Agile Project Management Using Large Language Models

Dhruva G
Department of Computer Science & Engineering
PES University
Bengaluru, India
pes1ug20cs131@pesu.pes.edu

Ishaan Shettigar
Department of Computer Science & Engineering
PES University
Bengaluru, India
pes1ug20cs170@pesu.pes.edu

Srikrshna Parthasarthy
Department of Computer Science & Engineering
PES University
Bengaluru, India
pes1ug20cs692@pesu.pes.edu

Sapna V M
Department of Computer Science & Engineering
PES University
Bengaluru, India
sapnavm@pes.edu

2024 5th International Conference on Innovative Trends in Information Technology (ICITIIT) | 979-8-3503-8681-3/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICITIIT61487.2024.10580873

***Abstract*—Agile data-driven methodology encourages engineering leaders to measure their teams' performance by leveraging metrics for improving visibility, identifying potential roadblocks, and increasing delivery velocity. The work presented here introduces a cutting-edge paradigm of a data-driven approach to Agile project management, contributing to the evolving research in project management methodologies. As organizations strive to consolidate the competitive market, their success is often measured by their agility and resilience. Such agility stems from the underlying management practices that an organization embraces and is crucial for the effective planning and delivery of large-scale projects. While management philosophies have continued to evolve, organizations specializing in software development have increasingly adopted Agile project management practices to keep up with a disruptive landscape inundated with rapidly emerging technological advancements. As organizations have continued to expand globally, the existing Agile practices have been laggard and sluggish, unable to keep up with the demands of a modern market. In this context, the authors introduce an Agile project management framework driven by Large Language Models (LLMs) to facilitate efficient management of large projects.**

Keywords—Agile methodologies, project management, generative AI, Large Language Models, Kanban

I. INTRODUCTION

As enterprises become entrenched in delivering large-scale and cross-functional software projects to global clients, the risk of failure looms large with several potential roadblocks. While the technological landscape is dotted with spectacular failures and mishits, there has been a concerted effort to identify and evolve a consistent software development lifecycle to meet the demands of a volatile market. Initial development frameworks and practices (such as the Waterfall model) relied extensively on a structured and rigid downstream development lifecycle. Such models mandated the need for collecting comprehensive requirements at the beginning of the project and allowed little to no change during software development and testing [1].

Such philosophy was rooted in the notion that frequent changes and modifications would derail complex projects, resulting in unreasonable budget overruns and missed deadlines. However, the evolution of the technological landscape and the increasing agility demanded by a volatile market made such rigid models unsuitable and archaic due to the quantum of specifications required and voluminous documentation, and monolithic models quickly fizzled out.

Subsequent innovations in software development gradually accommodated more realistic aspirations of consumers and the agility required to retain a viable competitive advantage. Most organizations today use a variant of the Agile philosophy [2], and it has continued to usher faster ways of taking software applications to the market.

The Agile management philosophy is founded on the principles of increased customer collaboration, reduced documentation, robust processes and the need to be constantly adaptive [3]. While the philosophy has catalyzed software innovation and disruption, the philosophy has failed to keep pace with a transformative age of software development and has continued to stagnate in the process. In this context, this paper explores the use of emerging technologies to advance existing Agile processes, to make them relevant to the current software development paradigm.

II. AGILE PROJECT MANAGEMENT

A. Introduction

The Agile Manifesto was proposed in the early 2000s as a contemporary alternative to a monolithic and documentation-driven software development approach, and it quickly became a forebearer for advancements in project management strategies. Four values and twelve principles form the bedrock of the Agile Manifesto, and they advocate a paradigm shift in software development, prioritizing human-centric and outcome-oriented approaches. The Agile philosophy is an umbrella of underlying management practices including the widespread Scrum, Kanban and XP [4].

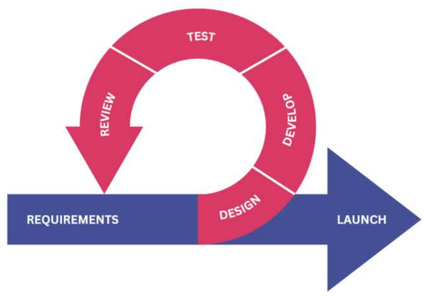

Fig. 1. Agile Software Development Lifecycle

The Agile philosophy outlines the need for delivering software in frequent bursts, increased customer collaboration, cross-functional synergy and flexibility in accommodating changing requirements as shown in Figure 1.

B. Agile Techniques

The Agile philosophy encompasses more than a dozen techniques within the software development lifecycle, allowing enterprises to tailor their implementation according to their specific requirements and the nature of their work. Some of these Agile practices have been elaborated below.

- Scrum – Scrum is a prominent agile project management framework characterized by iterative and incremental development methodologies. Within the Scrum framework, interdisciplinary teams engage in brief, time-bound sprints with the objective of delivering a potentially deployable product increment at the culmination of each iteration. The Scrum teams are characterized by a Product Owner and Scrum Master, who decide Sprint Planning, Daily Scrum, Product Backlogs and Sprint Backlogs [5].
- XP – Key principles of XP (eXtreme Programming) include pair programming, frequent code integration and testing, and test-driven development to ensure rigorous development and testing practices. Additionally, XP mandates collective code ownership coupled with disciplined development practices, making it efficient for large-scale projects with nebulous requirements resulting in frequent collaboration with customers. XP continues to thrive as it is a lightweight and highly flexible development framework [6].
- Feature-Driven Development – FDD involves a structured five-step process, comprising the development of an overarching model, delineation of a feature list, planning, designing, and building features iteratively. Central to FDD is the concept of features, denoting discrete, client-visible functionalities prioritized based on business value. This methodology fosters a refinement-based development framework based on a client-centric approach, ensuring efficient collaboration between the development team and stakeholders [7].

C. Kanban

The genesis of Kanban can be traced back to the late 1940s when Taiichi Ohno, a key architect of the Toyota Production System (TPS), developed Kanban as a visual scheduling and production control system to optimize manufacturing processes. The term "Kanban" (看板) in Japanese translates to "visual card." The tasks are represented as tiles on a Kanban board and this traces the task across the project lifecycle. As the task progresses through the development process, the tiles are accordingly moved across different phases on the Kanban board (as shown in Figure 2).

In the present milieu, the utilization of Kanban aims to enhance transparency throughout the software development lifecycle, and it serves as an efficient methodology for implementing Agile and DevOps principles [8]. The aforementioned framework emphasizes the need for instantaneous communication of capacity and the unhindered visibility of work progress. Visual representation using tiles on the Kanban board empowers cross-functional team members to glean real-time insights into the status of each work item, instilling a heightened level of transparency throughout the software development process.

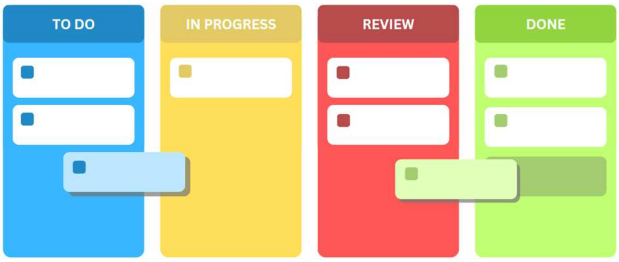

Fig. 2. A variant of the Kanban Board

III. Contemporary Challenges

Despite the remarkable leapfrogging in software development achieved with Agile practices, the existing framework has continued to lag behind a vigorously evolving technological landscape, threatening the philosophy with obsolescence in the absence of innovation. In this context, it becomes necessary to outline the contemporary challenges [9] in the effective implementation of the Agile principles, paving the way for a systematic enhancement and upgradation of the aforementioned practices.

As organizations expand globally, the workforce has increasingly become disparate and cross-national, enhancing the challenges in managing cycles of iterative software development across time zones and nationalities. Organizations rely on sub-optimal Agile project management practices, often attributable to the lack of trained Agile practitioners within the organization. Furthermore, the cultural shift required makes the adoption of Agile processes resource-consuming. Furthermore, the absence of data points and feedback loops make project management inherently heuristic, making it exceedingly vulnerable to human errors, inaccurate project estimates and insufficient resource allocation. These compounds significantly in the downstream processes.

The need for constant collaboration with customers or clients may significantly impede the pace of development due to ambiguous requirements or infrequent interactions. Some of these impediments include lack of communication among stakeholders, deferred replies and deferred communication, unavailability of customers and frequently evolving requirements from different stakeholders [10].

In an environment where consumer behaviour is dictated by the responsiveness, agility and turnaround time of enterprises, it becomes essential to evolve contemporary strategies and techniques to address the needs of a fiercely competitive market. This can be accelerated by leveraging emerging technologies and management philosophies to mitigate the roadblocks prevalent in the current Agile processes.

IV. Related Works

A. Literature Survey

Bharat Choudhary and Shanu K Rakesh (2016) [11] attempt to coalesce the diverging techniques used in Agile project management, and elaborately discuss the methodologies adopted by respondents. Their research outlines Scrum (evident in Figure 3) as the most popular Agile framework. The authors further enumerate the different techniques, advantages and limitations of Adaptive Software Development (ASD).

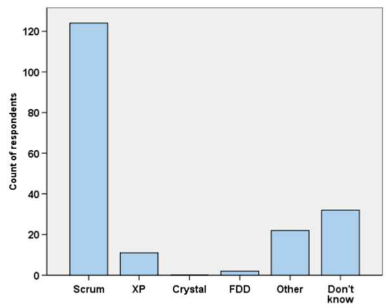

Fig. 3. 65% of the respondents used Scrum (N=125)

A. Ahmed et al. (2010) [12] explore the impact of Agile processes on the quality and productivity of software development and deduce inferences based on the data collected from enterprises in Pakistan. The authors considered knowledge sharing, self-organizing teams, participation of stakeholders, team size, formal training, responsiveness to change etc. as metrics for analysing productivity and quality. The authors propose a refined Agile lifecycle based on their inferential research, incorporating a renewed focus on refactoring and parallelized debugging. The research reports a positive correlation between the implementation of Agile processes and the productivity/quality obtained thereof. Figure 4 illustrates the refined Agile development lifecycle.

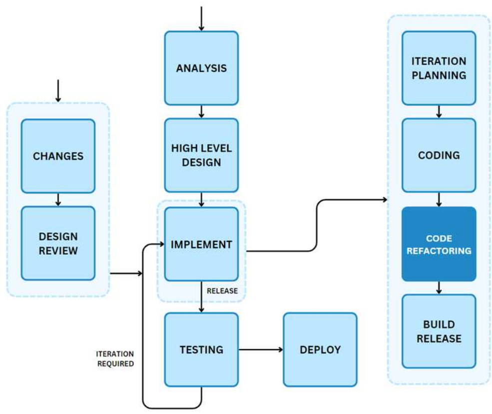

Fig. 4. Modified Agile lifecycle, with increased emphasis on Code Refactoring

Nazish Saleem et al. (2021) [13] present a comprehensive analysis on the evolutionary project management function across different levels of operations. The authors explore the evolutionary characteristics of project management in software development, examining traditional, agile, and global approaches. The research emphasizes the critical link between effective project management and success, and recognizes the evolving nature of project lifecycles and cross-cutting technological disruptions. The authors analyze the project management function across individual, project, team, and company levels to infer the levers needed for informed decision-making. Their work underscores the growing complexity of project management, especially in the global context where unique challenges such as geographical, temporal, socio-cultural, linguistic, and organizational distances need specialized project management practices., and advocates for tailored approaches based on project requirements, beckoning further empirical research to consolidate the existing research and address the nuances of complex multilevel project management dynamics.

Sergey Bushuyev et al. (2022) [14] introduce a novel metric for assessing the quality of project management processes using entropy. Project management quality is evaluated through the information project entropy, representing the team's confidence in project outcomes. A management quality scale and the entropy index are proposed for continuous monitoring throughout the project life cycle, as illustrated in Figure 5. Three pathways of project informational entropy are explored, emphasizing the ideal scenario representing a consistent decrease in entropy as the project lifecycle progresses. This research proposes the use of quantifiable metrics to evaluate project management practices.

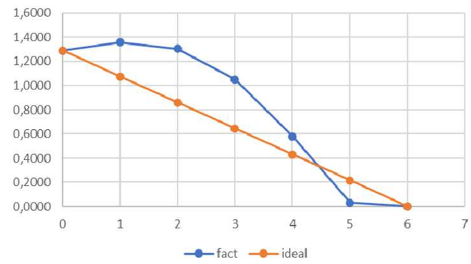

Fig. 5. Information entropy of the project across its lifecycle

Shuo Zhang and Lan Jin (2020) [15] employ the Monte Carlo method to enhance the accuracy of software project schedule management. The research attempts to address the dismal success rates in the software industry by providing a simulation-based solution to the uncertainty in project durations. The method involves decomposing tasks, and determining *optimistic* and *pessimistic* times. After 25,000 simulations, the authors achieve a reliable estimate of the project's total duration, demonstrating the potential feasibility and efficiency of Monte Carlo simulation in scheduling and management. The probabilistic result is shown in Figure 6.

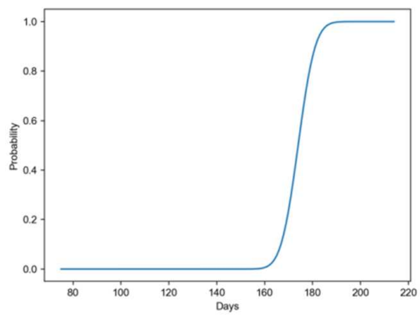

Fig. 6. Probabilstic schedule estimation using Monte Carlo method

B. Research Gaps

Agile project management continues to evoke academic interest, but existing research remains vastly inadequate. With the rapid proliferation of Large Language Models (LLMs), their impact on Agile project management and scheduling remains to be explored. Furthermore, the Agile processes fail to take into consideration relevant data points for continuous refinement within the ambit of every organization, making it a largely enterprise-agnostic technique with little customization. As customers demand near-instantaneous responsiveness, the current frameworks fail to consider the need for intelligent task creation, task management and workload management. Existing research can be attuned to enterprise data by coupling Agile processes with LLMs.

V. Generative AI & Large Language Models

A. *Introduction*

The rapid emergence and disruptive popularity of General Artificial Intelligence (GAI) has ushered in a new paradigm of efficient computing [16]. While it remains early days for the advent of GAI, it has quickly superseded traditional computational techniques, percolating to several industries. Unlike traditional machine learning algorithms trained to perform exceptionally well for specific tasks, GAI attempts to use large corpus of data to model realistic environments, with the ability of the model to replicate or accelerate human behaviour to a remarkable extent in several tasks.

A Large Language Model (LLM) is a statistical deep learning model that has undergone extensive training on a vast multimodal dataset, and it possesses the capability to generate text, generate multimedia content, assist in summarization and perform several challenging natural language processing (NLP) tasks. Such models are trained on several thousand gigabytes of multimodal data, to facilitate the development of a general artificial intelligence.

While safety and accuracy of such LLMs need further quantification and research, the models undergo continuous fine-tuning and refinement based on a robust feedback loop, enabling the model to stay relevant in the face of constant disruption and emergence of newer data. Industries continue to explore and leverage such large language models to automate several routine tasks. This paper attempts to explore the viability of using such large language models to accelerate Agile project management of large-scale software projects.

B. *PaLM LLM*

Pathways Language Model (PaLM) is a dense decoder-only Transformer model trained on 540-billion parameters. PaLM uses the Pathways system [17] that adeptly orchestrates diverse parallel computations across thousands of accelerators through efficient scheduling, while coordinating seamless data transfers. Through extensive evaluations on numerous language understanding and generation tasks, PaLM consistently demonstrates state-of-the-art few-shot performance, surpassing existing benchmarks by significant margins. Figure 7 [18] shows the performance of PaLM on scaling, benchmarked on the "Beyond the Imitation Game" scale against other LLMs. Additionally, PaLM performs well on multilingual NLP benchmarks, such as translation and text summarization.

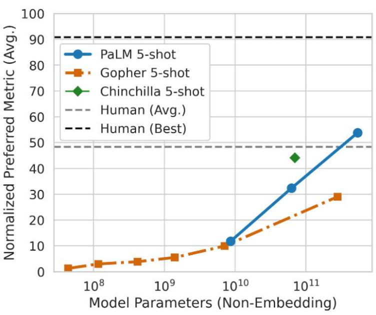

Fig. 7. Scaling behavior of PaLM on a subset of 58 BIG-bench tasks.

This paper introduces the use of PaLM to facilitate automated Agile task creation and management and explores the performance of the model in this context.

VI. Proposed Methodology

A. *Architecture*

Figure 8 introduces an Agile project management architecture using the PaLM large language model coupled with internal enterprise data.

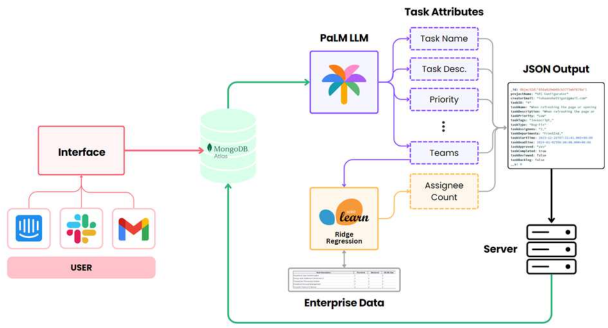

Fig. 8. Intelligent Agile project management using LLMs

The premise of an omni-channel interface facilitates customers and end users raising tickets, feature requests or other requirements on an *ad hoc* basis. These are stored internally on a cloud-based MongoDB Atlas database for further processing and refinement.

A large language model (PaLM in this instance) is used to augment the information obtained from the clients, end-users or customers using a large corpus of training data. A single query is augmented to include the following fields – task name, description, teams capable of resolving the ticket raised and the priority. The information obtained is used to forecast potential deadlines for the new ticket, enabling the project manager or Scrum master to leverage internal enterprise data to estimate the potential endpoint, enhancing the capabilities of existing enterprise-agnostic project management processes.

The proposed intelligent task augmentation and allocation methodology delves into relevant enterprise-specific details such as historical project timelines and resource allocation patterns. Such holistic perspective empowers Agile project managers to take informed decisions on task scheduling, optimize resource utilization, and mitigate risks effectively [19].

B. *Synthetic Dataset*

Due to the absence of a publicly available enterprise dataset, we were compelled to create a synthetic dataset [20] to simulate the necessary intelligent task allocation. Attributes such as the number of personnel allocated and priority were used to understand their correlation with the duration of the ticket raised on the Kanban board. Figure 9 is a sample of the dataset used.

Ticket #	Task Description	Frontend	Backend	ML/MLOps	QA	Priority	Duration (Days)
1	Implement User Authentication	2	1	0	1	High	5
2	Design and Implement Dashboard UI	3	0	0	0	High	7
3	Transaction Processing System	0	2	0	0	High	8
4	Implement Account Management	2	2	0	1	High	6
5	Integrate Payment Gateway	2	2	0	1	High	7

Fig. 9. Synthetic dataset used for estimating the duration of a new ticket

The dataset was synthetically created by using generative AI (i.e., ChatGPT), where a list of 250 tickets with their necessary attributes was generated along with their duration. These were validated by a project management professional to ensure that the synthetic dataset reflected the ground truth.

VII. Results & Discussion

The analysis of the results is two pronged, making it imperative to delve into the inferences of the results obtained. These inferences have guided downstream design and implementation choices.

A. *Regression Modelling*

In a departure from existing Agile project management processes, the proposed methodology hinges on leveraging enterprise data to facilitate data driven task allocation. The aforementioned synthetic dataset was subject to various regression modelling techniques to understand the correlation of a set of attributes on the duration of a new ticket. The mean square error (MSE) and R^2 score (tabulated in Table 1) were used to quantitatively determine the best performing regression models for further consideration. A consistent train-test split of 80:20 was used with a random split across the machine learning models during the analysis. Figures 10 – 14 visually illustrate the results obtained.

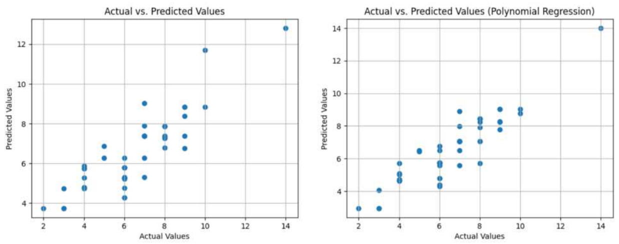

Fig. 10. Linear Regression and Polynomial Regression

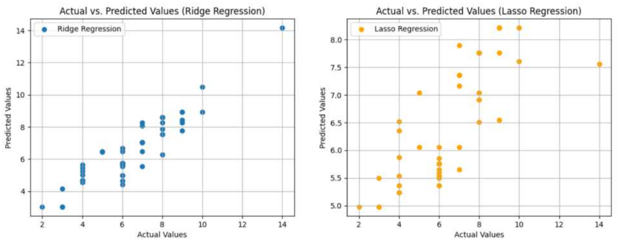

Fig. 11. Ridge Regression and Lasso Regression

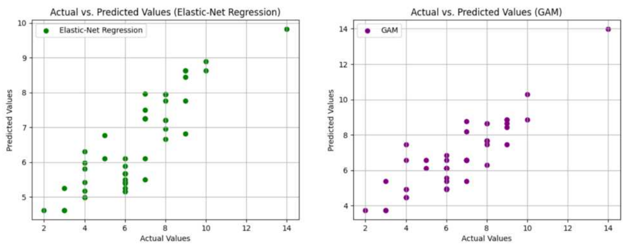

Fig. 12. Elastic-Net Regression and GAM Regression

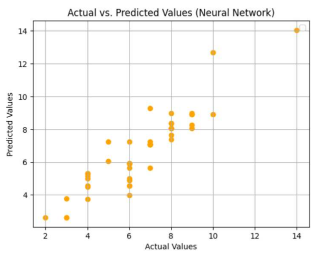

Fig. 13. Regression using Convolutional Neural Networks (1000 Epochs)

TABLE I. Accuracy Metrics

Sl. No.	Regression Model	MSE	R^2 Score
1	Linear Regression	1.234	0.7433
2	Polynomial Regression	0.8492	0.8234
3	Ridge Regression	0.7340	0.8473
4	Lasso Regression	2.449	0.4907
5	Elastic-Net Regression	1.5829	0.6708
6	GAM (Generalized Additive Model)	1.145	0.7618
7	CNN [1000 epochs]	0.9203	0.8086

The coefficient of determination (R^2) quantifies the proportion of variance in the dependent variable explained by the model, while Mean Squared Error (MSE) measures the average squared difference between predicted and observed values, serving as a measure of predictive accuracy in regression analysis. Lower MSE values and higher R2 Scores generally indicate better performance [21].

Ridge Regression (alternatively referred to as Tikhonov regularization or L2 regularization) [22], is a linear regression technique integrating a supplementary penalty term within the cost function. This augmentation is proportionate to the square of the coefficients' magnitudes, serving to mitigate the influence of large coefficient values. Ridge Regression aims to counteract overfitting by imposing constraints on the model parameters, resulting in a more stable and generalized predictive performance.

In the experiments conducted, the Ridge Regression performed exceedingly well across both the quantitative metrics measured. Therefore, the Ridge Regression model was selected for further downstream implementation.

B. *The Large Language Model*

The PaLM LLM was considered for the research due to its seamless integration into the programming environment via publicly available APIs. The augmentation of an input ticket ("*Server is crashing on executing the program*") has been tabulated in Table 2. Such augmentation by LLMs assist project managers by infusing more context to incoming tickets, and automates the routine hassle of raising tickets on project management platforms.

TABLE II. Sample Task Augmentation Using PaLM

Task Field	Output
Task Name	Program crash on server
Task Summary	The server is crashing when the program is executed.
Deadline	1
Priority	High
Assigned to	Backend (2), QA (1), Tech-Support (1)
Task types	Bug-fix
Tags	Java, Linux, Server, Crash

The above fields are auto-populated based on the incoming request from the client, customer or end-user. Based on the assignees suggested by the LLM, the duration is forecasted using the Ridge Regression model on the enterprise data.

VIII. CONCLUSION & FUTURE WORK

This paper attempts to identify the potential of LLMs for automating Agile project management. This delves into a new paradigm of data-driven Agile task scheduling and opens up new avenues of innovation in software management. Furthermore, the research introduces a novel coupling of large language models with enterprise data. In this context, several regression techniques for heuristic task scheduling have been elaborately discussed.

Future research on intelligent task-based scheduling and reinforcement-based task management can usher in a new era of Agile project management. Furthermore, LLMs can be effectively employed for analyzing screenshots uploaded by customers for problem resolution. The next decade beacons a monumental disruption in traditional project management processes.

REFERENCES

[1] A. Ahmed, S. Ahmad, N. Ehsan, E. Mirza and S. Z. Sarwar, "Agile software development: Impact on productivity and quality," 2010 IEEE International Conference on Management of Innovation & Technology, Singapore, 2010, pp. 287-291, doi: 10.1109/ICMIT.2010.5492703.

[2] B. Choudhary and S. K. Rakesh, "An approach using agile method for software development," 2016 International Conference on Innovation and Challenges in Cyber Security (ICICCS-INBUSH), Greater Noida, India, 2016, pp. 155-158, doi: 10.1109/ICICCS.2016.7542304.

[3] F. Paetsch, A. Eberlein and F. Maurer, "Requirements engineering and agile software development," WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003., Linz, Austria, 2003, pp. 308-313, doi: 10.1109/ENABL.2003.1231428.

[4] P. Sankhe, S. Mathur, T. B. Rehman and M. Dixit, "Review of an Agile Software Development Methodology with SCRUM & Extreme Programming," 2022 IEEE International Conference on Current Development in Engineering and Technology (CCET), Bhopal, India, 2022, pp. 1-6, doi: 10.1109/CCET56606.2022.10080640.

[5] A. Srivastava, S. Bhardwaj and S. Saraswat, "SCRUM model for agile methodology," 2017 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, India, 2017, pp. 864-869, doi: 10.1109/CCAA.2017.8229928.

[6] R. Juric, "Extreme programming and its development practices," ITI 2000. Proceedings of the 22nd International Conference on Information Technology Interfaces (Cat. No.00EX411), Pula, Croatia, 2000, pp. 97-104.

[7] S. Thakur and H. Singh, "FDRD: Feature driven reuse development process model," 2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies, Ramanathapuram, India, 2014, pp. 1593-1598, doi: 10.1109/ICACCCT.2014.7019376.

[8] M. O. Ahmad, J. Markkula and M. Oivo, "Kanban in software development: A systematic literature review," 2013 39th Euromicro Conference on Software Engineering and Advanced Applications, Santander, Spain, 2013, pp. 9-16, doi: 10.1109/SEAA.2013.28.

[9] W. R. Fitriani, P. Rahayu and D. I. Sensuse, "Challenges in agile software development: A systematic literature review," 2016 International Conference on Advanced Computer Science and Information Systems (ICACSIS), Malang, Indonesia, 2016, pp. 155-164, doi: 10.1109/ICACSIS.2016.7872736.

[10] A. K. Nisyak, K. Rizkiyah and T. Raharjo, "Human Related Challenges in Agile Software Development of Government Outsourcing Project," 2020 7th International Conference on Electrical Engineering, Computer Sciences and Informatics (EECSI), Yogyakarta, Indonesia, 2020, pp. 222-229, doi: 10.23919/EECSI50503.2020.9251899.

[11] B. Choudhary and S. K. Rakesh, "An approach using agile method for software development," 2016 International Conference on Innovation and Challenges in Cyber Security (ICICCS-INBUSH), Greater Noida, India, 2016, pp. 155-158, doi: 10.1109/ICICCS.2016.7542304.

[12] A. Ahmed, S. Ahmad, N. Ehsan, E. Mirza and S. Z. Sarwar, "Agile software development: Impact on productivity and quality," 2010 IEEE International Conference on Management of Innovation & Technology, Singapore, 2010, pp. 287-291, doi: 10.1109/ICMIT.2010.5492703.

[13] N. Saleem, S. Mathrani and A. Sajjad, "Exploring the Evolutionary Characteristics of Project Management Approaches at Different Levels of Operations," 2021 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), Brisbane, Australia, 2021, pp. 1-7, doi: 10.1109/CSDE53843.2021.9718471.

[14] S. Bushuyev, S. Onyshchenko, A. Bondar, N. Bushuyeva and A. Obronova, "Assessment and Monitoring Project Management Quality Based on the Entropy Approach," 2022 International Conference on Smart Information Systems and Technologies (SIST), Nur-Sultan, Kazakhstan, 2022, pp. 1-5, doi: 10.1109/SIST54437.2022.9945775.

[15] S. Zhang and L. Jin, "Research on Software Project Schedule Management Method based on Monte Carlo Simulation," 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 2020, pp. 1605-1608, doi: 10.1109/ITOEC49072.2020.9141570.

[16] M. Jovanović and M. Campbell, "Generative Artificial Intelligence: Trends and Prospects," in Computer, vol. 55, no. 10, pp. 107-112, Oct. 2022, doi: 10.1109/MC.2022.3192720.

[17] P. Barham, A. Chowdhery, J. Dean, S. Ghemawat, S. Hand, D. Hurt, M. Isard, H. Lim, R. Pang, S. Roy, B. Saeta, P. Schuh, R. Sepassi, L. El Shafey, C. A. Thekkath, and Y. Wu, "Pathways: Asynchronous Distributed Dataflow for ML," 2022. Available: https://arxiv.org/abs/2203.12533.

[18] S. Narang and A. Chowdhery, "Pathways Language Model (PaLM): Scaling to 540 Billion Parameters for Breakthrough Performance," https://blog.research.google/2022/04/pathways-language-model-palm-scaling-to.html (accessed Jan. 4, 2024)

[19] H. Wang, Z. Ren, Z. Yu, Y. Zhang, J. Liu and H. Cui, "CoupHM: Task Scheduling Using Gradient Based Optimization for Human-Machine Computing Systems," 2022 IEEE 28th International Conference on Parallel and Distributed Systems (ICPADS), Nanjing, China, 2023, pp. 859-866, doi: 10.1109/ICPADS56603.2022.00116.

[20] M. Abufadda and K. Mansour, "A Survey of Synthetic Data Generation for Machine Learning," 2021 22nd International Arab Conference on Information Technology (ACIT), Muscat, Oman, 2021, pp. 1-7, doi: 10.1109/ACIT53391.2021.9677302.

[21] D. Chicco, M. J. Warren, G. Jurman, "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation," PeerJ Comput Sci., vol. 7, e623, Jul 5, 2021. DOI: 10.7717/peerj-cs.623. PMID: 34307865; PMCID: PMC8279135.

[22] A. M. A. Sai, K. L. Sai Eswar, K. S. Sai Harshith, P. Raghavendra, G. Y. Kiran and M. V., "Study of Lasso and Ridge Regression using ADMM," 2022 2nd International Conference on Intelligent Technologies (CONIT), Hubli, India, 2022, pp. 1-8, doi: 10.1109/CONIT55038.2022.9847706.