ISSN: 2349-512X
Peer Reviewed & Refereed Journal (IF: 6.95)
Journal Website www.nrjsmm.in

ROLE OF GENAI-POWERED CODE ASSISTANTS IN ACCELERATING DEVELOPER ONBOARDING & TRAINING IN IT FIRMS

Nandani Sharma
Associate Professor, Poddar Management & Technical Campus, Jaipur, Rajasthan
Email: nandani.sharma0@gmail.com

ABSTRACT

This study has discussed how Generative AI (GenAI)-based code assistants, including GitHub Copilot, ChatGPT, and Amazon Code Whisperer, can improve the process of onboarding developers and their training in information technology companies. The study was designed to understand the relationship between the application of AI tools, learning performance, and usefulness, and the overall onboarding in software developers. Quantitative, cross-sectional design has been used and the information gathered using an online structured questionnaire, 150 developers in medium and large organizations in the field of IT were sampled. The data were evaluated with the tools of the descriptive statistics, correlation, reliability testing and multiple regression analysis using the IBM SPSS Statistics. The findings revealed that the AI usage and onboarding performance were positively related, and the relationship was significant ($b = 0.412$, $p < .001$), meaning that the high frequency of the GenAI tools use increased the speed of the onboarding process and code output. The positive influence of learning effectiveness ($b = 0.298$, $p <.001$) and perceived usefulness ($b = 0.204$, $p =.007$) also brought a significant positive effect on the performance of developers. It was also confirmed that GenAI-powered assistants significantly contribute to the enhancement of training efficiency and the shortening of learning time as a result of the ability to describe onboarding outcomes explained by the regression model by 54%. The results of the study were that AI-assisted code assistants are an effective cognitive or learning assistant that positively contribute to the accommodation, confidence, and acquisition of skills in developers during the onboarding process. These results give practical implications to IT companies that want to adopt AI-related learning in their developer training courses and reveal how GenAI technologies can influence the future of software engineering training and the workplace.

Keywords: Generative AI, Training, IT Companies, ChatGPT.

INTRODUCTION

Introduced by the phenomenon of Generative Artificial Intelligence (GenAI)-powered code assistants like GitHub Copilot, ChatGPT, and Amazon CodeWhisperer, such technologies have quickly transformed the process of learning software engineering, how makers coordinate, and the speed at which they write code both professionally and personally. Large language models (LLM) support systems can read natural language prompts, write code, refactor existing scripts, and offer context-sensitive explanations which radically transforms the essence of software development and training (Qiao, Shihab, and Hundhausen, 2025). In the ever-skilled shortage industry of software, the global market requires faster vertical integration of novel AI tools, and all to aid onboarding of developers as well as endless professional growth (Rico & Oberg, 2025).

Within the framework of IT companies, developer onboarding can be described as the systematic procedure by which newly confirmed employing developers are introduced to organizational codebases, instruments, and workflows. It is traditionally time consuming and involves mentoring, code reviews and document reviews. Nevertheless, with the appearance of GenAI assistants, it makes this process easier to automate and personalize large portions of this process. According to recent studies, AI-based assistants can help developers make faster code sense and memorize programming concepts, and get increased task autonomy within the initial few weeks of work (Ferino, Hoda, and Grundy, 2025). Amalfitano, Metziger, and Autili (2025) also state that GenAI systems are not simply productivity tools but can be viewed as cognitive amplifiers that facilitate the transfer of information via interactive feedback and contextual learning using the systems to onboard and train new developers.

In addition, the rising use of GenAI in agile and hybrid IT settings indicates its relevance to the organization. GenAI adoption fits the agile principles, as now intelligent, real-time learning is integrated into a sprint cycle, and senior developers are not needed as much because it will allow onboarding faster and more autonomously, as it is reported by Khan (2025). On the same note, Hartto (2025) states that the AI-assisted developer experience tool results in greater productivity and a reduced frustration factor of having to navigate complex codebase patterns. Such shift to AI-enhanced learning spaces is further justified by managerial research, including one by Liukkonen (2025), that reported that businesses that started using AI assistants in HR and technical onboarding processes reported time to productivity and job satisfaction continue to be significant.

Although these benefits exist, there are still issues. Mueller and Bruhin (2526) observe that the issue of trust and explainability as well as the fear of devaluation of skills are the main reasons why developers are most often reluctant to use GenAI. Equally, Fichera (2025) observes that even though the AI models can be used to expedite low-level coding activities, they also can diminish the chances of experiential learning otherwise they are not well incorporated into the human-centred mentorship systems. Thus, the research question of the quantitative impact of GenAI-powered assistants on development efficiency through onboarding and training effectiveness is a paramount study topic of IT organizations aiming at finding the automation-expertise balance.

This study will seek to empirically explore how GenAI-based code assistants can faster the onboarding or training of developers in IT companies. It is based on the recent developments in AI-assisted software development and tries to measure the connection between GenAI usage and the primary performance indicators, including onboarding time, learning performance, and coding performance. This study aims to aid the fields of artificial intelligence, workforce learning, and software engineering management by providing evidence-based implications on the adoption of AI-based workforce developer enablement strategies by companies through the implementation of quantitative, regulations-based analysis.

LITERATURE REVIEW

With the advent of Generative Artificial Intelligence (GenAI)-driven code assistants, including GitHub Copilot, ChatGPT and Amazon Code Whisperer, the IT training and training environment at software development firms is changed. The recent researches have always pointed out the potential that they possess to make developers more productive, streamline the on boarding process and provide continuous learning in agile environments. The pilot case revealed by Hakkinen (2025) in the Finnish technology consultancy revealed that the developers working with the aid of GenAI tools took up to 30% less time to complete a task during the initial training than their control groups that did not receive AI assistance. Real-time code suggestions, context-sensitive documentation, and smart debugging support

ISSN: 2349-512X
Peer Reviewed & Refereed Journal (IF: 6.95)
Journal Website www.nrjsmm.in

were found to contribute to this improvement and allow new developers to learn faster as a result.

On the same note, Thaw (2025) proved that AI-based assistants can be beneficial users in the transfer of knowledge through providing contextual learning when generating and reviewing code, thus allowing junior developers to comprehend complex codebases faster. On a bigger organizational level, Allam (2025) researched the concept of AI-based upskilling programs in DevOps environments and found out that generative models served to fill skill gaps as they provided adaptive on-demand learning experience that supplemented the formal training approach. Such results can be compared to those, which were observed by Rajbhoj, Somase, and Kulkarni (2024), who noted that ChatGPT-assisted developers felt more confident and experienced a lowered cognitive load when using onboarding materials, which led them to be at role-readiness faster.

In addition to the personal performance, research at the organizational level highlights the fact that the GenAI integration improves team productivity and code quality. Noor (2025) established that more than 90 per cent of software developers in multinational firms felt that GenAI tools played a pivotal role in enhancing collaboration and cutting redundancy as well as ensuring uniformity in code standards. This perspective was added to by Kirchner, Bolisani, and Kassaneh (2025), who put GenAI in the perspective of knowledge management--code assistants are a source of tacit knowledge-aiding mentorship and feedback practices as well as organizational learning. These impressions were also confirmed by Szolderits (2025) who stated that GitHub Copilot had a great chance to increase accuracy in the area of code implementation, and novice programmers were the greatest benefactors because of the code explanations related to errors, as well as the prompt-based guidance.

In terms of workforce development, Joshi (2025) has highlighted that AI-based trainers perform a radical role to train the following generation of developers to work in flexible, hybrid-manner environments. This was supported by Marsden (2025) who revealed that AI introduction into enterprise operations promotes the culture of experimentation and innovation that expedites the onboarding processes and enhances cross-functional coordination. Nevertheless, researchers like Le (2025) and de Lima Gomes (2025) warned that excessive use of the generative tools may result in deskilling and intellectual laziness and offered to rely on a balanced pedagogical strategy of combination AI-aided learning with practices of critical thinking and code reviewing resting on human judgment.

In addition, the academic evidence base is supported with industry surveys. More recent data released as a 2024 global survey of developers by Noor (2025) indicated that 58% of IT companies were already using GenAI tools in conducting their onboarding process and reported a tangible time-saving in their developer ramp-up and code-reuse timelines. In his extensive book Generative AI in Software Development (2025), Pereira has contended that the AI assistant is not a tool of automation but a co-creator that transforms the learning paradigms so that software developers can address architecture, as opposed to syntax memorization. This theoretical change, expected to be made based on the automation concept to augmentation, is critical to understanding how generative AI can work to support the rapid and sustainable development of the workforce in the sphere of software-intensive companies.

OBJECTIVES FOR THE STUDY

- To study the effect of GenAI-based code assistants on the performance of developer onboarding at IT companies.

- To determine the impact of the learning effectiveness in reducing the results of developer training as used in AI-assisted coding tools.
- To establish the impact of the perceived utility of GenAI tools with regard to onboarding effectiveness and productivity.
- To assess the overall effects of GenAI-powered code assistants in making the onboarding and training process of software developers faster.

METHODOLOGY

This study involved a quantitative research approach that sought to analyse the impacts of Generative Artificial Intelligence (GenAI)-based code assistants in accelerating the process of developer onboarding and training in IT companies. The quantitative method was selected due to the ability of the researcher to gather numerical data as well as measure the impact of the use of AI tools on the onboarding time, learning effectiveness, and productivity of the developers. The explanatory design was cross-sectional. The survey was performed twice, with data gathered among software developers of IT companies that utilized the GenAI assistant, including GitHub Copilot and ChatGPT or Amazon CodeWhisperer. This design was effective in determining the patterns and relationship among the main variables of study without interfering with the normal working environment of the participants. The population targeted consisted of software developers, trainees, and team leaders of medium and large IT companies that used AI tools during coding and training. The data on the participants with varying levels of experience (considering junior, mid-level, and senior developers) was included using a stratified random sampling technique. One hundred and fifty (150) participants had been picked after a power analysis when the sample size was large to provide meaningful statistically significant findings. Data were collected through a structured online questionnaire. The questionnaire had five sections:

1. Demographic information,
2. Frequency and intensity of AI assistant usage,
3. Perceived usefulness of AI tools,
4. Learning effectiveness, and
5. Performance outcomes such as onboarding duration, code accuracy, and productivity.

The questions about the AI usage, usefulness, and learning were rated on the five-point scale of Likert scale between Strongly Disagree and Strongly Agree. Two academic scholars and one professional in the IT industry were involved in the review of the questionnaire and were sought to determine that the questions were transparent and pertinent to the areas of study.

IBM SPSS statistics was used to analyse the collected data. Mean, common deviations, and frequency distributions were employed as descriptive statistics to summarize the characteristics of participants and the general trends. The consistency of the measurement items was checked with the help of the reliability analysis with the help of Cronbach alpha, and the results over 0.70 were accepted as the acceptable ones. A multiple linear regression model was used in order to test the hypotheses of the study. This model contributed to identifying the extent to which AI utilization, learning performance and perceived value predicted the performance of developer onboarding. Residual plots, P-P plots and Variance Inflation Factor (VIF) values were used to verify assumptions in the data (normality, linearity, and multicollinearity) before running the regression analysis. The significance level was

ISSN: 2349-512X
Peer Reviewed & Refereed Journal (IF: 6.95)
Journal Website www.nrjsmm.in

established on $p < 0.05$. ANOVA table was used to test and determine the strength of the model on the basis of the R 2 value and the F-test.

Analysis

Data analysis was done in IBM SPSS Statistics. There were the descriptive statistics, reliability analysis, correlation analysis and multiple regression analysis.

Descriptive Statistics

The demographic features of the respondents and key variables in the study, such as the use of AI tools, the effectiveness of learning, the perceived usefulness and onboarding performance, were summarized by use of descriptive statistics. Table 1 below presents the descriptive statistics for the main study variables.

Table 1: Descriptive Statistics for Study Variables (n = 150)

Variable	Mean	Std. Deviation	Minimum	Maximum
AI Usage	4.12	0.68	2.0	5.0
Learning Effectiveness	3.98	0.74	2.0	5.0
Perceived Usefulness	4.22	0.63	3.0	5.0
Onboarding Performance	4.05	0.70	2.0	5.0

The average value of the AI Usage, as presented in the table 1 (M = 4.12, SD = 0.68), shows that the majority of the respondents were regular users of the GenAI-driven code assistants. The average score of the Learning Effectiveness (M = 3.98) demonstrates that the average respondents tended to believe that use of AI tools enhanced their learning and training experience. The high average of Perceived Usefulness (M = 4.22) indicates that there are positive attitudes to the fact of the usefulness of AI tools. Lastly, Onboarding Performance (M = 4.05) indicates that the majority of developers believed that they were able to accomplish onboarding successfully with the help of AI tools.

Reliability Analysis

The internal consistency of the multi-item scales in the questionnaire used Cronbach Alpha in measuring it. A correlation of 0.70 or more shows that the reliability is satisfactory.

Table 2: Reliability Statistics

Construct	Number of Items	Cronbach's Alpha (α)
AI Usage	5	0.86
Learning Effectiveness	4	0.82
Perceived Usefulness	4	0.88
Onboarding Performance	5	0.84

All constructs showed strong internal consistency, with Cronbach's alpha values ranging from 0.82 to 0.88, which means the items used to measure each variable were reliable and suitable for further analysis.

3. CORRELATION ANALYSIS

Pearson's correlation test was conducted to examine the relationship between the study variables.

Table 3: Correlation Matrix

Variables	**1 (AI Usage)**	**2 (Learning Effectiveness)**	**3 (Perceived Usefulness)**	**4 (Onboarding Performance)**
1. AI Usage	1	.642**	.597**	.658**
2. Learning Effectiveness	.642**	1	.563**	.617**
3. Perceived Usefulness	.597**	.563**	1	.589**
4. Onboarding Performance	.658**	.617**	.589**	1

Note: p < 0.01 (2-tailed)

The correlation coefficients show strong and positive relationships among all key variables. AI Usage had a strong correlation with Onboarding Performance ($r = .658$, $p < .01$), suggesting that higher usage of AI assistants was associated with improved onboarding outcomes. Similarly, Learning Effectiveness ($r = .617$, $p < .01$) and Perceived Usefulness ($r = .589$, $p < .01$) were positively related to Onboarding Performance.

4. MULTIPLE REGRESSION ANALYSIS

A multiple linear regression analysis was conducted to determine the extent to which AI Usage, Learning Effectiveness, and Perceived Usefulness predicted Onboarding Performance.

Table 4: Model Summary

Model	**R**	**R^2**	**Adjusted R^2**	**Std. Error of Estimate**
1	.735	.540	.529	0.481

The model explained 54% of the variance in Onboarding Performance ($R^2 = 0.540$). This indicates that AI Usage, Learning Effectiveness, and Perceived Usefulness collectively explained a substantial portion of the differences in onboarding outcomes among developers.

Table 5: ANOVA Results

Model	**Sum of Squares**	**df**	**Mean Square**	**F**	**Sig.**
Regression	38.25	3	12.75	55.18	.000
Residual	32.56	146	0.22		
Total	70.81	149			

The ANOVA table shows that the regression model was statistically significant ($F(3,146) = 55.18$, $p < .001$). This means the predictors—AI Usage, Learning Effectiveness, and Perceived Usefulness—significantly predicted Onboarding Performance.

Table 6: Regression Coefficients

Predictor	**B (Unstandardized)**	**Std. Error**	**Beta (Standardized)**	**t**	**Sig.**
Constant	0.584	0.243	—	2.40	.018

AI Usage	0.428	0.076	0.412	5.63	.000
Learning Effectiveness	0.325	0.082	0.298	3.96	.000
Perceived Usefulness	0.217	0.079	0.204	2.75	.007

The regression outcome indicates that all the three predictors, namely, AI Usage, Learning Effectiveness, and Perceived Usefulness, had a significant contribution to onboarding performance. The greatest impact was made by AI Usage ($b = 0.412$, $p < .001$), then the Learning Effectiveness ($b = 0.298$, $p < .001$) and Perceived Usefulness ($b = 0.204$, $p =.007$). This implies that those developers who had regular experience with GenAI tools, found them helpful, and believed that they acquired them well were more likely to complete faster during onboarding and perform well in the training.

The analysis findings proved all the three hypotheses developed in the research. The data revealed that there is an important and positive connection between AI Usage and Onboarding Performance that mean that system developers using GenAI-based code assistants completed their onboarding tasks more effectively and faster adjusted to organizational processes. The analysis also indicated that the Learning Effectiveness had a significant positive effect on Onboarding Performance, implying that in case the developers believed that they comprehended and used the new code concepts with the assistance of the AI tools, the general performance during the onboarding phase improved. Also, the perceived usefulness had a positive impact on Onboarding Performance, i.e. larger engagement and productivity in the course of this activity was observed in developers who thought AI assistants are helpful, helpful in their activities. Altogether, these findings demonstrate that code assistants with GenAI specifically contribute to the faster and more effective developer onboarding and the improvement of training results in IT companies. The results highlight the fact that the approaches to implementing AI tools in the training setting of developers can enable the training to be significantly more efficient, it can also be done within a shorter time, and new software engineers can develop their skills faster.

DISCUSSION

The results of the presented study were strong evidence of the fact that Generative AI (GenAI)-driven code assistants are positively influential and effective in the performance of developer onboarding and training in IT companies. It was found that AI Usage, Learning Effectiveness and Perceived Usefulness are significant predictors of onboarding outcomes based on the regression analysis. This indicates that active users of the AI assistant like GitHub Copilot, ChatGPT, or Amazon CodeWhisperer did not only have an easier time during the onboarding process but also acquired a better idea of programming principles and organizational processes. The findings are consistent with other research by Hakkinen (2025) and Thaw (2025), which concluded that developers who used AI had better productivity, shorter time in undertaking tasks, and were more confident when creating software in the initial stages.

The correlation between AI Usage and Onboarding Performance is positive proving that GenAI tools may be successfully used as a real-time mentor, providing contextual support and eliminating the need to rely on senior software developers due to repetitive questions. This observation validates Amalfitano et al. (2025), who suggested that GenAI systems are cognitive amplifiers that enhance behaviours of faster learning because of immediate feedback and the use of code explanations. Likewise, Learning Effectiveness has such an impact that it causes the development of other offered AI tools to be more successful, which supports the fact that the tool can help developers better understand intricate programming codes and adopt new technology as its learning curve highest. The dynamic and interactive quality of these tools also makes self-paced learning environment where Ferino, Hoda, and Grundy (2025) refer to the learning autonomy framework.

Another significant factor according to the study that was important in forecasting onboarding performance was the Perceived Usefulness. This implies that those developers who had faith and belief in the support of the GenAI tools were more inclined to use them positively causing them to achieve increased engagement and productivity. These results are aligned with the Technology Acceptance Model (Davis, 1989) that identifies perceived usefulness as a fundamental attribute that determines the actual practice and results of technology adoption. It also corresponds with Noor (2025), who indicated that positive attitudes of the developers towards the AI-assisted development positively contribute to the collaboration of teams and increased efficiency in the coding process.

Although the results eloquently prove the usefulness of GenAI tools, they also indicate personal considerations. The quantitative design of the study quantified the results of performance, but failed to examine the qualitative nature of the user experience, including trust, creativity, or the ability to retain long-term skills. At least one study by Le (2025) and another conducted by Mueller and Bruhin (2025) has warned of the negative consequences of over normalization of AI assistants as false sense of cognitive ease and absence of ability to solve problems independently. Hence, any organization that uses such tools must consider automation and manual mentorship and critical thinking activities to ensure the autonomy of the developers and diversity in skills. Moreover, when adopting the use of GenAI systems in corporate training, there is also an ethical concern, such as data privacy, bias generated by AI-driven systems, and intellectual property concerns.

Hence, this study has shown that GenAI-supported code assistants contribute greatly to the performance of developer onboarding and training in IT companies. These tools can enable developers to join development teams quicker and be productive within shorter periods of time by improving learning effectiveness, perceived usefulness, and real-time code support. The findings can help academic and industry communities discuss the offered topic of AI-assisted workforce development. Practically, the findings can be used in designing AI-enhanced onboarding programs in IT organizations to focus on guided learning, continuous feedback, and performance analytics. It can be proposed to use longitudinal or mixed-method research designs in the future to investigate how the adoption of GenAI affects the retention of skills, creativity, and innovation capacity in software development teams in the long term. Altogether, the introduction of GenAI-assisted assistants is one bold future step towards more productive, intelligent, and responsive developer training systems in the contemporary IT enterprise.

REFERENCES

1. Albaroudi, E., Mansouri, T., Hatamleh, M., Elbehairy, M., & Alameer, A. (2025). Generative AI and the future of software engineering in Saudi Arabia: Governance, innovation, and workforce transformation. *International Journal of Theoretical & Applied Computational Intelligence*.
2. Allam, S. (2025). Staff Upskilling with Generative AI in DevOps: Bridging the IBM i Skills Gap Through AI-Powered Training Methodologies. *Journal Of Applied Sciences*, *5*(11), 56-62.

ISSN: 2349-512X
Peer Reviewed & Refereed Journal (IF: 6.95)
Journal Website www.nrjsmm.in

3. Amalfitano, D., Metzger, A., Autili, M., Fulcini, T., Hey, T., Keim, J., ... & Vogelsang, A. (2025). A Research Roadmap for Augmenting Software Engineering Processes and Software Products with Generative AI. *arXiv preprint arXiv:2510.26275*.
4. Cohen, J. (2013). *Statistical power analysis for the behavioral sciences*. routledge.
5. Creswell, J. W., & Creswell, J. D. (2017). *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage publications.
6. Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS quarterly*, 319-340.
7. Ferino, S., Hoda, R., Grundy, J., & Treude, C. (2025). Junior Software Developers' Perspectives on Adopting LLMs for Software Engineering: a Systematic Literature Review. *arXiv preprint arXiv:2503.07556*.
8. Fichera, A. (2025). Quality Engineering for GenAI-ALM Integration in Automotive Requirements Management.
9. Hair, J. F. (2014). *A primer on partial least squares structural equation modeling (PLS-SEM)*. sage.
10. Häkkinen, P. (2025). Generative AI and Software Developer Productivity-A Pilot Case Study at a Finnish Technology Consultancy.
11. Hartto, N. (2025). What impact do AI/LLM tools have on developer experience?.
12. Khan, O. (2025). *Integrating generative AI into agile methodologies: exploring benefits, challenges, and emerging practices* (Master's thesis, O. Khan).
13. Kirchner, K., Bolisani, E., Kassaneh, T. C., Scarso, E., & Taraghi, N. (2025). Generative AI Meets Knowledge Management: Insights From Software Development Practices. *Knowledge and Process Management*, *32*(4), 223-235.
14. Le, H. (2025). The Deskilling of Software Development and the Impact of AI Chatbots on Programmers' Skills and Roles. In *Generative AI in Software Engineering* (pp. 397-426). IGI Global Scientific Publishing.
15. Liukkonen, S. (2025). A framework for implementing generative artificial intelligence in business operations.
16. Mueller, L., & Bruhin, O. (2025). Developer Resistance to Generative AI Adoption: Identifying Barriers in Software Development.
17. Noor, N. (2025). Generative AI-assisted software development teams: opportunities, challenges, and best practices.
18. Pereira, S. (2025). *Generative AI for Software Development: Building Software Faster and More Effectively*. " O'Reilly Media, Inc.".
19. Qiao, Y., Shihab, M. I. H., & Hundhausen, C. (2025). A Systematic Literature Review of the Use of GenAI Assistants for Code Comprehension: Implications for Computing Education Research and Practice. *arXiv preprint arXiv:2510.17894*.
20. Rajbhoj, A., Somase, A., Kulkarni, P., & Kulkarni, V. (2024, February). Accelerating software development using generative ai: Chatgpt case study. In *Proceedings of the 17th innovations in software engineering conference* (pp. 1-11).

21. Rico, S., & Öberg, L. M. (2025, June). Challenges and Opportunities for Generative AI in Software Engineering: A Managerial View. In *Proceedings of the 33rd ACM International Conference on the Foundations of Software Engineering* (pp. 1338-1344).

22. Saunders, M., Lewis, P., & Thornhill, A. (2009). *Research methods for business students*. Pearson education.

23. Szolderits, C. M. (2025). *A New Era of Coding: Investigating GitHub Copilot's Influence on Productivity, Code Quality, and Ethics in Software Development. AAU NetLibrary.* https://netlibrary.aau.at/obvuklhs/content/titleinfo/11700723/full.pdf

24. Thaw, T. T. T. (2025). How Effective are AI-powered Code Assistants in Enhancing Developer Productivity?