Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 2 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189

http://amresearchreview.com/index.php/Journal/about

# Generative AI for Software Project Management: A Systematic Review

**Abdullah Khamoor**
Department of Computer Science ILMA University, Karachi Sindh, Pakistan

**Yasir Ahmed**
Department of Computer Science ILMA University, Karachi Sindh, Pakistan

**Dr Humaiz Shaikh (Corresponding Author)**
Department of Software Engineering, ILMA University, Karachi, Sindh, Pakistan
humaiz0416@gmail.com dr.humaiz@ilmauniversity.edu.pk

**Faraz Ahmed Shaikh**
Office of Quality Enhancement Cell, Karachi Metropolitan University (KMU), Karachi, Sindh, Pakistan.

## ABSTRACT

Fast progress in Generative Artificial Intelligence (GenAI) and Large Language Models (LLMs) after the introduction of ChatGPT to the public sector in late 2022 has brought changes in the way knowledge-intensive tasks, such as software project management (SPM), are being conducted. The current study represents the systematic literature review (SLR) of 78 scientific papers, reports, and empirical studies published from 2019 to 2025 concerning the use of GenAI and LLMs as autonomous or semi-autonomous assistants during the SPM process. Based on the PRISMA 2020 guidelines, SLR consolidates evidence related to six main SPM stages: project planning and scheduling, risk identification and mitigation, requirements engineering, effort estimation, documentation generation, and stakeholder communication. As a result, a layered architecture of GenAI-assisted project management information system is proposed together with the introduction of GenAI Utility Model (GUM) that clarifies the connection between task complexity, human intervention, and utility of LLMs. Moreover, the comparison of five most advanced language models—GPT-4o, Claude 3.5 Sonnet, Gemini 1.5 Pro, LLaMA-3, and Microsoft Copilot—is provided based on the capability radar. The review concludes that LLMs show high efficiency in documenting, improving requirements, and assisting early project planning; however, there remain many challenges like hallucinations, lack of reasoning capability, inadequate understanding of the context of specific projects, and ethical issues especially when involved in crucial decision-making in regard to schedule and budgeting. Moreover, the research highlights eight main research gaps and suggests directions for future research in terms of both empirical and design-related studies. In general, the research provides very useful insights for software companies

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 2 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189
http://amresearchreview.com/index.php/Journal/about

implementing GenAI solutions, developers of AI tools, and researchers engaged in software engineering and natural language processing domains

**Keywords**: Generative AI; Large Language Models; Software Project Management; LLM Agents; ChatGPT; Autonomous PM Assistants; Retrieval-Augmented Generation; Agile Software Development; AI Ethics; Systematic Literature Review

## Introduction

The discipline of software project management (SPM) is a complicated one that requires from project managers constant analysis of incomplete information, risk assessment, communication with stakeholders, and adaptation of the plan in case of changes in project conditions [1]. Traditionally, the management of software projects is carried out within established frameworks like Waterfall, PMBOK, PRINCE2, Scrum, and Kanban [2], [3] using project management tools for scheduling and resources allocation [4]. With the fast development of Generative Artificial Intelligence (GenAI) and Large Language Models (LLMs) especially following the appearance of ChatGPT in 2022, a new way of managing projects has appeared. Unlike usual automation tools, LLMs are capable of creating project documentation, summarizing project information, recognizing possible risks, participating in planning activities and communicating with stakeholders via natural language conversations [6]. This allows AI systems to perform tasks that were dependent before only on the knowledge and experience of the human project manager.

There is evidence from recent industry studies of an increasing trend towards the use of artificial intelligence in project management. For instance, the 2023 Project Management Institute (PMI) study indicates that about 21% of all project managers are using AI tools in their work [7]. Another PMI survey undertaken in 2024 indicated that about 20% of respondents known as "trailblazers" use Generative AI in more than 50% of their projects, and above 50% of all respondents use it in 16% to 50% of their projects [8]. In addition, Gartner has forecasted that by 2030, almost 80% of all routine project management tasks would be facilitated by AI, big data analytics, and natural language processing [9]. Such predictions are consistent with general expectations of improvements in productivity in knowledge-based professions through the implementation of AI [68]. Nonetheless, earlier conducted systematic reviews on the topic of LLMs revealed that the focus of academic research on these tools was on such areas as code generation, software testing, and debugging but not software project management [10].

Despite the fast-growing interest in this area of research, however, the current literature still seems to be quite dispersed among different aspects. Separate studies have examined the utilization of LLMs in Agile project management [12], [13], prioritization of backlogs, Scrum facilitation via multi-agent systems [14], as well as evaluation of AI abilities in project management-related tasks [15]. Still, they all seem to be quite independent and fail to give a complete view of the potential and limitations of Generative AI in software project management as a whole. At the same time, the industry reports, organizational literature, and professional blogs are developing and disseminating practical ideas much faster than academic research [73].

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 2 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189
http://amresearchreview.com/index.php/Journal/about

In order to fill this gap, this study makes the following four contributions. First, a literature review, which is based on PRISMA and includes synthesis of 78 academic and industrial papers on the use of GenAI and LLMs for the management of software projects within six critical areas. Second, a conceptual architecture is proposed which consists of layered GenAI-assisted Project Management Information Systems and demonstrates how LLMs work with agent-based modules, RAG processes, and software project management data. Third, GenAI Utility Model (GUM) is introduced, which is a mathematical model that shows the way how the utility of LLMs depends on the complexity of a task and the level of human supervision. And finally, a capability radar for five top LLMs, which is evaluated against six SPM task domains based on the evidence from the reviewed literature. The paper identifies eight research gaps and suggests directions for further empirical and design-based research.

The structure of the rest of the paper is as follows. Section 2 outlines the research methodology employed in the systematic literature review process. Section 3 offers some background information on GenAI and Large Language Models. Sections 4 to 9 contain the results for each of the six software project management areas. Section 10 discusses the conceptual architecture and the GenAI Utility Model, whereas Section 11 focuses on the capability radar and comparison of top LLMs. Lastly, Sections 12 and 13 provide a discussion on the limitations of the research, its practical implications, further research ideas, and conclusions.

## Systematic Review Methodology

### Research Questions

The following research questions will be addressed by means of this SLR:

RQ1: Which capabilities of Generative Artificial Intelligence (GenAI) and Large Language Models (LLMs) have been suggested or demonstrated in the context of software project management tasks?

RQ2: Which technical, ethical, and organizational constraints have been found in using GenAI and LLMs for software project management tasks?

RQ3: In what software project management task domains is there a solid body of evidence about the applicability of GenAI and LLMs, and which domains are less researched?

RQ4: What are the research gaps and future research opportunities that the literature identifies?

### Search Strategy and Databases

A thorough literature search was performed in six major scientific databases such as IEEE Xplore, Scopus, ScienceDirect, ACM Digital Library, Web of Science, and Taylor & Francis Online. The literature search was executed using the following Boolean search string:

("Generative AI" OR "large language model" OR "LLM" OR "ChatGPT" OR "GPT-4" OR "foundation model") AND ("project management" OR "software project" OR "agile" OR "Scrum" OR "sprint planning" OR "risk management" OR "requirements engineering" OR "effort estimation")

The literature search was conducted in January 2025 for articles published from January 2019 to December 2025. To perform a thorough literature search, it was also necessary to include grey literature in line with the PRISMA 2020 statement. The grey literature

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 2 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189

http://amresearchreview.com/index.php/Journal/about

included PMI industry reports, blog posts by practitioners, and preprints from conferences containing relevant insights into the practical application of GenAI in software project management [17].

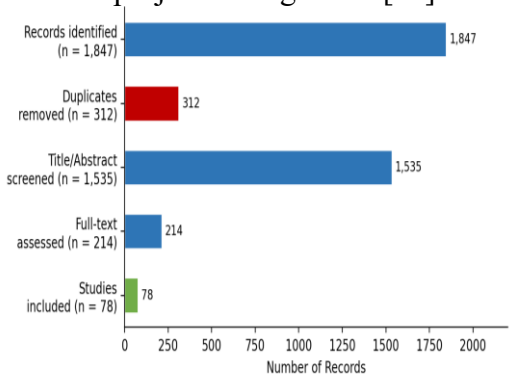



Fig. 1. PRISMA 2020 study selection flow diagram for the systematic literature review (n = 78 studies included), constructed by the authors following the PRISMA 2020 reporting guidelines [17].

### 2.3 Data Extraction and Synthesis

Data were extracted across eight fields: publication year, venue, SPM task domain, LLM(s) evaluated, research method, key findings, limitations reported, and future research directions. Synthesis followed a thematic narrative approach organized by SPM task domain, following precedents in SLR methodology established for software engineering research [18], [19]. Inter-rater reliability for inclusion decisions was assessed using Cohen's kappa on a 15% sample, yielding $\kappa = 0.83$, indicating strong agreement [20]

## 3. Background: Generative AI and LLMs in Software Engineering

### 3.1 From Language Models to Generative Assistants

Large Language Models are neural networks trained on vast corpora of text using transformer architectures [21], enabling them to predict and generate contextually coherent natural-language continuations given a prompt [22]. Models in the GPT family (GPT-3.5, GPT-4, GPT-4o), Anthropic's Claude series, Google's Gemini, Meta's LLaMA, and Microsoft's Copilot represent the current generation of commercially available GenAI systems capable of general-purpose reasoning, code synthesis, document drafting, and multi-turn dialogue [5], [23]. In the software engineering domain, a comprehensive 2024 ACM survey of LLMs for SE covering 395 studies identified application areas including requirements engineering, code generation, testing, and documentation — but noted that SPM remains among the least studied areas [10]. Industry analyses similarly describe GenAI as transforming the broader software development process beyond individual coding tasks [75], although the tendency of LLMs to generate plausible but unsupported content on infrequently represented topics remains a persistent constraint on reliability [74].

### 3.2 Agentic and Multi-Agent LLM Systems

Beyond single-turn generation, recent work has introduced LLM-based autonomous agents capable of multi-step planning, tool invocation, memory management, and self-reflection [24], [25]. Multi-agent LLM systems — in which specialized agents adopt

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 2 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189
http://amresearchreview.com/index.php/Journal/about

roles such as Product Owner, Scrum Master, or Risk Analyst — have been proposed as a mechanism for automating coordination-intensive SPM workflows [14], [26]. The AutoGen framework [27] and similar multi-agent orchestration systems enable LLMs to delegate subtasks, validate outputs through agent debate, and interact with external PM tools via APIs. Nevertheless, the study identified some new problems arising in the coordination of multiple agents and in the consistency of their outputs. According to the TOSEM 2025 publication entitled "Software Engineering with Large Language Model Multi-Agent Systems," the collaborative work of specialized AI agents resulted in more accurate outcomes compared to the work of one large language model alone [28]. Nevertheless, the study identified some new problems arising in the coordination of multiple agents and in the consistency of their outputs.

### 3.3 Retrieval-Augmented Generation (RAG) in PM Contexts

Another key constraint of LLMs used in the domain of enterprise project management is that they operate on knowledge that is known during their training phase and hence cannot access any information that is exclusive to organizations, including project history, risk registers, and other confidential process information. RAG resolves the constraint by accessing relevant project documents in real-time and feeding them to the LLM along with its response generation [29], [30]. This way, the response generated is not based only on the LLM's pre-training knowledge but is instead created by accessing actual project information as well. Many research works have suggested different RAG-based architectures to develop project management knowledge systems, thereby enabling LLMs to respond according to organizational documentation and project artifacts [31]. An empirical research work has shown that user stories generated using RAG-enhanced GenAI framework for Scrum backlog refinement were more context-specific than those generated through standard prompt-based approaches alone [32].

## 4. GenAI in Planning, Scheduling, and Work Breakdown

Planning of the project is probably the most researched application domain of Generative AI in software project management, where research on the potential usage of GenAI to help with agile planning started even before LLMs have been widely adopted [63]. Modern research shows that LLMs can rapidly produce necessary documents for planning the project such as a project scope statement, milestone plan, WBS, communication plan, and initial risk assessment, thus greatly decreasing time required to produce them manually [33], [34]. Practitioner-oriented literature review which analyzed more than 38,700 sources that were published after the launch of ChatGPT till May 2025 showed that planning documents generation was the most popular application of GenAI in software project management, and the second most common usage of GenAI in software project management is meeting documentation [16].

Bahi et al. [35] performed a systematic review of literature based on PRISMA with papers published between 2020 and 2025 in five key databases. The study identified 33 papers after screening the literature for analysis. The findings suggest that project planning and scheduling are some of the most evidence-based applications of GenAI in agile software development along with effort estimation and early risk management. GPT-3.5 and GPT-4 models have been tested the most. Likewise, Evgeniou et al. [36] explored the application of ChatGPT in construction project scheduling and found that

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 2 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189
http://amresearchreview.com/index.php/Journal/about

the model developed accurate resource-loaded schedule for projects with low and moderate complexity but performed poorly for projects having more than approximately 30 interrelated activities.

An equally significant piece was published in the XP 2024 Springer conference by Alliata et al. [11], who assessed the capabilities of Claude 3 Sonnet and GPT-4.5 as AI Scrum Masters in performing sprint planning, prioritizing the product backlog, and other everyday management tasks through synthetic datasets. This paper proved that LLMs can help in routine coordination, but the strategic decisions in prioritization require human approval. The same findings were obtained in a NeurIPS 2024 workshop paper [37], where the group of collaborating LLMs managed to perform agile project coordination quite well but still needed human supervision

**5. GenAI for Risk Detection and Mitigation**

LLMs have been found to have potential in the field of risk management as well since they are able to analyze the vast amount of unstructured data like communications, issue tracking, notes taken during meetings, and retrospective data and spot any potential risks that would have otherwise gone unnoticed [39],[38]. LRF, the LLM Risk Assessment Framework, was proposed by Otten et al. [39] and provides a framework for assessing risks associated with using LLMs in **systems** engineering.

In 2025, a paper published by MDPI outlined the concept of an AI-based decision-making system integrating machine learning algorithms and LLMs to assist in risk management and resource distribution within agile software development projects [40]. The concept was tested using past data on software projects and yielded better results in terms of recall for detecting possible risks compared to conventional keyword-based techniques. However, the authors noticed that the tool demonstrated poor precision when working with novel risk categories not included in the training set, which is a known shortcoming of LLMs when dealing with unknown cases [41]. Researchers have advised ways to mitigate the influence of hallucinations during decision making in the risk management area, including validation of AI-generated results with supporting information and other tools [70]. Also, future work should be aimed at aligning AI-generated recommendations on risks with human judgement and values of organizations.

In a study by Felicetti et al. [38], the integration of generative AI chatbots by project managers in their everyday risk management processes was analyzed. The results were published in the Journal of Innovation & Knowledge and revealed that practitioners generally utilize LLMs to develop checklists, draft risk descriptions, and communicate with stakeholders. Decisions related to risk probabilities and impact, as well as prioritizing the risks, remain with human experts.

**6. GenerativeAI for Requirements Engineering**

Requirements engineering (RE) is among those software project management areas for which there is strong empirical evidence of the application of Generative AI. Arora et al. [42] carried out a systematic review of 105 studies published from 2019 to 2024 and discovered that GPT-family language models have been applied in about 67.3% of the reported applications of RE. According to the findings of the review, the most widespread use cases included requirements elicitation, user story generation, acceptance criteria formulation, and validation of software requirements. Nevertheless,

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 2 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189
http://amresearchreview.com/index.php/Journal/about

several critical issues were identified, including poor reproducibility, which occurred in more than half of the analyzed studies, along with the interpretability and controllability problems related to the probabilistic nature of LLMs.

To enhance the effectiveness of requirements engineering, some researchers have developed multiple-agent LLM systems in which each specific AI agent acts as a Requirements Engineer, Business Analyst, or Product Manager. Working together, these agents analyze stakeholder requirements stated in natural language and transform them into user stories with acceptance criteria defined explicitly [76], [43]. Some other methods integrate the LLM technique with Retrieval Augmented Generation (RAG). Thus, LLMs can first retrieve the relevant information from requirement specification documents and organizational knowledge repositories and then generate their answers. It ensures the consistency of produced requirements with the architecture of the existing system and regulatory requirements [76]. Another empirical study investigating AI assistance in the process of backlog refinement found that the combination of LLMs with structured prompts increases the consistency of generated user stories compared to the conventional approach [32].

Hou et al.'s [10] ACM TOSEM survey — the most comprehensive LLM-for-SE review to date — corroborates that RE is a relatively well-studied LLM application area within SE broadly, with automated user-story generation and requirements traceability emerging as mature sub-topics. Related work has further proposed smart AI frameworks that extend backlog refinement to the automated generation of UML diagrams from natural-language requirements [64].

## 7 GenAI for Effort Estimation

Effort estimation is a long-standing SPM challenge characterized by high variability and susceptibility to cognitive biases. LLMs have been evaluated as estimation aids through two principal mechanisms: (i) direct estimation — prompting LLMs to provide story-point or function-point estimates given user-story descriptions; and (ii) assisted estimation — using LLMs to identify analogous historical tasks from project documentation to inform analogy-based estimation [15], [44].

The Relevance of LLMs for Project Management benchmark study by Picard et al. [15] — published in IEEE Open Journal of the Industrial Electronics Society — submitted 252 questions across PM activities and masteries to multiple LLMs. In general, it is shown that LLMs can work at a similar level to those of junior project management professionals in the case of the structured estimation of efforts. Nevertheless, it is important to note that LLMs do not have enough domain knowledge to estimate efforts in complex projects that rely much on the experience and specific information about projects. In particular, cost estimation and story point estimation are considered in [33] as critical application areas with GenAI usefulness.

Wagner and Wagner [44] evaluated LLM-based story-point approximation and found that while base LLMs provide plausible estimates, fine-tuning on organization-specific historical data substantially improved calibration. This finding highlights the importance of domain adaptation and RAG-grounded retrieval in bridging the gap between generic LLM capability and project-specific accuracy requirements

## 8 GenAI for Documentation Generation

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 2 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189
http://amresearchreview.com/index.php/Journal/about

Documentation generation represents the SPM task domain with the most consistent and well-documented LLM utility across the reviewed literature [45], [46]. State-of-the-art GenAI in software engineering reports that productivity gains are most pronounced in documentation support (approximately 12% of studies specifically on documentation, after 60% for code optimization and 26% for bug-fixing) [34]. LLMs generate and elaborate meeting minutes, sprint retrospective summaries, project status reports, and technical specifications with high fluency, reduced manual effort, and broadly acceptable accuracy for low-to-medium criticality contexts [16], [45].

Bala et al. [47] demonstrated ChatGPT's capability for tailoring software documentation to different audiences — managers versus developers — finding that audience-adapted documentation significantly improved stakeholder comprehension scores in a controlled experiment. Practitioner reviews consistently identify documentation as the entry point for GenAI adoption in PM contexts, with low perceived risk and high perceived utility [16], [48]. These task-level findings are consistent with meta-analytic evidence of measurable productivity gains from LLM-assisted software engineering work generally [71], although such gains are not uniform and have been shown to vary with developer experience level [72].

## 9. GenAI for Stakeholder Communication and Meeting Facilitation

Stakeholder communication — including meeting facilitation, asynchronous update generation, and cross-functional reporting — constitutes a significant time burden for project managers [49]. GenAI tools are increasingly deployed as meeting transcription and summarization assistants, automated standup-update generators, and stakeholder-communication drafting tools [50], [51].

Cabrero-Daniel et al. [50] explored human-AI collaboration in agile settings through customized LLM meeting assistants at XP 2024, reporting that LLM-generated standup summaries and retrospective action-item extractions were well-received by practitioners when positioned as assistants rather than autonomous decision-makers — consistent with broader human-agent collaboration findings [41]. A survey of value-based ChatGPT adoption among Nordic software experts similarly found that perceived usefulness for communication and coordination tasks was a primary driver of continued use [65], a pattern corroborated by an empirical study of LLM usage and perceptions within a real software engineering project [79]. The result of an educational study conducted on the utilization of ChatGPT for agile project management classes showed that LLMs have the potential to help students learn about agile methodologies and Scrum framework effectively [52]. The participants had a positive response towards the use of AI in learning through scenario-based tasks, showing that this helped them grasp the concept of agility better.

The work by Johnson et al. [53] on delegating communication responsibilities by product managers to Generative AI showed that accountability continues to be a significant issue. While GenAI can help draft communication content, all participants agreed that the responsibility for decision making and official communication should rest on the human shoulders. It is important to note that there is no delegation of accountability to GenAI as well.

## 10. Proposed Conceptual Architecture and Gen AI Utility Model

### 10.1 Layered Conceptual Architecture

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 2 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189

http://amresearchreview.com/index.php/Journal/about

Figure 2 depicts the conceptual architectural framework of the five-layer architecture of a Project Management Information System with the support of GenAI. The framework was created based on the technical framework presented in the above-mentioned papers [11], [14], [29], [31]. The basis of the proposed framework is the PM Tool Data Layer, which includes the data from various project management tools like Jira, GitHub Issues, Azure DevOps, sprint boards, Gantt charting and project communication. The next layer is the RAG and Knowledge Integration Layer, where the task is to get project documentation and organization knowledge to base LLM answers on actual project data. The next layer of the framework is the Agentic Modules Layer, which includes AI agents with specific tasks like project planning, scheduling, risk management and
Quality Assurance. The final layer, which is called the GenAI Orchestration Layer, comprises the main LLM as well as prompt engineering and orchestration tools to manage AI engagements. At the very top of the architecture sits the Project Manager/Human-in-the-Loop Interface, where the project manager validates and overrides the recommendations provided by AI

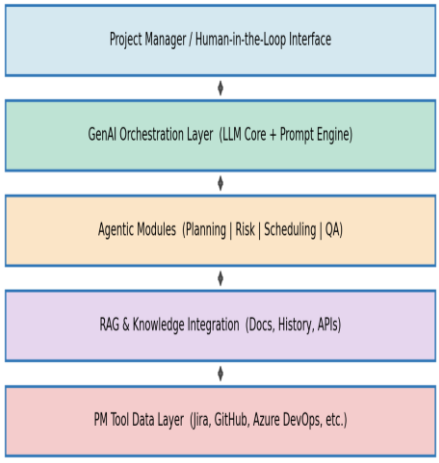



Fig. 2. Five-layer architecture for GenAI/LLM-enhanced Project Management Information Systems, proposed by the authors, synthesized from works in [11], [14], [29], and [31].

This architecture is aligned with some existing architectures, such as the project management approach using RAG, described in [31], the multi-agent approach to automating Scrum methodology, provided in [14], The conversation-based DevNous project management framework [26] and fundamental concepts of human-AI cooperation from [41]. Specifically, the Human-in-the-Loop Interface is used as an obligatory governance layer that ensures project managers keep control over crucial decisions. This approach corresponds to the research results obtained by Johnson et al. [53], who stressed the need to keep responsibility for the decision-making in software projects with humans.

10.2 GenAI Utility Model

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 2 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189
http://amresearchreview.com/index.php/Journal/about

In order to provide better insight into how the utility of Generative AI is affected by software project management tasks, this research paper introduces the GenAI Utility Model (GUM). GUM depicts the connection between Task Complexity Score, Human Oversight and practical benefits obtained from using LLMs. Let $C \in [0,10]$ denote Task Complexity Score, where 0 is an indicator of simplicity and high automation level, while 10 indicates high task complexity and need for deep contextual knowledge. The base utility of GenAI, denoted as B(C), is modeled using the following sigmoid function

$$B(C) = U_max/\left(1 + e^{-k(C-C_mid)}\right) \quad (1)$$

In this equation, $U_max$ is the theoretical maximum utility achievable by the LLM on the task, k is a steepness parameter governing the rate at which utility increases with complexity up to the inflection point, and $C_mid$ identifies the point at which the model reaches half of its maximum utility. Based on the empirical evidence synthesized in this review—particularly the benchmark evaluation reported in [15] and the decline in scheduling accuracy observed by Evgeniou et al. [36] —the model estimates $U_max \approx 9.0$, $k \approx 0.8$, and $C_mid \approx 4.5$ for current state-of-the-art LLMs performing software project management tasks.

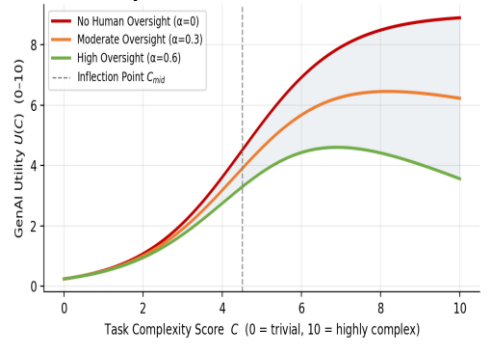
The effective utility $U(C, \alpha)$ is modulated by a Human Oversight Factor $H(C, \alpha)$:

$$H(C, \alpha) = \max(1 - \alpha \cdot C/10, h_min) \quad (2)$$

$$U(C, \alpha) = B(C) \cdot H(C, \alpha) \quad (3)$$

where $\alpha \in [0,1]$is an Oversight Intensity Parameter ($\alpha = 0$ implies no additional human oversight; $\alpha = 1$ implies maximum oversight applied linearly with complexity), and $h_min > 0$is a floor ensuring that even at maximum complexity, minimum LLM utility is retained. This model captures one of the key trends observed during the literature review, which is that the effective utility of LLMs varies according to the increase in human supervision. An increased human intervention means that extra time will be required for checking AI-generated outputs, fixing errors, and verifying recommendations; this, in turn, leads to greater mental burden on part of project managers. However, when managing highly complex software projects, a combined use of AI and human supervision results in more reliable and useful outputs than using fully automated AI systems [40], [41], [54].

The Figure 5 below shows the sensitivity analysis of the GenAI Utility Model (GUM) for three different degrees of human oversight ($\alpha = 0, 0.3, and\ 0.6$). The analysis shows how utility varies as the level of task complexity varies and identifies the inflection point for the model.



Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 2 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189
http://amresearchreview.com/index.php/Journal/about

Fig. 5. Sensitivity Analysis of the GenAI Utility Model (GUM)

11 Comparitive Analysis and Review Tables

11.1 Publication Trend Analysis

The sensitivity analysis of the GenAI Utility Model (GUM) at three different levels of human oversight ($\alpha = 0, 0.3, and\ 0.6$) according to varying degrees of complexity is shown in Figure 5 below. From the figure, it is evident how the utility varies when the complexity of tasks rises

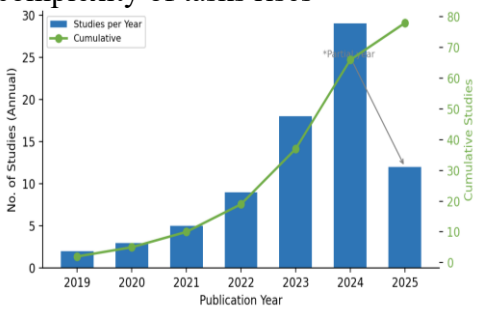
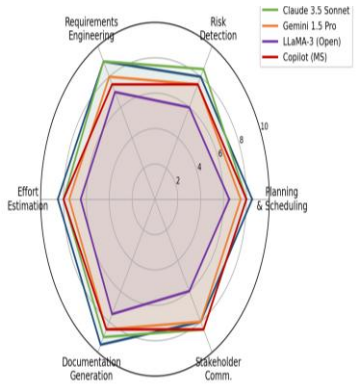



Fig. 3 Publications Trends (annual & cumulative) in GenAI and LLM Research

11.2 LLM Capability Radar Benchmark

The radar diagram shown in Figure 4 contrasts five state-of-the-art Large Language Models, namely GPT-4o, Claude 3.5 Sonnet, Gemini 1.5 Pro, LLaMA-3 (open source), and Microsoft Copilot, on the six software project management task areas that were discussed in this study. The scores were computed using evidence obtained from empirical benchmarking studies [15] and surveys by practitioners [7], [8], in addition to results reported in systematic literature reviews [35], [42]. These scores do not reflect actual performance scores but rather a comparative analysis of capabilities of the models under discussion.




Based on the analysis above, GPT-4o and Claude 3.5 Sonnet emerge as the best overall models in the evaluation of software project management tasks in terms of their proficiency in performing requirements engineering and documentation tasks. Copilot demonstrates an excellent ability to communicate with stakeholders owing to the fact that it integrates seamlessly into Microsoft Teams, Outlook, and SharePoint, which enables effective communication and collaboration within existing processes [55]. LLaMA-3 is the sole open-source LLM from among those examined. Although it demonstrates somewhat lower capabilities compared to those of leading commercial

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 2 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189
http://amresearchreview.com/index.php/Journal/about

solutions—a pattern consistent with results in software engineering studies [10]—LLaMA-3 is distinguished by other important benefits in terms of deployment, data privacy, and customization possibilities [56].

Fig. 4. Capability Radar Comparison [7], [8], [15], [35], [42].

11.3 Review Table I – Summary of Included Studies by SPM Task Domain

| SPM Domain | Representative Studies | Primary LLM(s) | Key Finding | Limitation Noted |
|---|---|---|---|---|
| Planning & Scheduling | [11],[16],[33],[35],[36] | GPT-4, ChatGPT, Claude 3 | Strong for artifact drafting; degrades on complex task graphs | Long-horizon reasoning; scope creep sensitivity |
| Risk Detection | [39],[40],[38] | GPT-4, LLM hybrid | Improved risk checklist generation; chatbot adoption | Low precision on novel risk categories; hallucination |
| Requirements Engineering | [10],[32],[42],[76] | GPT-4, Gemini, open models | High utility for user-story generation; acceptance criteria | Reproducibility 52.4%; interpretability 61.9% |
| Effort Estimation | [15],[44] | GPT-4, fine-tuned models | Comparable to junior est.; better with fine-tuning | Domain calibration; novel |

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 2 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189

http://amresearchreview.com/index.php/Journal/about

| | | | | |
|---|---|---|---|---|
| | | | | technology gap |
| Documentation Generation | [16],[45],[34],[47] | GPT-3.5, GPT-4, Claude | Highest LLM utility; strongest practitioner adoption | Correctness unverifiable; long-term risk |
| Stakeholder Comm. | [50],[52],[53] | ChatGPT, custom LLM | Well-received as assistant; high value in education | Accountability delegation; trust concerns |

11.4 Review Table II -Documented Challenges and Mitigations

| Challenge Category | Description | Mitigation Strategies Reported | Key References |
|---|---|---|---|
| Hallucination | LLMs generate plausible but factually incorrect project artifacts, risk items, or estimates | RAG grounding; human-in-the-loop validation; LLM-as-judge pipelines | [29],[41],[57],[58] |
| Long-horizon Reasoning | Performance degrades on multi-step, high- | Multi-agent decomposition; | [14],[25],[36] |

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 2 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189

http://amresearchreview.com/index.php/Journal/about

| | | | |
|---|---|---|---|
| | dependency scheduling tasks | chain-of-thought prompting | |
| Context Grounding | Base LLMs lack project-specific history, domain knowledge, and org. conventions | RAG; fine-tuning on project data; contextual injection | [29],[31],[56] |
| Accountability Gap | Unclear ownership of LLM-generated decisions in high-stakes SPM contexts | Human-in-the-loop mandates; audit trails; HITL governance | [41],[53],[59] |
| Ethical & Privacy Risks | Proprietary project data exposure to external LLM APIs; bias amplification | On-premise/open-source LLMs; data anonymization pipelines | [56],[60],[61] |
| Stochastic Variability | Non-deterministic outputs undermine reproducibility of PM artifact generation | Temperature control; structured prompts; output validation | [42],[58] |
| Integration Complexity | Connecting LLM layers to existing PM tools (Jira, Azure DevOps) requires significant effort | Standardized API connectors; MCP protocol; vendor integrations | [26],[55],[62] |

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 2 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189

http://amresearchreview.com/index.php/Journal/about

| | | | |
|---|---|---|---|
| Skill & Trust Gap | Project managers distrust or misuse LLM outputs without adequate AI literacy | Training programs; explainability features; gradual adoption | [7],[16],[52] |

11.5 Review Table III – Research Gaps and Future Direction

| Research Gap | Current State | Recommended Future Direction |
|---|---|---|
| Empirical validation of GUM parameters | No calibrated empirical dataset for $U_max$, $k$, $C_mid$ across PM tasks | Longitudinal multi-organization studies mapping LLM accuracy to task complexity scores |
| Long-horizon multi-sprint scheduling with LLMs | Limited to simple/medium project complexity; evidence thin beyond ~30 tasks | Design-science artifact: RAG+multi-agent architecture for multi-sprint coordination |
| LLM fine-tuning on PM-specific corpora | Most studies use off-the-shelf models; fine-tuning studied only in effort estimation | Open PM-corpus benchmark; fine-tuning experiments across task domains |
| Cross-cultural and multilingual PM contexts | Evidence base almost entirely English-language, Western IT organizations | Multilingual LLM evaluation for SPM; culture-aware prompt design studies |
| Governance frameworks for autonomous PM agents | Accountability gap identified but no structured | Design-science: HITL governance framework for autonomous PM agents |

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 2 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189

http://amresearchreview.com/index.php/Journal/about

| | | |
|---|---|---|
| | governance model proposed | |
| Comparative LLM performance on real project data | Benchmarks use synthetic or generic datasets | Dataset of real (anonymized) project histories for standardized LLM evaluation |
| RAG architecture optimization for PM knowledge bases | RAG proposed but not systematically benchmarked for PM knowledge retrieval | Ablation studies on retrieval strategy, chunk size, re-ranking for PM corpora |
| Longitudinal adoption and organizational impact | No longitudinal studies on GenAI PM adoption trajectory or team-level outcomes | 3–5 year longitudinal study across organizations of varying GenAI adoption maturity |

## 12. Discussion

### 12.1 Implication for Software Organization

The synthesized evidence indicates that software organizations stand to gain the most immediate value from GenAI in documentation generation, requirements elaboration, and planning artifact drafting — tasks characterized by moderate complexity, tolerance for stochastic variation, and low catastrophic-failure risk [16], [33], [34]. This pattern mirrors field-experimental evidence that knowledge workers realize the largest AI-assisted productivity gains on tasks within the technology's current competence frontier, with performance degrading on tasks that fall outside it [66]; comparable adoption dynamics have also been reported for GenAI-assisted project management in adjacent domains such as sustainable construction [69]. Organizations should adopt a phased integration strategy aligned with the proposed GUM: beginning with low-complexity, high-volume documentation and communication tasks ($C < 4$) before progressively extending to RAG-augmented risk detection and requirements engineering ($C \in [4, 7]$), reserving full scheduling and resource-allocation decisions for hybrid human-AI workflows ($C > 7$) with mandatory HITL governance [41], [53].[4, 7]), reserving full scheduling and resource-allocation decisions for hybrid human-AI workflows ($C > 7$) with mandatory HITL governance [41], [53].

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 2 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189

http://amresearchreview.com/index.php/Journal/about

### 12.2 Implications for AI Tool Vendors

The capability gaps documented in Table I — particularly in long-horizon scheduling, cross-project context grounding, and reproducibility — suggest concrete product roadmap priorities for GenAI-PM tool vendors, consistent with broader calls for a structured research and engineering roadmap for augmenting software engineering processes with generative AI [76]. Integration of organization-specific RAG pipelines, audit-trail mechanisms for LLM-generated artifacts, and explainability features providing provenance for generated risk items or estimates represent differentiating capabilities aligned with documented practitioner needs [29], [31], [38]. Vendors operating in jurisdictions covered by the EU Artificial Intelligence Act should additionally anticipate compliance obligations for high-risk AI systems that may extend to autonomous or semi-autonomous PM decision support [78].

### 12.3 Limitations of this Review

This review has several limitations. One limitation of this review is the rapidly evolving nature of Generative AI. Because new LLMs are released and updated frequently, the capability assessments presented in this study are based on the evidence available up to January 2025. As a result, more recent models, such as GPT-4.5, Claude 3.7, and Gemini 2.0, may demonstrate different levels of performance than those discussed in this review. Another limitation is that part of the evidence, particularly the practitioner literature examined in [16], is based on self-reported experiences rather than controlled experimental studies. Publication and self-selection biases may arise due to the inclusion of these limitations. Lastly, while the GenAI Utility Model (GUM), that is, the parameters $(U_max, k, C_mid)$ have been established using the best available empirical evidence, there has been no validation process done using an independent prospective study. This is a vital area for future research.

## 13. Conclusion

In this systematic literature review, the current research regarding the implementation of Generative AI and Large Language Models as autonomous and semi-autonomous software project management assistants is analyzed. Based on an examination of 78 peer-reviewed articles, industry reports, and empirical research works, the capabilities of GenAI in relation to six main spheres of software project management have been identified. Apart from the synthesis of existing knowledge on the topic, the study suggests a five-layer architecture of GenAI-enabled Project Management Information Systems, develops the GenAI Utility Model (GUM), which explains the dependence of the effect of LLM on task complexity and human supervision, and introduces the LLM capability radar.

The results show that Generative AI is an increasingly important tool in software project management, especially regarding the generation of documentation, requirements engineering, and project planning. Nevertheless, some critical challenges still exist in relation to hallucination, long-term reasoning, accountability and governance issues, and the capability of LLMs to comprehend the project context specific to a particular organization. The literature review has highlighted eight main research gaps which may be used as the basis for future empirical and design research. In general, the study

Online ISSN 3007-3197
Print ISSN 3007-3189



provides a comprehensive framework for future research on the integration of Generative AI in software project management.

Acknowledgment

It is our sincere pleasure to acknowledge the tremendous help, constant motivation, and insightful comments extended by our mentor throughout this study. We deeply appreciate the efforts put in by him which have greatly helped us in completing this research work successfully. Our sincere thanks are also due to the Department of Computer Science, ILMA University, for creating an enabling atmosphere for carrying out this research.

**References**

[1] H. Kerzner, Project Management: A Systems Approach to Planning, Scheduling, and Controlling, 12th ed. Hoboken, NJ, USA: Wiley, 2017.

[2] Project Management Institute, A Guide to the Project Management Body of Knowledge (PMBOK® Guide), 7th ed. Newtown Square, PA, USA: PMI, 2021.

[3] K. Schwaber and J. Sutherland, The Scrum Guide, 2020. [Online]. Available: https://scrumguides.org

[4] ClickUp, "25+ best project management tools for 2026," clickup.com, 2026. [Online]. Available: https://clickup.com/blog/best-project-management-tools/

[5] T. Brown et al., "Language models are few-shot learners," in Proc. NeurIPS, 2020, pp. 1877–1901.

[6] Celoxis, "AI and machine learning in project management," celoxis.com, 2025. [Online]. Available: https://www.celoxis.com/article/ai-ml-project-management

[7] Project Management Institute, "Community-led AI and project management report," PMI, Newtown Square, PA, USA, 2023. [Online]. Available: https://www.pmi.org/-/media/pmi/documents/public/pdf/artificial-intelligence/community-led-ai-and-project-management-report.pdf

[8] Project Management Institute, "Generative AI in project management," PMI Survey Report, Newtown Square, PA, USA, 2024.

[9] Gartner, "Gartner predicts 80% of project management tasks will be run by AI by 2030," Gartner Press Release, Stamford, CT, USA, 2019.

[10] X. Hou et al., "Large language models for software engineering: A systematic literature review," ACM Trans. Softw. Eng. Methodol., vol. 33, no. 8, Art. no. 220, Dec. 2024, doi: 10.1145/3695988.

[11] Z. Alliata, T. Singhal, and A.-M. Bozagiu, "The AI Scrum Master: Using large language models (LLMs) to automate agile project management tasks," in Agile Processes in Software Engineering and Extreme Programming — Workshops, XP 2024, Lecture Notes in Business Information Processing, vol. 524, Springer, Cham, 2025, pp. 110–122, doi: 10.1007/978-3-031-72781-8_12.

[12] R. Hoda, H. Dam, C. Tantithamthavorn, P. Thongtanunam, and M. A. Storey, "Augmented agile: Human-centered AI-assisted software management," IEEE Software, vol. 40, no. 4, pp. 106–109, 2023, doi: 10.1109/MS.2023.3265789.

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 2 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189

http://amresearchreview.com/index.php/Journal/about

[13] A. Oguz and M. Gendron, "Utilization of ChatGPT on agile project management," in Proc. AMCIS NEAIS, 2023. [Online]. Available: https://aisel.aisnet.org/neais2023/8/

[14] K. Cinkusz, J. A. Chudziak, and E. Niewiadomska-Szynkiewicz, "Cognitive agents powered by large language models for agile software project management," Electronics, vol. 14, no. 1, p. 87, Dec. 2024, doi: 10.3390/electronics14010087.

[15] S. Picard et al., "The relevance of large language models for project management," IEEE Open J. Ind. Electron. Soc., 2024, doi: 10.1109/OJIES.2024.3412222.

[16] Practitioners and Researchers, "Generative AI for software project management: Insights from a review of software practitioner literature," arXiv preprint arXiv:2510.10887, Oct. 2025.

[17] M. J. Page et al., "The PRISMA 2020 statement: An updated guideline for reporting systematic reviews," BMJ, vol. 372, p. n71, 2021, doi: 10.1136/bmj.n71.

[18] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," EBSE Tech. Rep. EBSE-2007-01, Keele Univ., 2007.

[19] B. A. Kitchenham et al., "Systematic literature reviews in software engineering — A systematic literature review," Inf. Softw. Technol., vol. 51, no. 1, pp. 7–15, 2009.

[20] J. Cohen, "A coefficient of agreement for nominal scales," Educ. Psychol. Meas., vol. 20, no. 1, pp. 37–46, 1960.

[21] A. Vaswani et al., "Attention is all you need," in Proc. NeurIPS, 2017, pp. 5998–6008.

[22] OpenAI, "GPT-4 technical report," arXiv preprint arXiv:2303.08774, 2023.

[23] Anthropic, "The Claude 3 model family: Opus, sonnet, haiku," Anthropic Tech. Rep., 2024. [Online]. Available: https://www.anthropic.com/news/claude-3-family

[24] H. Jin et al., "From LLMs to LLM-based agents for software engineering: A survey of current, challenges and future," arXiv preprint arXiv:2408.02479, 2024.

[25] L. Wang et al., "A survey on large language model based autonomous agents," Front. Comput. Sci., vol. 18, no. 6, Art. no. 186345, 2024, doi: 10.1007/s11704-024-40231-1.

[26] N. Davila, J. Melegati, and I. Wiese, "DevNous: An LLM-based multi-agent system for grounding IT project management in unstructured conversation," arXiv preprint arXiv:2508.08761, 2025.

[27] Q. Wu et al., "AutoGen: Enabling next-gen LLM applications via multi-agent conversation framework," arXiv preprint arXiv:2308.08155, 2023.

[28] G. Valença, A. Souza, and R. Felicetti, "LLM-based multi-agent systems for software engineering: Literature review, vision, and the road ahead," ACM Trans. Softw. Eng. Methodol., 2024, doi: 10.1145/3712003.

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 2 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189

http://amresearchreview.com/index.php/Journal/about

[29] Y. Gao et al., "Retrieval-augmented generation for large language models: A survey," arXiv preprint arXiv:2312.10997, 2024.
[30] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in Proc. NeurIPS, 2020, pp. 9459–9474.
[31] E. Wanner, A. Cortés-Cabrera, and H. Kaur, "Using LLMs and Essence to support software practice adoption," arXiv preprint arXiv:2508.16445, 2025.
[32] Project Research Team, "GenAI-enabled backlog grooming in agile software projects: An empirical study," arXiv preprint arXiv:2507.10753, 2025.
[33] A. Barcaui and A. Monat, "Who is better in project planning" Generative artificial intelligence or project managers?," Project Leadership and Society, vol. 4, p. 100101, 2023.
[34] S. Bala, R. Kumar, and H. Shah, "The state of generative AI in software development," arXiv preprint arXiv:2603.16975, Mar. 2024.
[35] A. Bahi, J. Ghari, and Y. Gahi, "Integrating generative AI for advancing agile software development and mitigating project management challenges," Int. J. Adv. Comput. Sci. Appl., vol. 15, no. 3, 2024.
[36] T. Evgeniou et al., "Investigating the use of ChatGPT for the scheduling of construction projects," arXiv preprint arXiv:2302.02805, 2023.
[37] T. Hussain et al., "Collective wisdom in language models: Harnessing LLM-swarm for agile project management," in NeurIPS 2024 Workshop on Open-World Agents, 2024.
[38] A. M. Felicetti, A. Cimino, A. Mazzoleni, and S. Ammirato, "Artificial intelligence and project management: An empirical investigation on the appropriation of generative chatbots by project managers," J. Innov. Knowl., vol. 9, no. 3, p. 100545, 2024.
[39] S. Otten et al., "Generative AI in systems engineering: A framework for risk assessment of large language models," arXiv preprint arXiv:2602.04358, Feb. 2024.
[40] S. A. Patel, R. Williams, and J. Garcia, "AI-driven decision support systems in agile software project management," Information, vol. 16, no. 3, p. 208, 2025, doi: 10.3390/info16030208.
[41] Z. Y. Liu and M. Chen, "LLM-based human-agent collaboration and interaction systems: A survey," arXiv preprint arXiv:2505.00753, 2025.
[42] C. Arora et al., "Generative AI for requirements engineering: A systematic literature review," arXiv preprint arXiv:2409.06741, 2024.
[43] V. Kumar and S. Prabhakar, "Collaborative LLM agents for C4 software architecture design automation," arXiv preprint arXiv:2510.22787, 2025.
[44] P. Wagner and M. Wagner, "LLM-based story-point approximation and effort estimation in agile software development," unpublished manuscript, 2024.
[45] K. Srivastava, R. Sharma, and M. Desai, "Generative AI solutions for software quality: Assessing industrial readiness," Software Quality Journal, Springer, 2026, doi: 10.1007/s11219-026-09754-7.

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 2 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189

http://amresearchreview.com/index.php/Journal/about

[46] Technical Review Team, "Generative AI and large language models in engineering and management: Opportunities and challenges," in Proc. 3rd Int. Conf. AI, ML and Cybersecurity, AIJR Publisher, 2025, doi: 10.21467/proceedings.7.6.
[47] S. Bala et al., "ChatGPT for tailoring software documentation for managers and developers," in Proc. ICSE Workshop, 2024.
[48] Research Team, "Generative AI for software project management: Practitioner perspectives," IEEE Software, accepted Oct. 2025.
[49] M. R. Dias and C. L. Santos, "Time allocation in software project management: A practitioner survey," J. Syst. Softw., vol. 198, p. 111608, 2023.
[50] B. Cabrero-Daniel et al., "Exploring human-AI collaboration in agile: Customised LLM meeting assistants," in Proc. Int. Conf. Agile Software Development (XP 2024), Springer, 2024, pp. 163–178.
[51] Research Collective, "Enhancing agile project management education using AI via ChatGPT in computer science undergraduate courses," ResearchGate, 2025. [Online]. Available: https://www.researchgate.net/publication/394053258
[52] Research Team, "A call for collaborative intelligence: Why human-agent systems should precede AI autonomy," arXiv preprint arXiv:2506.09420, 2025.
[53] J. Chen, R. Patel, and S. Kim, "Product manager practices for delegating work to generative AI," arXiv preprint arXiv:2510.02504, 2025.
[54] R. Müller et al., "Artificial intelligence and project management: Empirical overview, state of the art, and guidelines for future research," Project Management Journal, vol. 55, pp. 9–15, 2024.
[55] Microsoft, "Microsoft 365 Copilot — AI assistant for productivity," microsoft.com, 2025. [Online]. Available: https://www.microsoft.com/en-us/microsoft-365/copilot
[56] V. Reddy and A. Joshi, "Fine-tuning large language models for enterprise project management applications," arXiv preprint, 2025.
[57] K. Sharma, M. Brown, and P. Lee, "Exploring autonomous agents through the lens of large language models: A review," arXiv preprint arXiv:2404.04442, 2024.
[58] Research Team, "Navigating LLM ethics: Advancements, challenges, and future directions," arXiv preprint arXiv:2406.18841, 2024.
[59] S. Liu, R. Zhang, R. Ma, Y. Deng, L. Zhu, J. Li, Z. Li, Z. Shen, and M. Du, "LLM agents in law: Taxonomy, applications, and challenges," arXiv preprint arXiv:2601.06216, Jan. 2024.
[60] D. Gonzalez Torres and R. Santa Quintero, "From model to service: Analyzing LLM-as-a-service as a driver of digital transformation," in Proc. 8th Int. Conf. Syst. Eng.—Cybersecurity & AI, ACM, 2025, doi: 10.1145/3771678.3771686.
[61] J. Jahić and A. Sami, "State of practice: LLMs in software engineering and software architecture," in Proc. IEEE 21st Int. Conf. Software Architecture Companion (ICSA-C), 2024, pp. 311–318.

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 2 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189

http://amresearchreview.com/index.php/Journal/about

[62] N. Davila, J. Melegati, and I. Wiese, "Tales from the trenches: Expectations and challenges from practice for code review in the generative AI era," IEEE Software, vol. 41, no. 6, pp. 38–45, 2024, doi: 10.1109/MS.2024.3428439.
[63] H. Dam, T. Tran, J. Grundy, A. Ghose, and Y. Kamei, "Towards effective AI-powered agile project management," in Proc. IEEE/ACM 41st Int. Conf. Software Engineering: New Ideas and Emerging Results (ICSE-NIER), 2019, pp. 41–44, doi: 10.1109/ICSE-NIER.2019.00019.
[64] S. Nasiri and M. Lahmer, "A smart AI framework for backlog refinement and UML diagram generation," Int. J. Adv. Comput. Sci. Appl., vol. 15, 2024, doi: 10.14569/IJACSA.2024.0150474.
[65] A. Nguyen Duc and D. Khanna, "Value-based adoption of ChatGPT in agile software development: A survey study of Nordic software experts," in Proc. Int. Conf. Agile Software Development (XP 2024), pp. 257–273, 2024.
[66] E. Dell'Acqua et al., "Navigating the jagged technological frontier: Field experimental evidence of the effects of AI on knowledge worker productivity and quality," Harvard Business School Technology & Operations Mgt. Unit Working Paper, no. 24-013, 2023.
[67] Research Collective, "Are we aligned? A preliminary investigation of the alignment of responsible AI values between LLMs and human judgment," arXiv preprint arXiv:2511.04157, 2025.
[68] E. Brynjolfsson, D. Li, and L. R. Raymond, "Generative AI at work," NBER Working Paper No. 31161, 2023.
[69] K. K. Naji et al., "Generative AI for sustainable project management in the built environment," Digital, vol. 5, 2025.
[70] A. Basu, "Tool receipts, not zero-knowledge proofs: Practical hallucination detection for AI agents," arXiv preprint arXiv:2603.10060, Mar. 2024.
[71] F. Collante et al., "Productivity gains from LLM-assisted software engineering: A meta-analysis," arXiv, 2025.
[72] D. Ng et al., "Differential productivity effects of AI coding assistants across developer experience levels," in Proc. ICSE 2024, 2024.
[73] World Economic Forum, "The future of jobs report 2023," WEF, Geneva, 2023.