Benefits and Challenges of Generative AI in Software Development: A Survey-Based Study

Ilona Pawełoszek
Czestochowa University of Technology
Faculty of Management
Czestochowa, Poland *ilona.paweloszek@pcz.pl*

Abstract

This paper investigates how generative artificial intelligence (GenAI) tools affect software development from the perspective of 62 software professionals. Using a mixed-methods approach, the study combines survey-based quantitative data with thematic analysis of open-ended responses. Results show that experienced GenAI users report enhanced efficiency and creativity, but also raise concerns about code quality, overreliance, and increased expectations. Younger and less experienced developers feel more job insecurity. Organizational support appears to have limited influence on perceived pressure. The paper offers practical recommendations for the adaptive integration of GenAI tools and highlights directions for future research.

Keywords: generative AI, software development, developer perceptions, productivity, job expectations.

1. Introduction and background

Generative artificial intelligence (GenAI) is reshaping how developers write and maintain code. Tools such as GitHub Copilot, Tabnine, and OpenAI Codex assist with code generation, testing, and refactoring, and are increasingly seen as co-creators in the development process [1], [10]. According to McKinsey & Company [9], GenAI can reduce software development costs by up to 45%, and productivity gains of 30–50% have been reported in field studies [5].

Despite these benefits, the literature highlights persistent limitations, such as poor contextual understanding, suboptimal code quality, and risks associated with overreliance on AI tools [2], [10]. Researchers also point to a lack of evidence on how GenAI influences collaboration, decision-making, and developers' changing roles [3,4].

Most existing studies concentrate on the implementation stage of software development, where GenAI demonstrates the most impact [1], [5]. Stages such as requirements analysis or system design remain less explored due to the context-sensitive nature of these tasks [2], [4]. Systematic reviews have begun to organize this growing body of work [7], but human and organizational factors remain under-researched.

In summary, while GenAI promises transformative changes in software engineering, its actual impact depends on the usage context, developer experience, and organizational culture. This article addresses existing research gaps by analyzing how software professionals perceive and apply GenAI tools in their daily work. We examine not only perceived benefits and limitations but also job-related concerns and organizational responses.

2. Research questions and methodology

The study aimed to investigate the impact of generative AI tools on software development practices from the perspective of developers. Five specific research questions were addressed: how developers' profiles relate to perceived usefulness of GenAI (RQ1); which aspects of their work are most supported (RQ2); whether developers express job insecurity

due to GenAI (RQ3); the role of organizational support in shaping perceptions (RQ4); and which challenges users most frequently report (RQ5).

A structured online questionnaire was used, based on the CAWI method (Computer-Assisted Web Interviewing), and made available in both Polish and English. Participants were recruited via targeted invitations shared in social media groups for programmers, LinkedIn communities, and mailing lists. The sampling approach combined convenience and snowball sampling: initial participants were encouraged to share the survey with professional contacts. Due to the exploratory nature of the study and the niche topic, the recruitment period lasted from January to March 2025 and yielded 62 valid responses.

The questionnaire comprised both closed-ended and open-ended questions. Likert scales were used to assess agreement with various statements about GenAI (1 = strongly disagree, 5 = strongly agree), while ordinal scales captured years of experience and usage frequency. The internal consistency of the five items measuring perceived usefulness was tested using Cronbach's alpha, which showed acceptable reliability ($\alpha = 0.77$). The four sections of the survey included: (1) demographics, (2) declared experience with GenAI tools, (3) usefulness and limitations of GenAI across software development stages, and (4) open questions about perceptions, problems, and suggestions. The response scales were selected to balance granularity with respondent clarity. A 5-point Likert scale was chosen for attitudinal questions to capture direction and intensity of opinions while ensuring ease of use. For experience with GenAI tools, predefined categories (e.g., "occasional use," "regular use in work," "intensive use") were designed to reflect common usage patterns observed in prior studies and developer forums.

Quantitative analysis involved descriptive statistics and nonparametric tests (Kruskal-Wallis and post-hoc Dunn's test with Bonferroni correction). Qualitative responses were analyzed thematically using inductive coding. Analytical procedures were supported by Python scripts (pandas, numpy, regex) and basic visualizations. Although the dataset provided diverse insights, the sample was not representative of the global developer population. Possible limitations include selection bias, language-related interpretation differences, and the self-reported nature of the responses. These constraints were considered when interpreting results and generalizing findings.

3. Research results

3.1. Respondent profile

The final sample included 62 software professionals of varying age, experience, and roles. The largest age group was 45–54 (n = 19), followed by 35–44 (n = 16) and 25–34 (n = 15). Only four respondents were aged 55 or older. Figure 1 illustrates the age distribution and GenAI usage level within the sample.

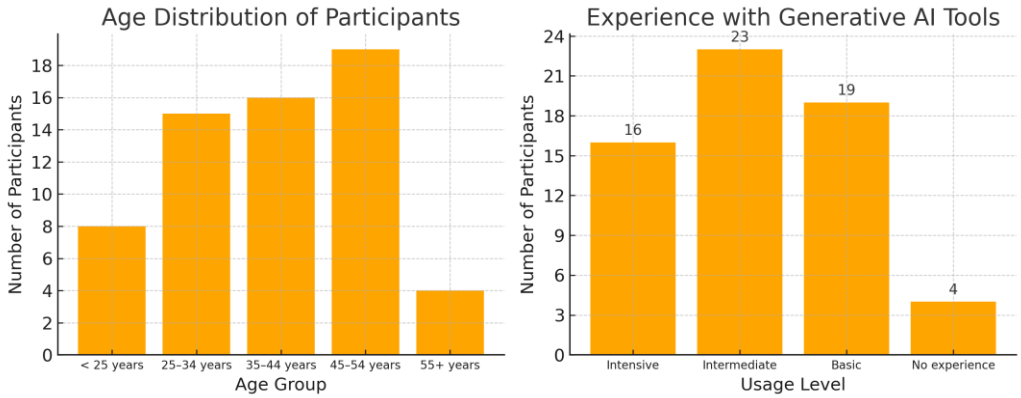

Fig. 1. Age distribution and GenAI usage level among participants

Participants held diverse job positions, with the majority working as senior or lead developers. Junior developers, freelancers, and individuals with entrepreneurial or academic roles were also represented. Most had between 4 and 15 years of IT experience,

and over a quarter reported more than 15 years of professional background.

In terms of GenAI usage, the largest group consisted of intermediate users (n = 23), followed by occasional (basic) users (n = 19), and advanced users (n = 16) engaged in intensive, multi-project use. Only four participants had no prior experience.

3.2. Usefulness of generative AI tools in relation to experience

To address RQ2, participants rated five statements about GenAI's impact on their work using a 5-point Likert scale. These statements assessed support in routine tasks, technical problem-solving, code quality, creative enablement, and reduction in the need for in-depth technical knowledge. The highest ratings were given to GenAI's usefulness in solving technical problems, simplifying routine tasks, and enhancing creative work. Participants were more skeptical about its potential to reduce the need for technical knowledge. Kruskal–Wallis tests revealed significant differences in responses depending on GenAI usage level (Table 1).

Table 1. Kruskal–Wallis test results for differences in perceived usefulness by AI usage level

Statement	H	p-value	Interpretation
Makes it easier to perform routine tasks	15.58	0.0014	Highly significant differences
Reduces need for in-depth technical knowledge	1.71	0.6348	No significant differences
Improves code quality	9.54	0.0229	Significant differences
Helps solve technical problems faster	18.28	0.0004	Highly significant differences
Enables focus on creative aspects	14.50	0.0023	Significant differences

Post-hoc tests confirmed that advanced users rated GenAI tools significantly more favorably than basic users and non-users, particularly in routine, technical, and creative tasks. No notable differences were found for intermediate users. Figure 2 shows group-level comparisons for the four statements with significant effects.

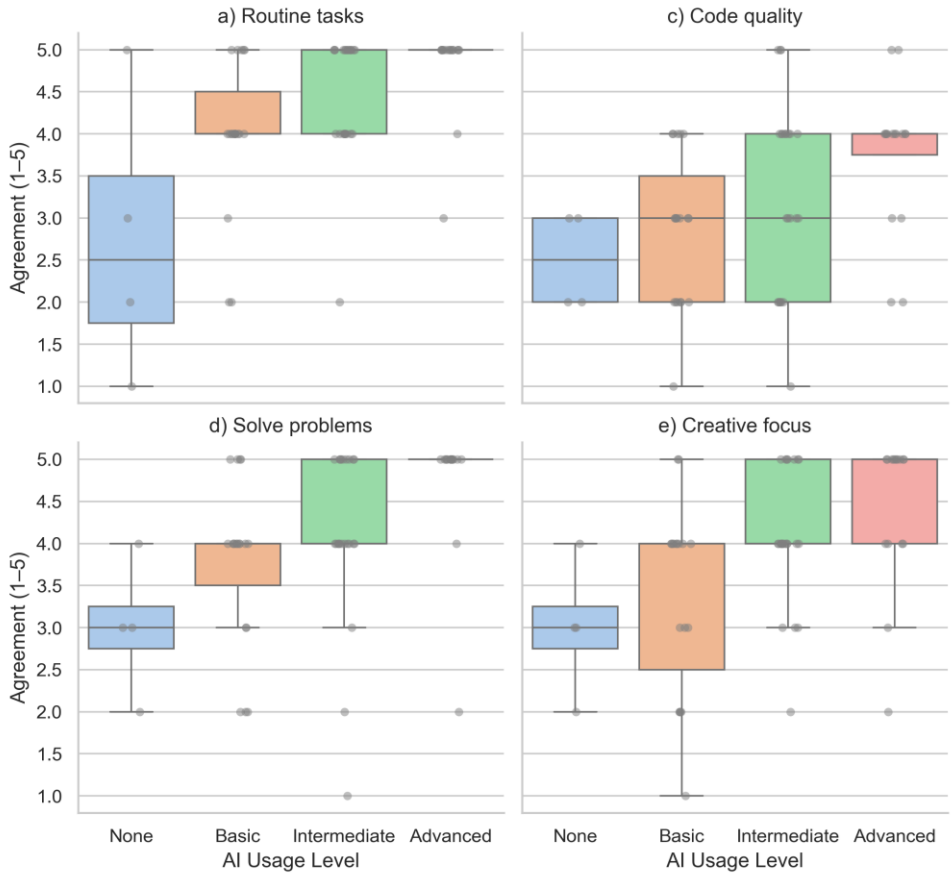

Fig. 2. Perceived usefulness of generative AI tools by usage level: (a) Performing routine programming tasks, (b) Improving code quality, (c) Solving technical problems faster, (d) Focusing on creative work

As illustrated in Figure 2, advanced users consistently rated GenAI tools more positively across all four dimensions. The largest visual gaps are evident in technical problem-solving (c) and creative focus (d), where advanced users' scores are notably higher than those of basic users and non-users. Routine task support (a) also shows a clear gradient by usage level, while opinions on code quality (b) appear more varied but still favor advanced users. These patterns indicate that developers' profiles, particularly their experience with GenAI tools, are strongly associated with perceived usefulness, thereby addressing Research Question 1.

3.3. Developer tasks most supported by generative AI

To address Research Question 2 – *Which aspects of developers' work are most supported by generative AI?* – participants were asked to assess the usefulness of generative AI tools across five key stages of the software development process: planning, system design, implementation, testing, and maintenance. Responses were recorded using a 5-point Likert scale and analyzed based on average scores for each stage.

The results indicate that implementation is the phase where generative AI tools are perceived as most helpful, followed by testing and system design. In contrast, the planning and maintenance stages received the lowest ratings for usefulness. Figure 3 visualizes these findings in the form of a radar chart, highlighting the distribution of perceived usefulness across the development lifecycle.

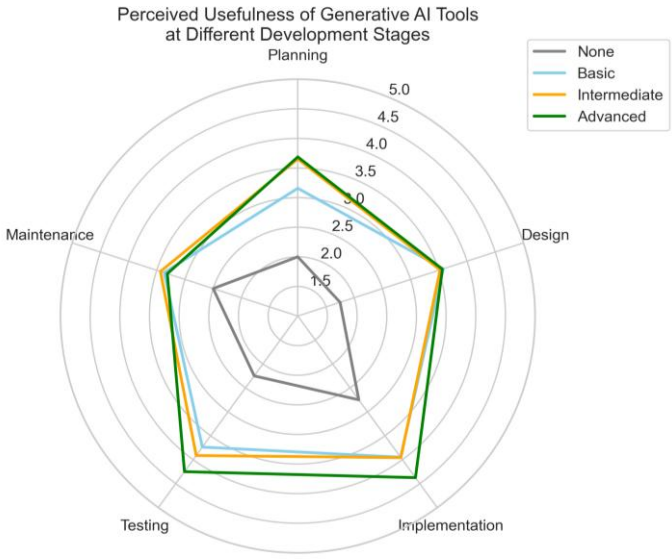

Fig. 3. Perceived usefulness of generative AI tools at different stages of software development grouped by AI usage experience

Developers with more experience using generative AI tools rated their usefulness higher at all stages of the software lifecycle. The biggest differences were observed in the implementation and testing stages, but even in the planning and maintenance stages — which were rated lower overall — advanced users were significantly more optimistic than less experienced users.

3.4. Perceived Risks and Changing Work Expectations

To address RQ3, participants were asked whether they feared job loss due to GenAI. Responses revealed that less experienced developers (under 3 years) expressed the highest concern, while more experienced professionals were less worried, likely due to greater familiarity with technological change. Figure 4 illustrates this trend.

Fig. 4. Perceived concern about job loss due to generative AI by work experience in IT

Another issue explored was the perceived rise in productivity expectations following the adoption of GenAI. Respondents were asked whether they felt increased pressure to deliver tasks faster or take on more projects. The highest pressure was reported by those who used GenAI tools regularly or intensively (Fig. 5).

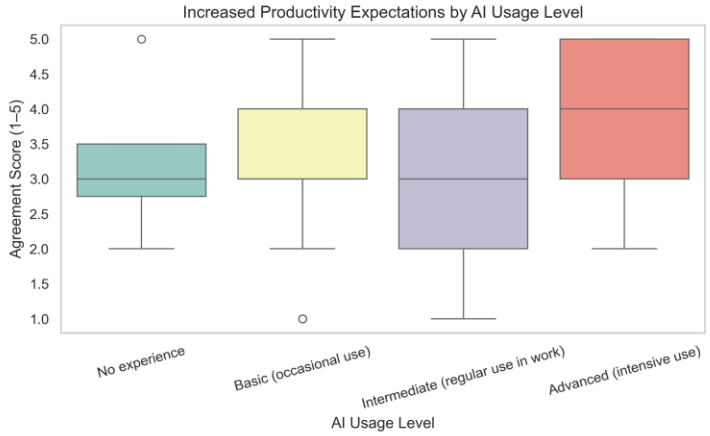

Fig. 5. Perceived increase in productivity expectations after the introduction of generative AI tools, by intensity of AI tool usage

Figure 6 further highlights that early-career professionals (less than three years of experience) perceived the highest pressure to meet increased productivity expectations. This suggests that the introduction of GenAI may disproportionately affect those at the beginning of their careers.

Fig. 6. Perceived increase in productivity expectations after the introduction of generative AI tools, by level of work experience in the IT industry

Overall, the findings suggest that GenAI adoption may contribute to shifting performance standards, particularly affecting younger and less experienced developers, as well as intensive tool users. These results are consistent with those of [8], who observed that less experienced developers report greater uncertainty and pressure when adapting to AI-enhanced workflows.

3.5. Organizational support for generative AI tools

To address RQ4. *Do organizations support the use of generative AI tools, and does this affect employees' perceptions of pressure or job security?* respondents were asked whether their company or team encourages the adoption of such tools in projects. Their responses were then analyzed in relation to two key concerns: increased performance demands and fear of job loss. A Kruskal–Wallis test was conducted to compare perceptions among respondents whose organizations support, do not support, or are uncertain about the use of generative AI.

The analysis revealed no statistically significant differences in either the perception of increased productivity expectations ($H = 3.93$, $p = 0.1399$) or fear of job loss ($H = 1.64$, $p = 0.4415$). These findings suggest that an organization's stance alone may not directly influence individual experiences of pressure or insecurity related to AI adoption.

These findings suggest that both emotional and performance-related impacts of GenAI adoption are shaped by user experience and seniority, not only by organizational policy.

3.6. Developer perspectives on generative AI tools

To answer RQ5, we analyzed responses to four open-ended questions concerning the perceived benefits, challenges, work-related changes, and suggestions for improvement associated with the use of generative AI in software development (Questions 12–15).
Using inductive thematic analysis, we identified recurring patterns and grouped them into four main categories. Representative responses provided by participants support each theme.

Participants emphasized a wide range of benefits associated with the use of GenAI:

- Time-saving and efficiency – accelerated coding, reduced time spent on documentation, debugging, or repetitive tasks.
- Problem-solving assistance – especially valuable when dealing with unfamiliar technologies or solving well-known challenges more quickly.
- Learning support – AI was often used as a companion in exploring new tools, programming languages, or technical concepts.
- Creative enablement – by automating routine tasks, developers could focus more on prototyping and high-level design.

Example response: *"AI helps me in programming – I now use over 7 languages, some of which I barely know, but with ChatGPT's help, I can solve complex problems."*

At the same time, many participants reported challenges related to generative AI use. The most frequently mentioned issues included:

- Hallucinations and low code quality – incorrect or logically inconsistent outputs requiring significant revision.
- Lack of contextual awareness – AI systems often failed to consider project-specific architecture, business requirements, or code structure.
- Overgeneralization and vague guidance – leading to misleading or incomplete answers.
- Security and privacy concerns – particularly when tools required access to full codebases (e.g., GitHub Copilot).
- Overreliance on AI – fears that constant AI use could erode problem-solving skills and motivation to learn.

Example response: *"Sometimes, the AI guesses and sends code that can't possibly work. It would be better if it simply said it doesn't know."*

Participants also described noticeable changes in their daily work routines. Commonly

reported changes included:

- Faster task execution and reduced effort in searching for technical information.
- Delegation of basic coding tasks to AI, allowing more time for creative, strategic, or architectural thinking.
- In some cases, these improvements came at a cost of increased pressure and higher expectations from managers or clients.

Example response:*"Along with faster work, expectations regarding task completion time and efficiency have also increased."*

Finally, respondents offered several thoughtful suggestions aimed at enhancing the usability and trustworthiness of generative AI tools:

- Stronger integration with development environments and version control systems.
- Improved contextual understanding across the entire project.
- Support for complex and non-standard problems beyond basic templates or known solutions.
- Greater transparency – such as explanations for suggested code or alternative options with trade-offs.
- Clearer data-handling and privacy safeguards.

Example response: *"When there are multiple possible solutions, it should present them along with their pros and cons, rather than suggesting only one."*

4. Summary and discussion

The study confirms that GenAI has practical applications in programmers' daily work, especially for intensive users, who report clear gains in efficiency, creativity, and problem-solving. Importantly, it was not confirmed that Gen AI reduces the need for technical knowledge - this belief was independent of the level of advancement of users. These findings may inform how organizations structure onboarding and continuous training in GenAI-assisted environments, and suggest directions for further research on differentiated support strategies.

The results obtained correspond with previous studies [1], [5], which emphasize the importance of intensive, practical implementation as a key factor influencing positive perceptions of AI. The study also noted a difference in the level of fear related to job loss, which was felt most strongly by less experienced programmers. This may indicate the need for adaptive support and redefinition of competencies in environments that implement new technologies.

Despite the numerous advantages, the study also revealed the limitations of the tools used – lack of context, low quality of some results, or difficulties in solving non-standard problems. These findings align with [7], who noted that current applications of GenAI in software engineering still concentrate primarily on code generation, with limited impact on broader development stages. These conclusions are consistent with empirical findings reported by Campos et al. [2] and underscore the need for further development of GenAI tools, particularly in terms of integration with the design context, personalization of results, and enhanced reproducibility and governance, as emphasized by [4]. From a practical perspective, the results suggest that successful implementation of generative AI tools in development teams requires:

- a differentiated approach to users – from training for less experienced users to advanced integrations for intensive users,
- transparency in communicating the benefits and limitations of these tools,
- and developing metacognitive skills – such as critical thinking, assessing the reliability of results, and the ability to correct AI errors.

5. Conclusions, study limitations, and future research

The study offered empirical insights into GenAI's role in programmers' work, showing its predominant use in implementation and testing, and stronger reception among experienced users. Advanced users perceive these tools as real support not only in routine

tasks but also in problem-solving and creative work.

The study had some limitations. First, the number of respondents was relatively small, which limits the possibility of generalizing the results. Second, the sample selection was purposeful and partly network-based (snowball sampling), which may be associated with biases resulting from the respondents' activity in environments open to new technologies. Third, although the questionnaire included open-ended questions, the scope of qualitative analysis could have been further deepened by employing methods of triangulation of sources, such as analyzing observations or case studies. Additionally, although participants represented a broad spectrum of job roles, most categories were represented by only one or two individuals, which made subgroup analysis by role infeasible. The study also treated GenAI tools as a unified category, without differentiating between specific tools used. It also did not capture participants' geographic location, which may influence adoption patterns and organizational practices.

In future research, it would be worthwhile to expand the scope of the analysis to include additional perspectives, such as the impact of generative AI on team collaboration, the development of soft skills, or decision-making processes in agile environments. A valuable direction would also be to examine the long-term impact of AI on the formation of programmers' career paths and the transformation of education models in the fields of computer science and software engineering. Finally, an important area for future analysis remains the development of methods for assessing the quality of generated solutions and the creation of transparent mechanisms for human-AI interaction.

References

1. Calegario, F., Burégio, V., Erivaldo, F., Andrade, D.M.C., Felix, K., Barbosa, N., da Silva Lucena, P.L., França, C.: Exploring the Intersection of Generative AI and Software Development. arXiv preprint (2023). https://doi.org/10.48550/arXiv.2312.14262
2. Campos, A., Melegati, J., Nascimento, N., Chanin, R., Sales, A., Wiese, I.: Some Things Never Change: How Far Generative AI Can Really Change Software Engineering Practice. arXiv preprint (2024). https://doi.org/10.48550/arXiv.2406.09725
3. Chen, A., Huo, T., Nam, Y., Port, D., Peruma, A.: The Impact of Generative AI-Powered Code Generation Tools on Software Engineer Hiring: Recruiters' Experiences, Perceptions, and Strategies. arXiv preprint (2024). https://doi.org/10.48550/arXiv.2409.00875
4. Cheng, H., Husen, J.H., Lu, Y., Racharak, T., Yoshioka, N., Ubayashi, N., Washizaki, H.: Generative AI for Requirements Engineering: A Systematic Literature Review. arXiv preprint arXiv (2024). https://doi.org/10.48550/arXiv.2409.06741
5. Cui, Z., Demirer, M., Jaffe, S., Musolff, L., Peng, S., Salz, T.: The Effects of Generative AI on High-Skilled Work: Evidence from Three Field Experiments with Software Developers. SSRN (2025). https://doi.org/10.2139/ssrn.4945566
6. Donvir, A., Sharma, G.: Ethical Challenges and Frameworks in AI-Driven Software Development and Testing. In: Proceedings of the 2025 15th IEEE Annual Computing and Communication Workshop and Conference (CCWC), pp. 569–576. IEEE, Las Vegas (2025). https://doi.org/10.1109/CCWC62904.2025.10903892
7. Karlovs-Karlovskis, U.: Generative Artificial Intelligence Use in Optimising Software Engineering Process: A Systematic Literature Review. Applied Computer Systems 29(1), 68–77 (2024). https://doi.org/10.2478/acss-2024-0009
8. Kuhail, M.A., Mathew, S.S., Khalil, A., Berengueres, J., Shah, S.J.H.: "Will I be replaced?" Assessing ChatGPT's Effect on Software Development and Programmer Perceptions of AI Tools. Sci. Comput. Program. 235, 103111 (2024). https://doi.org/10.1016/j.scico.2024.103111
9. McKinsey & Company: Unleashing Developer Productivity with Generative AI (2023), https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/unleashing-developer-productivity-with-generative-ai
10. Nguyen-Duc, A., Wang, X., Abrahamsson, P., Zhang, H.: How Will Generative AI Change the Software Developer Role? arXiv preprint (2023), https://arxiv.org/abs/2409.06741