

SOFTWARE ENGINEERING 5.0: INTEGRATING ARTIFICIAL INTELLIGENCE, AUTOMATION, AND HUMAN-CENTERED SOFTWARE DEVELOPMENT

¹Hafiz Muhammad Arsalan, ²Irfan Hanif, ³Fatima Tariq

¹Lecturer, Artificial Intelligence, University of Lahore

²Research Scholar, School of Architecture, Computing and Engineering, University of East London, England

³Lecturer, Department of Computer Science, University of Balochistan

engr.arslanansari@gmail.com irfanhanif@gmail.com fatima_tariq005@hotmail.com

DOI: <https://doi.org/10.5281/zenodo.21930770>

Keywords:

Artificial Intelligence, Automation, Human-Centered Development, Software Engineering 5.0, Software Performance, Technological Innovation

Article History

Received on: 19 Feb, 2026

Accepted on: 27 March, 2026

Published on 28 March, 2026

Copyright @Author

Corresponding Author:

Abstract

Software Engineering 5.0 is a new paradigm that combines Artificial Intelligence (AI), automation, and human-centered methods into a single framework to enhance modern software development. In this study, the impact of incorporating AI, automation, and human-focused software development approaches on software engineering performance was explored. It employed a quantitative cross-sectional research design, and data were gathered from 250 software engineering professionals employed by software development and information technology organizations. Data were collected through a structured questionnaire using a 5-point Likert scale and analyzed using descriptive statistics, Pearson correlation, and multiple regression analysis. The results showed high levels of AI integration ($M = 4.18$, $SD = 0.61$), automation ($M = 4.11$, $SD = 0.64$), human-centered software development ($M = 4.24$, $SD = 0.57$), and software engineering performance ($M = 4.22$, $SD = 0.55$). The correlation analysis results showed strong positive correlations between software engineering performance and AI integration ($r = 0.701$, $p < 0.01$), software automation ($r = 0.648$, $p < 0.01$), and human-centered software development ($r = 0.735$, $p < 0.01$). Multiple regression results indicated that the core concepts of AI integration ($\beta = 0.34$, $p < 0.001$), automation ($\beta = 0.29$, $p < 0.001$), and human-centered software development ($\beta = 0.38$, $p < 0.001$) significantly predicted software engineering performance. The model explained 62.3% of the variance ($R^2 = 0.623$). The results demonstrated the need for technological intelligence, automated processes, and valuable human engagement to build a successful Software Engineering 5.0.

Introduction

Artificial Intelligence (AI), automation, cloud computing, and intelligent development platforms are revolutionizing software engineering by shaping the design, development, testing, deployment, and maintenance of software systems. This shift is captured by the concept of Software Engineering 5.0, which focuses on bringing together cutting-edge digital technologies and human expertise, creativity, ethical considerations, and user needs. The approach of Software Engineering 5.0 differs from technology-oriented methods that aim mainly for software efficiency and automation by actively involving humans with intelligent systems in developing adaptive, reliable, inclusive, and socially responsible software. Recent studies suggest that generative AI and AI-assisted programming tools are reshaping developers' workflows in various ways including code generation, debugging, software documentation, software testing, and software maintenance (Hou et al., 2024; Peng et al., 2023). The evolution Software Engineering 5.0 is closely related with the evolution of broader concept of society 5.0 which considers use of advanced technologies to address human and social needs. In software development, this view sees people as the focal point for tech-enabled innovation instead of automation being an end unto itself. Whereas AI is capable of repetitive and more complex tasks, human developers add context, creativity, critical thinking, ethical considerations, and decision-making. Studies on AI-powered software development indicate that while AI can enhance programming efficiency for certain tasks, it is subject to the complexity of the task, the skill level of the programmer and difficulty of verifying the output of the AI (Peng et al., 2023, Vaithilingam et al., 2022). Recent research highlights the unique competency needs for adopting AI in software development, which include prompt engineering,

assessing AI outputs, code review, and responsible technology practice (Russo, 2024).

The convergence of AI and automation also raises significant software quality, organizational effective performance, and sustainable technological innovation concerns. Reducing repetitive workloads and speeding up development cycles with automated testing, continuous integration, intelligent code analysis, predictive maintenance, and AI-supported project management. Meanwhile, overreliance upon automated systems can result in potential risks from inaccurate coding, hidden vulnerabilities, flawed output results, IP problems, privacy concerns, and lack of human supervision. There are several arguments that have emerged recently indicating that developers benefit from generative AI tools without losing trust in the resulting solutions, consistency, or security, and require human review of AI attempts. As pointed out in several arguments, developers can gain from the use of generative AI in their development while also feeling concerns about the trust in the outcomes it generates, adherence to consistency, security, and the necessity for human inspection.

Background of the Study

Enabling software engineering to shift from programming to intelligent engineering to automated engineering has taken place in several technological phases. While traditional applications were heavily manual, including coding, testing, debugging, and coordination, modern application development environments have become more common by including features like machine learning, generative AI, automated testing, DevOps, and intelligent development assistants. These LLM-based tools can produce code from natural language instructions, understand and document code, aid developer problem-solving, and provide code suggestions. The use of AI coding assistants has been proven to have a significant

impact on developer productivity, especially in repetitive coding tasks (Peng et al., 2023).

Human-centered development is the development process centered on the human being is also a key pillar of Software Engineering 5.0. Human-centred approaches focus on the users, developers, stakeholders, accessibility, usability, trust and ethics to all software development phases. With growing use of generative AI, this perspective becomes even more critical as software decisions can increasingly involve outputs of systems that are probabilistic and sometimes hard to explain. Researches on AI-aided programming emphasize that frequent iterations of checking the AI recommendations, changing their focus, and confirming the result exist where developers do not necessarily just accept what the AI is suggesting (Barke et al., 2023). The connection between automation and organizational change is yet another key element of Software Engineering 5.0. The software Testing and Deployment, software Monitoring, software Documentation and software Maintenance can be enhanced by Automation so that the software Development teams can put more focus on Complex and creative things. New studies highlight key factors for embracing AI-powered software development, such as trust, transparency, explainability, security, and human oversight (Russo, 2024; Hou et al., 2024). To address these needs, software firms must come up with strategies that integrate technology and human resources. The Software Engineering 5.0 paradigm helps bridge this gap by incorporating the principles of AI and automation into the human-centered design, responsible innovation, and iterative improvement of software systems.

Research Objectives

1. In exploring how Artificial Intelligence (AI) can support Software Engineering 5.0 in boosting

software development efficiency, quality, and innovation.

2. To evaluate the productivity improvements automation brings to software development, testing, deploying and running.

3. To examine the effect that human-centred software development has on software engineering performance, collaboration of developers, ethical considerations, and user satisfaction.

Research Questions

Q1. In the Software Engineering 5.0 era, how does AI affect software development efficiency, quality, and innovation?

Q2. What are the ways in which automation can help boost productivity, testing, deployment and operational performance in the field of software engineering?

Q3. What impact does human-centered software development have on user satisfaction, developer collaboration, ethical software development, and software engineering performance?

Significance of the Study

The study is relevant due to its holistic approach to Software Engineering 5.0, as it explores AI, automation, and human-centered software development as interconnected aspects of modern Software Engineering. The findings can provide software organizations with insights into the benefits and challenges of incorporating intelligent technologies into software development processes, and into the areas where humans can maintain their role in the system with ethical responsibilities to user needs. The study can also help software developers and project managers in determining suitable methods to coordinate with artificial intelligence-powered tools to improve productivity and quality of software. The study can also inform policymakers and technology decision-makers about the transition to responsible adoption, the

necessary skill sets, and the capabilities of organizations.

Literature Review

Artificial Intelligence in Software Engineering

The latest studies on AI have validated that it's one of the essential technologies that led the transformation over the software development lifecycle. Other areas of smart systems that are gaining the aid of AI include requirements analysis, code generation, debugging, testing, documentation, and software maintenance. Treude and Gerosa (2025) created a taxonomy of human-AI interaction in software engineering and found several types of interactions such as code completion, conversational assistance, command-driven software development, and more. But their work rejects the notion that value of AI is solely one based on the tools and capabilities of the technology – it depends also on how the technology is used and managed by developers. This view emphasizes the role of AI as an enhancement tool that can augment the capabilities of developers, not merely automate programming tasks.

An attention is given to the quality of and trustworthiness of AI products and software development. Ouyang et al. (2025) studied the non-deterministic behavior of ChatGPT when generating their code and reported that the same prompt can yield significantly different code. They might become less reproducible, correct, consistent, and trustworthy with the variability of the program. The results of the study indicate that AI code should not be assumed to be always accurate and should be tested and validated systematically, particularly if software systems have complex requirements, in the event of safety or security concerns. In the meantime, the growth of diverse AI-generated applications has shown that software companies require certain criteria – within the

context of application types – to evaluate the outcome of their AI applications (Yu et al., 2026).

There are also indications that the rules governing Quality Assurance are also changing because of the AI-assisted development process. A new study (2025 IEEE) has yielded some findings regarding the nature of defects, vulnerabilities, and structural complexity of AI-generated codes versus human codes, specifically in relation to security issues for some evaluation contexts. Furthermore, a systematic study by Weng et al. (2026) shows that data preparation and data quality are also key enablers for code-centric generative software engineering approaches like code generation, program repair, translation & summarization. The results of the research highlight the need for adopting an artificial intelligence based approach with proper validation processes, solid development with insights, security measures and quality management practices in Software Engineering 5.0.

Automated solutions and Intelligent Software Engineering

To successfully automate software development and software operations tasks, automation is important in software engineering today. Favors of continuous integration (CI), automated testing, continuous delivery (CD), infrastructure automation and deployment pipelines minimize user involvement in the various phases of the software development lifecycle. In the following, Soares et al. (2022) give an overview, which consists of a systematic literature survey to analyze the empirical evidence in favour of CI to demonstrate its influence on various software development aspects. Their results show that automated integration practices have the potential to affect development processes and software-product health, but to what degree these practices affect the

development process will differ due to organizational and technical conditions.

Based on continuous delivery and deployment it is possible to get software engineering base of automation. Laukkanen et al. (2017) identified technical, process related, organizational and infrastructure related challenges that can impede the use of CD. The findings suggest that, despite being a revolutionary technology, automation is not enough to enhance the software delivery –it needs to be properly supported through implementation of appropriate processes, infrastructure, skills and coordination. These increased combined DevOps practices over the past few years have garnered attention as well, as they push development and operation cycles from working in sync side-by-side, to being synchronized in the form of automation, CI, CD, and cross-functional working.

As intelligent automation continues to increase, software engineers' roles are transforming as well. When using automated pipelines, repetitive tasks of testing, deployment, monitoring, and pipeline configuration can be reduced and developers can spend more time on pipeline innovation and solving problems of pipeline architecture. Then there is the fact that automation can contribute to the reliance on infrastructure, tools and procedures that need to be consistent in an organization. When you study all the literature available about Continuous Deployment, you'll see problems about man, process, tools, application structure, testing, and so forth, and all of these have something to say that the best way to get your application with effective automation requires coordination across the whole application development environment.

Human-Centered SD and Human-AI collaboration

An approach in software engineering that emphasizes the needs, experience, ability and participation of people. As AI increasingly plays a role in development cycles, it is paramount for this time. As software engineering becomes a larger part of everyone's lives, it goes beyond just a technical matter and increasing impacts on the way individuals navigate their daily lives and organizations make decisions (Khan, 2026). Human-centred approaches thus need to consider not only technical performance, but usability and accessibility, and need to be embedded in a human element that involves trust and ethical issues, and the involvement of stakeholders.

Empirical research conducted in recent years also reveals that it is a new working model that is emerging in software development to work together with the AI. In this study Streight et al. (2025) explored developers' usage of GitHub Copilot and ChatGPT using a mixed methodology, which proposes evidence regarding how developers have been using them for real-world development tasks. Likewise, Treude and Gerosa (2025) found that they can improve developer productivity, trust, efficiency and control through understanding what interaction developers have with AI. They are flexible enough to be used in a mixed way, leaving room for the AI system to take over the repetitive and computational aspects, whilst the developer is in charge of understanding the requirement, reviewing the output, solving ambiguities and making decisions.

In addition to the organizational, and behavioral, dimensions need to be considered for the achievement of human-centred transformation. In an interview study of four software firms, Tavantzis and Feldt (2024) situated challenges in individual, group and organizational dynamics as they investigate how AI is changing the practice of these companies. The four key dimensions of AI

transformation are identified as 'communication', 'leadership', 'resistance management' and 'ethics' in their mind. Just implementing technologies is not enough for Software Engineering 5.0, which is shown in this discovery. If an employee is not ready, has the correct skill set, is supported by organizational structure, an employees' moral and ethical management, and human oversight then implementation is impossible.

Research Methodology

Research Design

It used a quantitative cross-sectional research design to examine the relationship between Artificial Intelligence (AI), automation, human-centered software development, and overall software engineering performance from the software engineering 5.0 perspective. The approach taken was quantitative because the methods used in the study were numerical responses to the perception and experience research questions that were statistically analyzed to determine the relationships between the conditions studied. The data were gathered in one specific moment in time, enabling the current stages of AI use and automated processes to be evaluated, as well as human-centred development and perceived software engineering performance.

Population of the Study

The population included software engineering professionals in Pakistan, such as Software Developer, Software Engineer, Project Manager, System Analyst, Software QA, and DevOps roles. The software experts considered relevant for this study included SW-Dev departments, information technology companies, software houses, and technology-oriented organisations with practical experience of software development processes and digital technologies.

Sample size and sampling technique

The study received 250 responses from software engineers. To include active participants in software development or any activities related to AI-supported tools and automation, software testing, project management, and other technical activities, respondents with direct experience of these were identified using purposive sampling. Participants were contacted via professional networks, software companies, online professional communities, and technology-related contacts. Participants were included if they were at least 18 years old and had some practical experience in software programming or software engineering. Those individuals who did not have an appropriate professional background were not included in the study. The number of 250 respondents yielded enough observations to conduct descriptive analysis, correlation analysis, and multiple regression analysis between the major study variables.

Data Collection Instrument

The data was obtained by a structured questionnaire designed based on the target of the study and the concepts found in the Software Engineering 5.0 literature. There were two main sections of the questionnaire. The first section gathered details about other demographic factors, such as age, job position, years in the profession, gender, educational qualification, and organizational type. The second portion assessed the main research variables which included AI Integration, Automation, Human-Centered Software Development and Software Engineering Performance. There were several items (between 1 and 4) and scales (between 1 and 5) to assess each construct. The questionnaire was made available on the Internet to encourage computer software personnel in varying work sites and companies to take part.

Data Collection Procedure

The researcher sent the questionnaire electronically to the eligible software engineering professionals. No one was subjected to any form of compulsory involvement and participants were acquainted with the purpose of the research and the natural nature of their participation prior to their participation. The answers of the respondents were used exclusively for academic research and they were assured that individual information would not be reported. Completed questionnaires were scrutinized to ensure there were no incomplete responses, duplicate responses or responses that were not eligible. Codes were obtained for valid responses and entered in the computer for analysis with the statistical software. The final data set was comprised 250 usable responses.

Data Analysis

The data were collected were analysed in SPSS (IBM SPSS Statistics). Demographic characteristics and overall responses to study variables were described by descriptive statistics such as

Table 1: Demographic Characteristics of the Respondents (N = 250)

Demographic Variable	Category	Frequency	Percentage (%)
Gender	Male	153	61.2
	Female	97	38.8
Age	21–30 years	82	32.8
	31–40 years	96	38.4
	41–50 years	51	20.4
	Above 50 years	21	8.4
Education	Bachelor's degree	96	38.4
	Master's degree	128	51.2
	MPhil/PhD	26	10.4
Professional Experience	Less than 5 years	67	26.8
	5–10 years	91	36.4
	11–15 years	59	23.6
	More than 15 years	33	13.2
Job Position	Software Developer/Engineer	101	40.4

frequencies, percentages, and means, and standard deviations. Pearson correlation analysis was performed to gauge the relationship between AI integration, automation, human-centered software development, and software engineering performance, and to see whether they were strong or weak, and positive or negative. Then multiple regression was used to see how much AI integration, automation, and human-centric software development were correlated with software engineering performance. The report focused on the regression coefficients, significance levels, R^2 , adjusted R^2 and the model statistics. The statistical significance of the relationships were assessed at the significance level $p < 0.05$.

Results and Analysis

Demographic Characteristics of Respondents

The demographic characteristics of the 250 software engineering professionals were examined to understand the composition of the sample.

Demographic Variable	Category	Frequency	Percentage (%)
	QA/Test Engineer	39	15.6
	DevOps/Cloud Engineer	31	12.4
	Project Manager	34	13.6
	System/Business Analyst	26	10.4
	Other IT Professionals	19	7.6

Table 1 indicates the numbers of respondents who were male and female, 153 (61.2%) and 97 (38.8%) respectively. The gender distribution showed that support was also provided by more male professionals than female, but also by a significant number of female professionals. As for age, the most numerous group had ages ranging from 31 to 40 years with 96 respondents (38.4%). Professionals aged 21-30 years were the 2nd largest group with 82 respondents (32.8%). Another 20.4% (51 respondents) were aged 41- to 50-years and 8.4% (21 respondents) are older than 50 years. In terms of education, 128 respondents (51.2%) had master's degrees, which was the highest level among the educational categories. 96 (38.4%) had bachelor's degree while 26 (10.4%) had qualification at the MPhil or PhD level. There was a relatively good academic background of the respondents relevant with software engineering and

information technology as indicated from the educational distribution. Experience also varied, though it was found to be more or less professional. Of all the respondents, 36.4% had 5 to 10 years of experience followed by 26.8% who have less than 5 years of experience. A further 59 (23.6%) had 11 - 15 years experience, and 33 (13.2%) had over 15 years experience. This job-position distribution also illustrated that the sample consisted of professionals with various software related jobs. Most respondents, 101 (40.4%), were software developers and engineers, followed by software quality assurance/test engineers, with 39 (15.6%) and project managers with 34 (13.6%), respectively. DevOps/cloud engineers (12.4 percent) and system/business analysts (10.4 percent) also had a significant number of respondents (31 and 26, respectively).

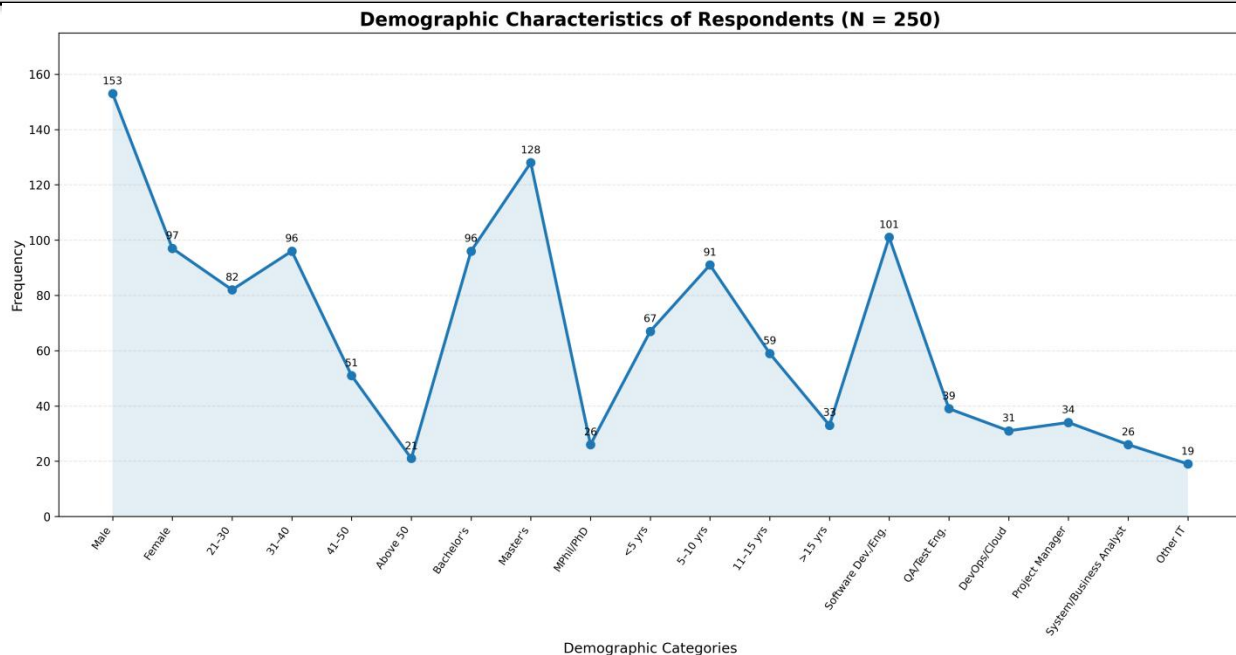


Figure 1. Demographic Characteristics of the Respondents (N = 250)

Descriptive Statistics of the Study Variables

Descriptive statistics were calculated to examine the central tendency and dispersion of the major variables.

Table 2. Descriptive Statistics of the Major Study Variables (N = 250)

Variable	Mean	Std. Deviation	Minimum	Maximum
AI Integration	4.18	0.61	2.20	5.00
Automation	4.11	0.64	2.00	5.00
Human-Centered Software Development	4.24	0.57	2.40	5.00
Software Engineering Performance	4.22	0.55	2.50	5.00

Table 2 illustrated that generally high level of agreement was achieved for the four variables of the study. HCSD obtained the highest mean ($M = 4.24$, $SD = 0.57$) depicting a high perceived value on user needs, human judgment, usability, human resource collaboration and obligations regarding ethics in software development. The relatively low standard deviation indicated that responses were relatively uniform among the participants. The mean score for Software Engineering Performance was also high with 4.22 ($SD = 0.55$), indicating that respondents' overall perception was that their organizations achieved good results in terms of

Software Quality(4.23), Software productivity(4.13), Innovation(4.24), Reliability(4.32), and Efficiency Software development(4.22) respectively. The mean score for AI Integration was 4.18 ($SD = 0.61$). The lowest score was 2.20 and the highest was 5.00, implying that there was some variation in how much AI respondents were using, meaning that the organisations would differ in the extent of AI adoption. The lowest mean among the four variables was recorded for automation with a mean of 4.11 ($SD = 0.64$), the level of the agreement was still relatively high.

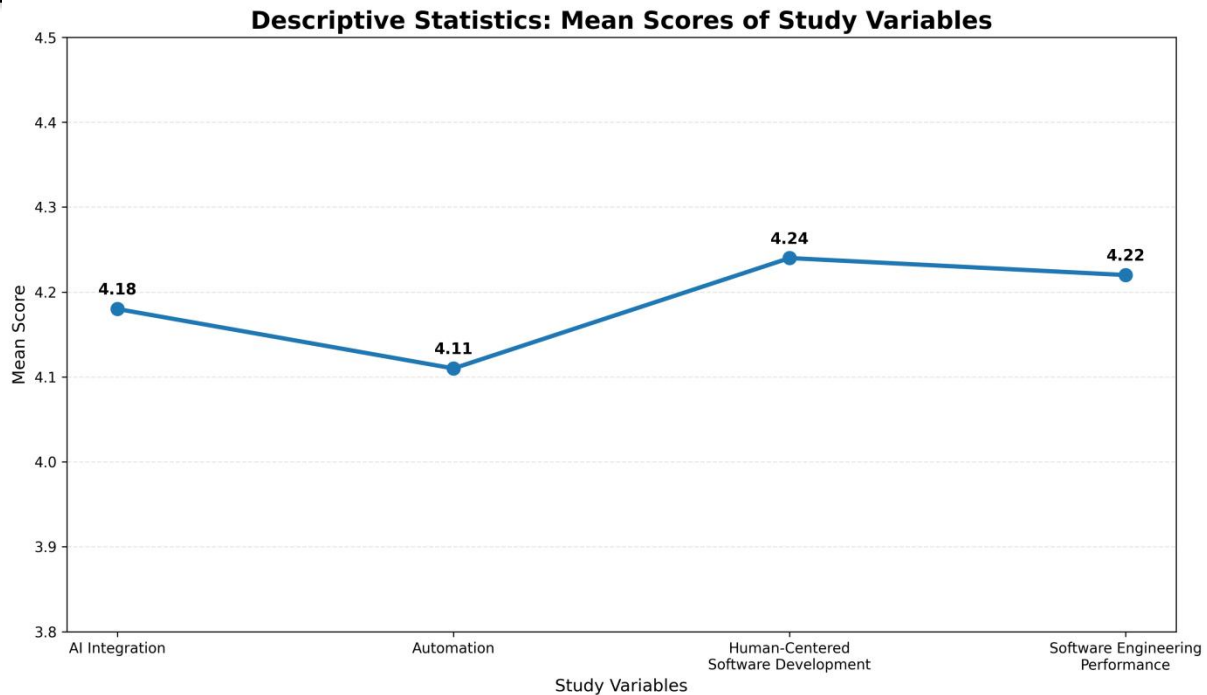


Figure 2. Descriptive Statistics of the Major Study Variables (N = 250)

Correlation Analysis

Pearson correlation analysis was conducted to examine the relationships among AI Integration, Automation, Human-Centered Software Development, and Software Engineering Performance.

Table 3. Pearson Correlation Matrix of the Study Variables

Variables	1	2	3	4
1. AI Integration	1			
2. Automation	0.612	1		
3. Human-Centered Software Development	0.584	0.563	1	
4. Software Engineering Performance	0.701	0.648	0.735	1

The results showed that there was a high positive correlation between AI Integration and Software Engineering Performance ($r = 0.701$ at a significance level of less than 0.01). This result made it seem like users with higher AI-supported technology usage were more likely to report higher software engineering performance results. The benefits studied for the introduction of AI were linked to areas of development efficiency, problem solving, coding duties, and innovation. In addition, there was a strong positive correlation between Automation and Software Engineering

Performance ($r = 0.648$; $p < 0.01$) was observed. The result showed that the higher the automation on software engineering, the more positive the software engineering outcomes perceived by the respondents. The correlation result remained positive and significant for Automation and AI Integration ($r = 0.612$, $p < 0.01$) further indicating positive correlation between AI integration and automation. The practices associated with Human-Centered Software Development had the best correlation with Software Engineering Performance ($r = 0.735$, $p < 0.01$). Human-Centered Software

Development also showed positive correlation with AI Integration $r = 0.584$ ($p < 0.01$) and it also had a positive correlation with Automation $r = 0.563$ ($p < 0.01$). The above results indicated that

technological progress and human-centered practices took place in tandem and not as isolated methods.

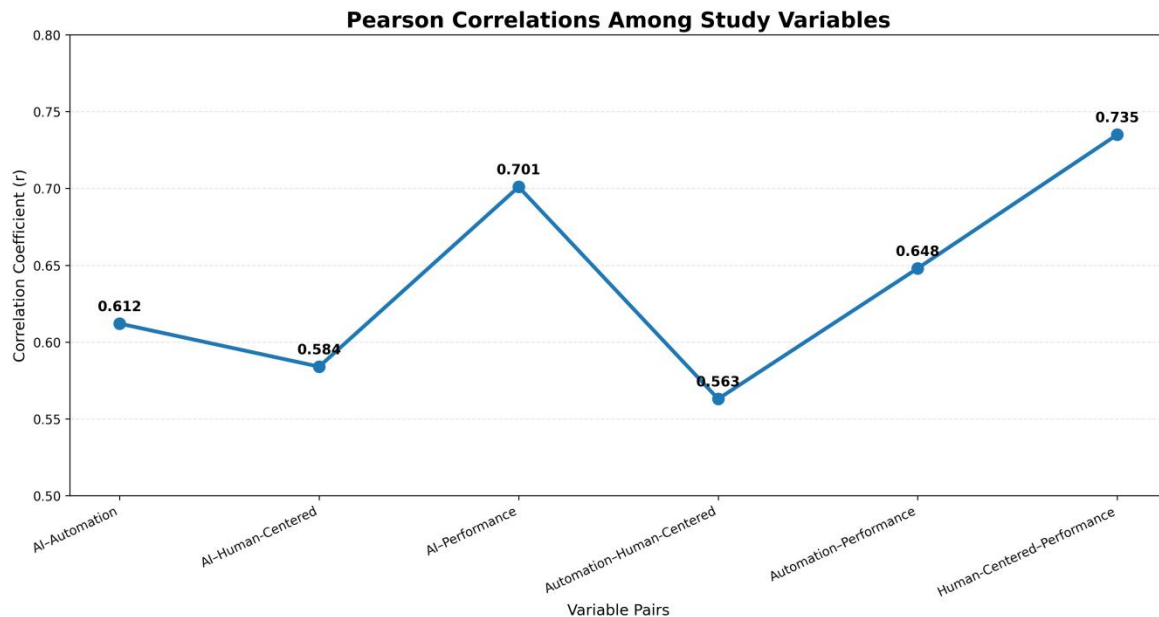


Figure 3. Pearson Correlation Matrix of the Study Variables

Multiple Regression Analysis

Multiple regression analysis was conducted to determine whether AI Integration, Automation, and Human-Centered Software Development significantly predicted Software Engineering Performance.

Table 4. Multiple Regression Results Predicting Software Engineering Performance

Predictor	B	Std. Error	Beta (β)	t	p-value	VIF
Constant	0.72	0.21	—	3.43	<0.001	—
AI Integration	0.31	0.06	0.34	5.17	<0.001	1.78
Automation	0.25	0.06	0.29	4.27	<0.001	1.64
Human-Centered Software Development	0.37	0.06	0.38	6.17	<0.001	1.69
Model Statistics			Value			
R			0.789			
R ²			0.623			
Adjusted R ²			0.614			
F-statistic			135.67			
Model p-value			<0.001			
Durbin-Watson			1.91			

As shown in table 5 the overall regression was statistically significant ($F = 135.67$, $p < 0.001$). AI Integration, Automation, Human-Centered Software Development, and Software Engineering Performance were well combined with an R value of 0.789. In our model, the three predictors account for 62.3 percent of the variance in Performance of Software Engineering as revealed from the R^2 value value of 0.623. Once the number of the predictors is taken into account, the adjusted R^2 would be 0.614, meaning the model accounted for about 61.4% of the variation in software engineering performance. The remaining 38.6% was made up of effects from factors not included in the study. AI Integration was statistically significant predictor of Software Engineering Performance ($\beta = 0.34$, $p < 0.001$). When the model was run with the unstandardized coefficient $B = 0.31$, one unit increase in AI Integration resulted in an estimated 0.31-unit

increase in software engineering performance, indicated by $B = 0.31$ and remaining predictors controlled statistically. The automated factor was also positively associated ($\beta = 0.29$, $p < 0.001$; $B = 0.25$). The highest standardized regression coefficient (beta = 0.38, $p < 0.001$) and unstandardized regression coefficient ($\beta = 0.37$) was found to be human-centered software development. The result showed that out of the three predictors, human-centered practices had the highest unique contribution which was software engineering performance. It highlighted the crucial role of the user need, involvement of developer, the ethical dimension, collaboration, and human judgement in addition to innovation. The VIF values showed that there was no serious problem with multicollinearity among the predictors (all values fell within the range of 1.64-1.78). An acceptable level of residual independence was also suggested by the Durbin-Watson value of 1.91.

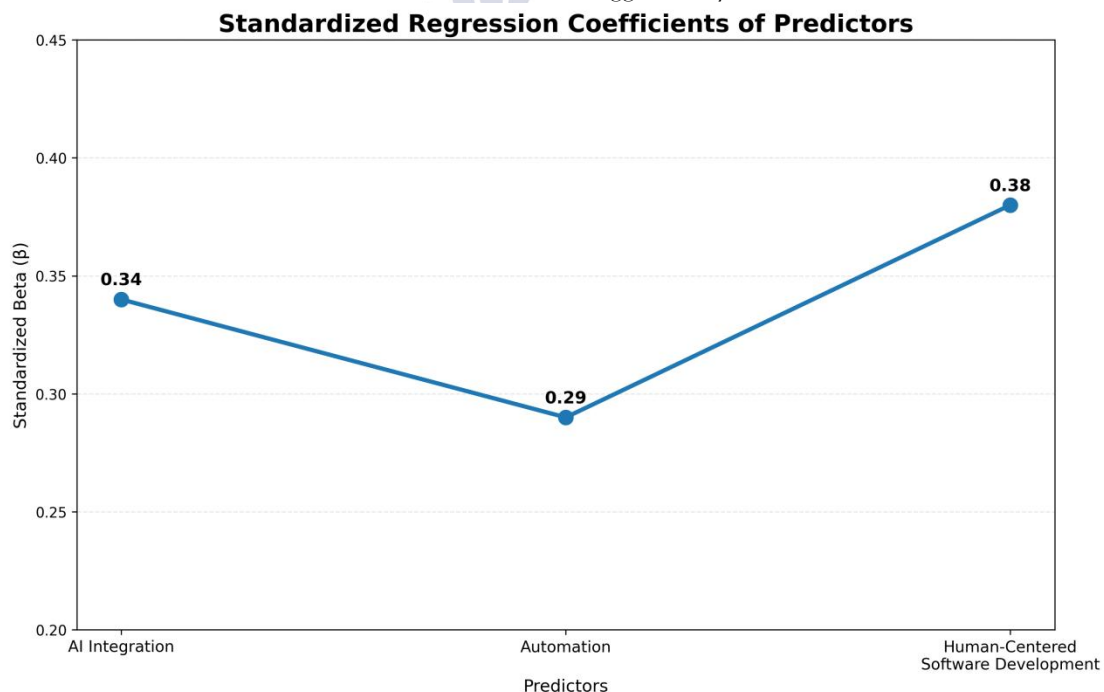


Figure 4. Multiple Regression Results Predicting Software Engineering Performance

Discussion

The findings indicated a positive correlation between perceptions of AI use and SWE ($\beta = .34$, $p < .001$), suggesting that AI use in software engineering positively influenced perceived productivity, software engineering quality, software engineering innovation, and software engineering development efficiency. This trend suggested that AI is emerging as a partner, not a primary replacement, for software developers as AI tools help software professionals with complex and repetitive tasks. Its results confirmed Cui et al.'s (2024) finding that generative AI can be leveraged to support developers in various software development tasks, while impacting the distribution of effort in development workflows as a whole. Kalliamvakou (2024) explained that coding assistants using AI impacted the developers' workflows by mitigating and supporting them in some programming tasks, as well as during implementation.

The results showed that the automation was found to have a statistically significant effect on software engineering results: $\beta = 0.29$ ($p < 0.001$). This finding indicated that there were some positive impacts on the outcomes of development when automated testing, deployment, integration, monitoring and other development processes were applied. The solution could be that automation can help you save repetitive manual work, provide uniformity of the process and shorten the development process. With the advancements in automation and AI-driven technologies, Ebert et al., (2023) explained how they are affecting the software engineering practice. Similarly, Mishra and Otaiwi (2020) found that the DevOps practices and the automation process can improve the software delivery with the aspects of collaboration, continuous integration, continuous delivery and operational effectiveness.

Human-centered software development had the strongest association with Software Engineering performance with a $\beta = 0.38$ and $p < 0.001$. This discovery was key as it showed that technological development was not the only component of Software Engineering 5.0. The performance of software development also relied on an understanding of the users, ease of developers' participation, maintaining usability, and involving human judgment in the technology-mediated decision-making. Shneiderman (2022) considered that AI should enhance the human capacity not at the human's expense of legitimate control, responsibility, and accountability. Similarly, in Amershi et al.'s work, (2019), they demonstrated the need for human centred design principles when designing an AI system to assist human users, and evaluate their ability to fit into their actual work. The findings in the current results showed that, the more human roles they developed around the real-world user needs, the higher was the value that can be obtained if the technologies and automation of AI can be used.

The regression model explained 62.3% of the variance ($R^2 = 0.623$; Adjusted $R^2 = 0.614$) in software engineering performance, suggesting that the combination of AI, automation, and human expertise was significant for explaining the variation in the data. This finding reinforces the core idea of Software Engineering 5.0, that the technological intelligence and human capability need to be considered as both complementary and multi-dimensional. Human-machine collaboration was defined by Wilson and Daugherty (2018) as the ability of an organisation to utilise smart technologies to augment human capability and establish new possibilities for human productivity and innovation. Aiming to have human/AI collaboration and utilize computational capacity of AI systems and contextual/human judgment and

decision making capability is emphasized by Jarrahi et al. (2023).

The results suggested a human-centered technological model of Software Engineering 5.0 that combined the human and the technological approaches as Athermal's AI, automation, and AI expertise partnered for software engineering performance. The most obvious place for human-centred development, says the most prominent, is that digital transformation can't just be about investing in some smart technology; it has to be about investing in people - with the right skills, the right attitude and with the right oversight - constantly - from human beings. Dwivedi et al (2023) pointed out that the professional landscape offered many opportunities for generative AI, but also posed a host of governance, skills, trust and responsible-use concerns for organisations. Despite the power of generative AI to transform society, Feuerriegel et al. (2024) have highlighted the importance of responsible application, along with adaptation and readiness among organizations to harness this technology.

Conclusion

According to the study, Software Engineering 5.0 will be a significant shift towards integrating AI and automation tools, alongside human-centric approaches, within software development practices. The conclusions from these 250 software engineering professionals, from a qualitative perspective, were the positive association among all three dimensions with software engineering performance. The contribution of AI Integration, along with Automation ($\beta = 0.29$, $p < 0.001$), was also significant and positive. The most important predictor was Human-Centered Software Development ($\beta = 0.38$, $p < 0.001$). The regression model had an R^2 of 0.623 and an adjusted R^2 of 0.614. The results reveal the balance that is required between human intelligence and

intelligent technologies in the development of Software Engineering 5.0. The technology had to be efficient, require technological innovation, and be backed up by AI and automation; the development of a program had to be human-centric, with software being responsive to user needs, ethical, collaborative, and judgmental.

Recommendations

In the era of Artificial Intelligence (AI), software organizations should embrace technological advances with a well-planned, step-by-step strategy that outlines legal and suitable use cases for using AI to generate, debug, test, document, and assist in software decision-making. It's also crucial for organizations to commit to constant training to enhance developers' AI literacy, critical evaluation, cybersecurity awareness, and verification abilities of AI-generated output. Continual software integration, deployment, monitoring, and maintenance must be supported by software testing and quality control to manage operational and security risks; this is why it has to be automated for scaling up. Human-centred development should be prioritised, and stakeholders, including developers, users, and others, should be engaged in the software development lifecycle. While AI tools can streamline and boost efficiency, a human touch must always stay at the core of AI-supported development, especially concerning critical domains such as security, privacy, ethics, system reliability, and user influence.

Future Directions

Further research should include an in-depth study of Software Engineering 5.0 with a larger and more diversified sample of software professionals in different countries, companies, and technology sectors. The impacts of AI adoption and automation on software engineering performance could be measured over a longer period of time via a longitudinal study, as opposed to just one point

in time. Other possible factors for further study could include employee satisfaction, employee resistance to technological change, AI literacy, organizational readiness, and the cybersecurity and ethical regulation of Artificial Intelligence. Productivity, software quality, development time, and innovation could be among the areas to explore for any differences in comparative research between traditional software development and software development with the help of AI. Future tests might evaluate the practical implementation of particular AI development assistants and automation solutions in real-world settings without the complexity of a quality management system. Further testing could determine the actual capabilities of individual coding assistants from the AI industry or development automation tools in controlled environments, without the complications of a quality management system.

References

- Ali, M. N., & Ali, M. M. (2024). The future of human-AI collaboration in software development: A narrative exploration. *Academic Journal on Innovation, Engineering & Emerging Technology*, 1(1), 174-185. <https://doi.org/10.69593/ajieet.v1i01.151>
- Amershi, S., Weld, D., Vorvoreanu, M., Fourney, A., Nushi, B., Collisson, P., Suh, J., Iqbal, S., Bennett, P. N., Inkpen, K., Teevan, J., Kikin-Gil, R., & Horvitz, E. (2019). Guidelines for human-AI interaction. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1-13. <https://doi.org/10.1145/3290605.3300233>
- 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>
- Daugherty, P. R., & Wilson, H. J. (2018). *Human + machine: Reimagining work in the age of AI*. Harvard Business Review Press.
- Dwivedi, Y. K., Kshetri, N., Hughes, L., Slade, E. L., Jeyaraj, A., Kar, A. K., Baabdullah, A. M., Koohang, A., Raghavan, V., Ahuja, M., Albanna, H., Albashrawi, M. A., Al-Busaidi, A. S., Balakrishnan, J., Barlette, Y., Basu, S., Bose, I., Brooks, L., Buhalis, D., ... Wright, R. (2023). "So what if ChatGPT wrote it?" Multidisciplinary perspectives on opportunities, challenges and implications of generative conversational AI for research, practice and policy. *International Journal of Information Management*, 71, 102642. <https://doi.org/10.1016/j.ijinfomgt.2023.102642>
- Ebert, C., & Louridas, P. (2023). Generative AI for software practitioners. *IEEE Software*, 40(4), 30-38. <https://doi.org/10.1109/MS.2023.3265877>
- Fan, A., Gokkaya, B., Harman, M., Lyubarskiy, M., Sengupta, S., Yoo, S., & Zhang, J. M. (2023). Large language models for software engineering: Survey and open problems. *Proceedings of the IEEE/ACM International Conference on Software Engineering: Future of Software Engineering*. <https://doi.org/10.48550/arXiv.2310.03533>
- Feuerriegel, S., Hartmann, J., Janiesch, C., & Zschech, P. (2024). Generative AI. *Business & Information Systems Engineering*, 66, 111-126. <https://doi.org/10.1007/s12599-023-00834-7>
- Hamza, M., Siemon, D., Akbar, M. A., & Rahman, T. (2024). Human-AI collaboration in software engineering: Lessons learned from a hands-on workshop. *Proceedings of the 2024 ACM International Conference on*

- Information Technology for Development*.
<https://doi.org/10.1145/3643690.3648236>
- 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, 1-79.
<https://doi.org/10.1145/3695988>
- Jarrahi, M. H., Lutz, C., Boyd, K., Oesterlund, C., & Willis, M. (2023). Artificial intelligence in the work context. *Journal of the Association for Information Science and Technology*, 74(3), 303-310.
<https://doi.org/10.1002/asi.24730>
- Laukkanen, E., Itkonen, J., & Lassenius, C. (2017). Problems, causes and solutions when adopting continuous delivery—A systematic literature review. *Information and Software Technology*, 82, 55-79.
<https://doi.org/10.1016/j.infsof.2016.10.001>
- Mishra, A., & Otaiwi, Z. (2020). DevOps and software quality: A systematic mapping. *Computer Science Review*, 38, 100308.
<https://doi.org/10.1016/j.cosrev.2020.100308>
- Ouyang, S., Zhang, J. M., Harman, M., & Wang, M. (2025). An empirical study of the non-determinism of ChatGPT in code generation. *ACM Transactions on Software Engineering and Methodology*, 34(2), Article 42, 1-28.
<https://doi.org/10.1145/3697010>
- 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>
- Russo, D. (2024). Navigating the complexity of generative AI adoption in software engineering—RCR report. *ACM Transactions on Software Engineering and Methodology*, 33(8), Article 221, 1-5.
<https://doi.org/10.1145/3680471>
- Shahin, M., Babar, M. A., & Zhu, L. (2017). Continuous integration, delivery and deployment: A systematic review on approaches, tools, challenges and practices. *IEEE Access*, 5, 3909-3943.
<https://doi.org/10.1109/ACCESS.2017.2685629>
- Shneiderman, B. (2022). *Human-centered AI*. Oxford University Press.
<https://doi.org/10.1093/oso/9780192845290.001.0001>
- Soares, E., Sizilio, G., Santos, J., Costa, D. A., & Kulesza, U. (2022). The effects of continuous integration on software development: A systematic literature review. *Empirical Software Engineering*, 27, Article 78.
<https://doi.org/10.1007/s10664-021-10114-4>
- Treude, C., & Gerosa, M. A. (2025). How developers interact with AI: A taxonomy of human-AI collaboration in software engineering. In *Proceedings of the 2025 IEEE/ACM 2nd International Conference on AI Foundation Models and Software Engineering (FORGE)* (pp. 236-240). IEEE.
<https://doi.org/10.1109/FORGE66646.2025.00033>
- Weng, S., Feng, Y., Yin, Y., Zhang, Z., & Xu, B. (2026). Data preparation and quality for code-centric generative software engineering tasks: A systematic literature review. *Frontiers of Computer Science*, 20, Article 2009203.
<https://doi.org/10.1007/s11704-025-41376-3>