Agile Project Management Tools: A Systematic Review of Features and Effectiveness in Software Development Contexts

First Author[1,2*], Second Author[2,3†] and Third Author[1,2†]

[1*]Department, Organization, Street, City, 100190, State, Country.
[2]Department, Organization, Street, City, 10587, State, Country.
[3]Department, Organization, Street, City, 610101, State, Country.

*Corresponding author(s). E-mail(s): iauthor@gmail.com;
Contributing authors: iiauthor@gmail.com; iiiauthor@gmail.com;
†These authors contributed equally to this work.

Abstract

Agile methodology is becoming really popular because of its adaptability, customer satisfaction, collaboration, transparency, and incremental delivery according to customer needs. There are many project management tools for software teams and non-technical teams. But not every project management system has all the necessary features. Some have very complex functionalities that are difficult for beginners. Some are not suitable for small teams. This study reviewed four popular project management tools and two test management tools to understand how project management and test management work. Based on the findings, this study proposes a new system for managing software projects. The system incorporates key tools for managing Scrum and Kanban projects, such as Kanban boards, workflow management, reports, a Gantt chart for visualizing dependency and schedule tasks, product backlogs and sprint backlogs, and tools for managing current sprints. This system also includes some new features like Scrum Review Board, Live issue Board and Management tool for Test Driven Development.

Keywords: Agile, Kanban, Scrum, Sprint, Project Management

1 Introduction

Agile methods represent a significant advancement in software engineering. Over the last 30 years, they have significantly increased software development success rates, IT team motivation and productivity, and improved market agility and quality.[1]

Agile methodology is an incremental and iterative approach to software development that prioritizes cooperation and flexibility amongst cross-functional teams. The Agile technique is a project management style that prioritizes ongoing communication and improvement while segmenting the work into manageable chunks. Teams work in cycles of planning, carrying out, and reviewing. The Agile Manifesto's primary tenets include producing high-quality software more quickly and efficiently. It consists of four core values and twelve supporting principles that emphasizes the importance of individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, and responding to change over following a plan.[2]

1.1 Evolution of Project Management Frameworks

Project management frameworks in software development have evolved significantly over time to reflect the changing environment of technology and user needs.

The waterfall methodology, first in 1970, revolutionized software development by providing formal specifications adapted to slow-changing demands and monolithic systems. However, its inflexible sequential approach became obsolete in the internet age, when adaptability and speed became critical, leading to the introduction of more flexible techniques. Internet-driven software development, particularly in startups, saw fast transformation as tiny teams without traditional experience sought quick market entrance under financial pressure. This transition encouraged the emergence of development tools and platforms to enable quicker cycles. Startups discarded the waterfall paradigm in favor of agile, collaborative approaches to meet budget restrictions and maximize productivity. They prioritized iterative processes, experimentation, and user feedback above detailed planning, preferring to develop solutions internally and leverage user feedback for business growth. Agile promotes flexibility and adaptation by dividing tasks into digestible chunks. Unlike waterfall, it does not demand full answers up front. Instead, it supports iterative development and testing with users, allowing for fast tweaks and course corrections in response to feedback and changing demands.[3]
In the late 1990s and early 2000s, as traditional methodologies were challenged, Agile emerged. In a pivotal meeting in Oregon in 2000, influential figures aimed to accelerate development, improve delivery, and prioritize customer satisfaction, marking a significant shift in approach. Less than a year later, on February 11-13, 2001, the history of the Agile Manifesto was written. In February 2001, 17 developers gathered in Utah to discuss the rigidity of the conventional waterfall method, which marked an important turning point in the history of software development. Their concerns prompted the creation of the Agile Manifesto, which was released on February 17 and emphasized collaboration, functioning software, customer participation, and flexibility. Since then, Agile has altered the industry, introducing frameworks such as Scrum, Kanban, and Extreme Programming, Lean, Feature Driven Development which have become

the preferred strategy for many firms and continue to shape the landscape of software development today.[4] Most well-known Agile frameworks are Lean, Kanban, Scrum, Extreme Programming, Feature Driven Development.

Lean development is the implementation of Lean principles in software development. Lean principles originated from the Japanese automotive industry. Lean principle aims to maximize value to the customer and minimize waste. The Lean software development process includes seven principles- Eliminate waste, Build in quality, Amplify learning, Decide as late as possible, Deliver as fast as possible, Empower the team, Build integrity in, See the whole.[5]

In Lean Development the scope of creating waste is defined. These includes writing unnecessary functions, starting more than completed, delaying in development, frequently changing requirements or unclear requirements, inefficient communications, partially completed work, defects, frequently switching tasks. The main advantages of Lean development is that it's primary focus is adding value to the customer and minimize waste. Team will eliminate unnecessary cost which will reduce overall cost. Teams are empowered to make decisions. Effective communication and collaboration is also prioritized in Lean. Lean software development comes with some limitations as well. It is not scalable for larger projects. It requires many documentation and can lead to very complex process that can be overwhelming for team members.

Kanban is one of the most simple and easy to implement project management framework. It is built on the philosophy of continuous improvement where each task are organized in phases or stages. The main element of Kanban framework is "Kanban Board" which visualizes the workflow. Team can get at-a-glance insight which work items in which stages. Kanban Boards are customizable according to the organization needs and teams can put "Work In Progress" limit to make the project manageable at a time. It helps to focus in the ongoing work and avoid burnout. But the simplicity of Kanban maybe not efficient in complex projects which needs proper planning, scheduling and co-ordination. Kanban is focused on continuous improvement and responding to change, it will be less effective for long-term planning and lacks of continuous feedback. It does not have any time-boxed period like Sprint. It delivers works continuously not iteratively which can be difficult where functional increment is necessary.

Scrum works very well for complex and dynamic software projects. It divides the entire project into time boxed period called Sprints. It is really flexible, adaptive way to manage projects. There are three important roles in Scrum- Product Owner, Scrum Master, Development Team. Scrum is useful for breaking down large projects into smaller parts, as well as testing and reviewing during Sprint Review. It also prioritizes feedback from customers. Short sprints enable adjustments based on client feedback. Daily Scrum meetings allow the team to stay up to current on their progress. Overall, Scrum project management works well for dynamic projects. However, like with every project management system, it has limits. Scrum implementation requires experienced team members. Training everybody on how to efficiently implement scum may be time-consuming. It is more effective in smaller groups and difficult to scale for larger team. Scrum is more effective for dynamic projects, but it is not appropriate for projects requiring predictability and a well-defined strategy.

Extreme Programming prioritizes client satisfaction and producing high-quality code. Extreme programming works best when the requirements are initially ambiguous and likely to change every few months. Extreme programming is used to lower project risk, especially for new features that might risk the entire software team.[6][7]Extreme programming includes writing high quality code through practices like Test-Driven-Development, continuous integration that improves reliability, maintainability. It also offers pair programming, daily stand ups and customer collaboration in the development phase for effective communication between customers and development team. Like any other Agile framework it frequently releases functional increments that enables customer to see the progress and provide feedback. But there are some drawbacks of Extreme Programming. In order to continuously adapt the system to the customer's needs, it requires a lot of efforts and perseverance. Practicing Extreme programming requires additional expenses due to pair programming and it is not suitable when customers and developers are working remotely. It also adds adequate amount of stress because of tight deadlines when code needed to designed and tested in the same day.

1.2 Comparison between Agile and Traditional Methods

- Conventional models, like the waterfall model, follow a preset and specific sequence of steps. Its structure is straightforward because all projects must follow the same set of procedures. This includes defining the requirements and designing, implementing, testing, and maintaining the system. Before moving on to the next phase, a team must finish the previous one completely, highlighting any obstacles.[8] Agile methods, on the other hand, are becoming more and more popular in dynamic fields like software development since they are flexible and iterative in nature. Agile approaches like Scrum, Extreme Programming, and Kanban place a strong emphasis on working with clients to adapt to change and deliver functional increments on time and effectively. These techniques are particularly effective in situations where the criteria are unclear or flexible.[9]
- Traditional project management relies on formal communication and documentation to discuss project needs and status. Agile project management encourages team members to communicate and collaborate in person.
- Traditional project management employs a predetermined strategy that is difficult to adjust. The linear, methodical technique and extended project phases make adjustments difficult to implement, and any divergence from the plan is viewed as a failure. However, agile project management allows for changes to be made as the project progresses.
- Traditional techniques may have a hierarchical team structure with well-defined roles and responsibilities. Agile approaches encourage self-organizing, cross-functional teams that work closely together throughout the development process. Team members frequently have overlapping duties and work together to achieve project objectives.
- Traditional techniques frequently include extensive initial planning and risk analysis, with mitigation tactics employed early in the project. Agile methodologies reduce risk through incremental delivery, allowing teams to discover and resolve issues early

in the development process. Agile teams accept uncertainty and alter their plans in response to feedback and changing requirement.

1.3 Agile Project Management Frameworks

Agile methodologies have revolutionized the landscape of software development, offering iterative and collaborative approaches to project management that prioritize adaptability, responsiveness, and continuous improvement. Within the realm of agile, various frameworks have emerged, each with its own principles, practices, and methodologies aimed at enhancing team efficiency, product quality, and customer satisfaction. Based on the 16th Annual State of Agile Report, Scrum continues to lead, rising from 58% to 87% since the 14th survey. Kanban surged from 7% to 56%. Scrumban grew from 10% to 27%. Extreme Programming increased from 1% to 7%. Lean startup saw a jump from 1% to 10%. Iterative methods quadrupled from 4% to 20%.[10]

1.3.1 Lean

Lean management is a concept that seeks to maximize value for consumers while eliminating waste. It evolved from Toyota's manufacturing method and has subsequently been adopted in a variety of industries. Womack and Jones (1996) developed the five principles of lean thinking, which are: value, value stream, flow, pull, and perfection. These five principles are crucial for developing lean processes and, eventually, lean enterprises. In lean, value is defined as everything that the customer values, whereas in regular business, value is frequently viewed as a monetary measurement. Value is only significant when presented in terms of a specific product that fits the needs of the customer at a particular cost and time. Waste, on the other hand, can be defined as anything an organization does that does not provide value to the end the customer. Wasteful operations are common in every process; yet, reducing waste is an essential component of the lean method.[11] Lean management strategies such as 5S (sort, set in order, shine, standardize, and sustain), Kaizen (continuous improvement), Just-in-Time (creating only what is required, when it is needed), and Kanban (visualizing workflow) are often used. Lean management emphasizes continual improvement, waste minimization, and customer attention in order to effectively offer high-quality goods or services.

1.3.2 Kanban

Kanban, derived from the Japanese phrase for "visual board," was developed by Toyota in the 1950s as a scheduling method for just-in-time manufacturing. In 2007, the uppercase "Kanban Method" emerged, refining Kanban ideas for wider use. Kanban was originally intended for manufacturing, but its productivity gains quickly expanded to IT, software development, R&D, and other areas. Kanban is used in software development to visualize work items on a Kanban board, which generally consists of columns indicating distinct phases of the workflow (for example, "To Do," "In Progress," and "Done"). Work items, represented by cards, travel between these columns as they advance through the workflow.

The Kanban approach recommends breaking down complicated team operations into smaller segments to facilitate validation. Kanban prioritizes tasks, sets delivery deadlines, and manages workflow to focus on what has to be done and when. Furthermore, Kanban limits the amount of work in progress to match the speed of the slowest step. By doing so, modifications will have a less overall impact while boosting teamwork.[12]

1.3.3 Extreme Programming

Extreme Programming (XP) promotes customer satisfaction by delivering essential software quickly, adjusting to changes. It enhances projects by promoting greater communication, simplicity, feedback, respect, and courage. XP includes regular communication with customers and peers, keeping designs simple, testing early for feedback, delivering swiftly, and being flexible to change.[13]

The Extreme Programming framework consists of five main stages: planning, designing, coding, testing, and listening, which are used to continually improve the development process. XP prioritizes interpersonal skills such as communication, teamwork, and feedback. Customers have important roles by establishing requirements, providing feedback, making essential choices, and leading the development process, whilst programmers perform tasks, trackers/managers guarantee project alignment, and coaches guide XP methods. By adopting these responsibilities, XP teams create software that fulfills stakeholder requirements.[14] Beck defined twelve principles that describe the XP strategy.[15] These practices include planning games, small releases, metaphors, simple design, tests, refactoring, pair programming, continuous integration, collective ownership, on-site customers, 40-hour weeks, and an open workspace, which together assure the delivery of high-quality software. XP emphasizes simplicity, common responsibility, and adherence to code standards, which promotes maintainability and scalability.[15, 16]

1.3.4 Scrum

Scrum is an agile framework for creating and maintaining complex products in order to generate software that fulfills business requirements incrementally.[16] Scrum is an agile framework for creating and maintaining complex products in order to generate software that fulfills business requirements incrementally. Scrum has three key roles: the Scrum Master, who facilitates and eliminates barriers; the Product Owner, who represents stakeholders and manages priorities; and the Development Team, which self-organizes and remains responsible for producing quality work. Collaboration among these positions is essential for project success.

Scrum framework contains backlogs, which specify functional needs in order of priority and can change throughout development. Sprints, which usually span 14-30 days, are time-boxed work units with the goal of meeting backlog requirements incrementally.[17] The tasks for sprint is decided by sprint backlog, it contains all the requirements for current sprint, and which could change during development.[18] Scrum meetings, facilitated by a Scrum Master, occur daily for 15 minutes and focus on answering three questions: what was done since the last meeting, what obstacles were encountered, and what will be accomplished by the next meeting.[17] Software

is delivered in increments, with the customer's desired functionality included in each increment. Customers may adjust their needs in each iteration. The scrum approach makes it extremely simple to manage change requirements.[18]

2 Related Study

2.1 Project Management Tools

Project management applications have become essential aids for teams and individuals managing project planning, organization, and execution on a variety of scales and complexity. These platforms offer a wide range of features and functionalities targeted at streamlining workflows, encouraging collaboration, and increasing overall project success. In this review, we will look at some of the most common project management systems used by teams from various sectors, emphasizing their essential features and benefits. This review focuses on four widely-used project management systems—Trello, JIRA, Asana, and Monday.com—highlighting their key features, advantages and limitations.

2.1.1 Trello

Findings:

Trello was one of the first Kanban-based applications, providing simple yet effective project visualization and drag-and-drop task management. It's key features include:

- **Kanban Boards:** Kanban Boards are the main element of any project in Trello. Each board represents a stage or category of a project, and cards can be added to those boards.
- **Subtasks and Progress Bar:** Each task can be divided into sub-tasks. When a subtask is completed, the user can mark it as completed.
- **Custom Fields in Cards:** Users can add custom fields such as text, numbers, dropdown boxes, and checkboxes.
- **Activity Log:** When a user updates or modifies a card, all the members in the project can see those updates using the activity log.
- **Group Assignment:** Each card can be assigned to more than one member.
- **Automation:** Users can create custom rules and commands to reduce repetitive tasks. Users can create rules, custom buttons, due date commands, and calendar commands.
- **Integration with Other Applications:** Trello integrates with a wide variety of third-party applications such as GitHub, Slack, Google Drive, etc.

Advantages

- Trello uses technology that operates in real-time. Therefore, all the members in a project can get updates instantly.
- It is beginner-friendly. Users do not need to have any prior project management knowledge to use it.
- Trello has a responsive design that ensures a good user experience on both computers and mobile devices.
- It eliminates the need for taking personal notes. Their free plan is also a great tool for individuals, freelancers, and small teams.

Limitations

- Trello is heavily dependent on third-party applications for enhanced functionalities. This may increase costs and complicate things further.
- It is not a great Agile project management tool. It does not have any predefined roles for Agile teams and does not follow Agile principles.
- It does not have advanced graphs, which are essential for Agile teams.
- It is not scalable for larger projects.

2.1.2 Asana

Findings:

Asana project management software provides features such as tasks, projects, discussions, and dashboards to help organize work. It also provides Team Pages for sharing ideas and conversations, as well as a 'Smart Box' for receiving only critical project updates and avoiding unnecessary messaging.[19] Asana's Workflow Builder is a visual tool that allows users to build customized workflows and collaborate seamlessly with team members.[20] It's key features include:

- **Task Management:** Asana excels in task management with robust features that enable task creation, assignment, subtask organization, recurring task scheduling, priority setting, and task dependencies. These capabilities ensure efficient project management and clarity in execution.
- **Communication:** Asana includes a built-in messaging application that facilitates seamless collaboration among team members. It supports real-time interaction and quick decision-making, enhancing communication effectiveness across global teams.
- **File Management:** Asana simplifies file storage and management with a centralized repository for project files. Team members can easily upload, access, and collaborate on documents, promoting efficiency and reducing version control issues.
- **Gantt Chart:** Integrated within Asana, the Gantt Chart feature offers visual project planning. Users can create timelines, set task dependencies, and track

progress effectively. It provides a comprehensive overview for informed decision-making and resource allocation.

- **Customizable Workflows:** Asana allows teams to customize workflows by creating custom fields, defining task stages, and establishing rules. This flexibility streamlines processes, ensures consistency, and aligns with unique project management methodologies.

Advantages

- Asana features a user-friendly interface designed for intuitive navigation.
- It integrates seamlessly with popular project management apps like Google Drive, Microsoft Teams, and Slack.
- Asana prioritizes collaboration with features like task alerts, personal task visibility, and task comments.
- Asana's task management is efficient, offering comprehensive task overviews, easy prioritization, assignments, and real-time updates.

Limitations

- It is not suitable for Agile Project Management.
- Asana's interface and features can be overwhelming for new users, potentially leading to a steep learning curve without proper onboarding.
- Task assignment in Asana is limited to one person per task, which can be restrictive for collaborative efforts or when team members are unavailable.
- The extensive features of Asana may make even simple projects feel complex, which may not align with the needs of smaller teams seeking simplicity.
- Asana lacks built-in time tracking, necessitating integration with third-party tools for comprehensive time management solutions.

2.1.3 JIRA

Findings:

JIRA is a prominent project management tool designed mainly for agile software development teams. Within the application, these teams have the option of creating scrum boards and kanban boards. It is a great tool for adopting Agile methodologies. It provides all the features as Trello. JIRA's other key features include customizable developer tool integrations, as well as issue and bug tracking. This well-established and dependable product, which is used by large enterprises throughout the world, has a strong user community.[21] It's key features include:

- **Scrum and Kanban Project Management:** JIRA provides Kanban Boards, Backlog, and Sprint planning for practicing Scrum and Kanban frameworks.

- **Time Tracking:** Users can set estimated time to complete a task. When a task is completed either partially or fully, users can log the amount of time spent on the task each day.
- **JIRA Query Language (JQL):** Users can create complex searches using JQL to find a set of issues. Queries consist of commands such as "Select," "And," "Or," "Not," etc., and relational operators based on factors like priority, status, assignee, created, updated, and other custom fields.
- **Reports:** Reports in JIRA provide visual representations that help teams measure progress, analyze work, and make data-driven decisions.
- **Project Milestones:** JIRA enables users to define project milestones by grouping a set of tasks together under one epic.
- **Workflow Customization:** Teams can customize workflows in JIRA by adding new stages and establishing transitions between these stages.

Advantages

- JIRA is a highly customizable tool that provides users with custom fields, workflow, and reports.
- JIRA is scalable for small and larger teams.
- JIRA integrates with other Atlassian tools such as Trello, Confluence, and Bitbucket, as well as other third-party tools.
- It is a great tool for practicing Agile methodologies such as Scrum and Kanban.
- It provides advanced reports to analyze performance, progress, and potential bottlenecks.

Limitations

- Managing projects using JIRA is quite difficult for new users. It requires a lot of time to learn to use JIRA, especially for those who are unfamiliar with software project management methodologies.
- JIRA offers many pricing plans, although it is costly for smaller teams. License fees and charges for plugins can further increase the budget.
- JIRA has limited built-in test management features.
- JIRA has a lot of features and a complex interface that may complicate the entire process.
- Permission and access control in JIRA can be complex.
- JIRA does not have a good communication tool.

2.1.4 Monday Dev

Findings:

Monday.com is a widely used cloud-based work operating system and project management platform that enables teams to plan, organize, and monitor work collaboratively. It offers a visual and user-friendly interface for creating and managing tasks, projects, and workflows. Monday Dev is an Agile Project Management tool designed for software development teams. It's key features include:

- **Scrum and Kanban Project Management:** JIRA provides Kanban Boards, Backlog, and Sprint planning for practicing Scrum and Kanban frameworks.
- **Main Table:** Main Table initially contains all the tasks of a project. User can create groups and move selected tasks from Main Table to appropriate group.
- **Custom View:** User can filter tasks from the Main Table and save that view.
- **Bugs Queue:** Users can manage all the bugs or flaws they encounter during development. User can create Bugs Queues Board where bugs can be organized by working status. Users can set priority, due date, status and other custom fields to track and solve bugs.
- **Retrospective:** Sprint Retrospective Boards can be used each stage of sprint and retrospective meetings. All the team members can put their thoughts and reflect on the previous sprints and decide accordingly. Members can vote on each retro topic to discuss, improve, or maintain the current practices.
- **Board's Discussion:** All the members can communicate through this feature. It is a great feature for team collaboration.

Advantages

- Monday.com is good for Agile Project Management. It supports Kanban and Scrum Project Management.
- It is lightweight and does not have too many features like JIRA.
- It is comparatively easier to learn for beginners.
- Live Chat option is a great communication tool.
- It also provides integration and automation.
- User interface is modern, user-friendly, and minimalistic.

Limitations

- Monday Dev does not have any free plans. JIRA provides free plans for Agile management for smaller teams and freelancers.
- Monday Dev does not have a card view to give a quick insight into all the attributes of a task.

- Monday Dev does not have any built-in or integrated Test Management Tool.
- Monday Devdoes not provide advanced reports like JIRA.

2.2 Problem Statement

- **Scrum and Kanban Project Management:** Trello and Asana do not provide project management features specifically designed for Scrum and Kanban frameworks.
- **Communication:** Trello and JIRA lack a built-in messaging application for direct communication within the platform.
- **Advanced Reports:** JIRA offers users advanced reports to analyze performance, efficiency, and identify bottlenecks. Monday.com also provides a BurnDown Chart for visualizing progress in a Sprint. Trello and Asana provide basic charts.
- **Test-Driven Management Tools:** While JIRA integrates with many testing tools, these integrations are generally not optimized for Test-Driven Development (TDD) methodologies.
- **Role-Based Authorization:** Scrum defines three roles—Product Owner, Scrum Master, and Development Team. However, JIRA, Monday.com, Asana, and Trello offer permission features but do not typically include a role-based authorization system specific to Scrum roles.
- **Daily Scrum and Sprint Review:** These applications do not provide Daily Scrum and Sprint Review functionalities with proper Role-Based Authorization.

2.3 Objectives

This study aims to pinpoint the essential features for managing software projects within Agile frameworks and how they align with Agile principles. By evaluating various project management tools, it seeks to provide insights to enhance project management efficiency and support successful project outcomes in software teams. The objectives are as follows:

- To create a user-friendly interface for seamless project management.
- To implement Kanban, Scrum, and TDD processes for efficient software development and product maintenance.
- To facilitate seamless communication among team members, managers, and stakeholders.
- To integrate Gantt Charts and various graphs/charts for better visualization to efficiently develop and maintain a software application.

3 Methodology

After conducting a comprehensive review of four popular project management software systems (Trello, JIRA, Monday.com, and Asana), as well as two test management platforms (Testlink and TestMonitor), we identified their relative strengths and limitations. Taking these results into consideration, we want to create a project management system that is especially designed to satisfy the requirements of agile teams. This proposed system will integrate both Scrum and Kanban frameworks for project management, supplemented by a test management tool that is meticulously aligned with the principles of Test-Driven Development (TDD).

3.1 High Level Architecture of Proposed System

In the proposed system, users will be able create two different project types: Scrum and Kanban. After successful login, users will be led to the homepage, where they will be able to access a variety of features. Users will be able view existing projects, create new projects, and update personal information.

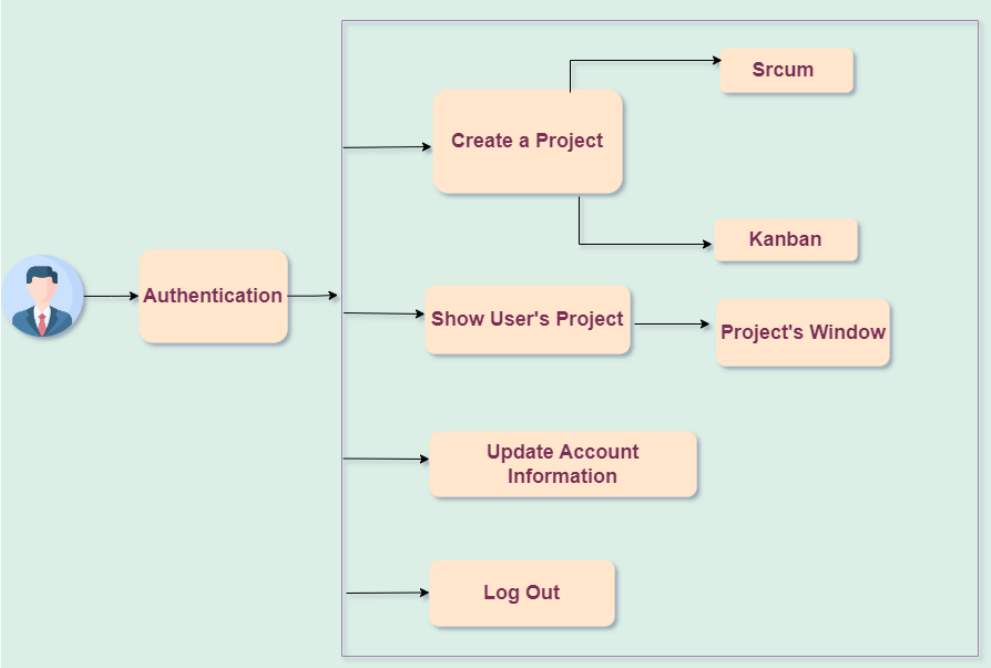

Fig. 1: Home Page

3.1.1 Kanban Module

The proposed system's Kanban module is aimed to help users view, organize, and manage their work activities more efficiently. This module, inspired by the Kanban

framework's principles,offers a flexible and simple method for tracking task progress through various stages of completion. The main features of the Kanban module are briefly discussed here.

Fig. 2: kanban Module

Kanban Boards

Users can develop and configure Kanban boards based on their workflows and procedures. Each board has columns for different levels of work, such as "To Do," "In Progress," and "Done."

Task Cards

On the Kanban board, tasks are represented as cards that contain important information such as the task title, description, assignees, start and due dates, tags, priority, dependency, sub-tasks, and status. Users can quickly move cards across columns to update their status and progress.

Work in Progress (WIP) Limits

Users may define WIP limits for each column to avoid overloading and keep the workflow balanced. WIP restrictions allow teams to focus on finishing jobs effectively and promotes a smoother flow through the process.

Role Based Authorization

Kanban roles are often more flexible than in other techniques such as Scrum. The emphasis is on the flow of work rather than particular roles. In the proposed system, some team members would be assigned the 'Admin' role to control the workflow. Admins can drag and drop between boards. Only those who are assigned of a given card have the ability to update its properties. Clients can only observe the workflow but they cannot alter anything personally. If they wish to change something, they must consult the team.

Reports

The proposed system incorporates five forms of graphical representations to give full insights on project performance and development. These graphs include Gantt Chart, Control Chart, Pie Chart, Recently Created Issue Chart, and Resolution Chart. By combining these visualizations, teams and stakeholders can quickly assess team performance and track job completion within specified timeframes. This makes it possible to analyze efficiency more thoroughly, which helps in decision-making and improves planning for next projects. Further details about the reports will be provided in later parts.

3.1.2 Scrum Module

The Scrum module implements the ideas of the Scrum framework to agile project management, providing a structured approach to iterative development cycles. It enables teams to collaborate effectively, adapt to changing demands, and prioritize activities to ensure project completion. This section briefly discusses the key aspects of the Scrum module.

Fig. 3: Scrum Module

Product Backlog

The product backlog is a list of all the tasks that must be executed for a product. In most cases, the product owner collaborates with the development team to manage and prioritize the backlog.

Sprint Backlog

The sprint backlog is a subset of the product backlog that includes the work items chosen by the development team for implementation within a given sprint.

Active Sprint

Tasks in an Active Sprint are grouped by state (e.g., To Do, In Progress, Hold, Testing, Done). A task can only progress if its predecessor has been performed. Users can change the status of tasks by clicking on buttons or dragging and dropping them between status boards.

Completed Sprints

User can see information of completed sprints. Product owner or Scrum master can restore any card to the Product Backlog or Sprint Backlog.

Scrum Review Board

Daily scrum meeting summaries and sprint meeting notes will be archived here for future reference.

Reports

The proposed system includes Gantt Chart, Velocity Chart, Burn Down Chart, Pie Chart, Recently Created Chart and Sprint Report enabling a thorough analysis of team performance throughout each sprint and the project lifetime.

4 Result and Discussion

4.1 Reports

4.1.1 Gantt Chart

Gantt Charts depict project schedules, tasks, and their dependencies over a period of time. The project schedule is shown on the chart's horizontal axis, which is often divided into days, weeks, and months. Each task in the project is represented by a horizontal bar or block on the Gantt chart. These bars are classified into three categories: normal (indicating present work/tasks completed on time), overdue (tasks completed over their due date), or finished after the due date. All the Boards with their cards are visible in the Gantt Chart. Dependencies between tasks are represented by arrows or lines connecting the bars, demonstrating the task sequence. Progress bars also track task progression, which is determined depending on the completion of sub-tasks. Each sub-task is allocated a point value, which indicates its relative time demand or complexity.

The Kanban module displays all project boards in Gantt Chart style. The Gantt Chart for the Scrum module only includes cards that have been committed to completion during the sprint. Furthermore, only the Admin in Kanban projects and the Scrum Master in Scrum projects have the authority to determine card dependencies and set start and deadline dates.

In terms of visual representation, cards that are past due are highlighted in red, whereas completed cards that were not completed by the deadline are highlighted in pink. Green cards indicate that they are on schedule but have not yet reached their due dates.

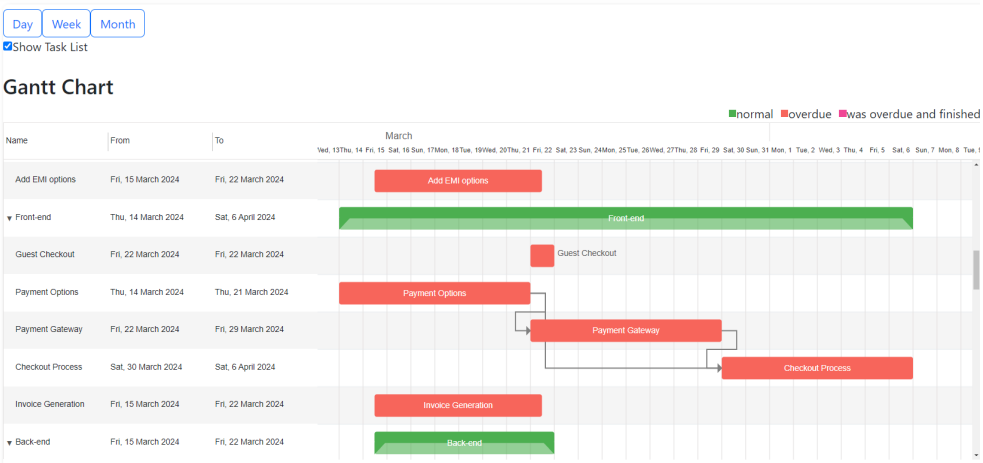

Fig. 4: Gantt Chart

4.1.2 Pie Chart

A pie chart in a project management application displays data in a circular arrangement, with each "slice" representing a piece or percentage of the total. In the proposed system users in the Kanban module may create pie charts depending on priority, assignees, and boards. These charts may be generated for the entire project or specific boards.

Furthermore, the Scrum module allows users to examine the percentage of remaining tasks in the backlog as well as the distribution of tasks within each sprint.

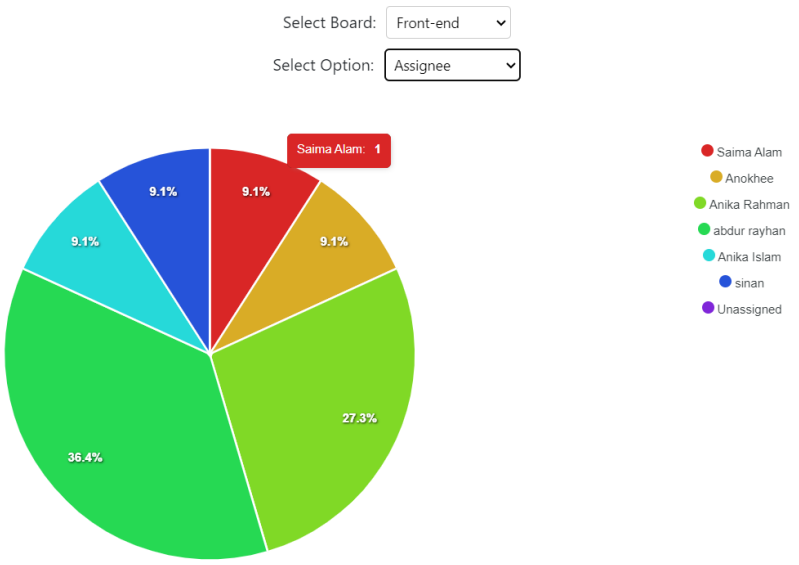

Fig. 5: Pie Chart

4.1.3 Control Chart

A control chart is a statistical tool for tracking and analyzing process change over time. It is often a line chart with data points reflecting a specific process or project parameter (such as cycle time, lead time, or throughput) across time. Examining the data patterns in the control chart allows us to determine if the process is 'in control' or 'out of control.'

The control chart displays the cycle time or lead time for your project. It maps the time spent by each issue in certain statuses over a set period of time. Cycle time is the amount of time spent working on a problem, often from when work begins to when work is done, but it can also include any additional time spent working on the issue. Lead time is similar to cycle time, but it refers to the period between when an issue is logged, rather than when work begins, and when work on that problem is completed.[22]

The control chart report indicates the extent to which work items spend in particular states over a time frame specified by the user. The control chart has two types of issues: single or clustered(Issues created on the same day and having equal elapsed time across the selected boards.) The red line on the control chart represents the average time it takes your team to move an issue through the selected statuses. The orange line graph shows the rolling average time it takes for your team to progress an issue through selected statuses. The rolling average is computed as follows: average of (cycle times of the current issue + cycle times of X issues before the current issue + cycle times of X issues after the issue.)[23]

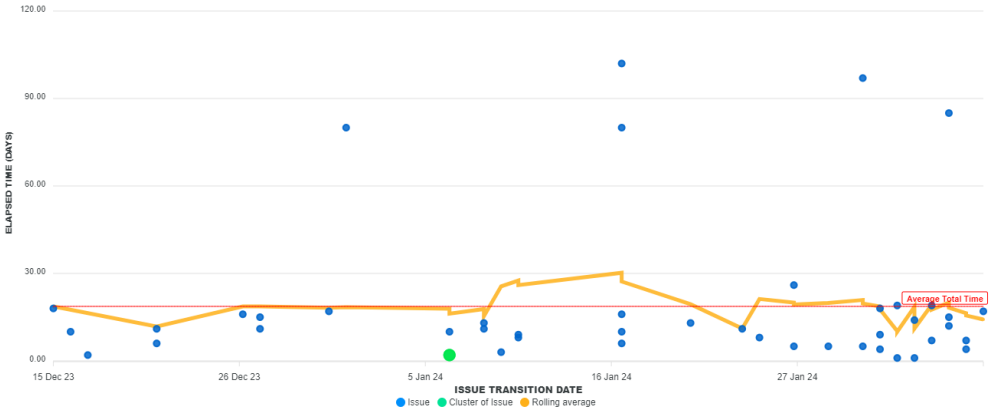

Fig. 6: Control Chart

4.1.4 Recently Created Issues Report

The "Recently Created Issues Report" displays the amount of issues for a project that were produced and resolved within a given time range (e.g., days, weeks, months, or 6 months). This report helps you assess whether the team is properly handling incoming tasks. The report normally shows a bar chart where

- X axis represents time. (Days, Weeks, Months)
- Y axis represents number of issues.

In each bar,

- Red color represents the number of unsolved issues.
- Green color represents the number of solved issues.

Users may see how many tasks were created and completed in a specific time period by looking at this chart. It assists in determining if earlier-created tasks are being quickly addressed. On the other hand, an excessive amount of unfinished work might be a sign of poor task management. The figure below is an example of a Recently Created Issues Report. It indicates poor team performance due to a significant amount of assigned work remaining unfinished for an extended period.

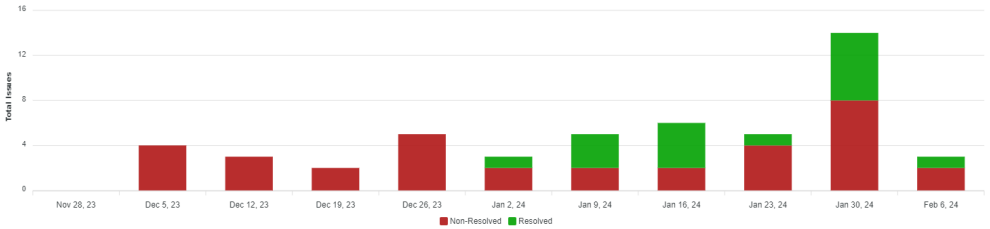

Fig. 7: Recently Created Issues Report

4.2 Resolution Time Chart

The Resolution Time Chart is a bar graph that provides information on the average amount of time it takes to resolve cards that were created within a given period. The X-axis shows time intervals (days, weeks, or months), and the Y-axis shows the average resolution time. Progress lagging is indicated by increasing bar, whereas decreasing bar chart indicates greater efficiency.

A detailed data table following the chart provides a comprehensive breakdown of card creation and resolution metrics for each time interval. This covers the quantity of cards created in a time frame(daily, weekly or monthly) and the overall amount of time needed to finish them. Administrators can use this information to perform a thorough comparative analysis between time intervals. This will help in the assessment of project effectiveness and facilitate informed decision-making.

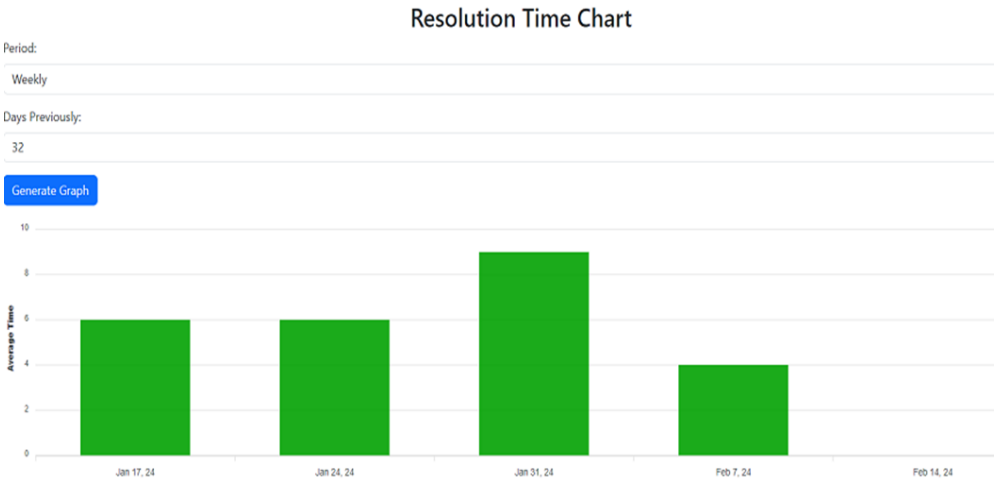

Fig. 8: Resolution Time Chart

Data Table

Period	Issues Resolved	Total Resolution	Avg. Resolution Time
Jan 17, 24 - Jan 24, 24	2	11	6
Jan 24, 24 - Jan 31, 24	1	6	6
Jan 31, 24 - Feb 7, 24	5	43	9
Feb 7, 24 - Feb 14, 24	4	14	4
Feb 14, 24 - Today	0	0	0

Fig. 9: Data Table for Resolution Time Chart

4.2.1 Velocity Chart

The velocity chart shows how much work a Scrum team commits to at the start of each sprint and how much work is accomplished during the sprint. The sprints are represented by the x-axis, and the statistics used to estimate stories/issues are shown on the y-axis. In the approach we propose, we use story points for estimation—a relative value that indicates a task's complexity, which might differ among developers. The orange bar in each sprint represents the total number of story points committed at the beginning of the sprint. The blue bar in each sprint represents the total story point completed throughout the sprint.

The Velocity chart is an effective tool for predicting both the average work completion within a sprint and the number of sprints necessary to complete a project. Consider a team assessing the latest eight sprints, with completed story points of 21.5, 20.5, 19, 22, 21, 18, 20, and 21. The average of these tale points is 20.375. As a result, the team should realistically expect to complete around 20 narrative points during the next sprint. Assuming the project's total story points are 400, and 163 have already been accomplished over the eight sprints, the team may estimate the number of more sprints required. This computation is done by subtracting the finished story points from the total (400 - 163) and dividing by the average story points completed each sprint (20.375). This produces an estimate of around 11 more sprints needed to complete the project at the present pace.

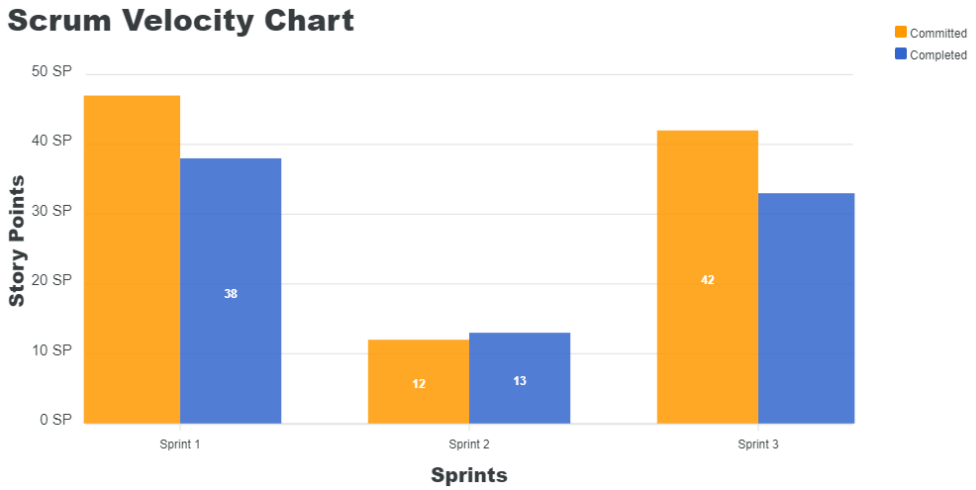

Fig. 10: Velocity Chart

4.2.2 Burndown Chart

A burndown chart displays the amount of work performed during an epic or sprint, as well as the overall amount of work remaining. The burndown chart's x axis indicates

each day of the sprint, while the y axis represents the estimation statistic. In the proposed system, we use story points as an estimation statistic.

A burndown chart has two line charts. The orange line is an approximation of where your team should be, assuming linear improvement. The blue line indicates the overall amount of work remaining in the sprint. If the blue line is below the guideline, it indicates that the team is on schedule to finish all tasks during the sprint.

The burndown chart is an extremely useful tool for obtaining insight into the team's performance. If the team frequently completes their work before the sprint's completion date, it may imply that they underestimated their capabilities, resulting in under commitment. On the contrary, if the team frequently fails to meet its projection, it may indicate that they have overcommitted, taking on more work than can be reasonably completed within the sprint period.

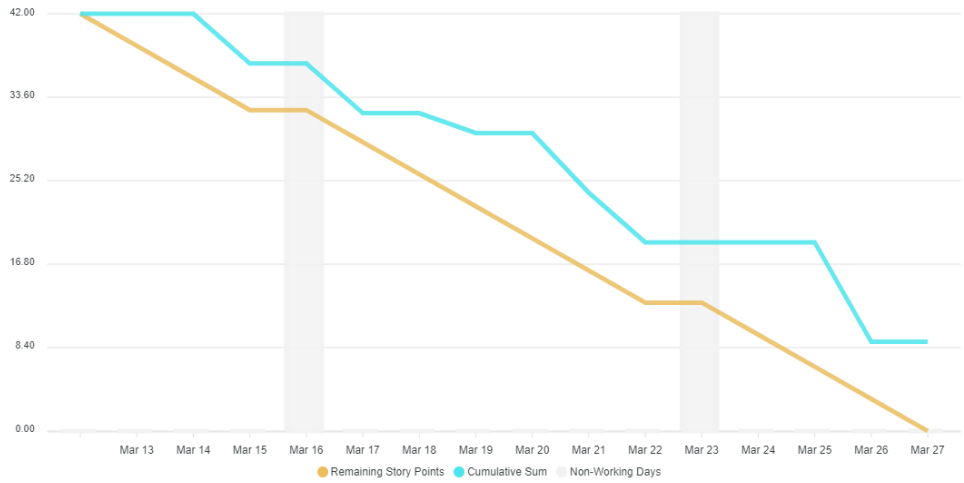

Fig. 11: Burndown Chart

4.2.3 Sprint Report

The sprint report provides an overview of the sprint. It displays information about finished, uncompleted, and deleted cards during a sprint.

4.3 Live Issue Board

In the proposed system, we've integrated a real-time communication feature called the "Live Issue Board" using Socket.IO. This tool allows team members to express questions and have discussions with their colleagues as soon as possible. In this context, team members can "like" responses that they believe are the most accurate. Furthermore, any team member may customize their notification settings for individual discussions. When the notification option is activated, users will get email

alerts. The Live Issue Board also contains a notice section where project managers or administrators may make important announcements.

4.4 Management Tool for Test-Driven-Development

Test-Driven Development (TDD) is an approach to development in which developers build tests for small sections of functionality before developing actual code. The technique involves repeatedly creating a failed test, implementing the code to make it pass, and then modifying the code to improve its design and maintainability. TDD ensures that developers' code meets the necessary standards while producing manageable code. The proposed system includes a management tool for Test-Driven Development (TDD). Each card or task has been divided into smaller subtasks. Before writing any code, developers create test cases for each subtask. Each test case goes through three phases: red (showing test case failure), green (representing test case passed), and blue (refactoring the code). Developers can mark a card/task as done after all subtasks have been performed and all related test cases have passed.

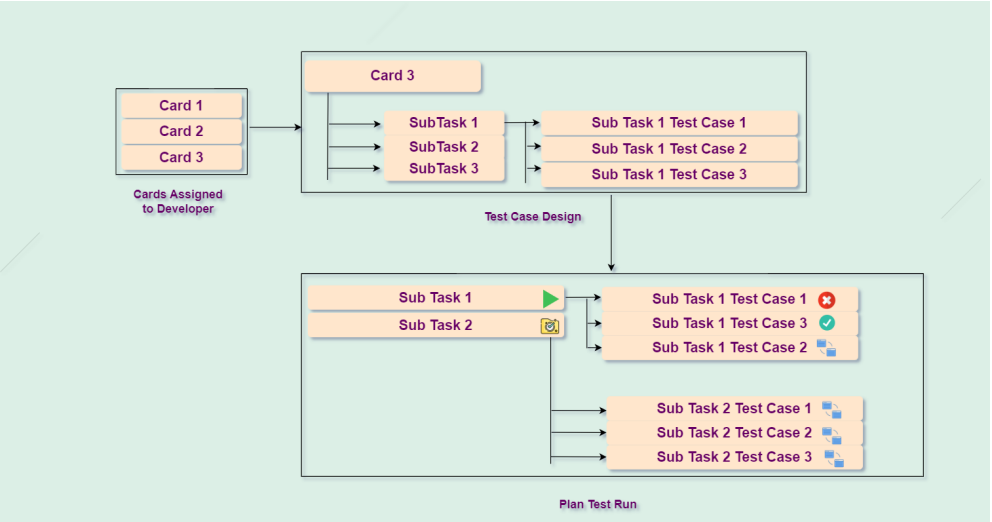

Fig. 12: High Level Architecture of Management Tool for Test-Driven-Development

4.5 Discussion

The primary goal of the proposed system is to provide agile project management capabilities to software teams. We've included several standard features like the Kanban Board, Product Backlog, Sprint Backlog, Gantt Chart, and Reports that are currently available in many established project management systems. In addition, we've introduced tools to improve management and communication. For example, assigning points to sub-tasks allows for a more precise measurement of task progress. The Scrum

Review Board records summaries of daily scrum and sprint sessions for future reference. We've also included a Management Tool for Test-Driven Development to help you practice this process. Furthermore, there is a Live Issue Board where developers can post questions to team members.

Table 1: Comparison Among Different Project Management Applications

Features	**Trello**	**JIRA**	**Asana**	**Monday.com**	**Proposed System**
Kanban Board	Yes	Yes	Yes	Yes	Yes
Reports	Basic	Advanced	Basic	Basic	Advanced
Scrum Project Management	No	Yes	No	No	Yes
Completed Sprint Boards	No	No	No	No	Yes
Scrum Review Board	No	No	No	No	Yes
Multiple Assignee in Tasks	Yes	Yes	No	Yes	Yes
Points to Sub Tasks	No	No	No	No	Yes
Gantt Chart	No	Yes	Yes	Yes	Yes
Live Issue Board	No	No	No	No	Yes
Management Tool for Test-Driven-Development	No	No	No	No	Yes

5 Conclusion

In conclusion, our goal is to provide effective software development and product maintenance by offering a user-friendly interface that combines Scrum, Kanban, and Test-Driven Development (TDD) techniques. Through the facilitation of smooth communication between managers, stakeholders, and team members, our goal is to improve transparency and collaboration throughout the project lifetime. Better visualization will also be provided by the integration of Gantt charts and other graphs and charts, which will help with the effective creation and upkeep of software applications.

5.1 Limitations

Project development is a continuous process. Our major objective is to make the program as user-friendly as possible. One significant limitation is that the proposed system is not appropriate for non-IT enterprises. Another drawback is that our project management system lacks a reliable file-sharing feature. We chose local file sharing over paid cloud storage due to the lack of available options. Typically, project management solutions work with third-party applications like Google Drive, Dropbox, and GitHub etc. Unfortunately, our present system cannot support integration.

5.2 Future Works

- Enhancing adaptability for non-IT organizations through user-friendly modifications.

- Implementation of automation features aimed at streamlining operations for general users.
- Integration with third-party applications to broaden accessibility and utility.
- Introduction of web-push notification functionality to enhance communication efficiency.

References

[1] Mihalache, A.: Project management tools for agile teams. Informatica Economica **21**(4), 85–93 (2017)

[2] Manifesto, A.: Manifesto for agile software development (2001)

[3] Sacolick, I.: A brief history of the agile methodology (2022). https://www.infoworld.com/article/3655646/a-brief-history-of-the-agile-methodology.html

[4] Quick, L.: History of Agile Methodology: How it was Developed (2024). https://www.knowledgehut.com/blog/agile/history-of-agile

[5] Poppendieck, M., Poppendieck, T.: Lean Software Development: An Agile Toolkit: An Agile Toolkit. Addison-Wesley, ??? (2003)

[6] Wells, D., Wells, D.: When should I use Extreme Programming. http://www.extremeprogramming.org/when.html

[7] McDonald, K.: What is Extreme Programming (XP)? — Agile Alliance (2023). https://www.agilealliance.org/glossary/xp/

[8] Khan, S., Mahadik, S.: A comparative study of agile and waterfall software development methodologies, 399–402 (2022) https://doi.org/10.48175/IJARSCT-5696

[9] Daraojimba, E.C., Nwasike, C.N., Adegbite, A.O., Ezeigweneme, C.A., Gidiagba, J.O.: Comprehensive review of agile methodologies in project management. Computer Science & IT Research Journal **5**(1), 190–218 (2024)

[10] 16th Annual State Of Agile Report (2022). https://info.digital.ai/rs/981-LQX-968/images/SOA16.pdf

[11] Kaikkonen, H., Härkönen, J., Haapasalo, H., Rodríguez, P.: Supporting lean software enterprises with agile development methods (2014)

[12] Rasool, N., Khan, S., Haseeb, U., Zubair, S., Iqbal, M.W., Hamid, K.: Scrum and the agile procedure's impact on software project management. Jilin Daxue Xuebao GongxuebanJournal Jilin Univ. Eng. Technol. Ed **42**, 380–392 (2023)

[13] Wells, D.: Extreme Programming: A gentle introduction. http://www.

extremeprogramming.org/

[14] Extreme programming: values, principles, and practices (2021). https://www.altexsoft.com/blog/extreme-programming-values-principles-and-practices/

[15] Beck, K.: Extreme Programming Explained: Embrace Change. addison-wesley professional, ??? (2000)

[16] Al-Zewairi, M., Biltawi, M., Etaiwi, W., Shaout, A., *et al.*: Agile software development methodologies: survey of surveys. Journal of Computer and Communications **5**(05), 74 (2017)

[17] Faniran, V.T., Badru, A., Ajayi, N.: Adopting scrum as an agile approach in distributed software development: A review of literature. In: 2017 1st International Conference on Next Generation Computing Applications (NextComp), pp. 36–40 (2017). IEEE

[18] Hayat, F., Rehman, A.U., Arif, K.S., Wahab, K., Abbas, M.: The influence of agile methodology (scrum) on software project management. In: 2019 20th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), pp. 145–149 (2019). IEEE

[19] Top 21 Project Management Tools: 2024 Guide. https://www.wrike.com/project-management-guide/faq/what-are-project-management-tools/

[20] Haije, E.G.: Top 20 Best Project Management Software in 2024 (2024). https://mopinion.com/top-20-best-project-management-software-an-overview/

[21] Özkan, D., Mishra, A.: Agile project management tools: a brief comprative view. Cybernetics and Information Technologies **19**(4), 17–25 (2019)

[22] View and understand the control chart — Jira Software Cloud — Atlassian Support. https://support.atlassian.com/jira-software-cloud/docs/view-and-understand-the-control-chart/

[23] Iterators, Iterators, Iterators: How to use the Jira Control Chart to take your business process to the next level? — Iterators (2023). https://www.iteratorshq.com/blog/how-to-use-the-jira-control-chart-to-take-your-business-process-to-the-next-level/