

Rapid Digital Transformation Using Agile Methodologies for Software Development Projects

¹Kausar Parveen, ²Maryam Daud

^{1,2}University of Engineering and Technology, Lahore

Email: Kausarnawaz6@gmail.com

(Received: 08 August 2021; Accepted: 03 Sep 2021; Issue Published: 12 Sep 2021)

ABSTRACT

Now a day's all organizations are moving towards digitalization. These consequences of the use of digital technologies made organizations seek for best and fast digital solutions. All software developer companies are also trying to draw consumer's attention by offering prompt services. In this regard, the critical issue in information technology and other areas of computation is how software can be created easily and rapidly for complex businesses. In this context, the main aim of the research is to show the agile methodology role in the rapid digital transformation. In this paper, we have surveyed different agile methodologies and tools for rapid software development and introduced an agile management tool having a backlog. We identified the key practices of agile methods and after a survey, it is suggested that the agile approach can help to achieve a balance between the applications generated by developers on customer demand. This paper illuminates and translates agile methodologies into agile project management tools for simple and rapid application development. Empirical research based on a case study is provided for better understanding and showing the importance of agility in software development.

KEYWORDS: Agile, Software development, Backlog, Agile Alliance, Success, Agile management tools.

1. INTRODUCTION

Today's market is very fast-paced and innovative; the only constant thing in today's industry is change. All organizations are seeking to draw consumer's attention by giving rapid services. Software developers are seeking quick ways to the development of quality software. In other words, it is right to say that software teams or companies are under pressure to complete the customers' demands. As a result, software development teams are under a lot of pressure to meet customer demands. [1,2].

Software Engineering is an ancient term, set up by a NATO study group in 1968 and focusing on the software problem [3]. IEEE characterized software engineering as the use in software creation, operation, and maintenance of a systems-driven, disciplined, quantifiable approach that is engineering software application [4].

The end product is interesting to see from the descriptions, but the approaches and behavior required for producing software must be carried out in a systematic, disciplined, and observable manner in all software life cycles. Following the characteristics of these approaches would result in more stable software, easy to maintain, and meets the software requirements, particularly when the software is large and contains many functionalities [4]. Software development and engineering is collaborative work; tasks can be spread through several teams, and these tasks must be coordinated and prioritized according to certain requirements. Tasks can be done in parallel, but they cannot be

started before the previous task has been completed.

These practices, procedures, and teams must be coordinated to achieve the best software or product at the lowest cost [3]. There are several traditional software development approaches, such as evolutionary, waterfall, spiral, iterative, and incremental development, and so on [5]. Often these approaches are referred to as expected software development approaches. These methods are very useful if huge complex software needs to be developed, it will help eradicate the old-style informal software creation and systematically produce high-quality software that meets user requirements [6].

There are many issues with these instance methods its plan should be made in advance, written software's requirements are required even at the coding level and a full test of the software at the end is required.

Many projects that have followed the conventional software development methods, especially maintenance and user request modifications, have addressed major problems. Some of these changes could result in significant changes, which are considered a major problem in software development.

Since all of these approaches require a lightweight software development process, the main aim is to accelerate development and react effectively to the changes requested. Therefore lightweight approach known as Agile methods of software development was introduced [3].

The agile software development system is the result of this demand but still, further improvements are

required. Agile architecture makes it simple to design software and use it in various settings [7]. Early software creation is aided by the agile approaches which are done in stages, with each release providing more working code and less documentation. To endorse agile approaches, a variety of agile methodologies are currently in use. We'll look at different agile methods, their similarities and differences, as well as their pitfalls, in this article.

This article offers an overview of various agile methodologies, key publications for these methodologies, and compares and contrasts the five methods focused on the approaches described. This paper looks at agile methodologies, their comparisons, practices and discusses how to integrate methodologies effectively.

The first part of this paper contains an introduction. Section 2 includes a literature analysis, which can aid in determining the present state of the issue domain. Section 3 provides the high-level methodology for determining the problem's solution. Section 4 is based on a case study while section 5 wraps up the whole paper and suggests further study.

2. THE AGILE METHODOLOGIES

Different organizations use various techniques for fast coding, many of which are labeled as agile systems. First, it's important to understand agile processes. Agile architecture is implemented via groups. According to a 2020 Agile report, larger organizations are now using agile for quicker, simpler, and wiser app development. The agile poll in 2020 showed that 94% of tech companies are using agile strategies, and this agile methodology usage ratio is expanding globally too [7].

In 2001, at the 17th meeting of agile process methodologies, the "Agile Alliance" was created. A variety of basic agile workflow concepts are outlined in this manifesto. This manifesto emphasizes iterative and gradual growth, client collaboration, regular and rapid implementation, a shorter development time, customer loyalty, lower mistakes, and adaptability to agile improvements [8, 9,10].

2.1. Categories of Agile Methodologies

Iterative and gradual methodologies are also used in Agile. Customers nowadays want their projects to be completed as quickly as possible [11]. Agile meets this need. Agile methodologies come in a variety of forms. Since these agile methodologies discard useless features or methods, therefore they are getting the attention of developers. Agile methods come in a variety of forms. In this section here in research, They're included in this research project.

2.1.1. Extreme Programming (XP)

Kent Beck first proposed the Extreme Programming method. XP's core principles are simplicity, collaboration, feedback, confidence, and respect [12]. The XP process begins with gathering consumer requirements. The XP method is broken down into a limited number of loops, each with a set of steps. The loop is divided based on the requirements of the customer, which are obtained first. The second step is preparation, which involves deciding on the number of cycles to be completed, assigning goals to the conditions, and estimating the commitment needed to complete each cycle. The pair programming principle has been used with iterations. New user specifications are often welcome, and the plan is updated to accommodate them. The created software is then checked for defects, and any bugs or errors discovered will be fixed in the next version [13].

Scrum: SCRUM started in 1993 at the Easel Corporation, where it was used to complete a software project. During this step, each sprint produces the commodity that completes the project's task. The product backlog is used to determine what the next condition is. Meetings are held to make decisions on the sprint backlog, which prioritizes which steps should be completed first in a sprint. Each sprint in this technique is given a time limit and must be completed on time. If a sprint is not finished according to the schedule, the next sprint will start after pausing. The paused sprint will be completed at the end, and switched to the next sprint from the backlog [14]. Scrum meetings will be continued are held. The Scrum Master holds a Sprint Review Meeting after each sprint.

2.1.2. Feature Driven Development

The Feature Driven Development (FDD) feature was created by Peter and Jeff De. According to Palmer and Felsing, this approach focuses only on the design and construction processes and does not encompass the whole architecture life cycle. Team members and consultants work together to design the overarching model of the domain, define the system's scope, and have a shared interpretation of the domain. The team is assigned a set of design by feature and build by feature tasks, after which they are split into two groups, one for features and the other for builds. The team selects a list of features in the Build a Features List process, and then groups these features into feature sets, which are then grouped into main feature sets. Both features are then prioritized in the Plan by Feature process, which determines the order in which feature sets are considered for the implementation plan. Features are checked in two-week period boxes during the Design by Feature and Build by Feature phases. This process is repeated until no more features are present [15].

2.1.3. The Dynamic Systems Development Methodology (DSDM)

This methodology is a system for controlling several processes at the same time and delivering a fast product in agile project management. The Moscow law is a strategy for prioritizing conditions based on whether they could have, should have, or could have but cannot occur at the same time. In the DSDM, the feasibility and market studies are completed one after the other. The remainder of the project's location is determined by these stages. The project's real progress is made in the final three iterative and incremental phases.

The issue is found, the cost is calculated, the system's functionality is discussed, and an implementation plan is created in the feasibility analysis. The prototype is designed to determine whether or not further steps should be taken [16]. In a business study, a business and technology review is carried out. This is a market field concept that describes business processes. The architecture specification and construction plan are the outputs of this process. The technique for prototyping, checking, and configuration management are also part of the implementation plan. The functional model iteration is the first gradual and iterative process. This process includes analysis, coding, and checking [17]. In Design and Build Iteration, a high-quality device is created and safely delivered to customers. This phase's goal is to clean up working designs to satisfy non-functional specifications. In the final deployment process, the product is shipped and installed in the real world.

2.1.4. Adaptive Software Development (ASD)

Highsmith is the creator of ASD. The specifications for this approach can be vague at the start of the project. The project's mission and goals are defined, and assumptions are used to determine the project's specifications and timeline. The feature like set parameters, scale, projections, and resource availability measures its tie box for projects. The length of the iteration is determined with the help of features like uncertainty degree, project size iteration of the given time box. Team participants create an authoritative argument with iterations. Each iteration is assigned functionality by developers and users. Collaboration is essential for teamwork, as well as mutual respect and confidence. Collaboration encourages learning and improves comprehension [18].

2.2 The Agile Project Management Concept of Industry

APM is a recent concept in the industry that refers to an iterative and systemic approach in which developers and project partners work together to understand the domain over time better. They also determine what features need to be built first and what needs to be constructed.

2.2.1 Agile Management Components

The components of the agile project management tool [19] are as follows.

Visual Controls: Each project will have its visual management system. This will be equivalent to a "cards-on-the-wall" planning method for aiding a team in project coordination. For eg, one successful agile project team hung different color groups of cards on the wall that represented the solution's features. One color represented features that had been developed, built, tested, and was in use, while the other color represented features that had been scheduled, built, and tested but not yet put into use (but were ready to go). The team could see where they were with each feature set at a glance. Visual control means that everyone on the team has the same perspective on the project.

2.2.2. Co-located high-performing teams

Employees, clients, and administrators would use a login mechanism to collaborate and communicate as if they are in the same space. Both main team members, including the customer/end-user, are co-located, ideally in a workspace, as in agile growth. This method vastly improves coordination and contact efficiency. However, IT developers can experience a major cultural shift as a result of this. Since project managers are in charge of assembling a high-performing team, they must ensure that they have been appointed developers who are capable of working collaboratively.

2.2.3 Adaptive control

The tool will have adaptive controls. Everything on the project is still changing and adjusting. Because of the complex nature of the workplace, the project manager must be seen as a leader rather than a taskmaster. Rather than giving the team strict orders to obey, the project manager assists the team in forming working relationships, defining ground limits, and encouraging cooperation. Agile team members refine their strategies over time by incorporating lessons learned from past iterations into the next version, which is a best practice for any project.

2.2.4. Collaborative development and feedback

There will be a feedback window from staff, customers, and managers since APM depends on teamwork among all team members to produce outcomes, capture candid feedback, and incorporate learning on the next version of the solution. Constant reviews and development are one of APM's strengths. In conjunction with the client and technology leaders, the project manager performs the initial preparation and the market analyst identifies and prioritizes the solution functions. The Agile project teams then cooperate on each incremental build's architecture, creation, testing, and reworking. The success of a project is aided by constant collaboration with the customer.

2.2.5. Feature-driven development enforced by Manage

On the manager's home page, he will be able to see all feature developments. This method eliminates uncertainty and helps the team to concentrate on one element at a time. One squad, for example, is entirely focused on Feature #4. They are unconcerned with Features 1 through 3. The business analyst and project manager ensure that the next item in the backlog is the next target in terms of business importance and risk. High-risk elements or key infrastructure parts are usually installed first, followed by everything else, which is prioritized depending on market importance. The aim is to develop feature-driven modules that only have a one-way dependence on the core system; as a result, customized components are self-contained and can be built in any order or even in parallel.

2.2.6. Move focus from Cost to Revenue

The importance of features is determined by their worth, such as increased sales or market share. The business analyst's job is to ensure that the agile project team does not overinvest in the implementation of the new approach. If this is the case, the economic case will have been eroded, and the project will cost more than the company will

earn. While the project manager is concerned with project expenses, the market analyst is concerned with the actual cost of ownership, which covers not just the cost of implementation or acquisition of the proposed solution but also the cost of maintaining the device until it has been implemented.

2.2.7. Lessons learned feature

For each loop, the team holds a lessons learned session to decide whether they can do best in the next iteration, and there will be a button at the end to monitor the lesson. If the team grows, it adapts to how the participants collaborate to boost overall team success.

2.2.8. The Current Tools for Agile Management

Currently, different tools for agile management are in use. The agile management tools assist in:

- Scrum project planning
- Scrum project monitoring
- Project management on a single framework

Scrum Tools are available in a range of sizes and forms. Some are open access (free), while others include are paid. Table 1 shows open source agile project management (APM) tools [20].

Table 1. Agile Project Management tools (Open Source)

Open Source APM Tools				
Agile Fant	Tackle	Scrum Factory	Digaboard	Agile Express
Retrospectiva	Scrum Dashboard	Scrinch	Fire Scrum	Ice Scrum

Table 2. Agile Project Management tools (Paid and Distilled)

APM Paid and Distilled Tools				
Scrum 2 Go	Scrumrf	Solution Factory	Prj Planner	Pango Scrum
Urban Turtle	Kunagi	Tweet Scrum	Project Cards	Pivotal Tracker
Rally Dev	Axosoft	EasyBacklog	Banana Scrum	pmScrum
Planbox	Quantum Whisper	Scrum Do	Agile Manager	Intervals
Assembla	Hansoft	Agile For Scrum	Scrumpy	iMeta Agility
ScrumTool	Daily-Scrum	Jira (GreenHopper)	Mingle	Agile Soup
Quick Scrum	Scrum Desk	Scrum'd	Scrum	Agile Cockpit
Acunote	Agile Agenda	Agile Bench	Agile Buddy	Quick Scrum
Agile Agenda	Agile Log	Scrum Pad	Scrum Edge	GravityDev
Redmine Backlogs	Version one	Airgile	Explain PMT	OnTime Now
AgileWrap	Agile Task	Scrumwise	Scrum Works	Tangy Orange Scrum

2.2.9. JIRA Agile -- Atlassian

JIRA aspires to be a tool that is more adaptable and has a shorter learning curve[21]. Jira is the best project management framework available today. Task supervisors and owners have access to comprehensive inventory statistics and analysis, such as a full-flow diagram showing the status and amount of project problems over time. The downside is that Jira is included in a daily subscription to the Atlassian race.

2.2.10. Axosoft

Axosoft is a scrum-based project management toolkit. Over 11,000 teams like NASA, Cisco, and Boeing have used it. Process specification, management of Backlog, and visualization are the three stages of Axosoft's methods, all consistent with the Scrum approach. Axosoft allows you to create an infinite number of programs, subprojects, tales, and epics[21].Axosoft takes a hybrid approach to technique, providing Scrum and

Kanban methods and a completely immersive Card View mode. It has a drag-and-drop interface that makes managing the queue, sprints, customer stories, and releases a breeze. It also comes with useful prioritization methods and work-in-progress benchmarks to help you keep your team on track.

2.2.11. Planbox

Planbox, a project management platform for tech Sponsorship Companies, was developed in 2009 for creating house solutions [22]. It's available in a cloud edition, which the developers claim would provide users continuous access to upgraded updates, new content, and so on. Planbox is a kit that caters to a wider range of people than any of its rivals. It's useful for marketing departments, IT, engineers, and agile newbies and veterans alike. Initiatives, campaigns, objects, and activities are the four main components of the job, including real-time alerts, reminders, time monitoring, velocity maps, storyboards, deadlines, and reports to help you and the staff handle it. Planbox has a feedback loop that assists in decisions based on the needs of customers.

2.2.12. Assembla

Assembla is an organization that focuses on mission management and problem tracking, specializing in Agile online project planning for distributed teams.[23] It was founded in 2005 and has already seen considerable success, with over one million customers in over 100 countries, and collaborated with businesses including HTC,eBay, Intui, and PayPal.it has created a niche in the project management industry. Through the Assembla Cloud Network, you can work with anyone in the big picture, regardless of venue.

2.2.13. Comparison

The comparisons of the four tools described above are made using the agile approach in table 3. The table3 shows that all aspects of agile philosophy are combined in the above agile software. In practice, they meet all of the agile methodology's requirements. Planbox is the most popular tool on the market, according to the table 4 ranking [24].

Table 3. Agile management parameters comparison of Jira, Axosoft, Planbox, and Assembla

	Jira	Axosoft	Planbox	Assembla
Backlog (Just Oral)	✓	✓	✓	✓
Cards	✓	✓	✓	✓
Card Walls	✓	✓	✓	✓
Planning	✓	✓	✓	✓
Burndown charts	✓	✓	✓	✓
Meetings	✓	✓	✓	✓
Sprint Management	✓	✓	_____	✓
Customer Feed Back loop	_____	_____	✓	✓

Table 4. Agile management tools rating

	Jira[20]	Axosoft	Planbox	Assembla
Avg User Rating[19]	3.98	4	4.7	3.65

2.2.14. Success or Failure of Software

We wanted to know the basic features within these systems that most effectively achieve these objectives [17]. As agile methodologies in project

management focus on speed, replica, and waste avoidance, different surveys are analyzed to learn which of these systems' basic functionalities are more useful in achieving those objectives. According to the 2013 survey [18], 39 percent of

projects were completed successfully, 18 percent were failed, and 43 percent were questioned. The project completion rate rose marginally from 2004, but the number of failed projects remained nearly unchanged.

2.2.15. Problem Statement

In APM backlog acts as an initial point for all development. The backlog is a comprehensive list of the functionality or technical activities which the team monitors and considers to be necessary and appropriate for the completion at all times of a project or publication. It is not uncommon for managers or sprint masters to be unable to the user's point of view because users are not familiar with technology or cannot effectively communicate their issues. As a result, the software fails. When this happens, development takes longer, and users must pay more to change their requirements.

3. METHODOLOGY

During our research, we discovered that developers frequently misunderstand customer terms or requirements, which leads to software failure. We look at eliminating the gap in APM tools and compensating for customer absence in this study.

3.1. Backlog and agile methodology

From the perspective of agile methodology, a backlog is a collection of features or technical tasks that the team keeps track of and considered necessary and sufficient to complete a project or release at any given time. If a backlog item does not contribute to the project's goal, it should be removed; however, it should be added if a task or feature becomes known at any point.

3.2. Current role of Backlog

A backlog of sprints is a collection of projects and requirements to be performed in an agile development process within a sprint. The sprint backlog includes the following items: A priority-ordered list of user stories within the sprint.

- The estimated relative effort for each user story.
- The tasks required to create each user story.
- The time it takes to complete each task in hours.
- You estimate the number of hours each task will take to complete at the task level. You can use the time each task takes to determine whether the tasks will fit into your sprint because it has a set length and thus a set number of available working hours.
- The development team should complete each task in one day or less.
- A burndown chart that depicts the progress of the development team's work. The sprint backlog is created and maintained collaboratively by the development team, and the development team can only modify it. The sprint backlog should reflect a current snapshot of the sprint's status.

3.3. New Feature

The backlog button is a screenshot explanation for customers. When you have the right requirements, you'll have the right solution. Customers' inability to explain technical terms is one of the reasons why their desired requirements cannot be communicated to developers. In an agile project management tool, the backlog button allows users to explain their requirements. The architecture of the backlog is shown in figure 1.

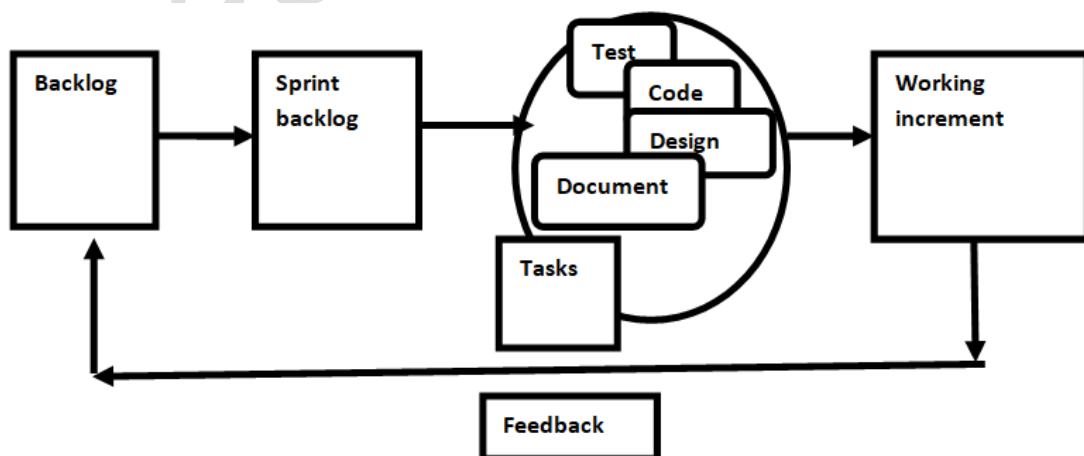


Figure 1: Architecture of Backlog in Agile tool

3.4. Working

In our proposed model, an agile project management tool has been created with a backlog

button. We can show various software views according to the customer's requirements by providing software with a backlog component button. Screenshots are shown according to the

customer's requirement using this backlog button. There can be different categories screenshots under the backlog button. This agile tool can show all categories sample screenshots like

- Games
- Education
- Tutorials
- TV websites
- Databases
- Desktop Applications

One is chosen by the customer and presented to the creator. The consumer may also add additional

functionality, either from his point of view or by selecting another backlog component.

3.5. Implementation

A real-world case study was conducted using two teams of software developers. Depending on the workload, the number of employees employed may be increased or decreased. Table 5 lists the production team members and their related qualifications and experience; each team has the same qualifications and experience. Table 6 depicts the two types of customers with the same level of experience and qualification.

Table 5. Team member's details

No	Role	Qualification	Years of Experience
Team A			
1	Project Manager	Ph.D Software Engineering	6
2	Team Lead	Master in Computer Science	6
3	Senior Software developer	Master in Computer Science	4
4	Software developer	Bachelors' of Computer Science	3
Team B			
1	Project Manager	Ph.D Software Engineering	6
2	Team Lead	Master in Computer Science	6
3	Senior Software developer	Master in Computer Science	4
4	Software developer	Bachelors' of Computer Science	3

Table 6. Customers details

No	Role	Years of Experience	Qualification
Customer A			
1	Customer	4	Master in Business Administrator
Customer B			
1	Customer	4	Master in Business Administrator

Two different teams tested the validity and results of the backlog button. The first team works without the backlog button, while the second team works with it. To measure success and time, a customer with the same requirements was sent to both teams from the same company. To obtain feedback on the case study, semi-structured interviews with the team were conducted. Over four weeks, we met with key individuals several times.

3.6. First Round

In round one, we gathered data on the product and customer and the product's history, development methodology, development team experience, and market success.

Because of their extensive experience with the company over the past several years, the team

members were well aware of the workflow. The customer was also aware of their requirements, but they lacked the technical knowledge to explain computer terms.

Team A gathers requirements through the backlog from customers and requirement specifications in the form of story cards, work schedules, effort distribution charts, and product burn-down charts. This provided team with a clear understanding of the details involved in product development and how it works. In the second round, The focus of the discussion was on clarifying and validating the information that had already been gathered. Table 7 shows results without using the backlog button and table 8 shows results after using the backlog button.

Table 7. Results without the backlog

	Time(min)	Understanding(percentage)	Success (percentage)
Customer	180	80	80
Project Manager	180	80	80
Team Lead	120	70	80
Senior Software developer	100	70	80
Software developer	100	70	80

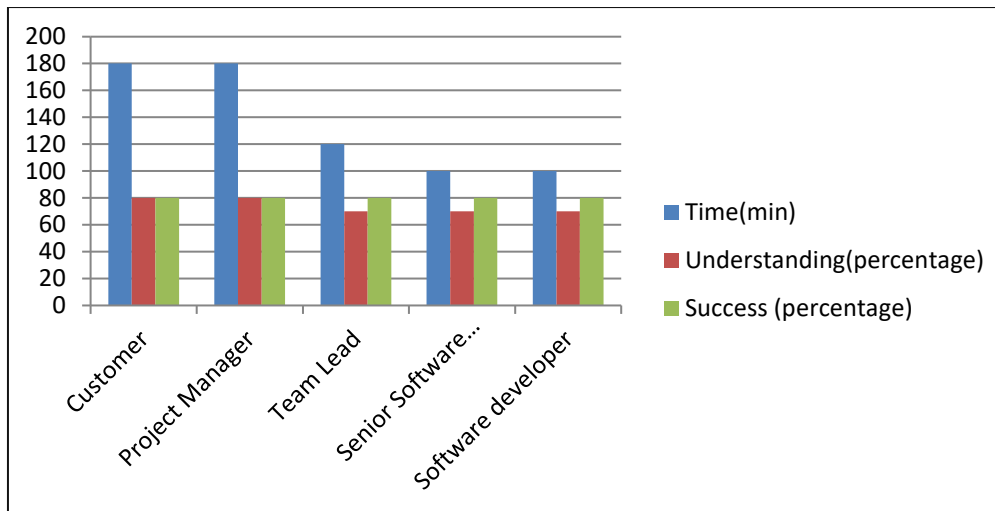


Figure 2: Results without the backlog

Table 8. Results with the backlog

	Time(min)	Understanding(percentage)	Success (percentage)
Customer	30	100	100
Project Manager	30	100	100
Team Lead	30	100	100
Senior Software developer	30	100	100
Software developer	30	100	100

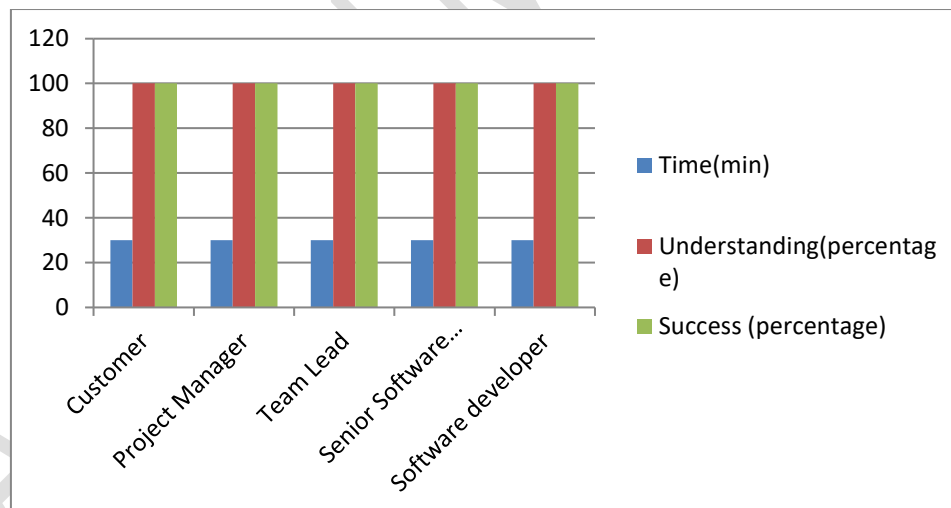


Figure 3: Results with the backlog

Table 9. Results Summary

	APM With Backlog	APM Without Backlog
Usage of time	Less	More
Clear Picture	Clear Picture	Not clear picture

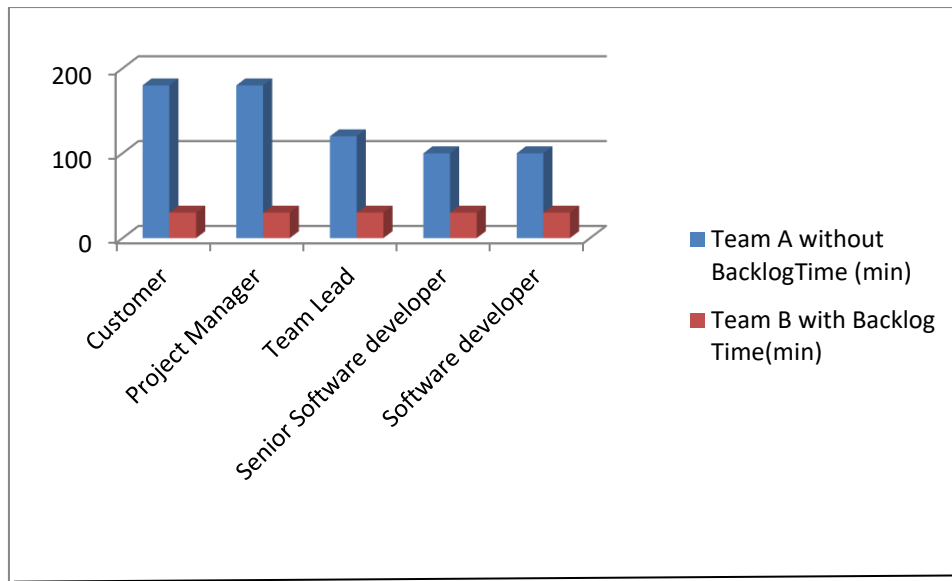


Figure 3: Results summary

3.7. Discussion/Findings

In the case of product growth, it has been found that consumer participation is important. The creators in this case adhere to the customer's standards and specifications. SCRUM, like many other agile growth technologies, necessitates full-time customer participation during Sprints. It has been noted that the Project Manager is still the individual to approach for collecting requirements.

4. CONCLUSION

According to information technology, computers invade every aspect of life, including kitchen compliances, large-scale manufacturing equipment, and educational institutions. All of these agencies want computerized systems, which necessitates the purchase of applications tailored to their needs. This demand drives the use of agile methodologies for accelerated software development and agile project management software. We addressed different 15-20 minute agile methodologies and agile project management tools in this article. We also add a new backlog feature. Screenshots and models are included in this feature. When customers came in for their products, they were amused by taking their specifications and showing related product screenshots and models, from which the consumer could select a design or request adjustments. As a result, the client will be able to express his needs more clearly. According to the case study between two teams, one of them used the feature of the enhanced backlog and the other team did not, and the findings showed that the team that used the enhanced backlog feature

performed better than the team that did not. As a result of the backlog button, which is new in project management, we discovered that using it would clear the customer's expectations to developers, reducing the risk of the desired product failing.

5. Future Work

There are many forms of agile management software for rapid software development on the market today, but they're all very complex and time-consuming, in my experience. Human-computer interaction methods can be used to solve this issue.

REFERENCES

- [1] M. Maynard, E. Falcone, K. Petersen, B. Fugate, and L. Bonney, "Conflicting paradigms in manufacturing and marketing decisions: The effects of situational awareness on team performance", *International Journal of Production Economics*, vol. 230, p. 107801, 2020.
- [2] C.Lindskog, and M.Magnusson. "Ambidexterity in Agile software development: a conceptual paper." *Journal of Organizational Effectiveness: People and Performance*, 2021.
- [3] S.Al-Saqqa, S. Sawalha, and H. AbdelNabi. "Agile Software Development: Methodologies and Trends." *International Journal of Interactive Mobile Technologies*, vol. 11, 2020.
- [4] A.Sablis, D.Smite, and N.Moe. "Team-external coordination in large-scale software development

projects." *Journal of Software: Evolution and Process*, 2020.

[5] P. Chandani., & M. A. Kumar, Review of Agile Software Development Methodology. *change*, 7, 2020

[6] G. Matharu, S. Mishra, A.Singh, and P.Upadhyay, "Empirical study of agile software development methodologies: A comparative analysis. *ACM SIGSOFT Software Engineering Notes*, Vol. 40, 1-6, 2015.

[7] Z.Dragičević, and S. Bošnjak,"The Role of agile software architect in the age of digital disruption and transformation", 2020.

[8] M. Hamid , F .Zeshan., A.Ahmad, F. Ahmad, F.,M. Hamza, A.Khan, Z. A., ... and H. Aljuaid, "An intelligent recommender and decision support system (IRDSS) for effective management of software projects". *IEEE Access*, 2020.

[9] J.Götze and A.Romanov, "Enabling Rapid Digital Transformation.In Architecting the Digital Transformation" , , *Springer*, p. 55-79, 2021.

[10] S.Munawar, R. Yousaf, R, and M.Hamid, M. "Extended Scrum process model using software reliability engineering concerns". *Journal of Information Communication Technologies and Robotic Applications*, p.1-10, 2018.

[11] V.Munteanu and P.Dragos, "The Case for Agile Methodologies against Traditional Ones in Financial Software Projects". *European Journal of Business and Management Research*, Vol 6, p.134-141 , 2021

[12] M.Holcombe, "Running an agile software development project". *John Wiley & Sons*, 2008.

[13] S.Sharma, D. Sarkar, and D. Gupta," Agile processes and methodologies: A conceptual study". *International journal on computer science and Engineering*, Vol 4, p.892, 2012.

[14] C.Malhotra and A. Chug, " Agile testing with Scrum-A survey". *International Journal of Advanced Research in Computer Science and Software Engineering*, vol.3,p. 452-459, 2013.

[15] D.E.Strode," The agile methods: An analytical comparison of five agile methods and an investigation of their target environment". *Unpublished Master of Information Sciences (Information Systems)*, Massey University, Palmerston North, 2005.

[16] C.B.Chapman and M.Pinfold, " The application of a knowledge based engineering approach to the rapid design and analysis of an automotive structure". *Advances in engineering software*, vol-32, p.903-912, 2001.

[17] G.BüyüKözkan, T. Derelİ, and A.. Baykasoğlu, A. "A survey on the methods and tools of concurrent new product development and agile manufacturing". *journal of Intelligent Manufacturing*, Vol.6,p. 731-751, 2004.

[18] J.Highsmith, and A.Cockburn, "Agile software development: *The business of innovation*". *Computer*, p. 120-127, 2001.

[19] K.B. Hass" The blending of traditional and agile project management". *PM world today*, p.1-8, 2007.

[20] A.Mishra, and D.Mishra, D. " Software project management tools: a brief comparative view". *ACM SIGSOFT Software Engineering Notes*, Vol.3, p.1-4, 2013.

[21] H.Sarkan, M.Ahmad and A. Bakar, "Using JIRA and Redmine in requirement development for agile methodology". In *IEEE 2011 Malaysian Conference in Software Engineering*, pp. 408-413, 2011.

[22] M. Younas, I. Ghani, I., D.Jawawi, and M. Khan, "A framework for agile development in cloud computing environment". *Journal of Internet Computing and Services*, Vol. 5,p. 67-74, 2016.

[23] M.Manole, and M.Avramescu," Comparative Analysis of Agile Project Management Tools". *Academy of Economic Studies. Economy Informatics*, Vol.1,p. 25-31, 2017.

[24] D. Özkan, and A. Mishra, " Agile Project Management Tools: A Brief Comparative View". *Cybernetics and Information Technologies*, Vol.4, 2019.