International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

TaskFlow: An AI-Augmented Software Project Management and Progress Analytics Platform

Panchakshari Chakor[1], Mr. Sagar Mahajan[2], Roshan Chordia[3], Aamir Aland[4]
[1,3,4]Undergraduate Students, Dept. of Information Technology, G H Raisoni College of Engineering and Management, Pune, India
[2]Assistant Professor, Dept. of Information Technology, G H Raisoni College of Engineering and Management, Pune, India

***Abstract*—** *Modern software projects face increasing complexity due to changing customer requirements, distributed development teams, shorter release cycles, and continuous collaboration. Conventional project management systems provide task organization and progress tracking but still depend considerably on manual task creation, manual status updates, and subjective progress reporting. This paper presents TaskFlow, an AI-augmented software project management and progress analytics platform designed to reduce such manual effort. The proposed platform uses the Groq API with the LLaMA3-8B-8192 model to analyse Software Requirement Specification (SRS) or client agreement documents, identify project modules and functional requirements, and automatically generate tasks, subtasks, and milestone recommendations. The system further integrates JWT-based authentication, role-based dashboards, workload visualization, task history, project discussions, automated email reminders, and GitHub REST API v3 based repository analytics. Repository activity and task completion information are used to derive an AI Progress Score intended to provide a more objective view of devel-opment progress than self-reported status updates. When actual progress deviates from the planned schedule, Smart Milestone Adjustment recommends changes to upcoming milestones. The platform is implemented using React.js, FastAPI, SQLAlchemy, SQLite for development, PostgreSQL for production, and Python SMTP for notifications. The modular architecture supports maintainability and future extensions such as CI/CD-based unit test coverage, build success rate tracking, and code complexity analysis. TaskFlow demonstrates how generative AI and repository analytics can be combined within a centralized software project management workflow to improve planning, transparency, collaboration, and progress monitoring*.

***Keywords*—** *Artificial Intelligence, Software Project Management, Large Language Model, LLaMA3, Task Generation, GitHub Analytics, Progress Analytics, Smart Milestone Adjustment*.

I. INTRODUCTION

Software development projects are increasingly difficult to manage because of rapidly changing requirements, distributed teams, short release cycles, and the adoption of Agile development practices. Effective project management requires continuous planning, task allocation, progress monitoring, communication, and timely delivery.

Although conventional project management platforms provide mechanisms for organizing work, project managers frequently remain responsible for converting requirements into actionable tasks, assigning responsibilities, updating status infor-mation, and interpreting development progress.

Artificial Intelligence (AI) and Large Language Models (LLMs) provide an opportunity to automate parts of this workflow. LLMs can process natural-language requirement documents and support planning activities by identifying project scope, modules, and functional requirements.

AI can therefore be used not only for communication but also for requirement analysis, task generation, project monitoring, and decision support.

TaskFlow is proposed as an integrated platform that combines AI-based requirement analysis with software development repository analytics.

A project manager can upload an SRS or client agreement document through a web dashboard. The document is processed and analysed by the Groq API using the LLaMA3-8B-8192 model.

The resulting project structure is converted into modules, tasks, subtasks, and milestone recommendations. This reduces the manual effort involved in translating requirements into an initial project plan.

The second major aspect of TaskFlow is development-progress analytics. Instead of depending only on manually entered task status, the platform integrates GitHub REST API v3 to obtain repository activity such as commit history and developer contributions.

The repository information is combined with task completion information to calculate an AI Progress Score. The system can also recommend Smart Milestone Adjustments when actual development progress differs from planned milestones.

TaskFlow additionally provides role-based access, workload visualization, task history, discussions, and automated email reminders.

These capabilities are presented through a centralized dashboard so that project managers and developers can monitor project execution within one platform.

II. OBJECTIVE

The main objectives of the proposed TaskFlow platform are:

1. To reduce the manual effort required for converting SRS or client agreement documents into software project plans.
2. To develop an AI-based requirement analysis module using the Groq API and LLaMA3-8B-8192 for identifying modules and generating tasks, subtasks, and milestone recommendations.
3. To implement secure JWT-based authentication and role-specific access for project managers and developers.
4. To provide workload visualization, task history, discussion management, and centralized project monitoring.
5. To integrate GitHub REST API v3 for collecting repository activity and developer contribution information.

Panchakshari Chakor, Mr. Sagar Mahajan, Roshan Chordia, and Aamir Aland, "TaskFlow: An AI-Augmented Software Project Management and Progress Analytics Platform," *International Research Journal of Advanced Engineering and Science*, Volume 11, Issue 3, pp. 106-111, 2026.

6. To derive an AI Progress Score using repository activity and task completion information.
7. To provide Smart Milestone Adjustment when actual progress deviates from the planned schedule.
8. To automate project communication through discussion threads and email reminders.

III. LITERATURE REVIEW

TABLE 1: Literature Review

Sr.	Paper	Author	Key Findings	Future Scope	Publisher/Year
1	AI-Powered IT Project Management: Analyzing the Effectiveness of Advanced Project Management Tools to Ensure Project Efficiency	Jiban Krishna Das, Rojena Khadaka	Evaluated AI-powered project management tools such as Jira, Asana, and Monday.com to improve task automation, predictive analytics, and resource optimization. The study showed improved project efficiency, collaboration, and decision-making while reducing project delays.	Develop advanced machine learning models for project outcome prediction and integrate NLP, predictive analytics, and intelligent automation for more efficient software project management.	IEEE, 2025
2	Generative AI for Software Project Management: Insights from a Review of Software Practitioner Literature	Rashina Honda, John Grundy, Zainab Masood	Reviewed 47 practitioner sources and found that Generative AI improves software project management through task automation, predictive analytics, communication, collaboration, and agile project support, enabling Intelligent decision-making.	Investigate Agentic AI for autonomous project management, improve Human–AI collaboration, and develop responsible AI systems for software project management.	IEEE Software, 2025
3	Agentic AI: Autonomous Intelligence for Complex Goals – A Comprehensive Survey	Deepak Bhaskar Acharya, Karthigeyan Kuppan	Presented a comprehensive survey of Agentic AI high-lighting autonomous planning, reasoning, adaptation, and decision-making across healthcare, finance, manufacturing, and software project management applications.	Improve scalability, adaptability, ethical governance, and Human–AI collaboration while developing secure Agentic AI systems.	IEEE Access, 2025
4	Software Project Management Using Machine Learning Technique: A Systematic Literature Review	Mohammed Najah Mahdi, Yunus Yusoff, Hayder Natiq	Reviewed 111 research papers from 2009–2020 and found that Machine Learning significantly improves software effort estimation, project risk prediction, cost estimation, resource allocation, and project success.	Develop hybrid and explainable ML models using real-world datasets to improve software effort estimation and project risk prediction.	Applied Sciences (MDPI), 2021
5	Co-Pilot for Project Managers: Developing a PDF-Driven AI Chatbot for Facilitating Project Management	Khorshed Alam, Zaheed Ahmed Bhuiyan	Proposed a PDF-driven AI chatbot using Large Language Models to extract project knowledge from SRS documents, enabling intelligent communication, decision support, and context-awareesponses with high semantic accuracy.	Extend the chatbot with image understanding, enhanced reasoning capabilities, and support for multiple software project domains.	IEEE Access, 2025
6	Sprint Management in Agile Approach: Progress and Velocity Evaluation Applying Machine Learning	Yadira Jazmin Preez Castillo, Patricio Letelier	Proposed a Machine Learning-based sprint monitoring framework using agile project data. Support Vector Machine achieved 88% accuracy in identifying sprint progress, delays, and low-performing sprints.	Improve prediction models using larger sprint datasets, advanced feature engineering, and real-time integration with agile project management tools.	Information (MDPI), 2024
7	ProjectION: An AI-Powered Decision Support Tool for Agile Release Planning	Pablo Bermejo, Jose A. Gamez	Developed ProjectION, an AI-powered decision support tool using multi-objective optimization algorithms. It automates backlog prioritization, release planning, forecasting, and achieved 90.48% us-ability during evaluation.	Integrate advanced AI optimization algorithms, improve forecasting techniques, support large-scale agile environments, and enhance intelligent decision-making for software release planning.	Automated Software Engineering (Springer), 2026
8	The Relevance of Large Language Models for Project Management	Stavros Karnouskos	Evaluated ChatGPT-3.5, ChatGPT-4, and Bard using 252 PMP certification questions. ChatGPT-4 achieved the best performance, demonstrating that LLMs effectively support project planning, scheduling, risk analysis, and decision-making.	Evaluate additional LLMs, improve prompt engineering, and strengthen Human–AI collaboration for practical project management.	IEEE Access, 2024

Panchakshari Chakor, Mr. Sagar Mahajan, Roshan Chordia, and Aamir Aland, "TaskFlow: An AI-Augmented Software Project Management and Progress Analytics Platform," *International Research Journal of Advanced Engineering and Science*, Volume 11, Issue 3, pp. 106-111, 2026.

9	Synergizing Global and Local Strategies for Dynamic Project Management: An Advanced Machine Learning-Enhanced Framework	George Sklias, Socratis Gkelios, Dimitrios Dimitriou	Proposed a Machine Learning-enhanced decision support framework combining global schedule optimization with local dynamic replanning to improve scheduling, resource allocation, and project efficiency.	Improve forecasting using larger datasets, integrate advanced ML models, and validate the framework across diverse project domains.	IEEE Access, 2024
10	Software Defect Prediction Using Ensemble Learning: A Systematic Literature Review	Muhammad Adnan Khan, Faseeha Matloob, Munir Ahmad, Sagheer Abbas	Reviewed ensemble learning approaches for software defect prediction and found that ensemble models outperform individual classifiers. Feature selection and SMOTE significantly improved prediction performance.	Investigate advanced feature selection methods, classifier diversity, and robust ensemble frameworks for software defect prediction.	IEEE Access, 2021
11	AgES: A Machine Learning-Based Frame-work for Agile Software Effort Estimation	Rojeena Khadka, Ifeoluwa Elegbe	Proposed the AgES frame-work using nine Machine Learning models. XGBoost achieved the best perfor-mance for Agile effort esti-mation, improving project planning and resource allo-cation.	Integrate AgES with JIRA and GitHub, expand datasets, explore hybrid ML models, and incor-porate Large Language Models.	Scientific Reports (Springer Nature), 2026
12	Artificial Intelligence in Open Innovation Project Management: A Systematic Literature Review on Technologies, Applications, and Integration Requirements	Moonita Limiany Prasetyo, Randall Aginta Peranginangin, Nada Martinovic, Mohammad Ichsan, Hendro Wicak-sono	Conducted a systematic literature review of 365 studies to identify AI technologies, project management applications, industrial sectors, and technical and non-technical requirements for AI adoption in open innovation project management. Machine Learning and NLP were widely used for predictive analytics, resource optimization, risk management, communication, scheduling, and knowledge management.	Future research should conduct quantitative studies to measure AI effectiveness, investigate emerging technologies, and further examine ethical, organizational, data privacy, and human-AI collaboration issues.	Journal of Open Innovation: Technology, Market, and Complexity, 2025
13	How Artificial Intelligence Will Transform Project Management: A Systematic Literature Review	M. E. Nenni, F. De Felice, A. Lambiase, G. M. Marseglia, A. M. Rossi	Conducted a Systematic Literature Review of AI applications in project management, analysing 170 primary studies to identify how AI supports project risk management, decision-making, prediction, classification, and other PM processes.	Future research should develop practical frameworks for integrating AI into project management, investigate effective AI adoption pathways, and validate AI-based approaches across different industries and project environments.	Soft Computing, 2024
14	AI-Powered IT Project Management: Analyzing the Effectiveness of Advanced Project Management Tools to Ensure Project Efficiency	Rojeena Khadka, Lord Coffie, Jiban Krisna Das	Compared AI-powered project management tools including JIRA, Asana, and Monday.com using IT project datasets. AI improved task automation, predictive analytics, workflow management, and project efficiency.	Develop advanced ML models for project outcome prediction and integrate NLP with predictive analytics for intelligent project management.	IEEE, 2025
15	Artificial Intelligence Models in Project Planning for IT/Software Projects: A Systematic Literature Review	Alshamrani and Chiclana	Systematically reviewed AI models used in software project planning and found that AI improves effort estimation, scheduling, risk prediction, and decision-making while highlighting challenges related to explainability and ethics.	Develop self-evolving AI models, improve explainability, ensure ethical AI governance, and validate AI frameworks across diverse software projects.	Digital (MDPI), 2024

IV. Proposed Methodology

The proposed methodology follows an integrated workflow beginning with secure user authentication and requirement acqui-sition and continuing through AI-based project planning, development monitoring, repository analytics, milestone adjustment, and communication.

The modules are designed to work together while remaining loosely coupled.

A. Authentication and Role Management

TaskFlow uses JWT-based authentication to manage user sessions. After successful authentication, a signed token authorizes subsequent requests.

Panchakshari Chakor, Mr. Sagar Mahajan, Roshan Chordia, and Aamir Aland, "TaskFlow: An AI-Augmented Software Project Management and Progress Analytics Platform," *International Research Journal of Advanced Engineering and Science*, Volume 11, Issue 3, pp. 106-111, 2026.

Users are assigned roles such as Project Manager or Developer. The role determines the dashboard, permissions, and actions available to the user.

B. AI-Based Requirement Analysis

The Project Manager uploads an SRS or client agreement through the web dashboard.

The uploaded document is processed and its textual content is passed to the Groq API.

The LLaMA3-8B-8192 model analyses the document to identify project modules and functional requirements.

C. Document Parsing

Document parsing extracts relevant textual information from the uploaded requirement document.

The extracted content is supplied to the language model with a structured prompt.

The purpose of this stage is to identify distinct modules, functional requirements, and their scope before task generation.

D. Task and Milestone Generation

The identified modules are transformed into actionable development work.

The AI module generates tasks and subtasks associated with the corresponding modules and recommends project milestones.

This automated decomposition reduces the effort required to manually transform an SRS into an initial project plan and improves consistency in task structure.

E. Task Assignment and Workload Visualization

Generated tasks can be assigned to project team members.

The centralized dashboard presents workload information so that responsibilities can be monitored and distributed among team members.

Task history also maintains project-level and task-level activity information.

F. GitHub Repository Integration

GitHub REST API v3 is integrated to obtain repository activity, including commit history and developer contributions.

The collected repository information provides an additional source of evidence about development activity instead of depending only on manually entered status information.

G. AI Progress Score

TaskFlow combines repository activity with task completion information to calculate an AI Progress Score.

The purpose of this score is to provide an objective representation of development progress.

When repository activity and task completion indicate that actual progress has deviated from the planned schedule, the platform can invoke Smart Milestone Adjustment.

H. Smart Milestone Adjustment

The Smart Milestone Adjustment component recommends revised milestone dates when observed progress differs from the planned schedule.

This provides an adaptive mechanism for project monitoring rather than treating the original schedule as fixed throughout the development lifecycle.

I. Communication and Notifications

TaskFlow includes discussion threads scoped to projects and tasks.

Automated email reminders are implemented using Python SMTP to notify team members about task assignments, deadlines, and status changes.

These facilities centralize project communication and reduce dependence on separate communication channels.

V. System Architecture

The architecture of TaskFlow is organized around four functional areas:

1. Project Planning and Setup
2. Real-Time Monitoring
3. Analytics and AI Agents
4. Collaboration and Alerts

The Project Manager initiates the workflow by uploading an SRS or client requirement PDF.

The AI module generates modules, tasks, and milestones, while manual task and milestone management remains available. During execution, users and developers access the centralized dashboard, task history, milestone progress, workload information, and GitHub activity.

The analytics layer provides software quality metrics, AI Progress Score, automatic project risk prediction, schedule adjustment recommendations, and sprint performance monitoring.

The collaboration layer supports project discussions and automated email reminders through AI agents.

Figure 1 represents the interaction between the Project Manager, User, Developer, and AI Agents.

VI. Implementation Details

The implementation uses a modular web architecture.

React.js is used for the frontend and dashboard interface, while FastAPI provides the Python-based backend and REST API layer.

SQLAlchemy provides object-relational mapping for database interaction.

SQLite is used during development and PostgreSQL is intended for production deployment.

The AI layer uses the Groq API with the LLaMA3-8B-8192 model for SRS parsing, task generation, and milestone recommendation.

GitHub REST API v3 provides repository activity and contribution information.

JWT provides secure stateless authentication, while Python SMTP is used for automated email notifications.

Panchakshari Chakor, Mr. Sagar Mahajan, Roshan Chordia, and Aamir Aland, "TaskFlow: An AI-Augmented Software Project Management and Progress Analytics Platform," *International Research Journal of Advanced Engineering and Science*, Volume 11, Issue 3, pp. 106-111, 2026.

International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

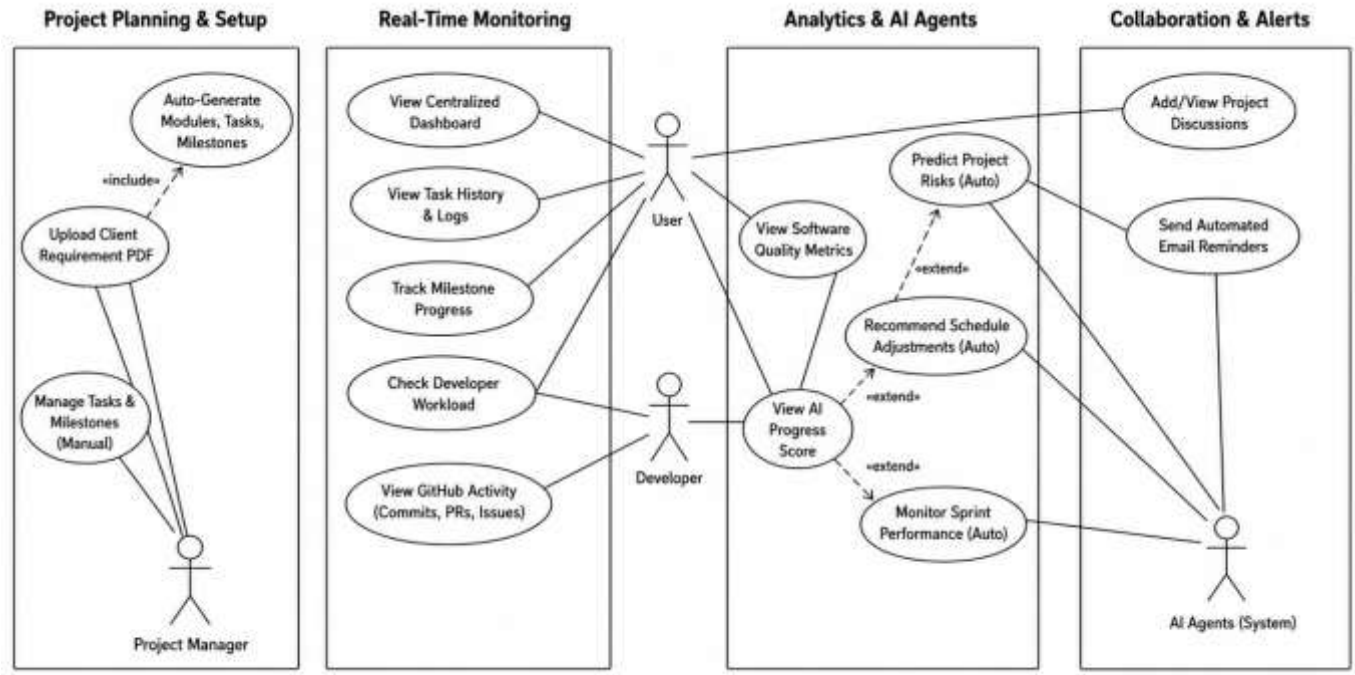

Figure 1: System Architecture of the Proposed TaskFlow Platform

TABLE 2: Technology Stack of TaskFlow

Technology	Purpose
React.js	Frontend framework for building the project dashboard and user interfaces
FastAPI (Python)	Backend framework for REST API development and business logic
SQL Alchemy ORM	Object-relational mapping for database interaction
SQLite	Database used during development
PostgreSQL	Database used for production deployment
Groq API (LLaMA3-8B-8192)	Large Language Model used for SRS parsing, task generation, and milestone recommendation
GitHub REST API v3	Repository integration for commit history and contribution analytics
JWT	Secure, stateless user authentication
Python SMTP	Automated email notifications and reminders

VII. Results and Discussion

The supplied TaskFlow project material describes the proposed outcomes rather than a completed experimental benchmark.

Therefore, the results in this paper are presented as expected system-level outcomes rather than experimentally measured accuracy values.

A. Automated Task Generation

The first expected outcome is automated task generation from requirement documents.

The system is intended to parse SRS or client agreement documents and generate modules, tasks, subtasks, and milestone recommendations using the Groq API and LLaMA3-8B-8192 model.

B. Secure Role-Based Access

The second expected outcome is secure role-based access.

Project Managers and Developers are expected to receive role-specific dashboards and permissions through JWT-based authentication.

C. Objective Progress Measurement

The third outcome is objective progress measurement.

GitHub commit history and contribution information are combined with task completion data to derive the AI Progress Score.

This is intended to provide a development-progress view that is less dependent on manually reported status.

D. Adaptive Scheduling

The fourth outcome is adaptive scheduling through Smart Milestone Adjustment.

When observed progress deviates from the planned schedule, the system recommends revised milestone dates.

E. Centralized Collaboration

The fifth outcome is centralized collaboration.

Discussion threads, task history, and automated email reminders provide a common environment for project communication and monitoring.

F. Modular Architecture

Finally, the modular React.js, FastAPI, SQLAlchemy, and database architecture is intended to support future extensions without requiring major restructuring of the complete platform.

VIII. Future Scope

The future scope of TaskFlow includes extending the analytics layer with unit test coverage through CI/CD pipelines, build success rate tracking, and code complexity analysis.

Panchakshari Chakor, Mr. Sagar Mahajan, Roshan Chordia, and Aamir Aland, "TaskFlow: An AI-Augmented Software Project Management and Progress Analytics Platform," *International Research Journal of Advanced Engineering and Science*, Volume 11, Issue 3, pp. 106-111, 2026.

These capabilities can provide additional evidence for software quality and development progress.

The AI layer can also be refined to improve the quality and consistency of automatically generated tasks.

Future validation should evaluate task-generation accuracy and consistency using larger and more diverse project documents.

Further work can investigate advanced predictive models for project risk, schedule deviation, effort estimation, and sprint performance.

Human–AI collaboration, explainability, data privacy, and responsible AI practices are also important areas for future development.

The platform can additionally be evaluated across different software project domains and larger repositories to assess scalability, reliability, and practical usefulness in industrial project-management environments.

IX. Conclusion

TaskFlow presents an AI-augmented approach to software project management that combines requirement understanding, automated task generation, repository analytics, progress monitoring, and adaptive milestone management.

The proposed platform addresses the manual effort involved in converting SRS or client agreement documents into project plans by using the Groq API and LLaMA3-8B-8192 model to identify modules, tasks, subtasks, and milestones.

Unlike a workflow based only on manual status updates, TaskFlow incorporates GitHub repository activity and task com-pletion information to derive an AI Progress Score.

Smart Milestone Adjustment further enables the project schedule to respond to deviations in actual development progress.

JWT authentication, workload visualization, task history, project discussions, and automated email reminders provide supporting collaboration and monitoring facilities.

The reviewed literature demonstrates growing interest in AI for project planning, estimation, risk prediction, sprint monitoring, decision support, and project-management automation.

TaskFlow combines several of these directions into a centralized software project management workflow.

Its modular architecture is designed to support future extensions including CI/CD-based testing metrics, build analytics, code complexity analysis, and additional AI-driven project intelligence.

Acknowledgment

The authors would like to thank the faculty of G. H. Raisoni College of Engineering and Management, Pune, and the project guide, Mr. Sagar Mahajan, for their guidance and feedback during the development of the TaskFlow project.

References

[1] J. K. Das and R. Khadaka, "AI-Powered IT Project Management: Analyzing the Effectiveness of Advanced Project Management Tools to Ensure Project Efficiency," IEEE, 2025.

[2] R. Honda, J. Grundy, and Z. Masood, "Generative AI for Software Project Management: Insights from a Review of Software Practitioner Literature," IEEE Software, 2025.

[3] D. B. Acharya and K. Kuppan, "Agentic AI: Autonomous Intelligence for Complex Goals – A Comprehensive Survey," IEEE Access, 2025.

[4] M. N. Mahdi, Y. Yusoff, and H. Natiq, "Software Project Management Using Machine Learning Technique: A Systematic Literature Review," Applied Sciences (MDPI), 2021.

[5] K. Alam and Z. A. Bhuiyan, "Co-Pilot for Project Managers: Developing a PDF-Driven AI Chatbot for Facilitating Project Management," IEEE Access, 2025.

[6] Y. J. P. Castillo and P. Letelier, "Sprint Management in Agile Approach: Progress and Velocity Evaluation Applying Machine Learning," Information (MDPI), 2024.

[7] P. Bermejo and J. A. Gamez, "ProjectION: An AI-Powered Decision Support Tool for Agile Release Planning," Automated Software Engineering (Springer), 2026.

[8] S. Karnouskos,"The Relevance of Large Language Models for Project Management," IEEE Access, 2024.

[9] G. Sklias, S. Gkelios, and D. Dimitriou, "Synergizing Global and Local Strategies for Dynamic Project Management: An Advanced Machine Learning-Enhanced Framework," IEEE Access, 2024.

[10] M. A. Khan, F. Matloob, M. Ahmad, and S. Abbas, "Software Defect Prediction Using Ensemble Learning: A Systematic Literature Review," IEEE Access, 2021.

[11] R. Khadka and I. Elegbe, "AgES: A Machine Learning-Based Framework for Agile Software Effort Estimation," Scientific Reports (Springer Nature), 2026.

[12] M. L. Prasetyo, R. A. Peranginangin, N. Martinovic, M. Ichsan, and H. Wicaksono, "Artificial Intelligence in Open Innovation Project Management: A Systematic Literature Review on Technologies, Ap-plications, and Integration Requirements," Journal of Open Innovation: Technology, Market, and Complexity, 2025.

[13] M. E. Nenni, F. De Felice, A. Lambiase, G. M. Marseglia, and A. M. Rossi,"How Artificial Intelligence Will Transform Project Management: A Systematic Literature Review," Soft Computing, 2024.

[14] R. Khadka, L. Coffie, and J. K. Das, "AI-Powered IT Project Management: Analyzing the Effectiveness of Advanced Project Management Tools to Ensure Project Efficiency," IEEE, 2025.

[15] Alshamrani and F. Chiclana, "Artificial Intelligence Models in Project Planning for IT/Software Projects: A Systematic Literature Review," Digital (MDPI), 2024.