Agile Methodology for Product Development: A Conceptual Study

Kajal Singh

Abstract: Agile is a software development life cycle (SDLC) methodology that is based on smarter and faster operating principles and techniques. The paper aims to review the agile processes, principles, characteristics, and frameworks for the continuous development of the product while controlling the integration, scope, schedule, cost, quality, resources, risk, procurement, and stakeholder management. Traditional product development methods are not much efficient to adjust the rapid changes of the requirements and provide quality products to the stakeholders. Using agile frameworks, the software development team, customers, and team leaders work more closely to produce high-quality products responding to the changes rapidly. This paper will also present the background analysis of the agile manifesto and will serve as a guide to indicate the characteristics and framework of Agile which places customer satisfaction at the highest priority. Additionally, it will focus on the advantages and disadvantages of the agile projects for faster and frequent delivery of the development.

Index Terms: Software Development Life Cycle (SDLC), Agile Methodology, Scrum, Kanban, eXtreme Programming (XP), Dynamic Systems Development Method (DSDM), Crystal.

Keywords: The Integration, Scope, Schedule, Cost, Quality, Resources, Risk, Procurement, and Stakeholder Management.

I. INTRODUCTION

The agile process is an iterative, incremental, or overlapping-based software development life cycle method, where requirements are changeable according to the customer requests. The process deals with the uncertainties in a positive manner that can occur throughout the development of a product [1]. There are several SDLC models like Waterfall, V-Shaped, RAD, Big Bang, Spiral, and one of them can be approved by the organization based on their project capability. However, these SDLC models share few common phases of product development and management such as Planning, Analysis, Design, Building, Testing, Deployment, and Maintenance which are performed as per the customer's expectations [2]. Agile is itself a software development process that follows an iterative methodology in which settling the business expectations is the highest priority with cost, quality, and schedule constraints. Being one of the SDLC models, Agile provides partially implemented software for customer validation. After incorporating the suggested modifications and getting acceptance from stakeholders, the software is fully implemented in the production environment [3].

Customers can be included again in the discussion about the production instance development for their reviews and additional comments. The development team assimilates the suggested changes following an iterative, incremental, overlapping, or hybrid technology to produce quality products. In this way, agility projects can lead the way to closure [4].

The clients are exclusively involved throughout the process providing continuous feedback and, therefore, cut down project risks at several phases of the process [5]. While traditional product development methodologies focus on the completion of the documentation of the KPIs and the final product, they do not give importance to customer reviews throughout the product generation [6]. Whereas, the agile principle is based on the core value of satisfying stakeholders through early and continuous delivery of valuable software while incorporating the suggested changes at any critical juncture [7]. The clients, developers, and team leaders work together constantly throughout the project having efficient and effective communication to get the job done. Continuous attention to technical excellence, quality tasks, requirement fulfillment, embracing modifications enhances agility [8].

II. BACKGROUND

Before agile principles, system configurations were carried out where the role of the stakeholders was minimal as they were only brought in for a quick acceptance testing. These systems, usually, did not meet their final requirements due to which developers worked on the tasks again and again creating a bigger mess every time. On the other side of the coin, users did not possess a good understanding of what developers used to do. As a result, they mostly put unrealistic demands on them. This entire process was chaotic for both stakeholders and the development team. Mostly, organizations focused on the end-to-end process of delivering software but unfortunately, their mismanaged IT departments did not see the overall picture. They followed fixed procedures for all the projects and swapped people in and out of the projects. In this type of paradigm, managements are the control of the domain while stakeholders hold negligible responsibility. Even developers abandon the process when the situation of the project gets any worse.

In 2001, the term 'Agile' was coined in a formal meeting where 17 people agreed upon 12 Principles that ended up becoming the manifesto for Agile Software Development. However, many organizations were already using agile methodology even before 2001 where practitioners were mixing old and new processes that worked for their project in different situations.

Agile Manifesto was formulated which was based on several principles and values. This led to the formation of the Agile Alliance which helped organizations and teams to adopt principles and practices mentioned in Agile Manifesto. Other people borrowed these ideas and tweaked them to make them appropriate for their context. Further, these values were utilized to engender several frameworks like Scrum, Kanban, and many more. The essence of twelve principles mentioned in the Agile Manifesto are:

- The highest priority will be to meet user's needs by continuous and early delivery of valuable product.
- Accepting change requests, even late in the development phase. Agile processes utilize principle values to incorporate the changing market exigency.
- Deliver working product fast and frequently within a short duration.
- Stakeholders and developers must coordinate together daily until the project is completed.
- Develop and manage projects around inspired individuals. Provide them the environment and aid they require and believe them to get the work done.
- The most efficient way of communicating in a development team is through face-to-face conversation.
- A working product is the dominant measure of progress.
  Agile principles encourage sustainable development. The developers, stakeholders, and users collaborate to work at a persistent pace perpetually.
- Constant attention to technical eminence and satisfactory design enhances agility.
- Simplicity is essential to carry out the procedures for product development.
- The best architectures, demands, and designs emerge from self-organizing groups.
- At the constant interim, the teams can conduct review meetings to be more efficient then modulate and adapt their operations accordingly
  Four main values are based on the above-mentioned agile manifesto principles. Table 1 shows the values of the agile project preferred over traditional principles. In an agile context, organizations and teams will give more importance to:
- Individuals and connections over procedures and techniques
- Operating Model over exhaustive documentation
- Association and Teamwork over agreement negotiation
- Adapting to changes over adherence to the plan

Table 1 Agile vs Traditional Approaches

Imperative	Submissive
Individuals and Connections	Procedures and Techniques
Operating Model	Exhaustive Documentation
Association and Teamwork	Agreement Negotiation
Adapting to Changes	Adherence to the plan

Using the newer model, agile projects are about 28% more successful than traditional projects. Using the traditional approach, the communication gap between the clients and developers is unsatisfactory. Further, teams are overburdened with unnecessary documentation. But new agile principles have provided continuous delivery of valuable products promoting sustainable development.

III. CHARACTERISTICS

Agile Methodology places customer satisfaction at the highest priority with faster development time as the process requires comparatively lesser time by dividing the main epic (main project) into smaller increments (Subtasks/ technical stories). This makes it possible to include new features or requirements through multiple iterations. Agile development methodology involves several phases which can be adjusted by the team based on their project scopes. These phases include Requirement Gathering, Review, Registered, Development, Validation, Prod Ready, Go Live, Feedback from Customer.
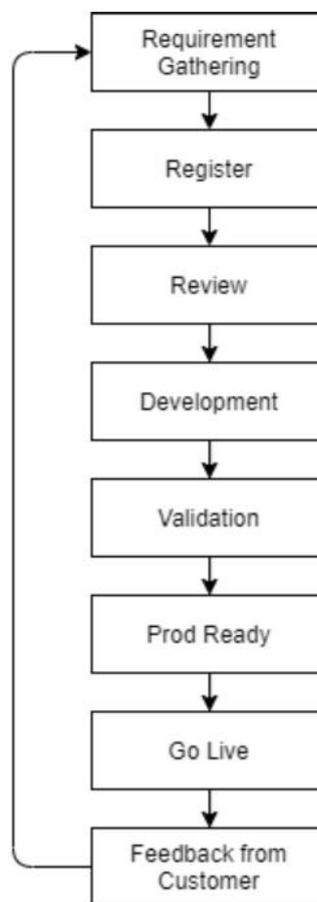

Fig. 1: Standard Phases of Agile Methodology

The Agile project can be established by analyzing the schedule, risks, resources, and other product management proceedings Under the register phase, team leaders and developers will take up the domain requirements based on the user's priority.

210

After this, it will move to the review phase where any team member can assign the task to themselves based on the bandwidth. After the initial summons, development work will be started. Upon completion, the development will be validated by the stakeholders before moving it to the production stage. Once validated, the requirement will be a part of Prod Ready state signifying tasks can be implemented on the production instance. After implementation, the requirement will be a part of the Go Live state. Finally, the developer will wait for feedback from the clients. If any changes are proposed, then these will be incorporated through multiple iterations. Following are the ten characteristics of carrying out a project with agility:

- *Iterative*

    The main objective of the agile software process is the satisfaction of customers, so it focuses on including all the suggested requirements even if multiple iterations are required.

- *Adaptive*

    A project might face several risk factors while incorporating new requirements. The agility nature of the project will adapt accordingly to produce efficient output.

- *Increment*

    When each successive version of a product is usable and builds upon the previous version by adding user-visible functionality, incremental development will assemble the superior product.

- *Collaborative*

    One of the unique features of agile is that it focuses on the people and how they work together in a software development team making good communication an important factor to integrate domains and modules at the end of the software development process.

- *Scope*

    In an agile context, scope management processes can be performed like define scope, create WBS (Work Breakdown Sheet), Structure control scope, validate scope, and are repeated for each iteration. This will generate an adaptive project lifecycle that will respond to the high level of change.

- *Schedule*

    Agile considerations for the schedule management include "Lifecycle Approach", "Resources Availability", "Project Dimensions", "Technology Support". In this way, the tasks and subtasks, which are part of an epic, are covered within short cycles which requires rapid feedback on the outcome.

- *Quality*

    Agile methods call for frequent quality and review checks throughout the project and iterations rather than just towards the end of the project. Recurring retrospectives continuously check on the effectiveness of the quality processes and suggests alternative approaches to improve quality.

- *Cost*

    Due to the frequent changes, lightweight cost estimates are generated to provide high-level forecasts of the project cost which can easily be adjusted when changes arise. Detailed estimates are provided for short-term planning horizons.

- *Risk*

    Frequent reviews of the incremental and additional tasks with the help of the cross-functional teams accelerate the knowledge sharing and make sure risks are managed and mitigated. It decreases the probability and impact of risks whereas increases the probability and impact of opportunities

- *People-Oriented*

    In an agile process, customer satisfaction is the priority over any technology and process. A software development team can increase the performance and capacity of the product depending on the effective communication with each other. Project with high variability benefits from self-organizing teams which focus on the collaborations.

IV. AGILE FRAMEWORKS

An agile framework follows a software development approach that is based on the twelve principles of the Agile Manifesto. Many organizations and agile teams use these frameworks to carry out their projects with agility. Eventually, they customize the framework suitable for their own unique needs. Following are the commonly used and well-registered frameworks for agile software development project:

A. Scrum

Scrum agile framework is an approach where projects are handled in a fixed, short, and time-boxed period known as "Sprint" which is usually of two weeks' time span. One of the key roles is played by Scrum Master who is responsible to organize and manage the daily stand-up call with the team. In the call, they can discuss their daily progress or bottlenecks related to the project. Team Leader or Technical Product Owner (TPO) communicates with the customers and stakeholders to maintain effective communication. In comparison to other principles and frameworks, it is easier to understand and allow for rapid feedback via reoccurring retrospections to ensure the effectiveness of a quality management plan. However, scrum methodology might create a higher extent of pressure on the team because of frequent and shorter deadlines. This framework might be an option if any team is looking for a lightweight, team-based, and Bottom-Up approach.
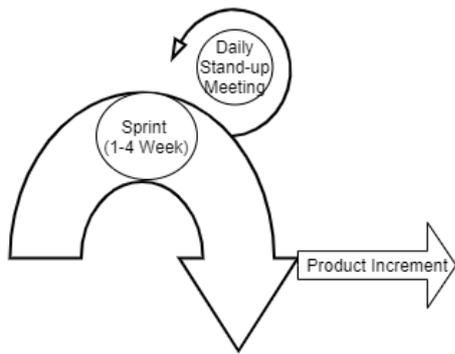
Fig. 2: Agile Product Development using Scrum

B.   Kanban

Kanban is an incremental agile methodology that virtually represents the stages of the Development Life Cycle to manage and control the flow of requirements. There is no sprint or fixed time-box limit within which the requirement must be completed. These items can start or end independently from each other and have no pre-determined fixed duration for that matter. Tasks can be moved from backlog to the Work In Progress (WIP) stage via register and review stages based on their priority. However, Kanban projects put a limit on the number of requirements which should be a part of WIP. When these tasks are completed and validated soundly by stakeholders only then new tasks are pulled into the cycle. Team Leaders organize meetings with the businesspeople to scrutinize the requirement details, their severity, fallback plans, risk owners, strategies to deal with opportunities and threats, and other important pointers. Later, these requirements are placed in backlog and are moved forward on the board based on their priority. Further, in the replenishment meetings, stakeholders or customers, team leaders, technical product owner, domain team talk through the progress of the tasks for schedule, cost, integration, risk and quality.
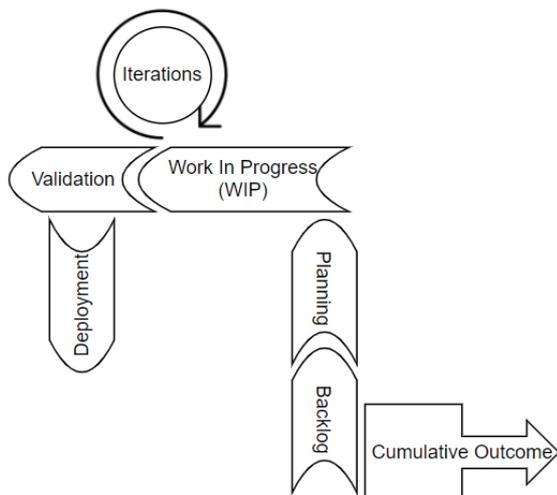

Fig. 3: Agile Development using Kanban

C.   eXtreme Programming (XP)

It is a popular agile framework that focuses on both providing

high-quality software to the customers and making the process

easier for engineers to develop. In this process, requirements

might change dynamically as customers are not able to articulate the requirements in a precise manner. These risks

are mitigated with the help of a relatively smaller group of developers working closely with managers, team leaders, and customers. In this way, it can help cut the loss for software development organizations. Also, the emphasis on improving the process for the developers helps to boost their morale, retention, and productivity. The main characteristics of XP include small releases, simple design, frequent testing, collective ownership, continuous integration, on-site customer, and coding standard.
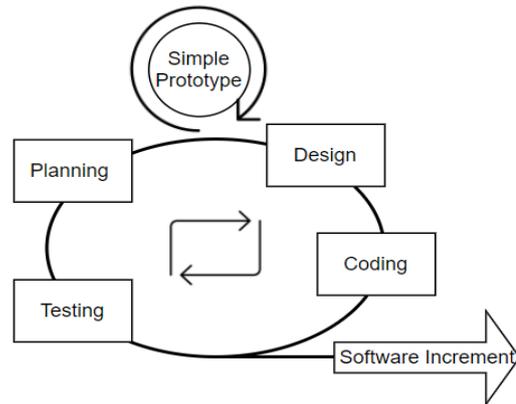

Fig. 4: Agile Development using eXtreme programming

D.   Dynamic Systems Development Method (DSDM)

DSDM is an agile method that focuses on the complete project lifecycle and delivers early benefits to the company with the help of clearly defined strategic goals and developments. The six principles of DSDM includes:

- Prioritize business requirements
- On-time delivery
- Collaborate
- Effective communication
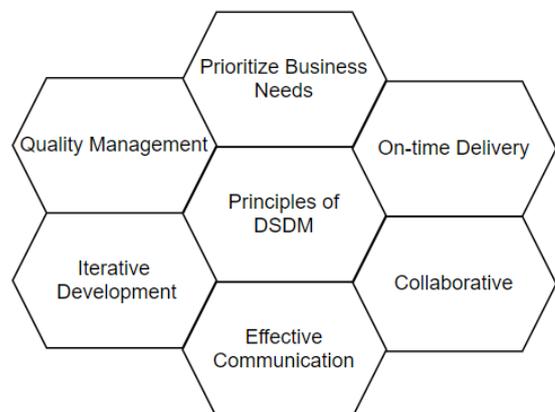- Iterative development
- Quality management

Fig. 5: Principles of DSDM Framework

This agile framework can address projects of all varieties and for any business domain. It champions the use of several proven practices including facilitated workshops and is a leading agile project management method. This was evolved based on Rapid Application Development (RAD) framework but with firm governance and stricter guidelines. Developers, customers, leaders are pro-actively involved in the project due to which basic business functionalities are delivered promptly. However, the DSDM framework is costly to implement and is not popular among small organizations.
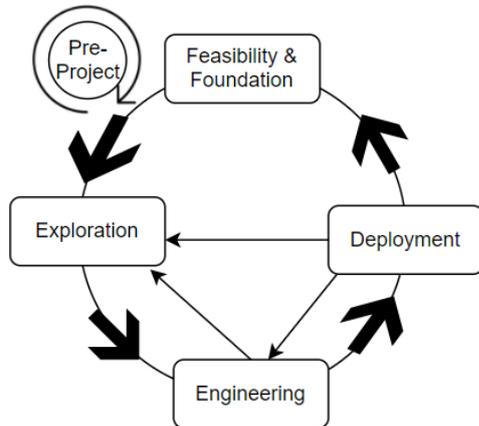

Fig. 6: Agile Development using DSDM Framework

E. Crystal

It is an agile framework that focuses on direct team collaboration and interactions instead of processes and tools. In short, it depends on teams to optimize and improve their workflow. Since most of the projects are unique and should be flexible as per the incoming requirements, teams are expected to be adaptive and modify the processes as per the ambiguous needs. An added benefit of this framework is that it is ultra-light. As a result, it does not require extensive reporting or documentation. There is complete transparency among the team members and are accountable for the workflow intimately. The adaptive approach of the Crystal framework assists the project to work in an effective way making it one of the most flexible frameworks. However, the lack of pre-determined plans and absence of any kind of documentation might make it onerous to track the team's progress.
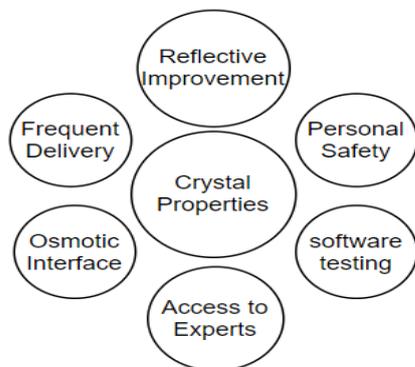

Fig. 6: Properties of Crystal framework

V. ADVANTAGES

Agile methodology has become one of the most popular approaches for product management. After the agile manifesto was declared in 2001, it has evolved and has become the most approved option among various organizations. The latest data shows that projects which utilize Agile principles are 28% more successful than traditional approaches-based projects and around 71% of the organizations use agile in their original or customized form. Following are some of the topmost reasons why companies adopt this methodology for managing their projects.

A. Product Quality

An agile framework can be iterative, incremental, or overlapping in nature due to which it can detect any bugs, enhancements, or adapt to the change requests at any stage of the process. Daily standup calls make it possible for the team to communicate among themselves to be cognizant of the project roadblocks. In the retrospection meetings, the team reviews the quality management for the workflow and suggests the way to incorporate improvements continuously. The flexible characteristic of agile allows the team to reach out to the customer for validation and accept modification requests constantly as the milestones list might change progressively depending on the market condition and business requirements.

B. Process Control

When any of the agile frameworks are followed, the stipulations are completed through different phases which generates a continuous workflow. This provides the project manager with the overall control of the process. Agile gives the clarity of the requirements while conducting continuous retrospections and ameliorates the quality, scope, schedule management for the team to work in the most efficient way possible. Agile teams get complete autonomy and are in charge of managing their task seeing that they are self-organized. Also, stakeholders get a frequent overview and the progress of the development tasks which gives them comparable supremacy.

C. Scope Management

Scope or domain-related processes like collecting requirements, defining scope, creating Work Breakdown Structure (WBS), controlling and validating scope are performed at the beginning of the process and are updated as per the requirement. This is possible because of the agility of the development lifecycle which will provide proper guidance and direction on how the scope will be managed throughout the project. Alternative analysis and decomposition are some of the tools and techniques which can be used to generate scope baseline and assumption log. Eventually, this will assist in the creation of a scope management plan.

D. Schedule Management

In an agile context, rolling wave planning will be followed in the adaptive lifecycle. This helps team members to create requirements in the user stories that are then prioritized accordingly.

213

Since each project is unique, the team leader can tailor the way the schedule is managed using several considerations like the lifecycle framework approach, resources availability, project dimensions, and technology support. The adaptive approaches use short and frequent cycles to perform the work, validate results, and adapt as necessary. These short cycles provide rapid feedback on the validation from stakeholders and suitability of the deliverables. This will generally manifest as iterative, on-demand, and pull-based scheduling.

- *Iterative Scheduling:* It is a procedure in which scheduling is performed iteratively until better results can be accomplished. Moreover, these iterations are stopped when no further improvement can be obtained in the task output.
- *On-Demand Scheduling:* In this kind, development is carried out only when there is a demand and availability of the necessary resources. For example, in the Kanban framework, the requirement can be pulled in the cycle on a demand basis from the backlog queue relying on the priority. It also keeps a check on the number of requirements in the work in progress phase so that the balance is maintained between the demand generation and the team's throughput.
- *Pull-Based Scheduling:* It follows the Just-In-Time (JIT) Approach. It pulls the task into the Work in Progress phase only when the requirement is raised to avoid the over-production and utilize the resources efficiently. Mostly, the lot sizes are reduced to assure the continuous flow of the resources and it exposes any expected risks or issues at the earlier stages of the production cycle.

E. Stakeholders Satisfaction

Mostly projects experience high level of change requests, iterations or incremental cycles which requires active engagement and participation with stakeholders. Agile frameworks allow regular interaction with client community which alleviates risks, strengthen trust and adjust new requirements early in the cycle which increases the probability of process success. The agile team gives a go-ahead to stakeholders to be involved in the constant feedback meetings, request generations and in all the crucial decisions which might affect the project flow. Constant consulting is carried out to manage the clients and match their needs. This allows the stakeholders to come up with new changes even at the late stages of the process. Team leaders can carry out retrospection meetings to capture their feedback. Consequently, team can work on the positive or negative value of the stakeholder engagement. Agile project team focuses on plenty of considerations for client satisfaction like stakeholder diversity, complexity of stakeholder relationship and communication technology.

F. Faster and Frequent Development Delivery

Agile frameworks follow customer centric approach which focuses on reviewing, developing, validating and releasing a continuous flow of valuable products and services to the customers on time. Mostly, semi-developed product is validated in advance with the clients to make sure the development is going in the right direction. It puts customer at the center of every crucial decision and reacts to

demands faster with agile tools and technologies. One of the technical workflows is Agile Release Train (ART). This continuous delivery pipeline consists of Continuous Exploration (CE), Continuous Integration (CI) and Continuous Deployment (CD). Indeed, they are the first elements of the pipeline which helps in the delivery of small batches of the new features with a clear vision and roadmap of deployment of incremental updates. This ensures the products are delivered as independently as possible so that faster delivery can be achieved.

VI. DISADVANTAGES

A. Limited Documentation

In the beginning of the agile projects, the team squander little time on the documentation. Whence, the KPIs and requirement logs are very limited or are not prepared at all. For the most part, corroboration is provided haphazardly when there are any predominant changes, or when the final product is generated. In this way, the documentations are very generic and less detailed which is of no great use for future references. In agile, it is recommended to make documents only if it is necessary and keep it as simple as possible to avoid the chaos of files and folders. Due to this, there is high chances of missing out any important details about the project. New team members, who might join the project later on, does not have much idea about the certain features of the product since documentations are created just in time after the development.

B. More Commitment

The project lifecycle is inter-dependent on customers, clients, team leaders, technical product owner, developers where they interact with each other incessantly. Several communication channels are used like face-to-face, virtual meetings, webinars, mail chains, social groups to perform close cooperation. This puts stakeholders in the spotlight to perform prompt testing and provide validation so that developers can go ahead to the next phase. Hence, agile project requires close collaboration and extensive team involvement which demands for more time and energy from everyone involved in the process to meet the set requirements. Though, each phase is engaging but the entire process is onerous and calls for a big commitment to ensure the favorable result. Any lack of engagement can impact product quality and success.

C. Fragmented Output

Mostly, agile frameworks follow iterative methodology where developers provide partially developed product to the stakeholders. Review meetings can be held to validate the product. Once, green signal is provided, developers work towards the full development of the product. Even though, this might save time as developers can incorporate changes suggested by the clients and bring the product to market faster but team works on the different features of the product in different cycle which might produce a fragmented output rather one cohesive unit.

Due to the lack of project boundaries and processes, this jeopardize the entire chain by creating under-developed or uncontrolled expansion of the product. The see-as-you-go nature of agile makes it difficult for the developers and stakeholders to assemble the features over different iterative cycles.

D. Difficult to Track

The agile methodology delivers in increments or iterations which makes it difficult to track the project progress. Team leads must look across the cycles for various features and developments.

This discourages most of the larger organizations to adopt the agility concept as lack of formal process description can lead to project failure. Additionally, less experienced team members might suffer more in order to maintain the checks and balances. The long term projects can be more problematic as team members might not always know what their end product will look like or the result of the next iterations, this will create perplexity in the team to accurately predict the cost, quality, schedule, resources, or project progress at any point in the development cycle. Sign-off is required at every phase to allow developers to move forward in the cycle. Sometimes, it might get difficult to manage and control the processes.

VII. CONCLUSION

In this paper, the Agile methodology for product development has been discussed. It explains the background, characteristics, frameworks, advantages, and limitations of agile projects. It has been concluded that agile is the most effective software development life cycle methodology. Almost every technical project seems to practice agile methods for software development, or a version of it. It has evolved since its formal launch in 2001 to overcome the challenges faced by organizations. The principles, frameworks, processes are designed as per the prevailing operating conditions. It adopts smarter and faster-operating methods to prioritize the product delivery, leverage feedback, and incorporate adjustments to improve the application. In this way, it encourages the ongoing procedure for continuous improvement while controlling the integration, scope, schedule, cost, quality, resources, risk, procurement, stakeholders for product development. Hence, the software development team, product owner, team lead, stakeholders are more productive and happier for being a part of the agile project. Agile provides an increased level of product competency which delivers extraordinary digital experience in a hypercompetitive world.

REFERENCES

1. A. Ahmed, S. Ahmad, Dr. N Ehsan, E. Mirza, S.Z. Sarwar, "Agile Software Development: Impact on Productivity and Quality", IEEE ICMIT (2010).
2. Peter Maher, "Weaving Agile Software Development Techniques into a Traditional Computer Science Curriculum", IEEE International Conference on Information Technology 2009, pp. 1687-1688.
3. Manifesto for Agile Software Development, http://agilemanifesto.org/
4. Sheetal Sharma, Darothi Sarkar, Divya Gupta, "Agile Processes and Methodologies: A Conceptual Study", International Journal on Computer Science and Engineering, (IJCSE), ISSN 0975-3397
5. How the kanban methodology applies to software development, Kanban vs scrum WIP www.atlassian.com/agile/kanban
6. Scott Ambler, "Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process", www.semanticscholar.org/
7. Scott W. Ambler "Agile Modeling: A Brief Overview", www.researchgate.net/publication/220868175_Agile_Modeling_A_Brief_Overview
8. Gaurav Kumar, Pradeep Kumar Bhatia, "Impact of Agile Methodology on Software Development Process", International Journal of Computer Technology and Electronics Engineering (IJCTEE), ISSN 2249-6343
9. Agile Manifesto and Essentials Overview, www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/
10. DSDM – Dynamic Systems Development Method, https://www.whatissixsigma.net/dsdm-dynamic-systems-development-method/

AUTHOR PROFILE

Kajal Singh, is a Business Intelligence Analyst, technologist, and researcher at Schneider Electric, Bangalore, India. She holds a Bachelor of Technology (B.Tech) degree in Electronics and Communication Engineering Department from Vellore Institute of Technology, Tamil Nadu, India (2020 graduated). She has gained experience in data modeling, mapping, and data insights and further holds technical expertise in Oracle Business Intelligence Enterprise Edition (OBIEE), Oracle Transactional Business Intelligence, Oracle Analytics Cloud (OAC), Oracle Data Integrator (ODI) tools and techniques. Kajal is an Oracle certified Business Analytics Expert, ThoughtSpot certified Data Expert and government certified personal fitness trainer. She is passionate about OAC Machine Learning, Artificial Intelligence, Operations Management, and Workflow Analysis. She aspires to work in the field of product development and is active on JIRA and Confluence tools to adopt the relevant agile frameworks like Kanban for her team projects.