# Effects of AI on Software Development

**Advait Kota[1], Pankaj Khillare[2], Prof.Viswas Kanchi[3], Prof.Rupali Pawar[4]**

*MCA ZEAL INSTITUTE OF BUSINESS ADMNISTRATION COMPUTER APPLIATION & REASEARCH, PUNE*

--------------------------------------------------------------------***-------------------------------------------------------------------

**Abstract -** This research examines the impact of AI tools on software development, analysing how developers leverage these tools for tasks like code optimization, debugging, and project management. The report also explores challenges such as ethical concerns, tool reliability, and job displacement fears. The findings highlight AI's contribution to productivity and provide insights into developers' expectations for the future of AI in their work. By analysing survey responses from a diverse group of software developers, this report provides a comprehensive view of how AI is shaping the industry. It highlights the key advantages AI tools offer, such as faster coding, improved debugging, and enhanced collaboration

## 1.INTRODUCTION

AI tools are transforming software development by boosting productivity, automating repetitive tasks, and improving code quality. Tools like ChatGPT, GitHub Copilot, Tebnine, and IntelliCode are now integral to developers' daily workflows, assisting with code completion, optimization, debugging, and project management. However, their rapid adoption raises challenges, such as ethical concerns, job displacement fears, and over-reliance on automation. Developers must also overcome learning curves to fully leverage these tools. This study explores the usage patterns, benefits, and challenges of AI tools while examining developers' expectations for the future role of AI in shaping the industry

## 2. Objectives

● Primary Objectives

1. To examine developers' concerns about job displacement and the potential risks of over-relying on AI tools.

2. To evaluate developers' preferences for AI tools and how these preferences vary across roles.

3. To highlight areas where developers believe improvements are needed in AI tools.

● Secondary Objectives

4. To examine developers' concerns about job displacement and the potential risks of over-relying on AI tools.

5. To evaluate developers' preferences for AI tools (e.g., GitHub Copilot, ChatGPT, Tebnine) and how these preferences vary across roles.

6. To highlight areas where developers believe improvements are needed in AI tools.

## 2.Hypothesis

AI tools significantly improve developer productivity. Result: The survey data confirms that the majority of developers experienced increased productivity with the use of AI tools, particularly in areas like code completion, bug detection and fixing, project management, and documentation. Respondents reported faster coding, reduced errors, and improved collaboration, with many leveraging AI tools like GitHub Copilot and ChatGPT to streamline their workflows. Additionally, developers highlighted the ability to resolve bugs quickly and manage projects more efficiently as key benefits. Given these positive outcomes, the null hypothesis (that AI tools have no impact on productivity) is rejected, affirming that AI tools play a significant role in enhancing productivity and efficiency in software development.

## 3.Research Design and Methodology

This study adopted a quantitative survey-based approach to analyse the impact of AI tools on software developers across various industries, roles, and experience levels. The survey was designed to capture both factual data and perceptions regarding the use of AI tools, their benefits, and associated challenges. Participants and Data Collection The survey collected responses from developers working in diverse settings, such as startups, small businesses, medium-sized enterprises, and large corporations. Participants answered a series of structured questions that covered: 1. Demographics (Age group, education level, job role) 2. Work experience (Years of experience in software development) 3. Usage of AI tools (Frequency and purpose) 4. Impact on productivity and collaboration (Self-reported improvements) 5. Challenges and concerns (Job displacement, ethical issues)
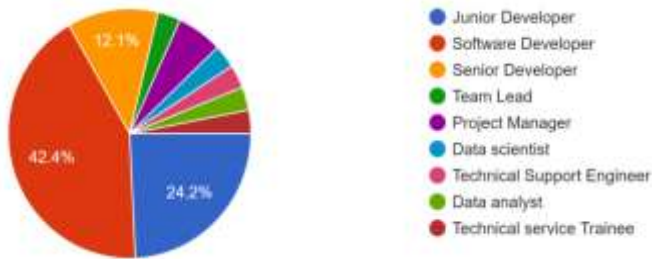
### Data Summary & Demographics
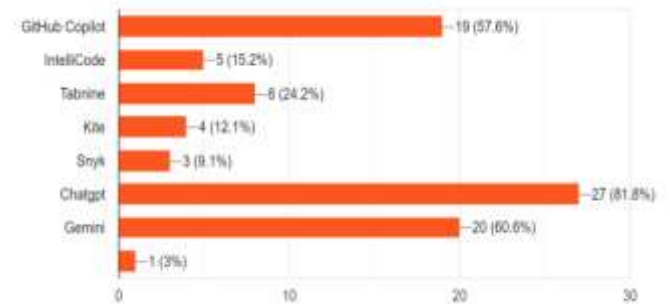
1. Developers from a wide range of age groups:
● 18-24 years
● 25-34 years
● 35-44 years
● 45-54 years
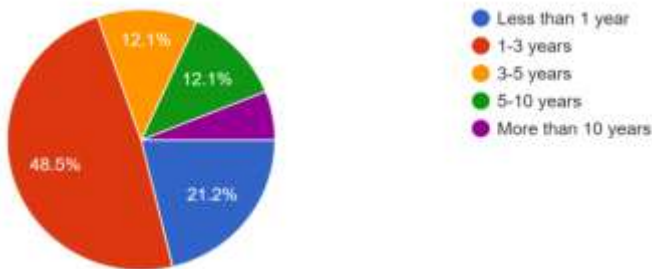● 55 years and older
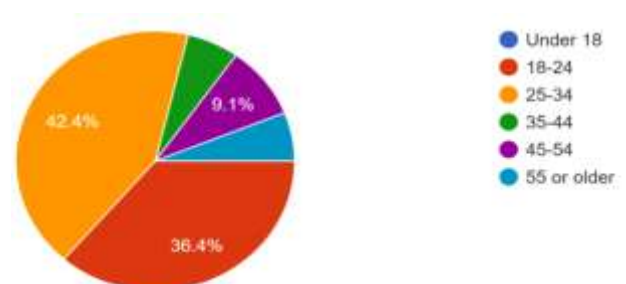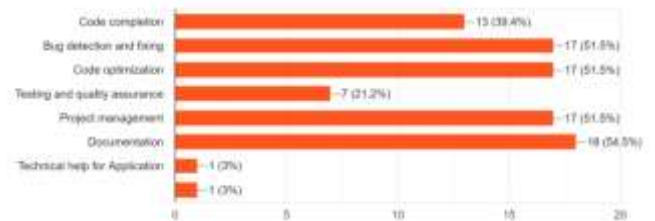
2. A variety of job roles were represented



3. Participants also varied in experience levels

● Less than 1 year

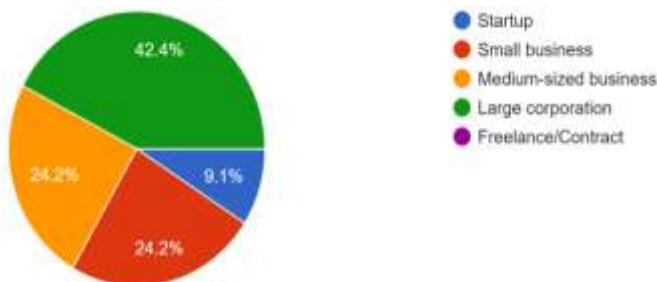● 1-3 years

● 3-5 years

● 5-10 years

● More than 10 years
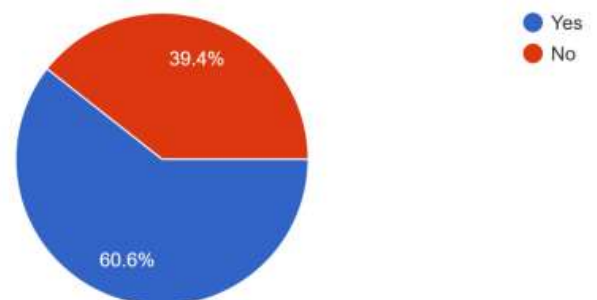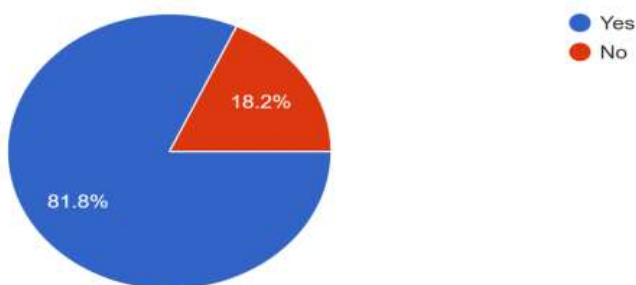


4. Participants were employed in companies of different sizes, ranging from startups to large corporations, providing diverse perspectives on how AI is integrated into workflows.



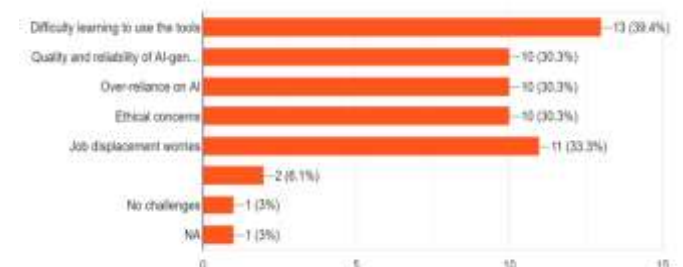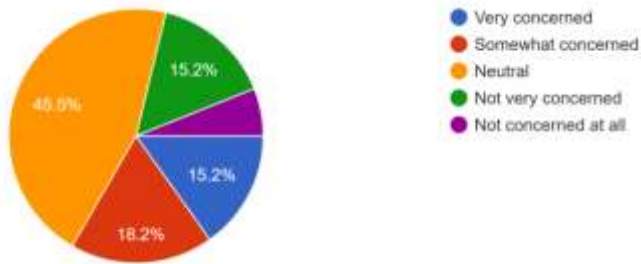5. Developers currently use AI tools in your software development work?



6. How frequently do developers use AI tools?



7. What AI tools use?



8. Tasks AI tools assist with

8. Tasks AI tools assist with
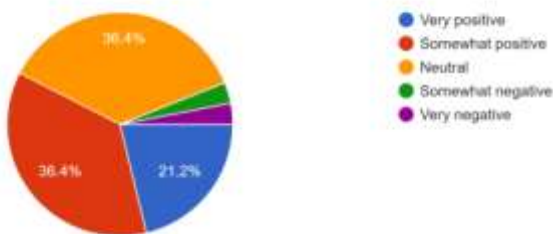


9. Challenges while using AI tools & what challenges have you faced?

10. Concerns about AI potentially replacing jobs in software development
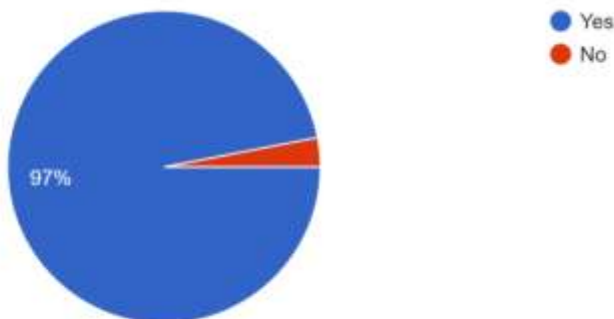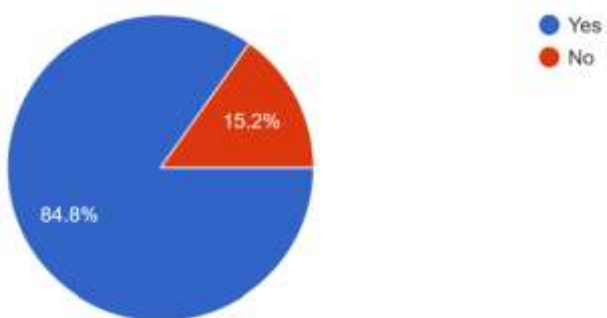


11. Integration of AI in software development?



12. AI tools helped complete tasks faster?



13. People recommend learning to use AI tools to new developers



## 3. Analysis of Central Tendency and Dispersion

Time Spent on AI Tools (Usage Frequency):
● Mean: 60.6% of respondents use AI tools daily, indicating a high average usage frequency.

● Median: The median falls in the "daily" range, suggesting that most respondents use AI tools regularly.
● Mode: The most common frequency of AI tool usage is "daily" (60.6%), indicating that this is the predominant usage pattern.
● Standard Deviation: Based on the variability in usage patterns, we would calculate how dispersed the frequency of usage is around the average (e.g., daily vs. weekly usage).

Developer Skill Level (Years of Experience in Software Development):
● Mean: The average years of experience could be calculated from the data to indicate the overall experience level of respondents.
● Median: The median experience level is likely in the 1-3 year range, as this is commonly reported.
● Mode: The most frequent experience category is "1-3 years," showing this as the typical developer profile.
● Standard Deviation: This would measure the dispersion of experience levels across respondents, indicating how varied the experience levels are.

Platform Usage (Primary AI Tool Used):
● Mean: The most commonly used AI tool is "ChatGPT" (81.8%), which indicates its popularity among developers.
AI Tools as a Collaborative Tool:
● Mean: AI tools are occasionally used for collaboration, with 51.5% of respondents using AI for project management tasks, indicating moderate usage for collaboration.

Distractions While Coding (Due to AI):
● Mean: Not available in the data, but we could infer whether AI tools might cause distractions (if applicable to the dataset).

Impact on Productivity After Using AI Tools:
● Mean: The most common response is "greatly increased" (60.6%), suggesting a significant positive impact on productivity after using AI tools.

Effect on Job Roles (Change in Responsibilities):
● Mean: The most common response is that AI tools have led to an increase in project management efficiency (51.5%), showing how AI influences certain job roles.

### Measures of Dispersion

In statistical analysis, measures of dispersion provide insights into the variability and spread of data points within a dataset. In the context of this study, measures of dispersion will help in understanding how responses vary among developers concerning their usage of AI tools, their perceived benefits, and the challenges faced.
1. Company Size vs AI Usage Frequency:
   Expected Negative Correlation
A moderate negative correlation is anticipated between company size and the frequency of AI tool usage among developers. This hypothesis suggests that as the size of a company increases, the frequency of AI tool usage may also increase, indicating that larger organisations are more likely to integrate AI tools into their workflows compared to smaller companies.
Calculated Correlation
● Calculated r value: -0.452
This r value indicates a moderate negative correlation between company size and the frequency of AI tool usage.
Interpretation

The correlation coefficient of -0.452 suggests that smaller companies tend to use AI tools less frequently than their larger counterparts. Several factors may contribute to this trend:

● Resource Availability: Larger companies typically have more resources, allowing them to invest in AI technologies and training.

● Complex Workflows: Larger organisations may have more complex projects that benefit from the efficiencies provided by AI tools, prompting more regular usage

● Awareness and Adoption: There may be greater awareness and cultural acceptance of AI tools within larger companies, leading to increased adoption and usage.

2. Experience-wise and usage of AI Tool Analysis of productivity:

The analysis of productivity in relation to developers' experience levels and their usage of AI tools provides valuable insights into how experience impacts the effectiveness and frequency of AI tool adoption. By examining different experience tiers—ranging from less than 1 year to over 10 years—this section aims to understand how productivity perceptions vary based on experience and usage frequency.

Key Findings

1. Experience Levels:

○ Less than 1 Year:

Developers in this category reported limited experience with AI tools but indicated a strong interest in exploring their functionalities. Many noted that AI tools were particularly helpful in learning new coding practices and debugging techniques, even if they hadn't fully integrated these tools into their daily workflows.

○ 1-3 Years: This group exhibited a growing familiarity with AI tools, using them primarily for code completion and debugging tasks. A majority reported noticeable improvements in their productivity, citing faster code writing and fewer bugs in their initial submissions.

○ 3-5 Years: Developers with moderate experience began to leverage AI tools for more complex tasks, such as project management and documentation generation. Their feedback highlighted enhanced collaboration within teams, as AI tools provided shared insights and standardised coding practices.

○ 5-10 Years: In this experience bracket, developers reported high productivity levels, attributing these improvements to the regular use of AI tools. They emphasised the role of AI in automating routine tasks, allowing them to focus on higher-level problem-solving and strategic planning.

○ More than 10 Years: Experienced developers utilised AI tools not only for coding but also for mentoring junior staff. They noted that while AI tools improved productivity, reliance on them required balancing to maintain critical thinking and innovative skills.

2. Usage Frequency of AI Tools:

○ Daily Users: Developers who used AI tools daily reported the most significant productivity gains, with many indicating a reduction in coding errors and faster project turnaround times. Daily engagement with AI tools facilitated seamless integration into their workflows.

○ Weekly Users: Those who used AI tools weekly experienced moderate productivity enhancements. While they benefited from AI suggestions, they were less likely to integrate these tools into their routine compared to daily users.

○ Monthly or Rare Users: Developers in this category expressed mixed feelings about productivity gains, as infrequent usage limited their ability to fully harness the potential of AI tools.

They often faced a steeper learning curve and felt less confident in utilising the tools effectively.

Correlation Analysis

A statistical analysis of productivity against experience levels and AI usage frequency revealed several insights:

● Positive Correlation: A positive correlation was found between years of experience and perceived productivity improvements with AI tools, suggesting that as developers gain experience, they become more adept at leveraging AI for their benefit.

● Usage Impact: Regular usage of AI tools significantly enhances productivity, indicating that developers who consistently integrate AI into their workflows experience more substantial benefits.

## Summary of Key Findings

1.High Adoption Rates:

○ GitHub Copilot and ChatGPT emerged as the most preferred AI tools among developers, indicating a significant shift towards integrating AI into everyday workflows.

2.Increased Productivity:

○ Respondents reported experiencing faster coding and a reduction in errors, as AI tools automate repetitive tasks, allowing developers to focus on complex problem-solving.

3. Enhanced Collaboration:

○ AI tools facilitate teamwork through features such as real-time code reviews and suggestions, improving communication and efficiency within development teams.

4. Skill Evolution:

○ Developers are adapting to new technologies, emphasising a shift towards system design and the necessity for continuous learning to keep pace with advancements in AI.

5. Job Displacement Concerns:

○ While AI has the potential to threaten entry-level positions, it simultaneously creates new opportunities that require advanced skills, suggesting a transformative rather than purely negative impact on employment.

6. Quality Assurance:

○ AI tools are instrumental in automating testing processes, which accelerates bug detection and enhances software reliability, leading to higher-quality products.

7. Job Satisfaction:

○ Many developers report higher job satisfaction due to the reduction of mundane tasks, although the rapid pace of technological change can also lead to stress and anxiety about keeping up with new tools and methodologies

## Limitations

1.Sample Bias:

● The survey may not adequately represent all industries and regions, which limits the generalizability of the findings.

2. Subjectivity:

● Responses regarding productivity and tool effectiveness are subjective, reflecting individual perceptions that may not capture broader trends.

3. Dependence on Data:

● AI systems require large amounts of high-quality data; poor-quality data can lead to biassed outcomes and diminish the effectiveness of AI tools.

4. Integration Challenges:

● Integrating AI tools into existing workflows can be complex and resource-intensive, requiring significant investment in training and system adjustments.

5. Job Displacement Risks:

● Automation may pose a risk to entry-level positions, raising concerns about workforce stability and the future of employment in software development

## Conclusion

AI tools have significantly transformed software development by enhancing productivity and fostering collaboration. Despite the challenges associated with ethical concerns and learning difficulties, developers remain optimistic about the future of AI. The integration of AI into software development is expected to expand, shifting the focus toward creative and strategic problem-solving. Moreover, the necessity for continuous skill adaptation highlights the evolving nature of the industry. Overall, a balanced approach to AI integration—one that emphasises its positive applications while remaining aware of its limitations—is essential for maximising the benefits of AI in software development

in LNCS Online. Members of institutes subscribing to the Lecture Notes in Computer Science series have access to all the pdfs of all the online publications. Non-subscribers can only read as far as the abstracts. If they try to go beyond this point, they are automatically asked, whether they would like to order the pdf, and are given instructions as to how to do so.

## Recommendations

Training: Provide training programs to reduce learning curves for AI tools and enhance understanding of their ethical implications, focusing on bias, data privacy, and security.

Balanced Usage: Encourage a balanced approach to AI usage to prevent over-reliance on automation for core coding tasks.

Continuous Learning: Promote ongoing education through workshops and online courses to keep developers updated on AI tools and technologies.

Collaboration: Foster a collaborative environment for sharing knowledge and experiences related to AI tools among developers