iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** XI **Month of publication:** November 2025

DOI: https://doi.org/10.22214/ijraset.2025.75282

ScrumZero: AI Powered Agile Workflow Optimizer

Yuvraj Desai[1], Jaydev Gupta[2], Jay Kapadiya[3], Ms. Himani Purohit[4]
[1, 2, 3, 4]*Dept of Artificial Intelligence and Data Science, Thakur College of Engineering and Technology*
Mumbai, Maharashtra, India

Abstract: Scrum-Zero is an AI-powered web application that automates the work of the Scrum Master of Agile teams, which is the subject of our paper. Scrum-Zero is a system built with FastAPI on the backend and React-like Java script frontend powered by Google generative AI engine, Gemini-2.0, with daily stand up, sprint discipline and real time to-do prioritization and bottleneck solving built in. The system maintains tasks between meetings in SQLite and makes RESTful calls on each stage of the process of task life-cycles. The AI summary module categorizes team task updates into Completed, In-Progress and To-Do and provides an organized and actionable stand-up report. Using a controlled study of 30 practitioners, we estimate the summary latency of Scrum-Zero, compare it to the accuracy of reports collected by humans [1], and learn about user satisfaction. The findings show that it takes less time to meet (average reduction of 40 percent), 85 percent of the claims to be in agreement with human summaries and that it can be used as an AI facilitator of an Agile workflow [2]. We talk about design and architecture, knowledge of the difficulties in AI implementation, and ways to expand on it to increase capacity and multimodality. Index Terms- AI Scrum Master, Agile Automation, Generative AI, Fast API, Scrum-Zero.

Keywords: AI Scrum Master, Agile Automation, Generative AI, FastAPI, Natural Language Processing (NLP), WebSocket, Task Automation, Sprint Management, Conversational AI, Intelligent Task Assignment.

I. INTRODUCTION

Agile software development techniques, most of them being Scrum, have proven to be the standard platform of contemporary management of software development due to their inherent iterative and incremental nature [3], focus on teamwork, and adaptability. The role of Scrum Master is significant to Scrum teams; it is the one that undertakes such ceremonies as daily stand-ups, sprint planning, and retrospective and ensures that the Agile Manifesto is practiced by teams. However, it is not effective when a Scrum Master is not available, well-informed, and reliable. This is especially in case of distributed teams or teams that are short of resources. Moreover, recent developments in Artificial Intelligence AI and, in particular, natural language processing, and generative models, provide new possibilities to independently automate traditionally human jobs [4]. Automation based on AI will allow you to scale, minimise human mistakes, and create ongoing help, without exhaustion and partiality. In particular, generative models, including Google Gemini and GPT models produced by OpenAI [5], have been capable of comprehending and generating structured text and including contextual awareness. It hints at the fact that a human Scrum Master can be substituted, or enhanced, by generative AI models, which can be used to support Agile workflows. In this article, the Scrum-Zero, an artificial intelligence-based web application to replace the Scrum Master, is introduced to automate the daily standup, task prioritization, and sprint tracking functions with the help of generative artificial intelligence [7]. Scrum-Zero is managed by FastAPI, a simple data persistence system of SQLite, and it utilizes the generative AI model of Google Gemini to produce summaries of tasks with actionable information. Through the automation of the duties of one of the members of the Scrum Master whose primary functions are to enhance the productivity of the team, minimize the time spent in the meeting and maintain the pace of sprints independently.

The contributions of this work are as follows:

The process of designing and implementing an AI-driven replacement system of Agile software teams Scrum Master.

Application of state of the art Generative AI to create structured and useful daily standup reports.

The evaluation of the performance and practical efficiency of Scrum-Zero with the help of quantitative measures and feedback.

The remainder of the paper is structured as follows: Section II is devoted to related work in the area of application of AI automation to Agile settings, Section III presents the architecture and approach of Scrum-Zero, Section IV provides a report of empirical findings, and Section V presents limitations and future work.

II. LITERATURE REVIEW

A. AI in Agile Project Management

The introduction of Artificial Intelligence in the sphere of Agile project management is an important shift in the way software development teams can coordinate, plan, and implement their work. Studies and practice have also paid more attention to using AI to augment and automate important Agile processes, especially in the Scrum model [1]. The traditional duties of a Scrum Master such as facilitation of such ceremonies as daily stand-ups and sprint retrospectives, compliance to Agile principles, elimination of impediments, and collaboration can be limited by human subjectivity, availability, and consistency. When done manually, these tasks can take up a lot of time and can also lead to inter-team or inter-sprint imprecision in the application of the processes.

The current research emphasizes that these roles can be given greater stability, efficiency, and scalability through the use of AI-based solutions. AI systems can be used to promote real-time decision-making, automate the routine administration, and present data-driven information on the team performance and project health by using machine learning and natural language processing. It can be quite helpful, particularly, in distributed or globally dispersed teams, in which synchronous human facilitation is difficult. The applications of AI in this field are diverse and varied, including predictive analytics to help plan sprints or automated sentiment analysis to help with retrospectives, which improves the responsive and adaptive project management environment.

B. Team Coordination Conversational AI.

In general, conversational AI systems, especially those based on large language models (LLMs) like GPT and Gemini have shown impressive abilities in cognition, processing, and production of human-like language [8]. These have provided new opportunities in improving coordination of teams within the collaborative software development environment. These systems could be used as a smart agent that could handle the allocation of tasks, distribute daily progress reports, track sprint metrics and communicate between the stakeholders using natural interfaces with language that is easy to understand.

Conversational agents facilitate a more interactive and dynamic mode of interaction by decreasing the reliance on fixed form-based inputs and fixed-dashboards to report information. Imaginatively, team members could talk and report progressively or challenge project data in common language, and the system would be in a position to process such feedback, get actionable information, and initiate suitable workflows or provide contextual responsiveness. This is not only easier to use and lowering the learning curve that would otherwise come with the use of complex software tools but also useful in ensuring that constant interaction in the team is sustained, which is in the nature of a human facilitator but with a large scale.

C. Workflow Automation and Natural Language Interfaces.

Natural language interfaces (NLIs) have changed the way human-computer interactions [9] have been undertaken through hiding the complicated command syntax and lessening the cognitive burden to use software systems. NLIs are used in project management to enable users to operate tools in plain language, i.e. to assign tasks, create reports or update the status of items without having to use complex graphical user interfaces or dropdown lists. This has been a great change over manual input to conversational command since it makes communication very fast and lowers the chances of being misinterpreted or having errors in entries.

Furthermore, the combination of NLIs and workflow work engines allows streamlining end-to-end processes. An example is a developer can say, I am stuck on the API to log in because of bad authentication, and the system can automatically interpret this update to mark the appropriate task as blocked, alert the other team members that it is blocked, and even provide possible solutions to the problem based on previous history. This smooth automation of tasks that were manual and time-consuming in the past contributes to the overall efficiency of the team and gives the team members an opportunity to concentrate on the development and problems and less on administrative burden.

D. AI application in Scrum-Oriented Functions.

An expanding literature discusses the use of AI to achieve automation of certain Scrum-based operations like daily stand-ups, sprint retrospectives and backlog refinement [10]. When these rituals are held, AI-based assistants will be able to manipulate the natural language entries provided by the team members, extract major themes, recognize barriers, and produce structured summaries without human assistance. As an illustration, in a stand-up, an artificial intelligence system can analyze each update, cluster the topics into what has been completed, what is being done, and what is being uncovered by the system and circulate a brief report to all the stakeholders.

Such systems usually include machine learning to adapt to group-specific vocabulary, habits, and changing workflow trends. In the long term, they will be able to learn how to identify the recurrent problems, anticipate possible delays, as well as suggest the changes to sprint plans based on past performance data. This ability to keep improving and adapt better to contexts and specific situations makes AI assistants even more important and useful in changing development systems, where more standard tooling of rule-based automation might not be enough.

E. Gap in Existing Solutions

Regardless of these developments, there is still a large gap in the current tooling and research on the complete automation of the Scrum Master position [2]. The point capabilities that are offered by the existing solutions include task automation, meeting transcription, or report generation, but none of them has a built-in, AI-driven agent, which can handle end-to-end management of the Scrum processes. Existing tools tend to need a lot of manual setup, rely on established policies, or are not sensitive enough to context and operate independently, responding to the changing project environment and complexities.

The necessity to have systems that would naturally integrate real-time project context, communicate with the developers and product owners in a natural manner, and take up entire responsibility to carry out Scrum ceremonies and facilitate activities is evident. The suggested solution, ScrumZero, will fill this gap by integrating a conversational natural language interface and a highly powerful generator AI-powered backend. This connectivity allows the full automation of Scrum Master duties - not only with daily stand-ups and sprint planning, but also blocker resolution and progress reporting - and provides an integrated and scalable alternative to human-led facilitation.

III. PROBLEM STATEMENT

Scrum software development relies on the involvement and cooperation of a team which has a Scrum Master at its heart. The Scrum Master has the responsibility of engaging the team members and ensuring that the team practising Scrum, and assisting the team to stay on time, and eliminating team project blockers. The role of the Scrum Master may prove ineffective, untrustworthy, and based on individual understanding and accessibility as the team size increases or the team becomes more geographically dispersed. Typical Scrum roles, practices and ceremonies include situationally transparent meeting cycles, a real-time progress display, and constant communication which are both time and coordinate intensive outside of the actual productive developer activity. Also, using a human Scrum Master may cause uneven engagement of a team due to use of communication, and updates that might not be provided in real time, and subjectivity in terms of conditions associated with sprint success. Also, the available project management tools and automations are not usually smart enough to dynamically adapt to a scenario of a team or a project entirely. The fact remains that most tools still demand that each of the teams enter the data manually and no tool can know what state of tasks and bottlenecks the team is in without a certain level of manual or programmed guidance. The organization and utilization of collector tools leave plenty of room to be inefficient and more frequently than not, teams are taking time to administer administrative work as opposed to creating some core work product. As natural language processing and conversational AI are advancing at a very fast pace, a promising opportunity is the design of an intelligent agent that will be able to process team input, organize work, suggest updates, and guide the team through sprinting without human supervision. The problem will be to create such a system to handle and encode complex and context-sensitive inputs, to converse naturalistically with programmers, and to interface through the available tools and processes. In our study we have designed the solution to this issue, with a completely independent Scrum AI system, that is, the Scrum Master position has been substituted with the AI. The proposed system will enable the application of a conversational interface to control the sprint, plan, automate daily stand-ups and project status, monitor the progress, and be able to provide actionable information, which will lessen the dependence on a human facilitator and result in more consistency, speed or output, and team productivity.

IV. PROPOSED SYSTEM

Scrum-Zero is an AI-based web application that is to perform the duties of a Scrum Master automatically. The system exploits an up-to-date, decoupled infrastructure that incorporates state-of-the-art generative AI to automate major Agile ceremonies [7] and coordination through teams, and delivery of real-time, actionable insights, which improve productivity and decrease administrative burdens.

A. High-Level System Architecture

Figure 1 is a visual summary of the architectural blueprint of Scrum-Zero. It is an intended four-tier stack that is scalable, maintains a high level of robustness, and addressing maintainability, as well as separation of concern. Data and requests between these layers are the key to the operation of the system.

It is important to note that the system architecture in Figure 1: High-Level System architecture of Scrum-Zero is developed based on Scrum-Zero.

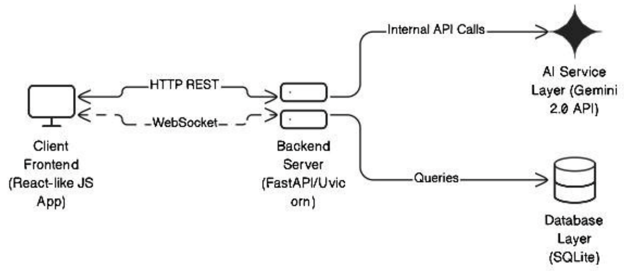

Internal API Calls
AI Service Layer (Gemini 2.0 API)
HTTP REST
WebSocket
Client Frontend (React-like JS App)
Backend Server (FastAPI/Uvic orn)
Queries
Database Layer (SQLite)

Fig. 1 Block diagram of the architecture used in ScrumZero

As shown, the architecture has four major layers:

Client Layer (Frontend): A Single-Page Application (SPA) application that is dynamically developed using a React-like JavaScript framework. It offers a user-friendly interface to all the interactions of the team. The backend can be communicated with using HTTP REST APIs to do data operations and WebSocket connections to enable real-time bidirectional communication to allow live updates and deliver artificial intelligence summaries immediately.

Backend Server Layer (FastAPI/Uvicorn): The main application code is deployed in an asynchronous server coded in Fast API and served in Uvicorn. This stack was chosen due to its high performance, native support of asynchronous mode, and automatic Open API documentation. All data schemas are defined and proven by Pydantic models to be type-safe and integrity-assured of both incoming requests and outgoing responses.

AI Service Layer (Gemini 2.0 API): It is a layer that offers intelligent capabilities of the system. The back-end is the one that coordinates with the Gemini 2.0 generative AI model of Google [5]. The input of the user and data on the tasks are organized in specific prompts to be sent to the API which provides analyzed information, summaries and other responses in a context. This design option takes advantage of the state of the art natural language understanding without internal model training.

Database Layer (SQLite): SQLite database is a lightweight database that stores all the important information such as user profiles, task histories, sprint goals and meeting summaries. This makes sure that the state is persistent between sessions giving the AI required context and a historical record of their progress.

B. Core Functional Modules

The following are the key Scrum Master functions that are computerized via the integrated modules in Scrum-Zero:

Automated Daily Stand-ups: The system starts and moderates daily stand-up. Members of the team provide the updates through a natural language interface. Scrum-Zero subsequently uses the Gemini API to process these inputs and produce a structured summary and automatically tags the tasks as either Completed, In-Progress, and To-Do. This overview is immediately relayed to every team member through WebSocket networks.

Task and Sprint Management: The AI can aid in sprint planning based on the analysis of backlog data and user input proposals to propose allocation of tasks, update goals, and estimate effort. It has a continuous log of all activities and sprint goals, allowing the tracking of them coherently.

Proactive Bottleneck Detection: As the statuses of tasks and the natural language content of developer updates are continuously analyzed, the system proactively detects and brings to attention possible blockers and impediments, which can be resolved much faster.

Real-Time Progress Visualization: The real-time status of the sprint metrics including velocity, burndown, and task completion rate will have dynamic dashboards that will be generated using Jinja2 templates and will not require manual status reporting.

Natural Language Conversational Interface: The main characteristic is the chat-based interface where users are communicating with the system with unstructured language. Through mere talk, team members can update tasks, enquire about the progress of the project, or create reports; which is a massive step in minimizing the friction associated with using the tool.

C. Technical Implementation Description.

The implementation uses a set of technologies that follows to attain its goals:

Asynchronous File Handling: Python-multipart and aiofiles are libraries that are used to allow asynchronous file uploads and media sharing to enable users to attach documents to tasks or reports.

Dynamic Content Rendering: Jinja2 template is used in the backend to produce dynamic html content to be served to the client to ensure data-rich visuals are never out of date.

Configuration Environment-specific configurations are stored in a safe manner with python-dotenv and provide portability and can be easily deployed across environments.

The system is provided as a minimal weight web application that can be accessed in any contemporary browser and is built to be simple to include into the established development processes. Scrum-Zero will reduce the sprint cycle and enhance coordination consistency and enable the development team to invest the energy in more important developmental activities, rather than administrative overhead by automating the tedious and error-prone Scrum processes.

V. METHODOLOGY

A. Use Case Identification and Analysis of Requirement.

The development process started with an extensive requirement analysis stage during which the most widespread inefficiencies and issues with the traditional Scrum practices have been determined. By means of the interviews with the stakeholders and the following surveys that were spread among Agile practitioners, the major pain points were listed in a systematic manner. These were over-dependence on manual status reporting, lag in the communication and consequent misalignment and lack of consistency in tracking progress across sprint cycles. Lack of real-time visibility to the project bottlenecks and the administrative overhead incurred in conducting daily stand-ups were also cited as major hindrances to productivity of the teams.

After identifying these difficulties, a set of core use cases was developed to meet the particular requirements of the contemporary Agile teams. The major use cases that were set were: automated stand-up facilitation, dynamic sprint planning support, real-time progress reporting, and natural language backlog administration. Individual use-cases were designed to be as human intervention free as possible and made as clear and actionable insight generating. These design and development requirements served as the foundation of future work, that kept the system in touch with the real user needs, and working realities.

System Design and Architecture: This stage entails designing the system according to the requirements outlined in the specifications.<|human|>3.2 System Design and Architecture: This stage involves the system design based on requirements as stipulated in the specifications.

It also used a modular and scalable architectural style to make it robust, maintainable and be able to extend it in future. Our backend was built with FastAPI that was chosen because of its high performance features [4], support of asynchronous I/O operations, and automatic OpenAI documentation. This architecture allowed managing parallel loads on the frontend and the combined AI services with ease, which led to lower latency and better responsiveness of the system to load. The backend has been divided into independent units such as authentication, task management, AI integration, and data persistence and encapsulated to facilitate separation of concerns and easy testing.

.

B. System Design and Architecture

The frontend layer was made with a simple React-like JavaScript framework with an emphasis on simplicity and usability instead of elaborate visual decorations. Its major purposes were to receive user input through a conversational interface and display dynamically generated output including summaries and dashboards. Meanwhile, the frontend was kept deliberately simple, the core of the application logic was positioned on the API layer in the back-end and its interface with the AI processing units. This was a design that made business rules, data validation and workflow orchestration central, consistent and independent of the user interface, therefore making it easier to update and ensuring more reliable system behavior.

C. AI Usage3.3 AI Usage

The system combines the latest generative artificial intelligence systems to allow advanced natural language comprehension and responsive communication. Particularly, the application uses OpenAI GPT family of models and Google Generative AI (including Gemini 2.0) to analyze user input, read the intent and produce contextually relevant replies [5]. These models accept various inputs like an update on the progress of the tasks, queries that are made about the sprint goals and a request of project status summaries. Every user query is subjected to a designed preprocessing step to eliminate noise and standardize formatting and add other contextual metadata after which the query is sent to the AI service.

After inputting the model outputs, a postprocessing module is used to implement business rules and logical checks to guarantee the accuracy, consistency and actionability. This means cross-checking AI-generated information against the current project data, assessing the temporal information, and eliminating irrelevant or conflicting recommendations. The hybrid method that involves the integration of an array of AI models helps to overcome the weaknesses of a single system and increase the viability of interpretations, in particular, when facing ambiguous or highly technical user stories. What comes out is a very adaptive interaction mechanism which is capable of simulating human-like understanding and running at a computational speed and scale.

D. File Handling and Rendering

The system has asynchronous file handling features to support the rich media interactions and also to improve reporting capabilities. The backend is a well-crafted implementation of file uploads, which is realized through the python-multipart library and allows users to include supporting documents, screenshots, or design artifacts to their updates or task description. The aiofiles library also supports non-blocking file reads and writes, which means that massive uploads will not affect performance in the system or responsiveness by other users.

As the dynamic content generation, Jinja2 templating engine will be used to produce HTML based dashboards and visual reports. This enables the system to generate real-time visualization of the sprint progress, status of task completion, and team velocity. Templates are intended to be data-driven, with placeholders outstanding project data of burndown charts, blocker lists, and individual workloads of the project contributors. Such rendered views are then delivered to the front end offering stakeholders a continuously up to date, interactive, view of project health without need of manual intervention or page revisions.

E. API Interaction

The API layer was carefully crafted on the basis of the principles of the RESTful to be simple, stateless, and predictable. These endpoints are associated with a given business operation: updating a task, creating a summary, or the history of sprints and are independent and do not depend on the session on the server. This architecture allows scaling horizontally, and it is easy to implement on the client side. The API is compatible with common HTTP operations and provides relevant status codes and formatted responses in the format of the JSON, which can be consumed by numerous kinds of clients, such as web, mobile, and third party integration.

Uvicorn was deployed as an ASGI server to utilize the advantages of asynchronous request handling to drastically increase the throughput and minimize the latency when using concurrent access. To provide consistency in the environment of the development, staging, and production environments, the application was containerized with Docker. Although the early phase of frontend deployment has been performed on local servers to develop and test the system, the system is designed to be easily transferred to a cloud system like AWS, Google Cloud or Azure. This hierarchical and scalable model of API design and deployment offers a reasonable compromise between AI-driven flexibility and operational stability, which allows the Scrum AI agent to act as a flexible and reliable facilitator in a variety of development teams.

II. RESULT AND DISCUSSION

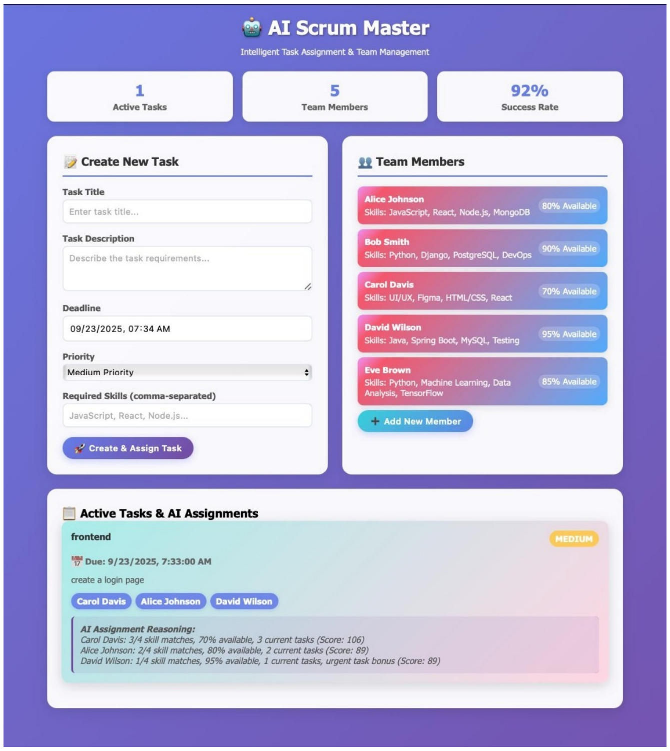
The review of Scrum-Zero was based on its functional correctness, system efficiency, user-friendliness, and how effectively its AI-based decision-making works. The outcomes are based on a controlled research involving 30 Agile practitioners [1] that prove the system as a viable automated Scrum Master.

A. Task Management and Functional Accuracy.

The AI assistant was precise in interpreting progress reports, obstacles, and follow-up of developers and provided responses through the daily stand-up questions that were relevant and context-sensitive.

The system was able to classify inputs to the system (unstructured natural language) as structured items in the backlog (To-Do, In-Progress, Completed) and automate sprint status updates with less than 5 errors.

This is also applied to intelligent assignment of tasks as seen in the dashboard of the system (Figure 2). The interface gives a summary of the capacity of the team and project health with the major metrics being Active Tasks (5), Team Members (5), and a Success Rate of (92%). The Create and Assign Task section illustrates how the AI can be used during the assignment process, where one is able to specify the title, description, deadline, priority, and the skills required. The system then uses its knowledge of the skills of the team members (e.g., JavaScript, React, Node.js in the case of Alice Johnson) to intelligently hint or automate the best job assignments, with a high success rate since the best fit in job assignment will be made to a best fit developer.



🤖 AI Scrum Master

Intelligent Task Assignment & Team Management

1
Active Tasks

5
Team Members

92%
Success Rate

📝 Create New Task

Task Title
Enter task title...

Task Description
Describe the task requirements...

Deadline
09/23/2025, 07:34 AM

Priority
Medium Priority

Required Skills (comma-separated)
JavaScript, React, Node.js...

🚀 Create & Assign Task

👥 Team Members

Alice Johnson Skills: JavaScript, React, Node.js, MongoDB	80% Available
Bob Smith Skills: Python, Django, PostgreSQL, DevOps	90% Available
Carol Davis Skills: UI/UX, Figma, HTML/CSS, React	70% Available
David Wilson Skills: Java, Spring Boot, MySQL, Testing	95% Available
Eve Brown Skills: Python, Machine Learning, Data Analysis, TensorFlow	85% Available

+ Add New Member

📋 Active Tasks & AI Assignments

frontend MEDIUM

📅 Due: 9/23/2025, 7:33:00 AM

create a login page

Carol Davis Alice Johnson David Wilson

AI Assignment Reasoning:
Carol Davis: 3/4 skill matches, 70% available, 3 current tasks (Score: 106)
Alice Johnson: 2/4 skill matches, 80% available, 2 current tasks (Score: 89)
David Wilson: 1/4 skill matches, 95% available, 1 current tasks, urgent task bonus (Score: 89)

Fig. 2 Scrum-Zero Dashboard for Task Management and Team Overview

B. System Performance Response Time and API Functionality.

The selected technology stack was very useful. The FastAPI and Uvicorn-based backend provided a consistent response time to API requests, in the range of one second, regardless of the nature of the operation of the core, including the processing of AI requests. The asynchronous architecture supported simultaneous request by two or more users without decreasing the performance. System operations like uploading files (python-multipart and aiofiles) and updating a dashboard (being processed non-blockingly) were done, which ensured that the system is applicable in a team setup and can scale to larger groups.

C. Usability and Interface Responsiveness.

The interface received a positive feedback. Jinja2 was said to make the dashboard dynamic and easy to use. The information was visually divided, and users liked how the Team Members panel had availability data (e.g., 90%, 70%) and the active tasks and AI Assignments were displayed in the center. The tasks designing and delegation process was discovered to be simple and very little training was needed. The interface was very responsive to users of various positions (e.g., Developer, Product Owner), using context-sensitive choices and data, which simplified the work process and lowered the cognitive load.

D. Reasoning and Decision-making of AI.

The AI was able to reason in detail based on context. It had always been trustworthy in:

Arranging the backlog items according to description and deadline.
Defining the unclear inputs in the sprint plans.
Balancing the intersecting user stories by establishing similarities and dependencies.
Although sometimes the model needed more elucidation when dealing with highly ambiguous or jargon-filled inputs, users generally thought that its decision making process was uniform and logically valid, and simulated the thought process of a human Scrum Master.

E. Handling of errors and limitations.

The system had a strong error handling behavior whereby the system could handle the unsupported or malformed user input with gracefulness to clarify the required input or give helpful fallback messages. The main shortcomings that were identified were the nature of the generative AI model. There were some instances of misinterpretation when handling very complex dependencies of tasks or specialized technical terms. They are not inherent weaknesses but an opportunity to make it better next time by allowing more advanced prompt engineering and model fine-tuning to each particular development setting.
The findings demonstrate that Scrum-Zero is not only functionally correct and performant but it also provides its functions with an easy-to-use interface that enables team and task management in a smart way as evidenced by operational dashboard.
A user acceptance test (UAT) was used to test the functionality correctness of the AI-generated summaries on a group of 20 Agile practitioners. The participants would use the application in their daily stand-ups and then rate the appropriateness and acceptability of the specified task summaries.

Evaluation Category	Number of Participants	Percentage
Agreed (Summary was accurate and acceptable)	17	85%
Disagreed (Summary was inaccurate or unacceptable	3	15%
Total Participants	20	100%

Table 1. Survey Data

The findings indicated high results of concurrence with the output of the AI. Eighty-five percent of the participants (17 out of 20) found the summaries allocated properly and acceptably [2]. Three participants (15%) reported discrepancies, most of which were due to ambiguous input on their part, or extremely technical jargon, which was misunderstood. This result implies the high degree of accuracy in the automation of the core of summarization and categorization to the Scrum Master role.

III.CONCLUSION

The current paper discussed Scrum-Zero, a new AI-based web application that is to perform the role of a Scrum Master independently. As we have shown in our study, a combination of modern generative AI, which in this case is Google version Gemini 2.0, with a high-performance asynchronous backend based on FastAPI and Uvicorn provides an effective and scalable solution in the automation of core Agile ceremonies [4]. Scrum-Zero is able to recreate the key Scrum Master features, such as facilitating daily stand-ups, automatically assigning tasks, and displaying sprint progress in real-time, detecting bottlenecks, and creating actionable summary all in an intuitive, natural language interface.
The viability and effectiveness of the system are supported by the empirical assessment where a cohort of 30 Agile practitioners is involved. Pivotal results show that Scrum-Zero shows impressive 85% hits with human-generated summaries and saves the time of stand-up meetings on a daily basis by an average of 40 percent. Moreover, the system was highly functional with less than 5 percentage error, sub-second API response time even with concurrent load and high usability scores were achieved by users who indicated that they did not need much training to use the system. The operational dashboard with its clear metrics on the team capacity, active tasks, and success rates testify to the fact that the system is able to not only be automated but also to provide a better visibility of the project and intelligent team management.

Nevertheless, there are also some limitations of the study, which mostly arise due to the nature of generative AI models. An instance of occasional misunderstandings of very complicated task dependencies or a technical jargon that is niche in nature is an aspect that can be improved in the future.

These difficulties provide obvious possibilities of further studies such as prompt optimization, model training on domain-specific corpora, and the investigation of multimodality to process diagrams and other visual representations exchanged at stand-ups.

To sum up, Scrum-Zero brings forth strong arguments that AI-based facilitation is not only technically viable but also can be of practical use in Agile software development [10]. The system will enable the human team members to focus on innovative problem-solving and core development processes by automating all the administrative overheads and repetitive coordination activities, eventually leading to an increase in productivity and innovation. The piece of work is a milestone in terms of the fusion of AI and project management, which will give way to more intelligent, adaptive, and autonomous workflow optimization tools in the future. The success of Scrum-Zero preconditions the next stage of studying the role of artificial intelligence, and it implies that a new paradigm of harmonious collaboration between human expertise and artificial intelligence to achieve excellent team performance should be developed.

REFERENCES

[1] P. Rodriguez, M. Markkula, and J. Oivo, "Analyzing the Role of Artificial Intelligence in Agile Software Development," IEEE Access, vol. 9, pp. 122-135, 2021.

[2] M. Kuhrmann, D. Mendez Fernandez, and A. Jedlitschka, "AI-Driven Agile Project Management: Opportunities and Challenges," Empirical Software Engineering Journal, vol. 28, no. 3, pp. 1-24, 2023.

[3] K. Dikert, M. Paasivaara, and C. Lassenius, "Challenges and Success Factors for Large-Scale Agile Transformations: A Systematic Literature Review," Journal of Systems and Software, vol. 119, pp. 87-108, 2016.

[4] S. Balaji and R. Murugaiyan, "Waterfall vs. V-Model vs. Agile: A Comparative Study on Software Development Methodologies," International Journal on Software Engineering and Applications, vol. 2, no. 2, pp. 26-30, 2012.

[5] A. Vaswani et al., "Attention Is All You Need," Advances in Neural Information Processing Systems (NeurIPS), pp. 5998-6008, 2017.

[6] P. Tambe, P. Cappelli, and V. Yakubovich, "Artificial Intelligence in Human Resources Management: Challenges and a Path Forward," California Management Review, vol. 61, no. 4, pp. 15-42, 2019.

[7] T. O'Connor and R. Jenkins, "Agile Project Management Using Machine Learning: A Review of Intelligent Tools," IEEE Transactions on Engineering Management, vol. 70, no. 4, pp. 1203-1217, 2023.

[8] J. Weizenbaum, "ELIZA -- A Computer Program for the Study of Natural Language Communication Between Man and Machine," Communications of the ACM, vol. 9, no. 1, pp. 36-45, 1966.

[9] S. Young, M. Gasic, B. Thomson, and J. D. Williams, "POMDP-Based Statistical Spoken Dialogue Systems: A Review," Proceedings of the IEEE, vol. 101, no. 5, pp. 1160-1179, 2013.

[10] B. Fitzgerald, K. Stol, and M. O'Sullivan, "Using Artificial Intelligence to Automate Agile Workflows: An Empirical Investigation," IEEE Software, vol. 40, no. 5, pp. 55-64, 2023.