Integrating Generative AI for Advancing Agile Software Development and Mitigating Project Management Challenges

Anas BAHI[1], Jihane GHARIB[2], Youssef GAHI[3]
Laboratory of Applied Geophysics, Geotechnics, Engineering Geology and Environment, Mohammadia School of Engineers, Mohammed V University in Rabat, Morocco[1]
Laboratory of Engineering Sciences, National School of Applied Sciences, Ibn Tofail University, Kenitra, Morocco[2, 3]
School of Electrical Engineering and Computer Science, University of Ottawa, 800 King Edward Ave., Ottawa, ON, Canada[3]

***Abstract*—Agile software development emphasizes iterative progress, adaptability, and stakeholder collaboration. It champions flexible planning, continuous improvement, and rapid delivery, aiming to respond swiftly to change and deliver value efficiently. Integrating Generative Artificial Intelligence (AI) into Agile software development processes presents a promising avenue for overcoming project management challenges and enhancing the efficiency and effectiveness of software development endeavors. This paper explores the potential benefits of leveraging Generative AI in Agile methodologies, aiming to streamline development workflows, foster innovation, and mitigate common project management challenges. By harnessing the capabilities of Generative AI for tasks such as code generation, automated testing, and predictive analytics, Agile teams can augment their productivity, accelerate delivery cycles, and improve the quality of software products. Additionally, Generative AI offers opportunities for enhancing collaboration, facilitating decision-making, and addressing uncertainties inherent in Agile project management. Through an in-depth analysis of the integration of Generative AI within Agile frameworks, this paper provides insights into how organizations can harness the transformative potential of AI to advance Agile software development practices and navigate the complexities of modern software projects more effectively.**

Keywords*—*Artificial intelligence; software engineering; Agile software development

I. Introduction

Project work that seeks to regulate and create the project's outputs in brief iterations and adapt to numerous shifts in circumstances during the project is made possible by agile project management approaches. Classical project management has shifted from managerial and administrative duties to project coaching in agile project management techniques. Many circumstances influence an agile project's success, and managing these pain points is essential. Software development for various systems is one of the many IT-related projects that agile project management (APM) has grown famous for handling. The project manager and the organization must exercise strict discipline when using APM. As Agile Project Management demands meticulous discipline for successful software development, exploring AI to enhance this process presents a novel and compelling research domain.

From a scientific perspective, artificial intelligence (AI) encompasses a variety of methods and strategies, including robotics, machine learning, and machine reasoning [1]. With AI's growing capabilities, there is curiosity about whether certain aspects of project management today may be automated or if some activities are delegated to AI, which would lessen the workload of typical project managers and assist in challenging circumstances [2]. One way to conceptualize intelligent project management is as artificial intelligence applied to automate certain aspects of the process when suitable. In this situation, Generative Artificial Intelligence (GenAI), which harnesses advanced algorithms to create content and solutions that mimic human-like ingenuity, could be beneficial in controlling the various pain points of projects and shifting to intelligent project management. Tools for generative AI, like CoPilot, GPT, and Bard, have become increasingly popular. They also increase the productivity of software engineering. [3]

GenAI introduces many positive impacts on Agile methodologies, revolutionizing how teams approach software development and project management. GenAI enhances Agile practices by augmenting creativity, efficiency, and decision-making processes. Through its ability to analyze vast datasets and generate novel solutions, GenAI aids in ideation sessions, helping teams overcome creative hurdles and foster innovation within Agile iterations. Additionally, GenAI streamlines development processes by automating repetitive tasks, such as code generation, testing, and documentation, thereby freeing up valuable time for team members to focus on high-value activities. Moreover, GenAI facilitates data-driven decision-making by providing insights into project metrics, team performance, and potential risks, enabling Agile teams to adapt and respond to changing requirements swiftly. Integrating GenAI into Agile methodologies leads to accelerated development cycles, improved collaboration, and enhanced product quality, ultimately driving tremendous success in software projects. In particular, GenAI may help with day-to-day tasks by providing advice, coaching, and recommendations.

This piece delves into the technologies and applications of Generative AI within the software sector. Software engineering is going to see significant changes as a result of GenAI.

However, research on the subject is still in its early stages. There aren't any significant case-studies found in the literature that tackle the integration of GenAI within the Agile methodologies This research agenda is crucial for guiding academics and practitioners by highlighting current applications and shaping future studies. This paper presents an adapted framework of explicit integration of Generative AI into one of the most commonly used agile methodology by developers. Our goal is to give a real case study of how this can be achieved, also discuss the pitfalls and the potential limitations.

This study's primary goal is to investigate how AI can alleviate the difficulties and pain points related to agile project management. Our objective is to delve deeper into the practical aspects of implementing generative AI in agile projects and to provide a balanced view by discussing both the benefits and challenges of Gen AI in software development. The following questions will be addressed and answered in the subsequent sections of this paper.

- RQ1: What problems do agile projects typically have?
- RQ2: How can GenAI assist in overcoming common challenges encountered in agile projects?
- RQ3: Can GenAI help with the software development process? Can GenAI used in Agile frameworks to enhance project management assistance?

To do so, we structured this paper based on four sections. The first one is this introduction; the second section delineates the background notions of this article, which are Agile project management (APM) & Generative Artificial Intelligence (GenAI). This includes the theoretical section, which introduces standard agile methodologies, both at small and large scales, and discusses the mechanism of artificial intelligence employed in this study. The following section helps ascertain the typical difficulties that different agile project management techniques could provide by conducting a literature review. A theoretical model of pain points is developed based on these issues. This is followed by the significant impact of GenAI in Software developments and applications of GenAI as a solution for several real-world project management problems. The last section encounters our developed framework to demonstrate how GenAI can enhance agile software development practices.

II. Literature Review of Agile Methodologies for Software Development

A. Background of Agile Project Management

Traditional project management is referred to by the Project Management Institute as "predictive project management," in contrast to agile methods. Non-agile projects are called "plan-driven" because they emphasize creating a detailed plan in advance and carrying it out throughout the project [4]. In software development, the term "agile" was first used in 2001 by seventeen software developers trying to address a critical problem: managing projects using the waterfall model, which divides the development into discrete phases [5]. Using four core values as a foundation, the seventeen developers created the Manifesto for Agile Software Development: "Individuals and interactions over processes and tools, working software over comprehensive documentation; customer collaboration over contract negotiation; and responding to change over following a plan." Any approach that upholds the values and tenets of the Agile Manifesto is included in the agile philosophy [6]. The agile framework is made to handle change effectively, in contrast to traditional systems, which emphasize rigorous change management and upfront planning. Agile software development involves self-organizing teams working with clients to evolve needs. As a result, agile projects need the customer to be heavily involved at every stage to give regular, honest, and transparent feedback [5] [7].

Agile project management is an iterative and flexible approach to project management that emphasizes adaptability, collaboration, and customer satisfaction. It is particularly well-suited for projects where requirements are expected to change frequently or need to be more well-defined initially. Agile methodologies focus on delivering small, incremental portions of a project in short time frames, called iterations or sprints, typically ranging from one to four weeks. Agile project management has gained widespread adoption in software development and is increasingly applied in various other industries where flexibility and responsiveness are valued. Its principles can be adapted and scaled to suit the needs of diverse projects and organizations. The key characteristics of Agile project management are:

- Customer Collaboration
- Emphasis on Individuals and Interactions
- Continuous Delivery
- Flexibility and Adaptability
- Cross-functional Teams
- Continuous Feedback
- Iterative Development

Projects are broken down into small, manageable iterations, allowing for continuous improvement and adaptation as the project progresses. Regular and close collaboration with stakeholders, including the end-users or customers, ensures that the delivered product meets their expectations and requirements. Agile embraces changes in requirements even late in the development process. This flexibility allows teams to respond quickly to evolving priorities or new insights. Agile encourages forming cross-functional, self-organizing teams with all the skills necessary to complete the project within the team, promoting better communication and collaboration. Regular reviews and feedback sessions are integral to Agile methodologies. This constant feedback loop helps identify and address issues promptly, ensuring the project remains aligned with stakeholders' expectations. Agile promotes delivering a potentially shippable product at the end of each iteration, ensuring that there is always a tangible output, even if the project still needs to be completed. Agile values individuals and interactions over processes and tools, emphasizing the importance of effective communication and collaboration within the team.

B. Background Frameworks

Agile frameworks such as Scrum, XP, Kanban, and Lean Software Development were initially developed for small-scale projects. Other large-scale frameworks exist, such as Scaled Agile (SAFe), Disciplined Agile (DA), and Nexus. Several popular Agile frameworks are widely used in project management to facilitate iterative and adaptive development. Each framework offers its own set of practices, roles, and ceremonies to guide teams in delivering value incrementally. These frameworks provide structure and guidance for teams adopting Agile principles, allowing them to tailor their approach based on their project's specific needs and context. Choosing a particular framework depends on the project's size, complexity, industry, and organizational culture. Agile methodologies employ various ceremonies and artifacts to facilitate effective project management and collaboration within teams:

1) Ceremonies:

a) Sprint planning: A meeting held at the beginning of each sprint where the team plans the work to be completed during the sprint. It involves selecting user stories from the backlog, estimating effort, and committing to a sprint goal.

b) Daily standup (Daily Scrum): A short meeting where team members discuss their progress, what they plan to work on next, and any obstacles they face. It promotes transparency, collaboration, and alignment within the team.

c) Sprint review: A meeting held at the end of each sprint where the team demonstrates the work completed during the sprint to stakeholders and collects feedback. It provides an opportunity to review the product increment and adjust priorities based on stakeholder input.

d) Sprint retrospective: A meeting held at the end of each sprint where the team reflects on the previous sprint, discusses what went well, what could be improved, and identifies actionable items for process enhancement in future sprints. It fosters continuous improvement and learning within the team.

2) Artifacts:

a) Product backlog: A prioritized list of a product's desired features, enhancements, and fixes. It serves as the single source of requirements for the team and is continuously refined and reprioritized.

b) Sprint backlog: A subset of the product backlog containing the user stories and tasks the team commits to completing during a sprint. It helps the team focus on the work in the current sprint.

c) Increment: The sum of all the completed and potentially shippable product backlog items at the end of a sprint. It represents the tangible outcome of a sprint and provides value to stakeholders.

d) Burn-down chart: A visual representation of the remaining work (usually measured in story points or tasks) throughout a sprint. It helps the team track progress towards completing the sprint goal and identifies potential issues early.

These ceremonies and artifacts are integral components of Agile methodologies, facilitating collaboration, transparency, and adaptability within teams to deliver high-quality products iteratively and incrementally.

C. Agile Pain Points in Software Development

Significant difficulties could arise in agile projects when the complexity, context, and scope evolve [8]. Several study reports have noted various problems: [9], [10], [11], [12], [13], [14], [15], [16], [17], [18]. Using a variety of publications, the literature study identifies the most prevalent issues that arise in agile initiatives.

Beginning with [12], it has been observed that development team members who have experience with rigid approaches, such as the waterfall approach, may be reluctant to adopt agile practices. Other difficulties that have been brought up include the development team's ignorance of the agile methodology, the lack of participation from upper management, problems with technical support, incompatibility with current infrastructure and procedures, and the existence of time and financial constraints [12]. The research in [9] is based on agile projects in the public sector found several potential trouble spots. These factors included the following: responsibilities within the agile setup, documentation, education, experience, and commitment; location of agile teams; complexity of software architecture and systems integration; and stakeholder communication and involvement [9]. Additionally, research has shown that many agile methodologies, including Scrum, XP, and Lean, are combined [10]. Agile methodology is thus characterized in a comprehensive sense. Agile practitioners encounter various complicated, context-specific challenges, some of which have persisted for years and are difficult to address successfully. On the other hand, although some problem areas still exist, their emphasis has changed. Although practitioners have embraced agile approaches, questions about their ongoing efficacy have yet to be raised. Many complex topics have yet to get much academic attention, including governance, business participation, transformation, failure, and the impact of claims and constraints [10]. Furthermore, some issues, like contracts and government, need more attention from researchers. In contrast, other problems, like business and IT transformation, have received attention but have yet to have the anticipated influence on industry practices [11]. It's also more complicated to choose a large-scale agile framework. The study in [14] claims that while various agile frameworks provide basic understandings, they soon become unhelpful when applied outside of the context of the framework in which they were designed. Due to the COVID-19 epidemic, there has been an increase in remote project work globally. Working on remote projects offers its own set of difficulties [13]. Based on research conducted by [15] on self-organizing teams, the challenges at the project, team, person, and task levels are categorized. Activities involving the team, senior management (SM), and clients are included at the project level; activities involving the core development team and their SM are included at the team level; team member activities are included at the individual level; and technical tasks are contained in the task level [19]. A particular pain point model has been developed to examine some of the most prevalent pain points mentioned in the literature about agile projects to help manage these difficulties more effectively. This model aims to explain the common causes of these difficulties and offer workable

ways to solve them, regardless of the project's scale. A fishbone diagram representing the primary pain spots has been created using the information presented above, as seen in Fig. 1. This graphic shows two unique problems for every problem category in the previous literature review. These challenges can be categorized based on the project, personnel, methods, persistence, and estimated effort.

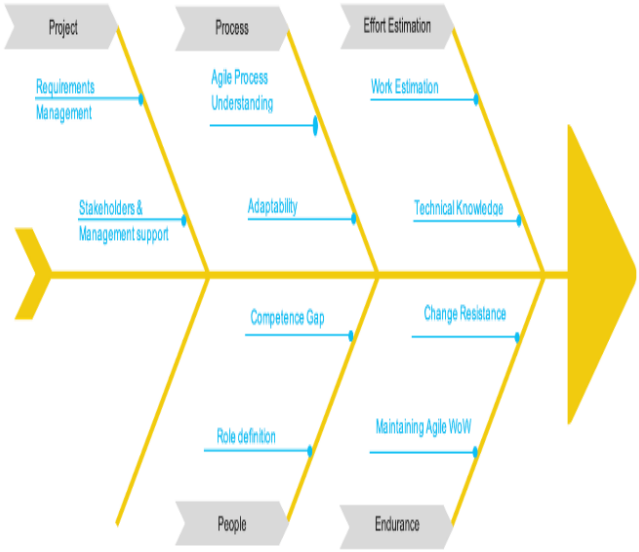

Fig. 1. Primary pain points arising from agile difficulties.

To answer RQ1: What issues do agile projects typically have? Common pain points in agile methodologies include problems with requirement management, managerial and stakeholder support, role clarity, team member overlap, understanding of agile processes, adaptability to various shifts, resistance to change, maintaining agile practices, effort prediction, and sufficient technical expertise. Every one of these difficulties may affect agile initiatives and their supervision of them.

III. GenAI Revolution to the Software Delivery

A. Background Historical Improvements

More than any other recent breakthrough, Generative Artificial Intelligence (GenAI) can drastically alter the software industry. According to Bill Gates, it represents the most significant advancement since the creation of the Internet. Software productivity can be enhanced through various means, including automating monotonous tasks like requirements traceability or testing, enhancing software quality by creating test suites from requirements, and automating workflows by directing work products to the appropriate stage in a production pipeline [20]. However, generative AI poses new risks because it is neither deterministic nor explicable. Prominent examples of restrictions on usage in professional software engineering include Intellectual Property Rights (IPR) and cybersecurity.

The field of generative AI has been around for a while. Researchers were reluctant to introduce such technology to the general public as they were unsure of its validity. People tend to close their eyes to apparent risks when faced with a projected gold rush, as we have seen repeatedly in the history of IT [21]. Ultimately, even well-intended tools can have disastrous effects. After GPT was eventually made available to the general public in 2022, the AI arms race began at a pace never seen before. GPT only needed two months to gain one hundred million users [3].

For years, using Google or StackOverflow has been a standard part of the work for any coder. Countless code repositories are available online, and search engines have gotten better at indexing them. Community advice sites like StackOverflow offer insightful commentary and well-reasoned answers to user questions. One feature that both search engines and Q&A websites have in common is the ability to pull up previously stored material.

GenAI differs. As the name implies, it can synthesize or generate responses to your queries. Instead of searching through a premade one, it will create a reaction for you, as traditional search engines do. The response is predicated on enormous training data, including search engine stored and indexed content [22]. Generative AI is trained further to deliver insightful responses using input from humans. Many human trainers ask questions and comment on the generated responses, rewarding correct responses and penalizing incorrect ones. While protecting against unfavorable replies, this type of reinforcement learning directs the system toward producing more accurate responses. This has given rise to glimmers of a new mode of operation that centers on "prompt engineering," determining the best way to phrase a question or an entire conversation. Generative AI does not work with individual questions and answers; it maintains a context window, which can guide the AI in generating contextually relevant and well-informed responses [3].

B. Examination of Specific Gen AI Technologies and their Direct Applications in Agile Software Environments.

Numerous generative software platforms enable the conversion of straightforward instructions into computer code. Development tools and editors, including Visual Studio Code, Visual Studio, NeoVim, and JetBrains IDEs, can be extended with GitHub Copilot. It provides code completion driven by OpenAI Codex, a Generative AI system created by OpenAI.

The recent release of Copilot X, powered by GPT-4, is promising. In addition to better autocompletion, it can help with other development activities, including code comprehension, pull request improvement, scripting, and shell tool support [22]. GPT-4 can produce code from docstrings in software engineering interviews and answer coding questions on par with or better than human performance. It is capable of interacting with LaTeX and front-end programming. It can run Python, Pseudo, and reverse engineer programs. The company that created GPT-4, OpenAI, provides programmatic access to its LLMs. These advanced LLMs, such as GPT-4, represent a paradigm shift in computing, harnessing vast knowledge and understanding of context to perform complex tasks. Their natural language processing and generation capabilities have significant implications, potentially revolutionizing how we interact with technology and automate sophisticated tasks. This implies that programmers can incorporate them into their apps and use them conversationally. Moreover, developing plugins, a means of integrating the underlying models with other

services capable of answering inquiries and taking appropriate action is feasible.

Unlike its competitors, Tabnine stands out from the crowd by giving special consideration to licensing and privacy considerations. It provides code completion at the line or function level. Only permissive license open-source software has been used to teach Tabnine. Additionally, it guarantees developers that it does not keep any of the code that uses it, and the underlying models can be downloaded and used locally rather than only being available as a service [23]. A summary of various standard technologies is given in Fig. 2 [3]. Take note of how quickly the terrain is changing. We should anticipate the introduction of new tools and the advancement of current ones.

To answer the first part of RQ3: Can GenAI help with the software development process? Can GenAI used in Agile frameworks to enhance project management assistance? Fig. 2 provides a detailed argument of how.

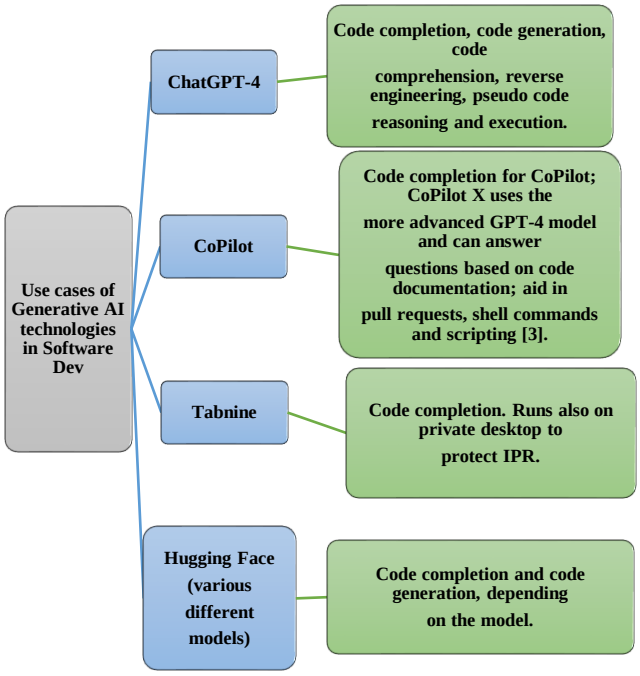

Fig. 2. Code development using generative AI technologies.

IV. GenAI and its Role in Enhancing Agile Software Development Practices

A. GenAI to Overcome Agile Management Pain Points

GenAI is emerging as a transformative force in addressing various pain points within Agile project management, offering innovative solutions to enhance efficiency, collaboration, and adaptability. Generative AI, through its ability to analyze patterns and generate novel ideas, can assist teams in brainstorming sessions, helping overcome creative hurdles and fostering a culture of continuous innovation. Moreover, Generative AI algorithms can analyze historical project data, team performance, and external factors to provide intelligent recommendations for optimizing resource distribution and refining task priorities. This streamlines decision-making processes and contributes to a more adaptive and responsive project management approach. Furthermore, Generative AI's data processing capabilities enable it to analyze and interpret large datasets, facilitating the extraction of valuable insights that can inform strategic decisions and project optimizations. This can significantly improve the project's overall performance and outcomes. Also, Generative AI-powered collaboration tools can enhance communication by offering real-time language translation, aiding cross-cultural collaboration, and minimizing misunderstandings.

Furthermore, Generative AI can contribute to automated testing and quality assurance processes, addressing the challenge of maintaining product quality within tight Agile development cycles. By generating and executing test scenarios, AI can identify potential issues early in the development process, reducing the risk of defects and enhancing overall product reliability. Following the pain points assembled in Fig. 1, Table I propose solutions to them using GenAI according to the literature:

This study draws on the highlighted challenges to align the answers. This systematic approach makes it easier to see how each suggested solution specifically tackles the given problems, which increases the solutions' overall efficacy. Furthermore, this alignment ensures that the suggested remedies successfully tackle the issues discovered, opening the door for a more comprehensive and successful implementation plan [34]. In conclusion, to answer RQ2, Generative AI is a valuable ally in overcoming Agile project pain points. Its ability to support creative imagination, optimize resource allocation, analyze large datasets, enhance collaboration, and automate testing processes makes it a versatile tool for Agile teams striving to achieve higher levels of efficiency, adaptability, and success in their project endeavors. As the field of Generative AI continues to evolve, its potential to revolutionize Agile project management practices is likely to grow, providing teams with innovative solutions to navigate the complexities of dynamic and iterative development processes. It can offer insightful information and recommendations consistent with findings from literature reviews. GenAI's responses can be shaped by particular prompt patterns, which makes it easier to create reviews and suggestions for addressing project difficulties. GenAI has excellent potential to alleviate typical pain points in agile projects as an auxiliary tool. It can help predict problems previously occurring in its pre-trained data by using its historical data to benchmark and identify similarities in each project. Though GenAI can offer valuable recommendations for handling pain points, it is nevertheless imperative that the project team continue to be skilled in agile techniques and only rely on GenAI's answers with rigorous evaluation [35].

TABLE. I SOLUTION MAPPING USING GENAI

Project	**Requirements management**	According to a predetermined specification pattern, GenAI is configured to generate requirements. Responding to a prompt, it can request explanations, provide recommendations, and generate high-level requirements tabularly. However, depending on the round they were formed, these needs may have different contents [4] [24].
	Stakeholders & management support	As a steering committee member, GenAI understands its role and provides presentations upon request. It seems to understand the subject at hand and provides useful advice. Most of these recommendations seem reasonable. Nonetheless, a notable degree of variation exists in the proposed activities for every round [25].
People	**Role definition**	Based on the comments given, GenAI seems to understand the prompt pattern and provides a preliminary role description that outlines the duties and skills required for the position. Furthermore, it indicates that the roles could need to be modified based on the project's actual requirements [26].
	Competence Gap	GenAI can quickly identify the task and provide a variety of analysis and suggestions. Even with the identical prompt pattern, the results can vary greatly, bringing a variety of careful factors into play [27].
Process	**Agile process understanding**	It seems that GenAI understands the prompt's context, which is based on agile coaching concepts. Even though the design differs greatly, the suggested concepts are still useful. The coaching places a strong emphasis on continuous improvement and learning, and it provides a detailed program [4].
	adaptability	It seems that GenAI understands the prompt context, which emphasizes adaptability. It can automatically create an action plan upon request and can adjust this plan with additional prompt elements. Though answers to the same prompt can differ, it effectively identifies problems and provides pertinent advice [4] [28].
Endurance	**Change resistance**	It seems that GenAI understands the prompt pattern associated with resistance to change and is able to create an action plan based on feedback from users. This approach includes tactics to address resistance to change, yet using the same stimulus more than once may yield different results [11] [29][30].
	Maintaining agile way of work	GenAI is knowledgeable with the fundamentals of agile processes and the jobs that go along with them, such as those of the Scrum Master. After gaining an understanding of the backdrop, it goes into detail on the significance of cross-functional roles and sprint planning. The initial question seems to be aimed mostly at a senior developer, with the intention of educating and motivating them in their work [4] [31].
Effort estimation	**Work estimation**	It seems like GenAI understands the purpose of an effort estimation prompt. It provides a list of tasks that it determines are required, together with descriptions and time estimates expressed in days. It also discusses the assumptions made in the estimates and how those assumptions might affect the amount of work needed. Repeating the inquiry, however, results in a substantial variation in the estimation [4] [32].
	Technical knowledge	The prompt is identified by GenAI as one that deals with team management of technical knowledge. It creates a suggestion based on the further details in the prompt about how the team could overcome this difficulty. It's interesting to note that different rounds of the same prompt produce diverse answers [4] [33].

B. GenAI to Enhance Software development using Agile Frameworks

As mentioned in Section II, the Scrum framework is a small-scale process used mainly by developing teams and IT engineers to deliver the best quality products with the highest adaptability. Scrum is a widely adopted Agile framework for project management that provides a structured and iterative approach to software development and other types of projects. Scrum is a widely adopted Agile framework for project management that provides a structured and iterative approach to software development and different kinds of projects.

Integrating Generative Artificial Intelligence (GenAI) into Scrum can enhance efficiency, innovation, and decision-making. We elaborated in Fig. 3 a framework illustrating how GenAI can assist at different stages of the Scrum process:

- Backlog Refinement and Clarification Through Automated User Story Generation: GenAI can analyze historical data, user feedback, and market trends to propose potential user stories for the backlog. It helps generate diverse and creative ideas, fostering innovation in feature development. Team members can interact with GenAI to seek clarification on user stories or acceptance criteria, ensuring a shared understanding of requirements [36].
- Estimation and research assistance: GenAI can assist in estimating the complexity of user stories by analyzing historical data and patterns. It can offer insights into potential risks and challenges associated with specific tasks, aiding the team in better planning. It can help gather information related to tasks or user stories, preparing sprint planning meetings.
- Progress tracking: GenAI tools can analyze individual and team progress based on task updates, identifying potential bottlenecks or areas needing additional support. It assists in providing a holistic view of the team's performance.
- Generating meeting summaries and Automated Testing: GenAI can assist in summarizing the sprint review meeting and highlighting key achievements and outcomes. GenAI can contribute to automated testing processes, helping identify defects and ensure the reliability of the delivered product increment. It accelerates the testing phase, allowing the team to focus on refining features.
- Feedback Analysis and Brainstorming Improvement Ideas: GenAI can analyze feedback received during the sprint and provide insights into areas of improvement. It helps in identifying patterns, potential process enhancements, and team dynamics. Teams can engage GenAI in brainstorming sessions to generate ideas for process improvements during retrospectives.
- Dynamic Prioritization and Answering Questions on Backlog Items: GenAI can dynamically prioritize backlog items based on changing requirements, user feedback, and business priorities. It assists in maintaining a backlog aligned with the project's evolving needs: GenAI can provide additional information or context for backlog items, helping in prioritization discussions.

- Code Suggestions and Reviews: GenAI tools can provide code suggestions, identify potential improvements, and even assist in code reviews. It fosters collaboration among developers and helps maintain coding standards.

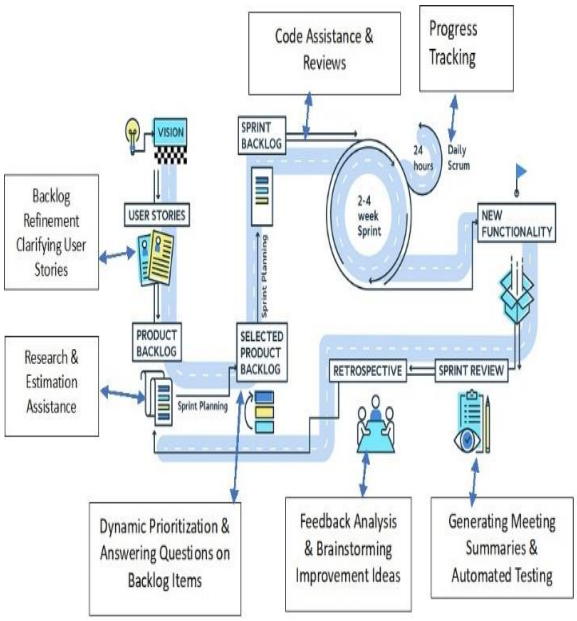

Fig. 3. Generative AI Integration across the scrum framework stages.

Another noticeable improvement of the integration of GenAI into the Scrum process is knowledge sharing through automated documentation; GenAI can assist in generating documentation for code changes, updates, and system architecture. It ensures that knowledge is captured and shared across the team efficiently. Also, there is continuous improvement using data-driven insights; GenAI analyzes historical project data to provide insights into team performance, sprint outcomes, and areas for constant improvement. It contributes to data-driven decision-making in retrospectives.

C. Considerations and Limits

To answer the second part of the RQ3: Can GenAI help with the software development process? Can GenAI used in Agile frameworks to enhance project management assistance? Our developed framework, which combines conversations with GenAI through prompt patterns in Section IV(B), demonstrates its usefulness in producing a variety of templates and rough designs for agile projects. Because of the data it was trained on, GenAI, a pre-trained language model, has a basic grasp of different kinds of agile projects. Nevertheless, it is not advisable to place complete faith in the data produced by GenAI due to the inherent variances in project needs and human resources. It's important to proceed cautiously and have agile project specialists analyze GenAI's results because they might not be entirely trustworthy. Some of the upcoming issues are listed below and are still open for discussion if future contributions:

- Ethical Use: Ensure that GenAI is used ethically, avoiding biases and adhering to privacy and security standards.
- Human Oversight: While GenAI can assist in various tasks, human oversight is crucial, especially in critical decision-making processes.
- Context Awareness: While GenAI can provide information, it may need to gain awareness of the specific context or nuances of the team's unique processes. Team members should validate information accordingly.
- Data Security: Ensure that sensitive information is not shared with GenAI, and be mindful of data security considerations.
- Complementary Role: GenAI should be seen as a complementary tool that aids communication and information retrieval, but human collaboration and decision-making remain essential.

Integrating GenAI into the Scrum process requires a collaborative approach, where the technology augments the capabilities of the Scrum team rather than replacing human interactions. Regular evaluations and adjustments should be made to optimize the use of GenAI based on the evolving needs of the team and project. Leveraging GenAI in the Scrum process can enhance communication, provide quick information access, and streamline certain aspects of collaboration. However, it's crucial to integrate it judiciously, considering the specific needs and dynamics of the Scrum team. Regular feedback and adjustments will help optimize its utility within the Scrum framework.

V. Conclusion

It's impossible to envisage a time when software programmers would be paid as much as they are now in a world of low code and generative artificial intelligence. There will be changes to several established roles, like programmer. Given the rate of advancement, we anticipate that, compared to today, very few software organizations will not have an AI-augmented development and testing approach over the next three years. Generative AI will control most mobile apps and internet content, either entirely or partially. Software engineers will require new skills, such as refining software that is generated automatically, feeding learning engines, and investigating behaviors that are not explicable. Generative AI will accelerate software development. However, be wary of marketing hype about quick fixes for developing secure and resilient software using non-deterministic technology.

Many recommendations for agile project management are provided by GenAI, which can be especially helpful early on when knowledge and experience with agile methods are scarce. How much GenAI can help with tasks involving highly experienced people still needs to be clarified? Although simple prompt structures seem to work well for now, there is a fascinating field for further research into more intricate prompt designs and their combinations.

ACKNOWLEDGMENT

This research was not funded by any grant.

REFERENCES

[1] "AI Smart Kit: Agile Decision-Making on AI (Abridged Version)." https://web.s.ebscohost.com/ehost/ebookviewer/ebook/ZTAwMHh3d19fMjkxMjA3OF9fQU41?sid=58909a3d49d04dfb8e4ad794808d7e26@redis&vid=1&format=EB&lpid=lp_4&rid=0&runquerystringmethod=eBookDownloadundefined_click_bookcheckouts%3acheckBookAvailability%3aready.

[2] "How AI Will Transform Project Management." https://hbr.org/2023/02/how-ai-will-transform-project-management.

[3] Ebert, C. and P.Louridas: "Generative AI for Software Practitioners". IEEE Software, Vol. 40, No. 4, pp. 30-38,Jul/Aug. 2023.https://doi.org/10.1109/MS.2023.3265877

[4] "Agile Practice Guide," Project Management Institute, Inc., pp. 57–59, 2017.

[5] Sara Hassan Ahmed Sallam, Mohamed Mostafa Fouad, Fahd Hemeida, Relationship between Agile Maturity and Digital Transformation Success, Journal of Advanced Research in Applied Sciences and Engineering Technology 33, Issue 3 (2024) 154-168, https://doi.org/10.37934/araset.33.3.154168

[6] Sarker, Iqbal H., Faisal Faruque, Ujjal Hossen, and Atikur Rahman. "A survey of software development process models in software engineering." International Journal of Software Engineering and Its Applications 9, no. 11 (2015): 55-70. https://doi.org/10.14257/ijseia.2015.9.11.05

[7] Moniruzzaman, A. B. M., and Dr Syed Akhter Hossain. "Comparative Study on Agile software development methodologies." arXiv preprint arXiv:1307.3356 (2013).

[8] Kari Sainio, Pekka Abrahamsson, Generative Artificial Intelligence Assisting In Agile Project Pain Points Empirical study using ChatGPT, Master's Thesis, Faculty of Management and Business, Tampere University, August 2023

[9] J. Nuottila, K. Aaltonen, and J. Kujala, "Challenges of adopting agile methods in a public organization," International Journal of Information Systems and Project Management, vol. 4, no. 3, pp. 65–85, Feb. 2022, doi: 10.12821/ijispm040304.

[10] K. Dikert, M. Paasivaara, and C. Lassenius, "The Journal of Systems and Software Challenges and success factors for large-scale agile transformations: A systematic literature review," J Syst Softw, vol. 119, pp. 87–108, 2016, doi: 10.1016/j.jss.2016.06.013.

[11] P. Gregory, L. Barroca, H. Sharp, A. Deshpande, and K. Taylor, "The challenges that challenge: Engaging with agile practitioners' concerns," Inf Softw Technol, vol. 77, pp. 92–104, 2016, doi: 10.1016/j.infsof.2016.04.006.88

[12] J. Patel and Y. Tymchenko, "Making Sense of Resistance to Agile Adoption Mak-ing Sense of Resistance to Agile Adoption in Waterfall Organizations: Social In-telligence and Leadership".

[13] R. Reunamäki and C. F. Fey, "Remote agile: Problems, solutions, and pitfalls to avoid," Bus Horiz, Oct. 2022, doi: 10.1016/J.BUSHOR.2022.10.003.

[14] K. Conboy and N. Carroll, "Implementing Large-Scale Agile Frameworks: Chal-lenges and Recommendations," IEEE Software. 2019. doi: 10.1109/MS.2018.2884865.

[15] R. Hoda and L. K. Murugesan, "Multi-level agile project management challenges: A self-organizing team perspective," Journal of Systems and Software, vol. 117, pp. 245–257, Jul. 2016, doi: 10.1016/J.JSS.2016.02.049.

[16] M. Shameem, R. Ranjan Kumar, C. Kumar, B. Chandra, and A. A. Khan, "Priori-tizing challenges of agile process in distributed software development environment using analytic hierarchy process," 2018, doi: 10.1002/smr.1979.

[17] E. Kula, E. Greuter, A. van Deursen, and G. Gousios, "Factors Affecting On-Time Delivery in Large-Scale Agile Software Development", doi: 10.1109/TSE.2021.3101192.

[18] J. Sithambaram, M. H. N. B. M. Nasir, and R. Ahmad, "Issues and challenges im-pacting the successful management of agile-hybrid projects: A grounded theory approach," International Journal of Project Management, vol. 39, no. 5, pp. 474–495, Jul. 2021, doi: 10.1016/j.ijproman.2021.03.002.

[19] M. Paasivaara, B. Behm, C. Lassenius, and M. Hallikainen, "Large-scale agile transformation at Ericsson: a case study," Empir Softw Eng, 2018, doi: 10.1007/s10664-017-9555-8.

[20] Devlin, Jacob, et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." arXiv preprint arXiv:1810.04805 (2018). https://arxiv.org/abs/1810.04805.

[21] Radford, Alec, et al. "Improving Language Understanding by Generative Pre-Training." OpenAI Blog, June 11,2018. https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.

[22] Wolfram, Stephen. "What is ChatGPT Doing... and Why Does it Work?" (2023). https://wolfr.am/SWChatGPT.

[23] Bubeck, Sébastien, et al. "Sparks of Artificial General Intelligence: Early experiments with GPT-4." arXivpreprint arXiv: 2303.12712 (2023). https://arxiv.org/abs/2303.12712.

[24] K. Vlaanderen, S. Jansen, S. Brinkkemper, and E. Jaspers, "The agile requirements refinery: Applying SCRUM principles to software product management," 2010, doi:10.1016/j.infsof.2010.08.004.

[25] T. Raharjo and B. Purwandari, "Agile project management challenges and map-ping solutions: A systematic literature review," ACM International Conference Proceeding Series, pp. 123–129, Jan. 2020, doi: 10.1145/3378936.3378949.

[26] H. Barke and L. Prechelt, "Role clarity deficiencies can wreck agile teams," PeerJ Comput Sci, vol. 5, pp. 1–20, 2019, doi: 10.7717/PEERJ-CS.241.

[27] N. B. Moe, T. Dingsøyr, and T. Dybå, "Understanding self-organizing teams in agile software development," Proceedings of the Australian Software Engineering Conference, ASWEC, pp. 76–85, 2008, doi: 10.1109/ASWEC.2008.4483195.

[28] Mr. O'Reilly, "In-Depth Report Engaging people and building processes to accelerate results The Drivers of Agility".

[29] R. T. Kreutzer, T. Neugebauer, and A. Pattloch, "Change Management: Shaping Change Successfully," 2018, pp. 197–218. doi: 10.1007/978-3-662-56548-3_3.

[30] D. Koutsikouri, S. Madsen, and N. B. Lindström, "Agile Transformation: How Employees Experience and Cope with Transformative Change," Springer, Cham, 2020, pp. 155–163. doi: 10.1007/978-3-030-58858-8_16.

[31] "Mastering ChatGPT: How to Craft Effective Prompts (Full Guide + Examples)." https://gptbot.io/master-chatgpt-prompting-techniques-guide/ (accessed Jun. 01, 2023).

[32] J. White et al., "A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT," 2023.

[33] E. Colonese, "Agile: The human factors as the weakest link in the chain," Communications in Computer and Information Science, vol. 422, pp. 59–73, 2016, doi: 10.1007/978-3-319-27896-4_6/FIGURES/3.

[34] Anggia Astridita, Teguh Raharjo and Anita Nur Fitriani, "Perceived Benefits and Challenges of Implementing CMMI on Agile Project Management: A Systematic Literature Review" International Journal of Advanced Computer Science and Applications (IJACSA), 15(1), 2024. http://dx.doi.org/10.14569/ IJACSA.2024.0150122.

[35] "Mastering Generative AI and Prompt Engineering: A Practical Guide for Data Scientists 1.1. Evolution of AI: From rule-based to generative models 1.2. Key generative AI models: RNNs, LSTMs, GPT, and more 1.3. Popular use cases for generative AI".

[36] "Managing Software Requirements the Agile Way | Managing Software Requirements the AgileWay."https://learning.oreilly.com/library/view/managingsoftwarerequirements/9781800206465/B16234_FM_Final_NM.xhtml (accessed Feb. 15, 2023).