
13) Transparency through Visual Management (e.g., Kanban Boards): Visual tools like Kanban boards, burndown charts, and information radiators provide transparency, making progress, bottlenecks, and workload visible to the entire team and stakeholders.

Agile has made significant strides, with several organizations and industries adopting Agile methodologies for project management and product development. Below are statistics from Notta.ai highlighting the adoption rates of various Agile frameworks [1].

a) The Scaled Agile Framework (SAFe) is the most likely choice for any enterprise among the various Agile frameworks, but only 26% of respondents stated they use SAFe. This marks a 50% decrease from the previous year.
b) 19% stated that they use Scrum@Scale or Scrum of Scrums, whereas 4% stated they use Enterprise Scrum.
c) In terms of the use of team-level Agile tools, 62% of companies use Atlassian Jira, 32% use Mural/Miro, and 25% use either Microsoft Excel or Microsoft Project.

d) According to the 16th Annual State of Agile Report, almost 9 out of 10 respondents stated that they are using Scrum, whereas about half are using Kanban.

e) The same report found that Kanban is the most closely followed Agile framework at 25% and ties with Scrum at 25%, While a Hybrid/Scrumban framework also stands at 25%.

f) Among the various Agile techniques and practices, 49% of marketers stated that they use Sprint/Iteration planning, 45% use Digital Kanban board, 44% use Daily standup, and 39% use user stories and/or epics.

In terms of the use of Agile marketing, the impact of various factors that influenced the decision is that 64% found that Agile marketing is more important in the competitive landscape, while 30% stated that it had no impact.

E. Challenges And Disadvantages Of Traditional Methodology

The Waterfall methodology offers a structured and predictable approach to software development, making it suitable for small projects with well-defined requirements. Its emphasis on documentation and clear deliverables contributes to effective project management. However, its lack of flexibility, limited customer involvement, and high-risk factors can be problematic, particularly in complex and dynamic development environments. Below are the disadvantages or challenges with the traditional/Waterfall methodology:

1) Rigid Nature: The Waterfall methodology's rigid and sequential nature is a significant drawback when changes or unforeseen circumstances arise during the project. Once a phase is completed, modifications to the requirements or design are impossible without disrupting the entire process.[7]

2) Customer Involvement is limited to the initial phase: Customer involvement is during the requirements gathering phase, which is in the early phases of project management. There is a lack of continuous feedback from the customer, making it challenging to accommodate changing customer needs and preferences during the project's later phases.

3) Increased Risk: Testing and validation occur towards the end of the development process in Waterfall, which contributes to an increased risk of identifying critical issues later in the project lifecycle. Rework at this stage can be very costly and time-consuming. [7]

4) Lengthy Development Cycle: Traditional methodology's sequential nature means the entire project life cycle might be lengthy. If a specific phase takes longer, it contributes to the overall delay.

5) Not Suitable for Complex Projects: Waterfall is not a practical methodology for complex projects with complex and evolving requirements. The inability to adapt to changing requirements leads to unsatisfied customers and unsatisfactory outcomes.

F. Challenges And Disadvantages Of Agile Methodology

Even though Agile adoption is increasing, there are certain disadvantages to it. The below is a consolidation of the reasons why Agile Methods fail:

1) Insufficient experience with agile methods: Agile looks simple but requires significant experience in many social and managerial disciplines to succeed in its usage. This is what most project management practitioners fail to understand. Regarding skills and experience of the team, Agile is rated high compared to low knowledge of the traditional methods. [9]

2) Little understanding of the broader organizational change needed: Adopting Agile requires a complete shift from and sometimes the curtailment of the status quo, and this understanding is not too clear to managers [9]

3) Philosophy or culture of the company in contradiction with Agile principles: Any form of an organization's culture contradicting the Agile culture will not augur well for the thriving of Agile projects. Regarding company culture, Agile thrives better in a low management control environment than in a high management control environment for traditional project management methods [2].

4) Iterative delivery leads to a limited vision of the complete product: Developers cannot quantify the full extent of required efforts for some software deliverables. This is principally true for more oversized products at the start of the development life cycle, eventually leading to frustrations.

5) More Time and Dedication: Testers, clients, and developers must communicate with each other continuously. This includes countless face-to-face discussions because they are the best way to communicate. Everyone engaged in the project must cooperate closely. Daily consumers need to be accessible at each stage for timely testing and signoff so designers can mark it off as complete before moving on to the following function. This may guarantee that the item meets the customer's expectations, but it is burdensome and time-consuming.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

This requires more time and energy from all involved. An agile project requires close collaboration and extensive team involvement, which demands more time and energy to meet the set requirements. Though each phase engages, the entire process demands a significant commitment to ensure a favorable result. Any lack of engagement can impact product quality and success.[5]

6) Active stakeholder involvement and changing requirements: According to Flora, Harleen K., and Swati V. Chande in their article "A systematic study on agile software development methodologies and practices." - Customer interaction is rewarding and taxing on the team because of the increased commitment and time needed to keep the customer engaged [10]. Customer engagement at the iterative level also results in a change of scope and requirements. Any lack of customer engagement impacts the deliverables and shifts the timelines.

7) Skilled Development Team: Agile encourages developers to take accountability for their work, which requires excellence on the part of the development team. The team must be able to collaborate and plan the sprint goals. They are responsible for delivering a high-quality product iteration at the end of each sprint. If there is a lack of collaboration, experience, or skill within the development team, it adversely affects progress and team morale [XXX].

8) Minimum to No documentation: Agile emphasizes not spending extensive time on documentation. It suggests that the teams can spend this time instead of collaborating directly with the customers on their requirements. All the requirements are documented as part of user stories, which do not contain much detail [8]. As the project or product evolves, the number of user stories also evolves. Someone who has been part of the project since its inception will have extensive knowledge based on experience, and someone who has joined the team newly will have to dig through the heap of user stories to understand the features and characteristics of the application [5]. If organizations lack the quality of their user stories, it furthers the documentation problem.

9) High dependency on Customer engagement: Agile methodology highly promotes the need for collaboration between the agile team and the customer. When the customer is fully engaged, the requirements are constantly fed as user stories and are consumed by the development team to build the incremental product. If customer engagement is lacking, it results in no requirements, and the team works on unrelated efforts. Over-engaged customers who constantly give and change requirements are also a significant challenge. [5]

G. *The Hybrid Approach*

It was surprising that Dr. Royce highlighted the iterative approach to software development in his article. He emphasized that software coding should not commence until the design is finalized, stating, “An acceptable written description forces the designer to take an unequivocal position and provide tangible evidence of completion.” He also advocated for customer interaction during the design phase, asserting that “Each designer must communicate with the interfacing designers, with his management, and possibly with the customer.” Dr. Royce believed that program design should be the responsibility of Program Designers rather than analysts. This phase should yield a System Overview document and a database design, among other things.

According to Janine, Jeff Reiff, and Dennis Schlegal, in the publication “Hybrid project management – a systematic literature review,” combining hybrid approaches is possible. These are a mix of specific characteristics of the Traditional and Agile methodologies [6]. They are detailed below.

1) *Water-Scrum-Fall:* According to the article "Water-scrum-fall is the reality of agile for most organizations today.", this methodology has three high-level phases.

The first phase is the “Water” phase, where the below activities are performed as in the “Waterfall Methodology” before starting the development work:

Requirements gathering – The detailed requirements are gathered and documented,

Analysis –Feasibility analysis is conducted on the requirements gathered and an initial design is developed,

Planning – Detailed Project plans are developed; timelines are documented, and resource allocation are performed.

The “Scrum” phase is the development phase where the actual development activities take place. The activities in this phase are performed iteratively for multiple sprints. The below activities are performed in this phase.

Sprint planning – Sprint goals are defined, and the Sprint backlog is populated with items to be worked on in the Sprint

Implementation (Development & Testing) – Here, the actual development activities are performed along with testing of the components developed in each sprint

Daily Scrum – This is the timeboxes event where the team gathers to review the work being performed by the team, and any blockers or issues are notified.

Sprint Review – This is performed at the end of the Sprint to demonstrate the component developed in the Sprint to the Stakeholders. It allows the stakeholders to identify and suggest any critical changes required.

Sprint Retrospective — This is where the team reviews the work done in the prior sprint and finds areas for improvement so that they can make the necessary changes in their work style and perform better in the oncoming sprints.

The "Fall" phase is the last phase of this methodology, during which the product is released for use by the customer/stakeholder. [19].

Final Testing – In this stage, comprehensive testing is performed, including Integration testing, end-to-end testing, and user acceptance testing.

Release preparation – In this stage, preparations are made to release the product for use. Complete user manuals, technical documentation, operational guides, and release notes for the product are prepared. Compliance and security checks are performed to ensure that the final product meets regulatory, security, and industry standards. Deployment strategy, rollback plans, and preparation of the environments for the release are also done in this stage. All the necessary signoffs are obtained in this stage.

Deployment – In this stage, the product is deployed to production for use and tested as soon as it goes live.

Post-Deployment Maintenance – After the product goes live, it is monitored, and any issues identified are resolved immediately. Feedback is collected from the users to track any problems or concerns from the end-users. Ongoing support and maintenance of the product for the long term are planned and set up.

2) *Waterfall-Agile:* According to this methodology, the Requirements and Planning are performed in the "Waterfall" phase. In the "Agile" phase, the Design, Development, Testing, and Implementation are done. [6]
   In this methodology, Waterfall and Agile do not overlap and are distinct. It is predominantly Waterfall, and the Agile methodology is employed for the iterative development approach. The below diagram represents the steps in this methodology. [20]
3) *Hybrid V-Model:* This model includes the elements of the V-model (Verification and Validation-focused approach) and the iterative approaches of Agile. This method primarily employs the strict phases of the waterfall, where each phase needs to be completed before the next. This methodology allows for an iterative approach to developing certain features or modules. It emphasizes thorough testing to be conducted early and often. [4]
4) *Agile-Stage-Gate:* This method involves splitting the work into multiple stages and conducting a stage-gate review at the end of each stage. The work is performed in an Agile fashion within the stages. [6] Below diagram represents the scrum sprints being conducted within each project phase, and at the end of each phase a stage-gate review is conducted for receiving feedback[21]

All these methods suggest that the project activities be split into three phases.

- Initial Phase,
- Development phase and
- Final phase.

Initial Phase: All these hybrid methods suggest that the Requirements, planning, and Scope definition must be done in the Initial Phase. This allows for the vision and scope of the project to be defined, which enables the Scrum team to have a clearer idea of the direction in which the work is progressing.

Development Phase: All these methods recommend that the Development phase, which includes the Design, development, Unit testing, and implementation activities, stay agile. This allows for iterative development, early detection of issues, and the ability to incorporate minor changes to the entire project during the development phase.

Final Phase: According to all these methods, the final phase should include the Integration, Testing, and launch of the product. In the last phase, all the components would be ready, and an integration would allow for a complete product to be available for testing, bug fixes, and launch.

Waterfall methodology was criticized for its linear approach, the time it took to deliver complex projects, and its limited flexibility to change. Customer feedback was also delayed, and if a change was required, it had to go through the entire linear flow again. Documentation was extensively performed in the waterfall methodology, which, even though it was cumbersome, added value to stakeholders, project managers, business analysts, the development team, and the production support team. Agile did not favor extensive documentation and the requirements were maintained as user stories.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

The iterative development, test, and change cycle was highly effective since the Stakeholders had a chance to review the product components early on and recommend any changes if required. Hybrid overcomes the cons of both methodologies by recommending requirements documentation and encourages Agile iteration to be employed for development and to incorporate change. This makes the hybrid method ideal for projects that need a structured framework but also need to adapt to change.

IV. CONCLUSIONS

Both Traditional and Agile methodologies have their advantages and disadvantages. For specific projects, either of them can work without any alterations made, but for some, they may not. The time it took from inception to delivery with the waterfall methodology was massive. The stakeholders or customers had a tough time when requesting changes in requirements; they went through change management processes, which were time-consuming. There was no direct interaction between the business and the development team, which resulted in the development team envisioning the design based on the documented requirements. In a world where technology is fast changing, customers' needs would also change frequently, and Waterfall was just not suited for that.

Even though agile also enabled collaboration between the customers and the development team, it also added constant changes in requirements. The iterative approach of agile ensures a high quality of products and limited risks or failures. But, with the lack of documentation, the entire project relied heavily on the development team's knowledge. Any change to the requirements, if implemented by a new team member, took time and added risk due to the lack of documentation they could refer to.

Hybrid methodology incorporates the best of both worlds and emphasizes the need to complete the design before beginning coding/development. Dr. Royce stated in his article, “The documentation is the specification and is the design.” We have agreed with him regarding the need for documentation in hybrid methodology. The design is iteratively collaborated with the customer; thus, it is final and cannot change once the design is complete. The development team then uses this design to produce incremental product iterations. The incremental components are thoroughly evaluated and fixed before the next iteration. The final integration happens after all the elements are ready and integration testing is performed. This approach will result in a viable product without requiring several design changes to be incorporated in the development phase, enabling the developers to have creative freedom

REFERENCES

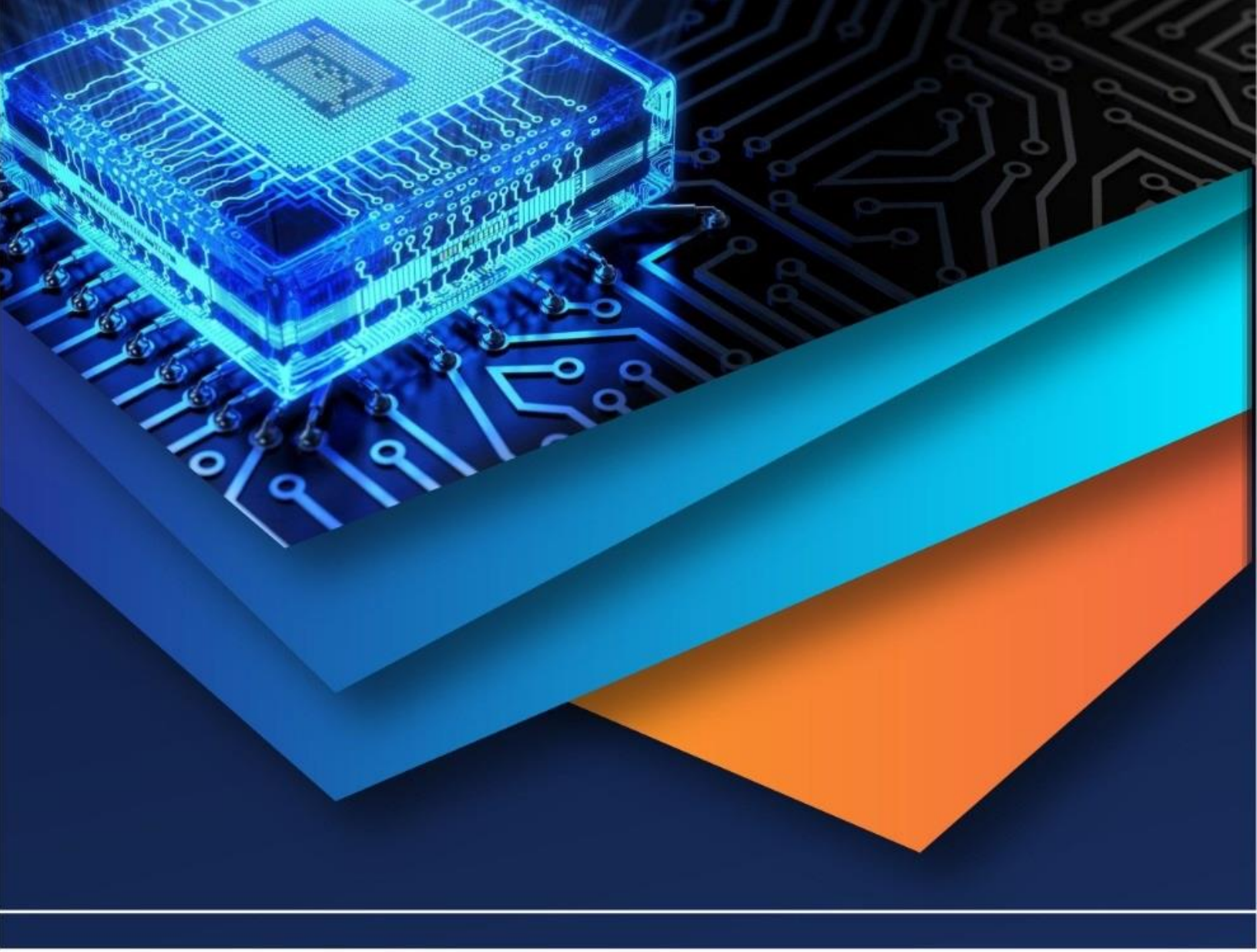
[1] Mahajan, V. (2024). 50+ Agile statistics you need to know in 2024. Notta.ai. Retrieved from https://www.notta.ai/en/blog/agile-statistics#agile-adoption-and-usage-statistics
[2] Koi-Akrofi, J., Matey, H. A., & Koi-Akrofi, G. (2019). Understanding the characteristics, benefits, and challenges of Agile IT project management: A literature-based perspective. International Journal of Software Engineering & Applications, 10(5), 25–44. https://doi.org/10.5121/ijsea.2019.10502
[3] Royce, W. W. (1987). Managing the development of large software systems: Concepts and techniques. In Proceedings of the 9th International Conference on Software Engineering (pp. 328–338).
[4] Koceska, N., & Koceski, S. (2022). Hybrid project management as a new form of project management. Journal of Applied Economics and Business, 10(4), 16–23.
[5] Singh, K. (2021). Agile methodology for product development: A conceptual study. International Journal of Recent Technology and Engineering, 10(5), 209–215. https://doi.org/10.35940/ijrte.A5899.0510121
[6] Reiff, J., & Schlegel, D. (2022). Hybrid project management – A systematic literature review. International Journal of Information Systems and Project Management, 10(2), Article 4.
[7] Pargaonkar, S. (2023). A comprehensive research analysis of software development life cycle (SDLC) Agile & Waterfall model advantages, disadvantages, and application suitability in software quality engineering. International Journal of Scientific and Research Publications, 13(8). https://doi.org/10.29322/IJSRP.13.08.2023.p14015
[8] Zhezherau, A. (2024, August). What are the disadvantages of Agile? Wrike. Retrieved from https://www.wrike.com/agile-guide/faq/disadvantages-of-agile/
[9] Miller, G. J. (2013). Agile problems, challenges, & failures. Project Management Institute.
[10] Flora, H. K., & Chande, S. V. (2014). A systematic study on agile software development methodologies and practices. International Journal of Computer Science and Information Technologies, 5(3), 3626–3637.
[11] Celigent. What is the role of the scrum master and project manager? Retrieved from https://celigent.com/what-is-the-role-of-the-scrum-master-and-project-manager/
[12] Moradikouchi, A. (2023). Terahertz sensing for pharmaceutical applications. Retrieved from https://core.ac.uk/download/561093213.pdf
[13] Yli-Pukka, R. (2022). Agile methods in hardware development. Retrieved from https://core.ac.uk/download/530246488.pdf
[14] Mathew, T., & Manocha, A. (2021). Author at Business Analyst Articles, Webinars, Templates, Jobs. Retrieved from https://www.batimes.com/author/thomson_mattyahoo-ca/
[15] Techgenies. Complete guide to software development life cycle (SDLC). Retrieved from https://techgenies.com/blog/software-development-life-cycle-sdlc/
[16] Project Management Institute. (2021). A guide to the project management body of knowledge (PMBOK Guide) and the standard for project management (7th ed.). Newtown Square, PA: Project Management Institute.
[17] Schwaber, K., & Sutherland, J. (2020). The Scrum Guide: The definitive guide to Scrum: The rules of the game. Scrum.org. Retrieved from https://scrumguides.org
[18] Scaled Agile, Inc. (2021). SAFe 5 for Lean Enterprises. Retrieved from https://scaledagileframework.com

[19] West, D., Gilpin, M., Grant, T., & Anderson, A. (2011). Water-scrum-fall is the reality of agile for most organizations today. Forrester Research, 26(2011), 1–17.

[20] Kuyeskar, M. (2023). Beyond traditional: Unlocking project success with hybrid project management. LinkedIn. Retrieved from https://www.linkedin.com/pulse/beyond-traditional-unlocking-project-success-hybrid-manali-kuyeskar-yfgye/

[21] Hansen, Camilla & Arlitt, R. & Eifler, Tobias & Deininger, M.. (2022). Design by Prototyping: Increasing Agility in Mechatronic Product Design through Prototyping Sprints. Proceedings of the Design Society. 2. 201-210. 10.1017/pds.2022.22.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)