Al-Farouk journal of sciences
Volume (2), Issue (3) (JUNE, 2026)
ISSN: 3135-2359

مجلة الفاروق للعلوم
المجلد (2) ملحق العدد 3 يونيو 2026
ردمد: 3135-2359

MEASURING THE IMPACT OF GENERATIVE AI ON SOFTWARE TEAM PRODUCTIVITY AND OUTPUT QUALITY IN AGILE ENVIRONMENTS

[1]ABDALMENAM KHALIF MASAUD ABUSWAH
abd85menam85@gmail.com
[2] ABDARRAHMAN KHALIF ALI ABOUSOWA
a.abuswa@hn.edu.ly
[3] ZIAD OMAR SALEM WAREG
Zwrrag2012@gmail.com
Higher Institute of Science and Technology Nalut - Libya
Received: 29/6/2026 Revised: 20/7/2026 Accepted: 28/7/2026 Publication: 29/8/2026

ABSTRACT

Generative artificial intelligence (GAI) is becoming more incorporated into software engineering functions like code creation, debugging, requirement analysis, testing, and sharing knowledge. This research looks at how GAI affects software teams in terms of productivity and quality of the output in Agile environments. The research design used is quantitative, cross-sectional survey type using a questionnaire prepared for this research. The data used consists of 35 responses, with 34 usable cases in analyzing 30 Likert items. The measuring instrument consists of six concepts: use of GAI, efficiency of the software team, quality of the software output, GAI in Agile, team collaboration, and communication, and overall impact perceived. The descriptive results show positive feelings about the six concepts. The values on the mean for the different concepts varied from 3.54 to 3.78 on a scale of five, with GAI being the concept that received the highest mean ($M = 3.78$, $SD = 0.47$) while productivity was the one that received the lowest ($M = 3.54$, $SD = 0.69$). The instrument has a high level of internal consistency with $\alpha = 0.799$ for the entire scale of 30 items. In terms of specific items, productivity, quality, and team collaboration had acceptable reliability, whereas GAI had low internal consistency and Agile and general have the upper limit of reliability therefore, construct-level findings should be interpreted cautiously.

Pearson correlation analysis showed statistically significant positive associations between overall perceived impact and software output quality ($r = 0.365$, $p = 0.034$) and team collaboration and communication ($r = 0.371$, $p = 0.031$). Productivity was positively associated with overall impact but did not reach the conventional 0.05 significance level ($r = 0.312$, $p = 0.073$). In a multiple regression model, the five dimensions explained 23.5% of the variance in overall perceived impact ($R^2 = 0.235$); however, the overall model was not statistically significant ($F(5, 28) = 1.719$, $p = 0.163$). These findings support a cautious interpretation: respondents generally perceive GAI positively, but the present small sample does not provide strong evidence for broad causal claims.

Keywords: Generative artificial intelligence; software engineering; software productivity; software quality; Agile development; team collaboration; empirical survey; software development.

1. INTRODUCTION

Generative AI has transitioned swiftly from being just an experimental tool to becoming an essential element of software development process. Large-scale language models and AI coding helpers help developers by creating programs, analyzing and fixing existing codes, and helping them in reflecting on programming challenges. The growing interest in these abilities has been confirmed by multiple empirical studies. One of them—a field experiment conducted in 2026 involving three randomized trials with 4,867 programmers—revealed that programmers who used AI helpers improved their productivity significantly in comparison to those who did not use them (Cui et al., 2026).

GAI's impact on software engineering goes beyond the ability to write code faster. Software development is actually a group process in which different parameters such as output quality, productivity, communication, changes in requirements and limitations are all combined. Principles of Agile development are flexibility and readiness for change, delivery, cooperation and feedback. Research done by Diva and Dingsøyr (2008) indicates that Agile studies are empirical and interdisciplinary and involve not only technical processes, but also organizational, human and social processes.

GAI may influence several of these dimensions simultaneously. In the context of software development, developers can use GAI to generate implementation ideas, reduce time spent on routine coding, identify defects, improve test cases, and support understanding of requirements. According to recent studies, AI performance may depend upon how developers combine AI output with their personal expertise and how the proficiency of AI corresponds with the task taken on (Chang & Huang, 2026). At the same time, having positive attitudes does not imply a causal effect of productivity and quality. AI output may contain mistakes and require validation by humans, with differing influence on the developers depending on their background and the specific task. Consequently, empirical studies should distinguish between perceived benefits, statistical associations, and causal effects. The present study adopts this cautious perspective and uses the actual questionnaire responses collected for the research to measure perceptions of GAI use, productivity, quality, Agile work, collaboration, and overall impact.

1.1 Problem Statement

While GAI's use becomes wider in programming, empirical evidence that takes productivity and quality of results into account together must still be sought for and connected to Agile practices of teamwork. Most of the discussions in the public domain are concentrated on how fast individual software developers are, when there are still many issues that a software team needs to solve, e.g., team knowledge coordination, adapting to changing requirements, quality preservation of the code, achieving Sprint goals, and others. This study therefore addresses the practical question of how software professionals perceive the contribution of GAI across these interconnected dimensions.

1.2 Research Objectives

- To measure the level of GAI use among software-development respondents.
- To assess respondents' perceptions of the effect of GAI on software team productivity.
- To assess respondents' perceptions of the effect of GAI on software output quality.
- To examine perceptions of GAI support for Agile development activities.
- To examine perceptions of GAI support for collaboration and communication within software teams.
- To assess the overall perceived impact of GAI on software team performance.
- To examine statistical associations among the measured constructs.

1.3 Research Questions

• What is the reported level of GAI use among the respondents?
• How do respondents perceive the effect of GAI on software team productivity?
• How do respondents perceive the effect of GAI on software output quality?
• How do respondents perceive the role of GAI in Agile development activities?
• How do respondents perceive the effect of GAI on team collaboration and communication?
• What is the overall perceived impact of GAI on software teams?
• What statistical relationships exist between GAI use and the measured outcome dimensions?

2. LITERATURE REVIEW

2.1 Generative AI in Software Engineering

GAI tools are increasingly embedded in software engineering tasks. Their value derives from their ability to transform natural-language instructions and contextual information into code suggestions, explanations, documentation, test cases, and problem-solving alternatives. In practical development environments, these capabilities can reduce the effort required for repetitive activities and provide developers with an interactive source of technical assistance. (Hou et al., 2024; Barke et al., 2023).

The empirical literature increasingly indicates that AI assistance can affect developer productivity, although the magnitude and consistency of the effect vary. Cui et al. (2026) used randomized field experiments across Microsoft, Accenture, and another large organization and found a positive average effect on completed tasks. This evidence is stronger than purely perceptual evidence because it uses experimental variation in access to the AI assistant. However, the present study is intentionally different: it measures professionals' perceptions across multiple dimensions rather than attempting to estimate a causal treatment effect.

Research has also shifted toward understanding how AI reliance, task–technology fit, experience, and knowledge integration shape performance. Chang and Huang (2026) reported that AI reliance can improve information-system development performance and emphasized the importance of task fit, personal experience, and knowledge integration. This supports the view that GAI should be understood as part of a human–technology work system rather than as an autonomous replacement for software engineers.

2.2 Software Productivity and Output Quality

Software productivity is multidimensional. It may include time required to complete tasks, throughput, ability to focus on high-value work, and delivery speed. Output quality similarly extends beyond whether code executes: readability, maintainability, defect reduction, and testing quality are relevant dimensions. This study operationalizes these concepts through five productivity items and five quality items, allowing the respondent to evaluate perceived changes in routine work, task completion, focus, development speed, code quality, errors, readability, maintainability, and testing.

The distinction between productivity and quality is particularly important for GAI. A tool can accelerate code production while simultaneously creating review or verification burdens. Conversely, better explanations, debugging support, and test generation may improve quality while also saving time. The questionnaire therefore avoids treating productivity as a single outcome and instead measures it alongside quality.

2.3 Agile Software Development

Agile development emphasizes iterative delivery, responding flexibly to change, collaboration, and obtaining regular feedback. Dybå and Dingsøyr (2008) synthesized empirical studies on Agile project coordination and indicated that the research in this area covers Agile development adoption, human and social factors, perceptions, and comparative outcomes. In recent studies, the advantages and disadvantages of Agile practices continue to be determined, particularly when organizations expand using Agile methods (Edison et al., 2022).

GAI may fit Agile workflows because many of its potential benefits occur at short task horizons: generating implementation ideas during a Sprint, explaining requirements, resolving technical problems, preparing tests, and helping developers respond to changing requirements. The present questionnaire captures these possibilities through five Agile-specific items concerning responsiveness, Sprint execution, problem solving, Sprint goals, and decision support. (Barke et al., 2023; Bird et al., 2023).

2.4 Open Communication and Wisdom Sharing Software development is a social process demanding knowledge transfer. Team members must exchange technical input, comprehend common issues, monitor their activities, and assist each other based on different levels of professionalism that team members possess. GAI can aid indirectly by sharing information such as explanations, instances, summaries, and other examples for discussion.

2.5 Research Gap

The present study addresses a practical gap by combining six related constructs in one questionnaire: actual GAI use, productivity, output quality, Agile work, collaboration, and overall impact. The instrument therefore provides a compact empirical picture of how respondents perceive GAI across individual task execution and team-level outcomes. The study is also grounded in the actual response dataset rather than hypothetical or simulated observations.

3. CONCEPTUAL FRAMEWORK AND HYPOTHESES

The conceptual framework treats GAI use as the primary technology-use construct and considers productivity, quality, Agile support, and team collaboration as outcome-related dimensions. Overall perceived impact is treated as a summary outcome reflecting respondents' global evaluation of GAI in software development.

Because the dataset is cross-sectional and perception-based, the hypotheses are tested as statistical associations rather than definitive causal effects.

Hypothesis	Statement
H1	Higher reported GAI use is positively associated with software team productivity.
H2	Higher reported GAI use is positively associated with perceived software output quality.
H3	Higher reported GAI use is positively associated with perceived support for Agile development.
H4	Higher reported GAI use is positively associated with perceived team collaboration and communication.
H5	The productivity, quality, Agile, and team-collaboration dimensions are positively associated with overall perceived GAI impact.

4. METHODOLOGY

4.1 Research Design

The study uses a quantitative, cross-sectional survey design. Data were collected through a structured questionnaire and exported to Microsoft Excel. The supplied dataset contains 35 response records collected between 12 and 14 August 2026. No additional responses were generated or imputed for this analysis.

4.2 Instrument

The survey consists of seven demographic/use items and then it presents 30 statements based on the Likert scale. The 30 statements are grouped in six groups consisting of 5 statements each; thus GAI use (GAI1-GAI5), productivity (PROD1-PROD5), quality (QUAL1-QUAL5), Agile support (AGILE1-AGILE5), team collaboration (TEAM1-TEAM5), and overall impact (IMP1-IMP5) constructs were measured. The responses obtain a five-point scale with codes from 1 (Strongly Disagree) to 5 (Strongly Agree).

4.3 Sample and Data Screening

The dataset consists of 35 records in total. In the dataset one record has no gender response, two records lack their age, one record does not contain the information about the education, one record does not contain the data on experience, one record has incomplete job-role information, one record lacks its GAI-experience, and one record has no record of using the GAI..

The Likert-scale section contains one incomplete respondent record, resulting in 34 complete cases for construct-level analysis. Percentages in demographic tables are calculated using the number of valid responses for each variable, while descriptive statistics for the constructs use available valid item responses and the inferential analyses use complete construct scores.

4.4 Data Analysis

The analysis was conducted directly from the supplied Excel dataset using the stated Likert coding. Frequencies and percentages were used for categorical variables. Means and standard deviations were used to summarize Likert items and construct scores. Cronbach's alpha was calculated to assess internal consistency. Pearson correlation coefficients were used to assess associations among construct scores. Finally, a multiple linear regression model was estimated with overall perceived impact as the dependent variable and GAI use, productivity, quality, Agile support, and team collaboration as predictors.

4.5 Ethical and Data-Integrity Considerations

The analysis presented in this document is limited to the supplied questionnaire dataset. No names, contact details, or other direct identifiers were used in the analysis. The results should be interpreted as evidence from the surveyed respondents rather than as population estimates. Because the sample is small and appears to be a convenience response set, statistical generalization should be made cautiously.

5. RESULTS

5.1 Demographic Profile

Gender

Category	n	Valid %
Male	17	50.0%
Female	17	50.0%

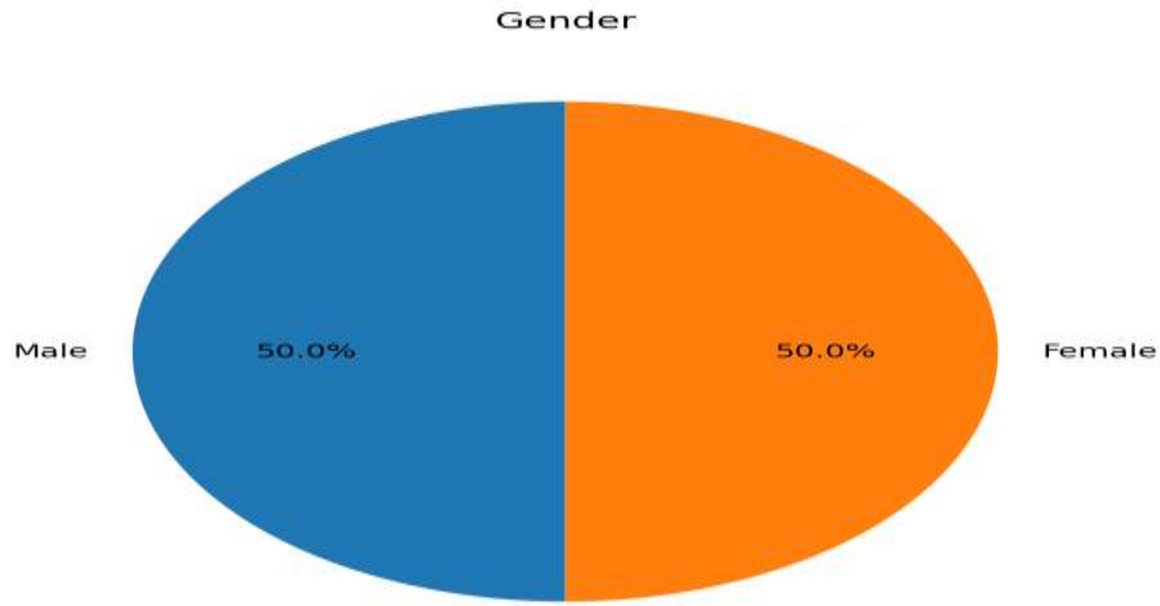

Figure 1. Gender Distribution (n = 34)

Age group

Category	n	Valid %
25–34 years	16	48.5%
35–44 years	14	42.4%
Less than 25 years	3	9.1%

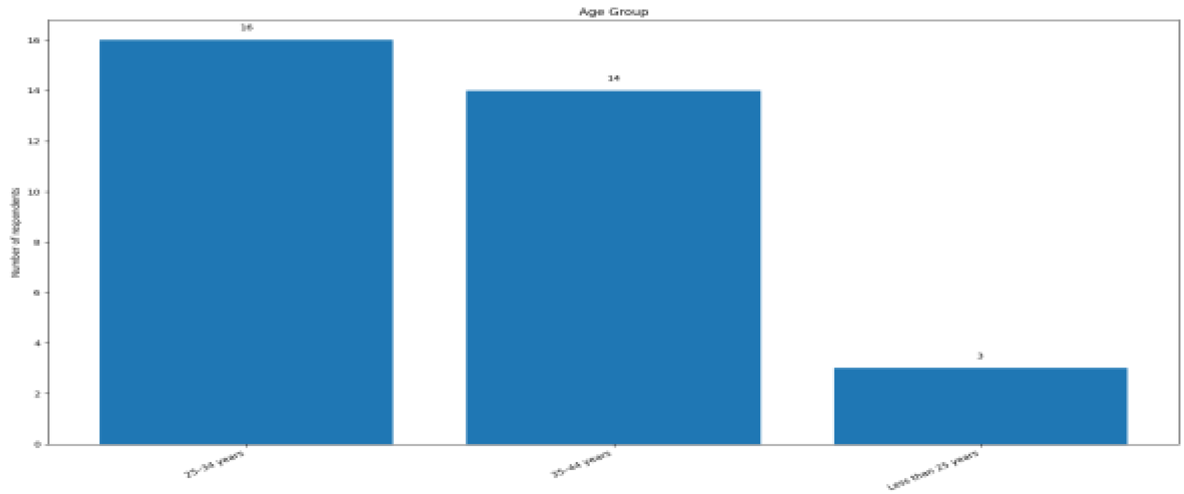

Figure 2. Age Group (n = 33)

Educational qualification

Category	n	Valid %
Master's	11	32.4%
Doctorate	11	32.4%
Bachelor's	9	26.5%
Diploma	2	5.9%
Other	1	2.9%

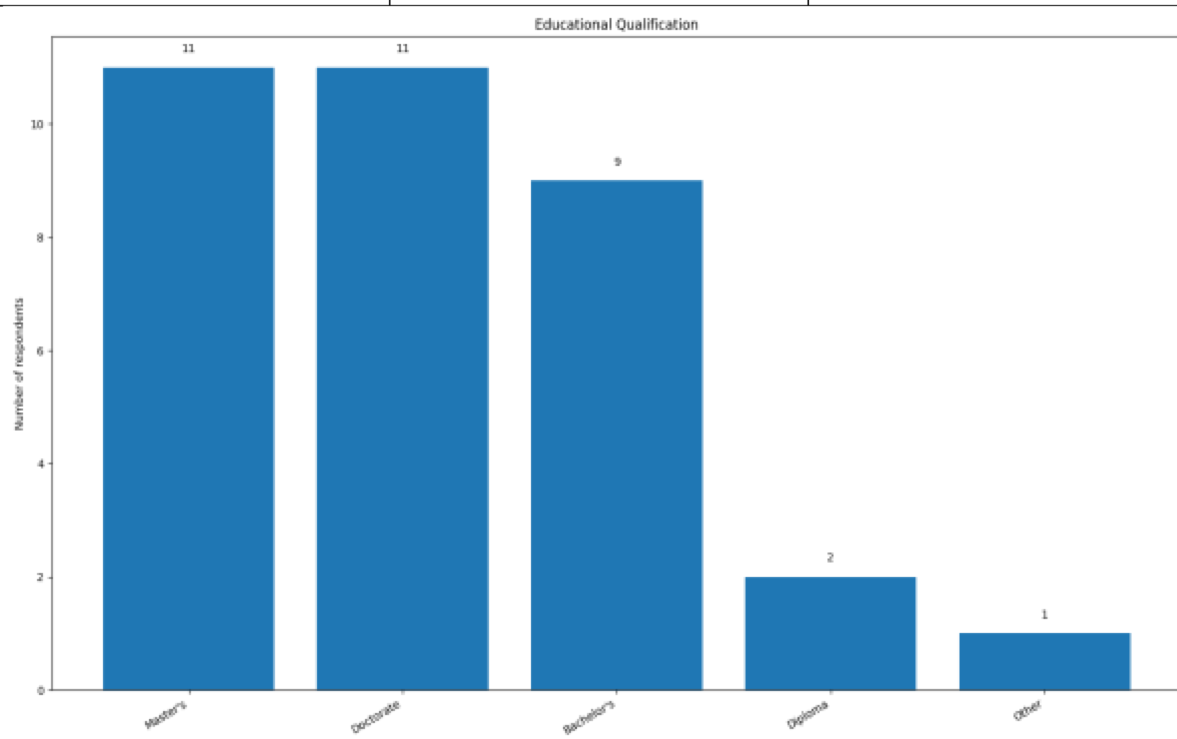

Figure 3. Educational Qualification (n = 34)

Software-development experience

Category	n	Valid %
More than 10 years	11	33.3%
7–10 years	9	27.3%
4–6 years	6	18.2%
Less than 1 year	5	15.2%
1–3 years	2	6.1%

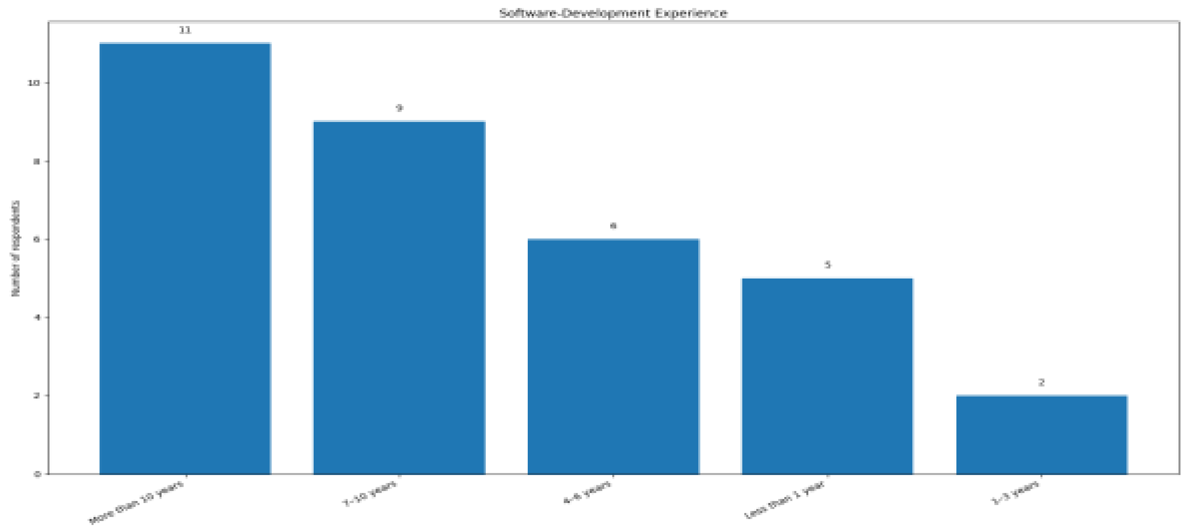

Figure 4. Software-Development Experience (n = 33)

Job role

Category	n	Valid %
Software Engineer	11	32.4%
Software Developer	10	29.4%
Other	7	20.6%
Systems Analyst	6	17.6%

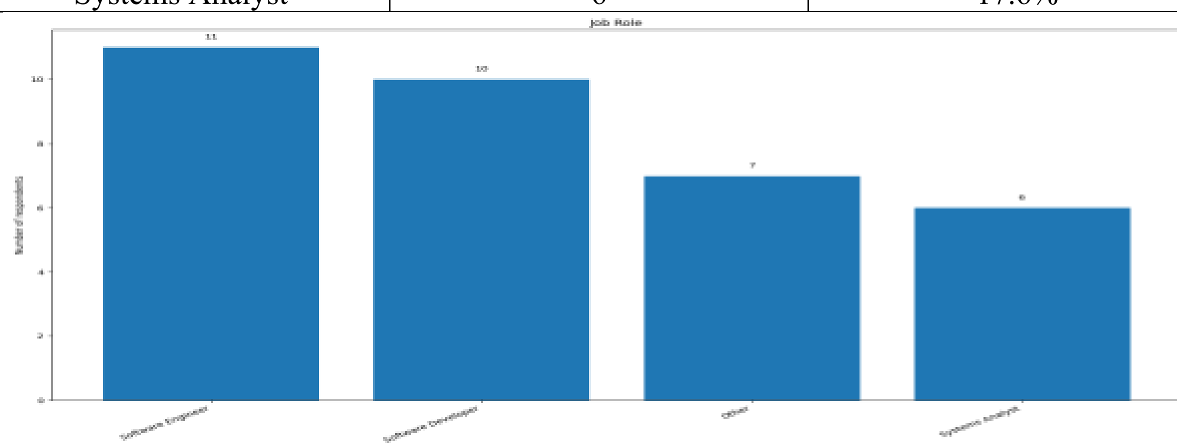

Figure 5. Job Role (n = 34)

GAI experience

Category	n	Valid %
Good	12	35.3%
Moderate	10	29.4%
Very high	8	23.5%
Limited experience	2	5.9%
No experience	2	5.9%

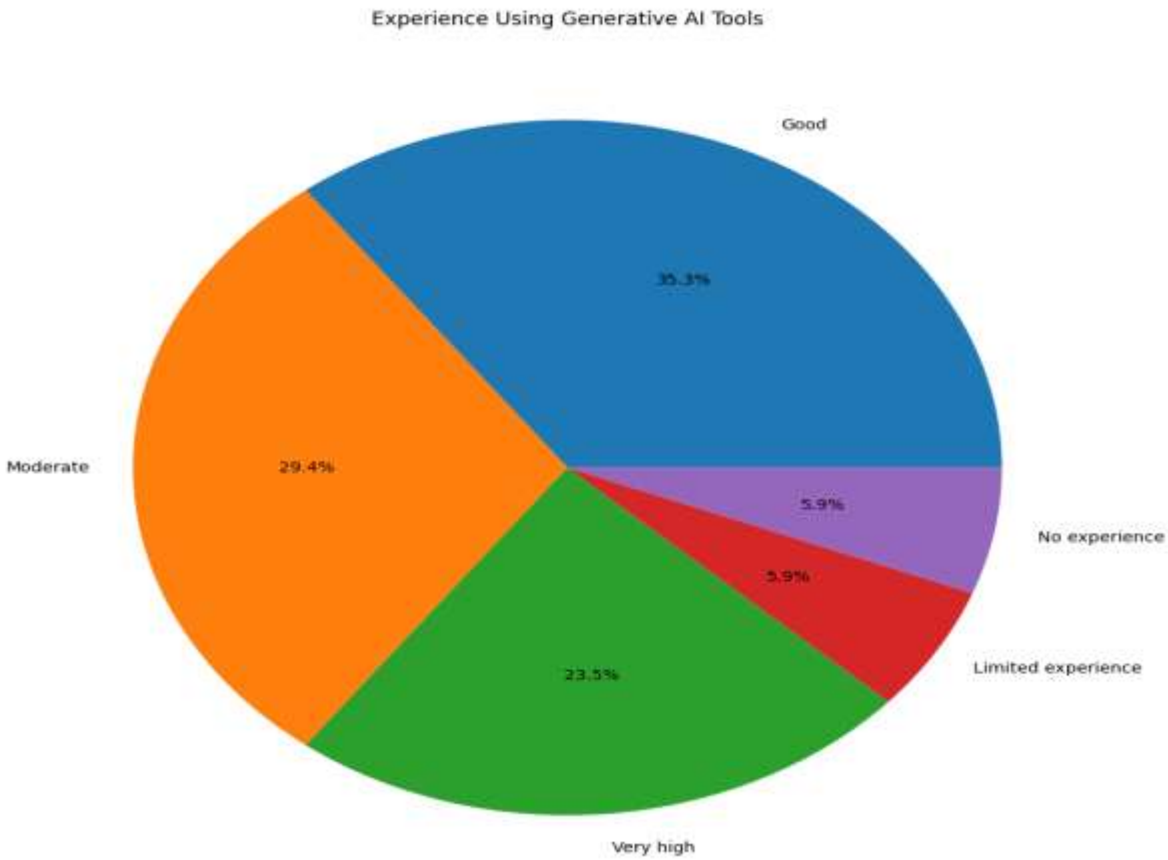

Figure 6. Experience Using Generative AI Tools (n = 34)

GAI use frequency

Category	n	Valid %
Often	17	50.0%
Sometimes	12	35.3%
Rarely	3	8.8%
Always	2	5.9%

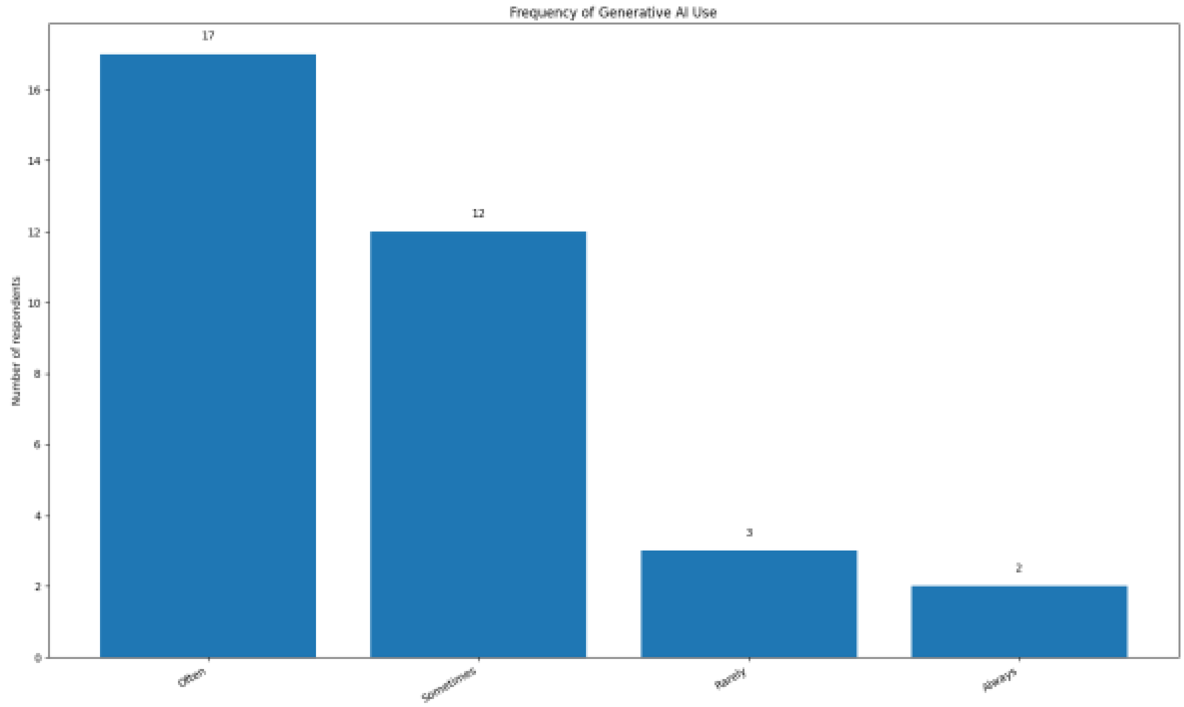

Figure 7. Frequency of Generative AI Use (n = 34)

5.2 Descriptive Statistics of the Main Constructs

Code	Construct	Mean	SD	Min	Max	Cronbach α
GAI	Generative AI Use	3.78	0.47	2.80	5.00	0.383
PROD	Software Team Productivity	3.54	0.69	1.40	5.00	0.721
QUAL	Software Output Quality	3.57	0.74	1.20	5.00	0.739
AGILE	Generative AI in Agile Development	3.55	0.58	1.80	4.60	0.615
TEAM	Team Collaboration and Communication	3.57	0.73	1.20	5.00	0.720
IMPACT	Overall Perceived Impact	3.62	0.58	1.80	5.00	0.632

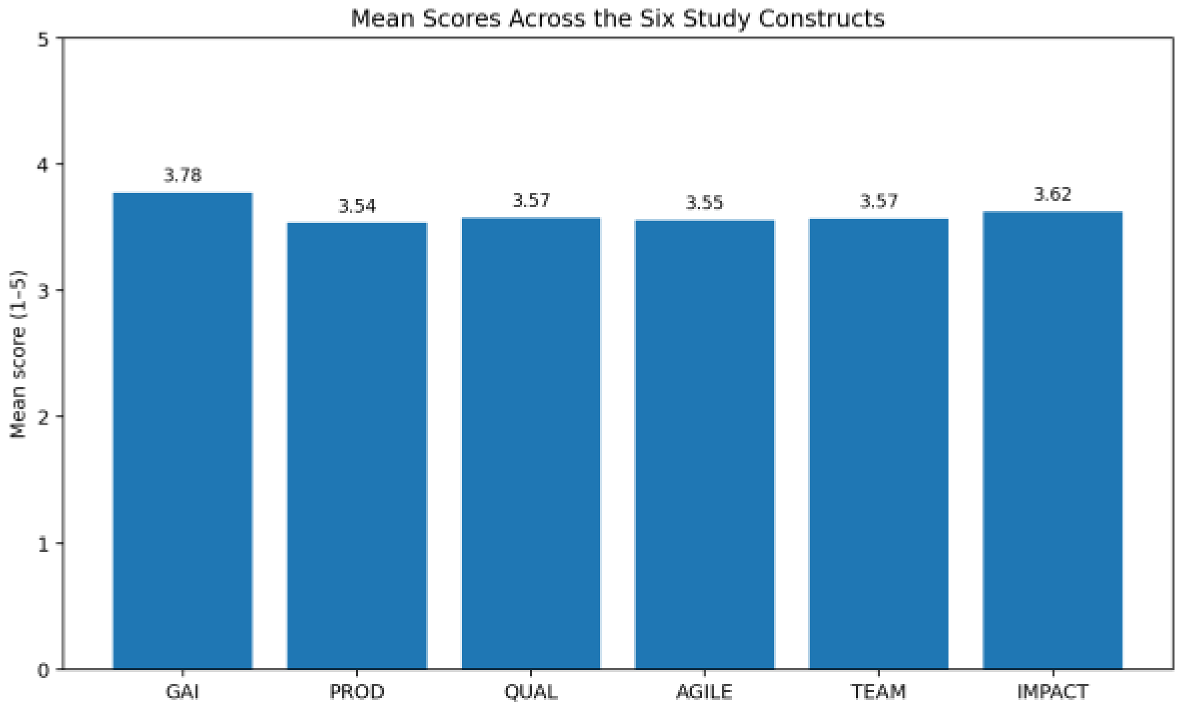

Figure 8. Mean Scores Across the Six Study Constructs

Overall, all six construct means were above the neutral midpoint of 3.00. GAI use had the highest mean score (M = 3.78), followed by overall perceived impact (M = 3.62). Productivity had the lowest construct mean (M = 3.54), although it remained above the neutral point. This pattern suggests generally favorable perceptions while also indicating that perceived productivity gains were less pronounced than perceptions of GAI use itself.

5.3 Item-Level Descriptive Statistics

Item	Statement (English translation)	Valid N	Mean	SD
GAI1	I regularly use generative AI tools while performing software-development tasks.	34	3.94	0.89
GAI2	I use generative AI to assist with writing source code.	34	3.76	0.78
GAI3	I use generative AI to identify and correct software errors.	34	3.74	0.75
GAI4	I use generative AI to generate ideas or solutions to software problems.	34	3.76	0.99
GAI5	I use generative AI to help understand and analyze software requirements.	33	3.67	1.02
PROD1	Generative AI helps reduce the time needed to complete programming tasks.	34	3.53	0.96
PROD2	Generative AI helps the software team complete more tasks within a given period.	34	3.74	0.83
PROD3	Generative AI reduces the time spent on routine software tasks.	34	3.44	0.99
PROD4	Generative AI helps team members focus on more important software tasks.	34	3.50	1.13
PROD5	Generative AI contributes to faster software development within the team.	34	3.47	1.08
QUAL1	Generative AI helps improve the quality of code produced by the team.	34	3.68	1.27
QUAL2	Generative AI helps reduce software errors in final outputs.	33	3.48	0.94
QUAL3	Generative AI helps improve code readability and	34	3.56	1.11

	understandability.			
QUAL4	Generative AI helps improve software maintainability.	34	3.62	0.92
QUAL5	Generative AI helps improve the quality of software testing.	34	3.50	0.93
AGILE1	Generative AI helps our team respond quickly to changes in project requirements.	34	3.74	0.99
AGILE2	Generative AI helps accelerate task execution within development Sprints.	33	3.48	0.80
AGILE3	Generative AI helps team members solve problems during the development cycle.	34	3.56	0.96
AGILE4	Generative AI contributes to the team's ability to meet Sprint goals.	33	3.36	0.82
AGILE5	Generative AI helps support decision-making related to software tasks.	34	3.59	0.99
TEAM1	Generative AI helps team members exchange software knowledge.	34	3.82	0.94
TEAM2	Generative AI helps team members understand software problems better.	34	3.41	1.02
TEAM3	Generative AI facilitates collaboration among software-development team members.	34	3.29	1.06
TEAM4	Generative AI helps less-experienced team members perform software tasks.	33	3.64	0.99
TEAM5	Using generative AI improves knowledge sharing within the software team.	33	3.70	0.95
IMP1	Using generative AI has improved the overall performance of the software team.	34	3.71	0.80
IMP2	Using generative AI has improved the efficiency of the software-development process.	34	3.76	0.96
IMP3	I believe the benefits of generative AI in software development outweigh its associated challenges.	34	3.24	0.92
IMP4	I believe generative AI will become an essential part of software development in the future.	34	3.65	0.88
IMP5	Overall, I believe generative AI has a positive impact on software team productivity and quality.	34	3.76	0.96

Table 4. Item-Level Descriptive Statistics

5.4 Reliability Analysis

The 30-item instrument produced an overall Cronbach's alpha of $\alpha = 0.799$, indicating good overall internal consistency for the complete set of items. However, reliability differs by construct. The productivity ($\alpha = 0.721$), quality ($\alpha = 0.739$), and team collaboration ($\alpha = 0.720$) scales show acceptable internal consistency. The Agile ($\alpha = 0.615$) and overall-impact ($\alpha = 0.632$) scales are marginal, while the GAI-use scale ($\alpha = 0.383$) is low.

The low alpha for GAI use is important. It indicates that the five GAI-use statements do not behave as a highly homogeneous scale in this sample. They may represent different practical uses of GAI (coding, debugging, idea generation, and requirements analysis) rather than one narrowly defined behavior. Consequently, the GAI scale should not be overinterpreted as a single unidimensional measure without further validation using a larger sample.

5.5 Correlation Analysis

Predictor	r with IMPACT	p-value	Significant at .05?
GAI	0.118	0.508	No
PROD	0.312	0.073	No
QUAL	0.365	0.034	Yes
AGILE	-0.056	0.755	No
TEAM	0.371	0.031	Yes

The strongest statistically significant associations with overall perceived impact were quality ($r = 0.365$, $p = 0.034$) and team collaboration ($r = 0.371$, $p = 0.031$). Productivity showed a moderate positive association that narrowly missed the conventional significance threshold ($r = 0.312$, $p = 0.073$). GAI use itself was weakly associated with overall impact ($r = 0.118$, $p = 0.508$), while the Agile-support scale showed a near-zero, nonsignificant association ($r = -0.056$, $p = 0.755$).

5.6 Multiple Regression Analysis

Predictor	B	t	p
GAI	-0.013	-0.052	0.959
PROD	0.037	0.158	0.876
QUAL	0.154	0.812	0.424
AGILE	-0.202	-1.021	0.316
TEAM	0.295	1.741	0.093

The regression model explained 23.5% of the variance in overall perceived impact ($R^2 = 0.235$; adjusted $R^2 = 0.098$). The overall model was not statistically significant, $F(5, 28) = 1.719$, $p = 0.163$. Team collaboration had the largest positive regression coefficient among the predictors ($B = 0.295$, $p = 0.093$), but it did not reach $p < .05$ in the simultaneous model. None of the five predictors was statistically significant at the 0.05 level after controlling for the other dimensions.
Because the sample is small ($n = 34$ complete cases) relative to the number of predictors, the regression should be viewed as exploratory. It should not be used to claim that any construct causes changes in overall impact.

5.7 Hypothesis Testing Summary

Hypothesis	Test	r / result	p / interpretation	Decision
H1	GAI ↔ PROD	0.539	0.001	Supported
H2	GAI ↔ QUAL	0.405	0.017	Supported
H3	GAI ↔ AGILE	0.046	0.798	Not supported
H4	GAI ↔ TEAM	0.089	0.616	Not supported
H5	PROD/QUAL/AGILE/TEAM ↔ IMPACT	See Table 11	Mixed	Partially supported

6. DISCUSSION

The findings provide a generally positive picture of GAI use among the respondents, but they also show why the technology should not be evaluated through a single productivity metric. The highest construct mean was GAI use, indicating that respondents report meaningful engagement with GAI across development tasks. This is consistent with the broader movement toward AI-assisted software engineering documented in recent empirical research.
The productivity mean was positive but lower than the GAI-use mean. This suggests that using GAI and perceiving a productivity benefit are related but not identical experiences. Developers may use AI frequently while still needing time for verification, integration, debugging, and adaptation. The result is therefore compatible with the emerging literature showing positive average productivity effects but variation across developers and tasks (Cui et al., 2026).

Quality was also rated positively and showed a statistically significant relationship with overall perceived impact. This is an important finding because it indicates that respondents' global evaluation of GAI is more closely related to perceived output quality than to raw GAI use frequency. Recent work on GAI for software quality similarly emphasizes the potential of AI-based solutions while noting that technological maturity and adoption challenges remain relevant (Gheventer et al., 2026).

Team collaboration and communication also showed a statistically significant positive association with overall impact. This supports the idea that the value of GAI may extend beyond individual code generation into knowledge sharing and collective problem solving. However, the Agile-specific construct did not show a significant association with overall impact in this sample. This may reflect the fact that Agile effectiveness depends on broader organizational, process, and social factors rather than on one technology alone.

The reliability analysis adds an important qualification. The GAI-use scale had low internal consistency, whereas several outcome scales were acceptable. The five GAI-use items intentionally cover distinct behaviors—regular use, code writing, debugging, ideation, and requirements analysis—so the low alpha may reflect multidimensionality rather than poor questionnaire quality alone. Nevertheless, future studies should validate the construct using a larger sample and potentially separate usage frequency from usage-purpose dimensions.

6.1 Practical Implications

• Organizations should evaluate GAI adoption using both productivity and quality indicators rather than coding speed alone.
• Teams should establish review and verification practices for AI-generated code and tests.
• GAI training should include prompt design, output evaluation, debugging, requirements analysis, and secure use—not only code generation.
• Agile teams should integrate GAI into existing Sprint and collaboration practices rather than treating it as a separate development process.
• Managers should monitor whether AI assistance genuinely reduces routine effort and creates time for higher-value engineering activities.

6.2 Limitations

• The sample is small (35 records; 34 complete construct cases), limiting statistical power.
• The study uses self-reported perceptions rather than objective measures such as repository commits, cycle time, defect density, or Sprint throughput.
• The cross-sectional design does not establish causality.
• The sample is not demonstrated to be probability-based, so generalization to all software teams is not justified.
• Some construct reliability coefficients are marginal or low, particularly the GAI-use scale.
• The study does not separately compare specific GAI tools, programming languages, project types, or Agile frameworks.

7. CONCLUSION

This study examined the perceived impact of generative AI on software team productivity and output quality in Agile environments using the actual questionnaire dataset supplied for the research. The 35 responses indicate generally favorable perceptions of GAI use, productivity, quality, Agile support, team collaboration, and overall impact. The strongest descriptive perception was associated with GAI use, while productivity received the lowest—though still positive—construct mean.

The inferential findings are more nuanced. Quality and team collaboration were significantly associated with overall perceived impact, while productivity showed a positive but no significant relationship. GAI use itself was not significantly associated with overall impact, and the Agile-support dimension showed no significant relationship in this sample. The multivariable regression model explained 23.5% of the variance in overall impact but was not statistically significant.

Accordingly, the evidence supports the conclusion that respondents perceive GAI as a useful and generally positive component of software development, but the present dataset does not justify strong causal claims that GAI independently increases productivity or quality. A larger follow-up study combining survey measures with objective software-engineering metrics would provide a stronger basis for testing these effects.

REFERENCES

Cui, K. Z., Demirer, M., Jaffe, S., Musolff, L., Peng, S., & Salz, T. (2026). The effects of generative AI on high-skilled work: Evidence from three field experiments with software developers. Management Science. https://doi.org/10.1287/mnsc.2025.00535

Chang, T.-S., & Huang, L.-G. (2026). The impact of generative artificial intelligence on software development performance: A study based on the theory of technology dominance. Information & Management, 63(3), 104306. https://doi.org/10.1016/j.im.2026.104306

Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. Information and Software Technology, 50(9–10), 833–859. https://doi.org/10.1016/j.infsof.2008.01.006

Edison, H., Wang, X., & Conboy, K. (2022). Comparing methods for large-scale agile software development: A systematic literature review. IEEE Transactions on Software Engineering, 48(8), 2709–2731. https://doi.org/10.1109/TSE.2021.3069039

Gheventer, A., Gurgel, P. do A., Brito, C. H., Maiani de Mello, R., Rocha, S., Feitosa, R., & Travassos, G. H. (2026). Generative AI solutions for software quality: Assessing industrial readiness. Software Quality Journal, 34, Article 24. https://doi.org/10.1007/s11219-026-09754-7

Barke, S., James, M. B., & Polikarpova, N. (2023). Grounded Copilot: How programmers interact with code-generating models. Proceedings of the ACM on Programming Languages, 7(OOPSLA1), 85–111. https://doi.org/10.1145/3586030

Bird, C., Ford, D., Zimmermann, T., Forsgren, N., Kalliamvakou, E., Lowdermilk, T., & Gazit, I. (2023). Taking flight with Copilot: Early insights and opportunities of AI-powered pair-programming tools. Communications of the ACM, 66(6), 56–62. https://doi.org/10.1145/3589996

Hou, X., Zhao, Y., Liu, Y., Yang, Z., Wang, K., Li, L., Luo, X., Lo, D., Grundy, J., & Wang, H. (2024). Large language models for software engineering: A systematic literature review. ACM Transactions on Software Engineering and Methodology, 33(8), Article 220. https://doi.org/10.1145/3695988

Imai, S. (2022). Is GitHub Copilot a substitute for human pair-programming? An empirical study. In Proceedings of the IEEE/ACM 44th International Conference on Software

Engineering: Companion Proceedings (pp. 319–321). Association for Computing Machinery. https://doi.org/10.1145/3510454.3522684

Kuhail, M. A., Mathew, S. S., Khalil, A., Berengueres, J., & Shah, S. J. H. (2024). "Will I be replaced?" Assessing ChatGPT's effect on software development and programmer perceptions of AI tools. Science of Computer Programming, 235, 103111. https://doi.org/10.1016/j.scico.2024.103111

Liang, J. T., Yang, C., & Myers, B. A. (2024). A large-scale survey on the usability of AI programming assistants: Successes and challenges. In Proceedings of the IEEE/ACM 46th International Conference on Software Engineering (ICSE 2024). Association for Computing Machinery. https://doi.org/10.1145/3597503.3608128

Peng, S., Kalliamvakou, E., Cihon, P., & Demirer, M. (2023). The impact of AI on developer productivity: Evidence from GitHub Copilot. arXiv. https://doi.org/10.48550/arXiv.2302.06590

Sauvola, J., Tarkoma, S., Klemettinen, M., Riekki, J., & Doermann, D. (2024). Future of software development with generative AI. Automated Software Engineering, 31, 26. https://doi.org/10.1007/s10515-024-00426-z