

The Impact of Generative AI-Powered Code Generation Tools on Software Engineer Hiring: Recruiters' Experiences, Perceptions, and Strategies

Alyssia Chen
University of Hawai'i at Mānoa
abc8@hawaii.edu

Timothy Huo
University of Hawai'i at Mānoa
thuo@hawaii.edu

Yunhee Nam
University of Hawai'i at Mānoa
yunheen@hawaii.edu

Dan Port
University of Hawai'i at Mānoa
dport@hawaii.edu

Anthony Peruma
University of Hawai'i at Mānoa
peruma@hawaii.edu

Abstract

The rapid advancements in Generative AI (GenAI) tools, such as ChatGPT and GitHub Copilot, are transforming software engineering by automating code generation tasks. While these tools improve developer productivity, they also present challenges for organizations and hiring professionals in evaluating software engineering candidates' true abilities and potential. Although there is existing research on these tools in both industry and academia, there is a lack of research on how these tools specifically affect the hiring process. Therefore, this study aims to explore recruiters' experiences and perceptions regarding GenAI-powered code generation tools, as well as their challenges and strategies for evaluating candidates. Findings from our survey of 32 industry professionals indicate that although most participants are familiar with such tools, the majority of organizations have not adjusted their candidate evaluation methods to account for candidates' use/knowledge of these tools. There are mixed opinions on whether candidates should be allowed to use these tools during interviews, with many participants valuing candidates who can effectively demonstrate their skills in using these tools. Additionally, most participants believe that it is important to incorporate GenAI-powered code generation tools into computer science curricula and mention the key risks and benefits of doing so.

Keywords: practitioner survey, code generation, recruiter, hiring in computing, generative AI

1. Introduction

Recent advancements in Artificial Intelligence (AI) research have resulted in significant progress

in Generative AI (GenAI) tools and techniques. These developments are significantly impacting various industries by automating processes that previously required human intervention, thereby streamlining operations and improving efficiencies (Singla et al., 2024). One such industry where the impact of GenAI is particularly noteworthy is the software engineering industry. With their ability to generate code and provide intelligent auto-completion, GenAI tools, such as ChatGPT¹ and GitHub Copilot² are revolutionizing the software industry by changing the way developers write code (Ebert and Louridas, 2023). These GenAI-powered code generation tools enhance developer productivity by automating repetitive tasks, identifying bugs or poor-quality code, and providing solutions (Liang et al., 2024; Sauvola et al., 2024). However, there are also drawbacks to consider, such as hallucinations, ethical and legal concerns related to using GenAI tools to generate code, potential biases in the generated code, and the risk of overreliance, which could lead to a decline in coding skills and a lack of understanding of underlying concepts (Bull and Kharrufa, 2024; Fan et al., 2023; Russo, 2024).

As these GenAI-powered code generation tools become more prevalent and widely accessible, it is crucial to understand how they influence the skills and qualifications required for software engineers and how organizations and hiring professionals have had to adapt their evaluation approaches. For instance, evaluating a candidate's programming, problem-solving, innovation, and creativity skills becomes challenging if they rely heavily on GenAI-powered code generation tools.

¹<https://chat.openai.com>

²<https://github.com/features/copilot>

1.1. Goal & Research Questions

This shift towards AI-augmented software development (Ozkaya, 2023) necessitates a reevaluation of traditional evaluation techniques to ensure that organizations are able to accurately assess a software engineering candidate's true abilities and potential. However, given the recent emergence of these tools, there is a lack of research on how the hiring process should evolve to assess candidates who have experience utilizing such tools. Therefore the goal of this study is to *investigate how organizations and hiring professionals have adapted their hiring strategies and policies to account for GenAI-powered code generation tools, and to understand their perceptions of such tools*. Hence, we answer the following research questions (RQs):

RQ1: How familiar are recruiters with GenAI code generation tools? This RQ aims to understand recruiters' awareness of the latest advancements in AI code generation tools and ascertain if they are prepared to evaluate candidates who utilize these tools.

RQ2: How are organizations and recruiters adapting their candidate screening and skills evaluation processes in response to GenAI code generation tools? As AI code generation tools become more widespread, this RQ explores how organizations and recruiters are adapting their evaluation strategies to ensure fair and accurate assessments and the factors they consider when making hiring decisions.

RQ3: To what extent is a candidate's experience in using GenAI code generation tools helpful in evaluating them? This RQ investigates the value recruiters place on a candidate's knowledge and experience with AI code generation tools to gain insights into changing skill requirements and the importance of proficiency with these tools in the hiring process.

RQ4: What is the perception within the software engineering industry on the importance of integrating GenAI code generation tools into the computer science (or related) curricula? As AI code generation tools become more prevalent in the software engineering industry, this RQ aims to gather insights on how academia should adapt to ensure students are prepared for the evolving demands of the software engineering industry.

1.2. Contribution

To answer our RQs, we surveyed 32 industry professionals who are involved in recruiting software engineers to understand how they evaluate candidates in the era of GenAI-powered code generation tools. We envision our findings providing valuable insights into

the changing landscape of software engineer recruitment and offering practical recommendations for employers, candidates, and academia to navigate the challenges and opportunities presented by such GenAI-powered tools.

2. Related Work

Existing research has studied the software engineering industry's hiring process, including candidate skills, resume evaluation, and the hiring process from the student perspective. However, limited research exists on how AI code generation tools have impacted the hiring process and candidate evaluation.

Singh and Singh (2021) investigated whether computer-science-oriented, engineering graduate students' knowledge in reliability and safety engineering were at the level industry professionals expected. To do so, both students and practitioners were interviewed in the study, and their analysis revealed that most of the graduate students' knowledge was not up to expectations. Petersheim et al. (2023) conducted a study with both recruiters and computer science students, and both groups were given entry-level computer science resumes to review. It was found that recruiters and students prioritized certain resume items differently and spent varying amounts of time in sections of the resume. Garousi et al. (2019) conducted a meta-analysis of 35 studies to identify the most important skills needed by software engineering graduates and found "requirements, design and testing" to be the most critical and found "configuration management, models and process" to be the knowledge graduates did not know enough about.

Chinn and Vandegrift (2008) examined students' responses to an in-class assignment requiring them to evaluate and hire fictitious candidates. This allowed the authors to identify the hiring criteria that students prioritized, which included technical skills, soft skills, personal traits, previous employment, and career considerations. Lunn et al. (2024) focused on the whole hiring process itself from a student's point of view and discovered how students prepare and get support, with emphasis on the inclusivity of the process. Adnin et al. (2022) examined the hiring process from an employer's perspective, and through interviews, they discovered sources of bias and significant preferences for strong technical skills. Stepanova et al. (2021) analyzed the survey responses of more than 200 recruiters in the US to learn more about the hiring process and recruiters' hiring choices. Alekseeva et al. (2021) found a significant increase in the demand for AI skills in most industries and occupations in the U.S. labor market from 2010 to 2019. A practitioner survey by Akdur

(2022) identified the most in-demand skills for software engineering. Scaffidi (2018) interviewed employers to understand the soft skills and personal attributes new graduates need to impress recruiters.

3. Method

This section presents our survey design, participant recruitment, and response analysis methodology. As our study involves working with human subjects, we obtained approval from our institution's Institutional Review Board of the Office of Research Compliance.

3.1. Survey Design

We utilized Qualtrics to design and host the survey and set it up to permit only one response per participant. Our survey comprised 24 questions aimed at collecting data on participants' demographics, their experiences with and perceptions of AI code generation tools for candidate evaluation, and their perspectives on integrating such tools into academic curricula. These questions were developed based on the objectives of our study, as discussed in Section 1.1, and our review of relevant literature, as discussed in Section 2. Finally, as best practice, we conducted a pilot run with a couple practitioners before publishing the survey to identify potential issues with the questionnaire (Linåker et al., 2015). Based on their feedback, we reworded some questions to improve clarity and re-ordered a few. Table 1 shows the published survey questions, the question type, whether a response is required, and any logic/notes applicable to each question. The complete questionnaire and survey responses are included in our artifact package (Chen et al., 2024).

3.2. Survey Participants

To ensure the reliability of the responses, we made sure that survey participants were involved in the recruitment process for software engineers or similar positions. To accomplish this, we contacted representatives of companies that took part in the Spring 2024 Career Fair hosted by the Information and Computer Sciences Department at the University of Hawai'i at Mānoa. The fair had 30 organizations from various sectors, with an average of two employees from each. Around 174 computer science students attended, looking for technology sector jobs.

To recruit participants, the authors met with industry representatives at the Career Fair. After confirming that they were involved in the hiring of software engineers, the representatives were invited to participate in the survey. They were provided with a QR code to access

the online survey, or offered a laptop or tablet to complete the survey. Additionally, some participants also referred colleagues and friends, who were then emailed the survey link. This approach of convenience and snowball sampling enabled us to identify and invite individuals with the necessary qualifications required for our study. Before starting the survey, participants had to agree to an informed consent document. Participation was voluntary, and no compensation was offered.

3.3. Data Analysis

We analyzed the survey data using both quantitative and qualitative methods (Wagner et al., 2020). Our quantitative analysis involved standard statistical techniques, while for qualitative analysis, we reviewed participants' open-ended responses to identify common themes. To ensure reliability, three authors independently reviewed and categorized the responses, discussed any discrepancies, and reached a consensus. We elaborate on the specific techniques when answering our RQs.

4. Results

This section presents our RQ results. We first report the number of responses received and participant demographics before answering our RQs. It should be noted that the responses received from participants in the pilot run are not in the RQ results.

Survey Responses

The survey was available to the public from February 2024 to May 2024, and we gathered 39 responses within this period. However, not all participants responded to all the questions. To maintain consistency in our analysis, we only included participants who answered all the required questions, leading to 32 valid responses. Hence, our results focus solely on these 32 responses.

Participant Demographics

We divide our reporting of participants' demographics (survey questions #1 to #8) into two parts: general demographics and work experience.

General Demographic: Among the 32 participants, 17 (53.13%) identified as male, 14 (43.75%) as female, and 1 (3.13 %) preferred not to disclose. In terms of age, 2 (6.25%) were between the age range of 18-24, 12 (37.5%) between 25-34, 9 (28.13%) between 35-44, 3 (9.38%) between 45-54, and 6 (18.75%) between 55-64.

Work Experience: For years of industry experience, almost half (46.88%) of participants reported more than

Table 1: Below are the questions that are part of the survey. The questionnaire including the answer options for the single-choice and multi-choice questions, is available in our artifact package Chen et al., 2024.

No.	Question	Type	Required	Notes
1	What is your gender?	Single-Choice	Yes	Includes "Other" free-text option
2	What is your age?	Single-Choice	Yes	
3	How many years of industry experience do you have?	Single-Choice	Yes	
4	Which job category closely matches your current position?	Single-Choice	Yes	Includes "Other" free-text option
5	Where are you employed?	Single-Choice	Yes	Includes "Other" free-text option
6	How many employees work in your establishment?	Single-Choice	Yes	
7	Which of the following industries most closely matches the one in which you are employed?	Single-Choice	Yes	Includes "Other" free-text option
8	What is your role in the hiring process?	Free Text	Yes	
9	Which of these AI code generation tools are you familiar with?	Multi-Choice	Yes	Includes "Other" free-text option
10	How frequently do you use an AI code generation tool for a work project or task?	Single-Choice	Yes	Show only if anything but "None" in Q9 was selected
11	With the increasing prevalence of AI code-generation tools, has your organization developed official guidelines or policies related to these specific tools when evaluating candidates?	Single-Choice	Yes	
12	If possible, can you provide a summary or key points of the guidelines/policies to evaluate candidates with regard to AI code generation tools?	Free Text	No	Shown only if "Yes" is selected in Q11
13	Have you changed the criteria you utilize to evaluate a candidate's coding skills due to the wide availability of AI code-generation skills?	Single-Choice	Yes	Shown only if "No" or "Not Sure" is selected in Q11
14	If possible, can you provide a summary or key points of your candidate code evaluation criteria?	Free Text	No	Shown only if "Yes" is selected in Q13
15	Have you permitted candidates to utilize AI code generation tools in a technical interview?	Single-Choice	Yes	
16	Should candidates be allowed to utilize AI code generation tools during technical interviews?	Single-Choice	Yes	
17	Please let us know why you selected this answer option.	Free Text	Yes	Shown only if "Yes" or "No" is selected in Q16
18	To what extent do you believe AI code generation tools pose challenges in accurately assessing a candidate's coding proficiency, analytical skills, and overall experience?	Single-Choice	Yes	
19	How frequently do you directly ask a candidate about their personal experience and skills in using AI code generation tools?	Single-Choice	Yes	
20	How much of a preference do you give to candidates who demonstrate skills in using AI code generation tools?	Single-Choice	Yes	
21	From your perspective, what critical skills should candidates possess to utilize AI code generation tools effectively?	Free Text	Yes	
22	How important is it for universities to integrate education on AI code generation tools into their curricula?	Single-Choice	Yes	
23	What, if any, benefits do you foresee in integrating AI code generation tools into academic courses?	Free Text	No	
24	What, if any, risks or concerns do you foresee in integrating AI code generation tools into academic courses?	Free Text	No	

15 years, 2 (6.25%) report 11-15 years, 6 (18.75%) report 6-10 years, 4 (12.5%) report 3-5 years, and 5 (15.63%) report 1-2 years. When asked which job category they closely match with, 13 (40.63%) reported “HR/Talent Acquisition/Hiring Manager/ Or Similar”, 5 reported “Software Engineering/Software Architect/ Or Similar”, 7 reported “Team Lead/Project Manager/Account Manager/ Or Similar”, and another 7 reported “Other”. Those who report “Other” listed other specific technical and communication roles such as Cybersecurity, Communications & Marketing, and Health Information Technology. Lastly, 12 (37.5%) participants are employed in the private-for-profit sector, 2 (6.25%) in the private-not-for-profit sector, 1 (3.13%) in local government, 11 (34.38%) in state government, and 6 (18.75%) in federal government.

The demographic details highlight that, despite the relatively small sample size of 32 participants, they represent various industries and possess extensive experience within their respective fields. As such, their feedback offers valuable insights for our study.

RQ1: How familiar are recruiters with GenAI code generation tools?

For this RQ, we analyze responses from survey questions #9 and #10 regarding their familiarity and usage of AI code generation tools.

Table 2: Familiarity with AI code generation tools.

Answer Options	Count	Percentage
ChatGPT	28	60.87%
Github Copilot	8	17.39%
Amazon CodeWhisperer	2	4.35%
Replit AI	2	4.35%
Other	2	4.35%

Survey question #9 is a multi-choice question with an “Other” free text option, aiming to gauge recruiters’ familiarity with AI code generation tools. As shown in Table 2, the AI code generation tools participants were familiar with include “ChatGPT” (28 or 60.87%), “Github Copilot” (8 or 17.39%), “Amazon CodeWhisperer” (2 or 4.35%), “Replit AI” (2 or 4.35%), and “Other” (2 or 4.35%). The choices under “Other” were “Bard (Gemini)” and “Pilot AI”. Additionally, 4 (8.70%) participants reported they were not familiar with AI code generation tools. As this is a multi-choice question, respondents had an average of 1.31 answer options chosen, with the most common combination being “ChatGPT” and “Github Copilot” selected eight times. The next most frequent combinations were

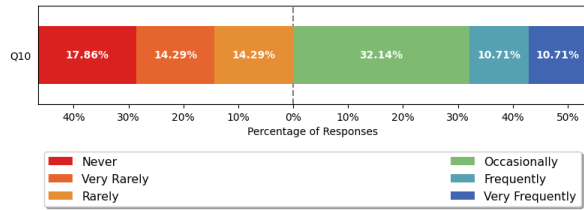


Figure 1: Frequency of AI code generation tool usage for work

“ChatGPT” and “Amazon CodeWhisperer”, “ChatGPT” and “Replit AI”, and “Github Copilot” and “Replit AI”, each occurring twice.

Survey question #10 is a Likert (i.e., single-choice) question examining the frequency with which recruiters use AI code generation tools for work projects and tasks. As shown in Figure 1, more than half of the participants reported using AI code generation tools, with 3 (10.71%) selecting “Very Frequently”, another 3 (10.71%) selecting “Frequently”, and 9 (32.14%) selecting “Occasionally”. In contrast, 4 (14.28%) report “Rarely”, another 4 (14.28%) report “Very Rarely”, and 5 (17.86%) report “Never”.

Summary for RQ1. ChatGPT stood out as the most familiar AI code generation tool, with recognition at 60.87%, but recognition significantly dropped for other tools, followed by Github Copilot at 17.39%. Moreover, a majority (53.57%) of respondents indicated occasional to very frequent usage of AI code generation tools for work. Conversely, 28.57% reported rarely or very rarely using such tools, while another 17.86% reported never using them.

RQ2: How are organizations and recruiters adapting their candidate screening and skills evaluation processes in response to GenAI code generation tools?

This RQ has two parts. The first examines how organizations approach AI code generation tools, while the second examines the evaluator’s experience and perception of such tools when evaluating candidates.

Organizational Policy:

Through survey questions #11 and #12, we gain insight into how organizations have adapted to these tools when evaluating candidates. Survey question #11 is a single-choice question designed to determine if organizations are adjusting to the emergence of AI code generation tools when hiring software engineers. We found that 5 (15.63%) participants selected “Yes”,

21(65.63%) selected “No”, and 6(18.75%) selected “Not Sure”. Participants who selected “Yes” were then asked to provide a summary of their organization’s guidelines/policies regarding the use of AI code generation tools in candidate evaluation. Only one participant responded, cautioning against the use of AI.

Recruiter Perception & Experience:

Next, we examine how individual recruiters evaluate candidates regarding AI code generation tools. First, from survey question #13, 4 participants (14.81%) reported that they have changed the criteria they use to evaluate a candidate’s coding skills, 17 (62.96%) reported that they have not, and 6 (22.22%) selected ‘Not applicable.’. Participants who said they had changed their evaluation criteria were prompted with an optional survey question, #14, and asked, if possible, to provide a summary of their evaluation criteria. There were only three responses to this question, which we reviewed and identified the following two main categories:

- **Standardized Evaluation Criteria:** One participant mentioned they follow a set of predefined and uniform standards to evaluate candidate performance (e.g., “Federal government uses KSAs”).
- **Verbal Discussion:** Two participants mention they rely on verbal techniques to evaluate candidates’ understanding of coding and problem-solving skills (e.g., “We rely more on verbal interviews to discuss methodology and general approaches to tasks and problems rather than coding or written exercises”).

Moving on, survey question #15 asks participants whether they allow candidates to utilize AI code generation tools in technical interviews. Only 1 (3.13%) participant selected “Yes”, while 20 (62.50%) selected “No”, and 11 (34.38%) selected “Not applicable”. Next, in survey question #16, we ask participants if candidates should be allowed to utilize AI code generation tools during technical interviews. The results show that 5 participants (25%) report “Yes”, 7 participants (35%) report “No”, and 8 participants (40%) report “Not Sure”.

In survey question #17, participants who reported “Yes” and “No” in question #16 were asked to explain their choices.

Examining the free text responses for the “Yes” option, we identified the below three categories:

- **AI is a tool:** Three participants see AI code generation tools as another resource for software engineers. They believe candidates should be permitted to use these tools in interviews as they are widely available and prevalent. (e.g., “AI is another tool that can be used by the applicant”).
- **Promotes Fairness:** One participant mentioned that allowing AI code generation tools in technical

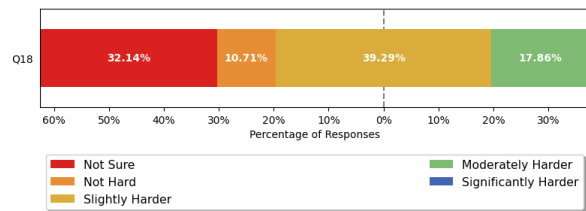


Figure 2: Perceived challenges in assessing candidate skills with AI code generation tools

interviews is a fair approach and unbiased.

- **Demonstrates Proficiency of Current Technologies:** One participant mentions AI enables candidates to showcase their familiarity with the most recent technologies during the interview (e.g., “It allows candidates to display their knowledge of the most current technologies out there”).

For responses associated with “No”, we identified the following three main categories:

- **Challenge Evaluating Candidates True Technical Skills:** Four participants mention that they want to evaluate a candidate’s knowledge and skill without the assistance of AI tools. These tools hinder an evaluator’s ability to determine if candidates possess the necessary expertise to perform their roles independently (e.g., “Certain roles need people to know the information off the top of their head without the use of AI”).
- **Concerns Over Quality of AI-Generated Code:** One participant expressed doubts about the accuracy and reliability of AI-generated code. (e.g., “Because being able to code without AI is a skill necessary to determine AI hallucinations”).
- **Challenge Evaluating Candidates Cognitive Skills:** Two participants highlighted the importance of assessing candidates’ critical thinking and problem-solving skills during technical interviews. AI code generation tools may hinder the evaluation of these crucial abilities (e.g., “Our technical interviews isn’t about writing a lot of code, its about your thought process to solve the problem”).

Lastly, survey question #18 examines the challenge AI code generation tools have in assessing a candidate’s coding proficiency, analytical skills, and overall experience. According to Figure 2, excluding 4 participants who chose “Not Applicable”, half of the participants report these tools will pose more of a challenge, with 5 (17.86%) selecting “Moderately Harder” and 11 (39.29%) selecting “Slightly Harder”. In contrast, only 3 (10.71%) selected “Not Hard” and 9 (32.14%) report “Not Sure”.

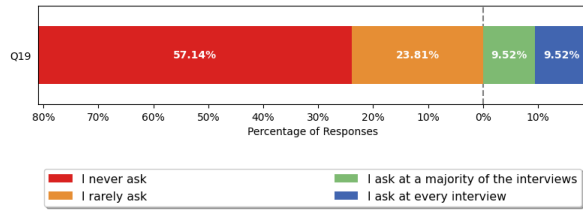


Figure 3: Frequency of directly asking a candidate about their personal experience and skills in using AI code generation tools

Summary for RQ2. Most organizations and recruiters have not adapted their candidate evaluation techniques to account for AI code generation tools. For those that have, one common means to evaluate candidates is a verbal discussion about their code. Opinions on allowing candidates to use AI code generation tools during interviews were divided. Those in favor highlighted AI tools as additional resources, promoting fairness and showcasing candidates' familiarity with modern technology. Those against emphasized the importance of assessing candidates' genuine skills without AI assistance and prioritizing problem-solving approaches over extensive code writing during interviews. Lastly, participants responded that assessing candidates' skills with AI code generation tools as moderately or slightly more challenging, with only a minority expressing confidence.

RQ3: To what extent is a candidate's experience in using GenAI code generation tools helpful in evaluating them?

To answer this RQ, we analyzed participants' responses to question #19 to determine how frequently they inquire about candidates' experience and skills in AI code generation tools. Subsequently, with survey question #20, we investigated whether any preference is shown towards candidates demonstrating expertise in AI code generation tools. Lastly, in survey question #21, we explored their opinions on the critical skills that candidates should possess to effectively utilize AI code generation tools.

As shown in Figure 3, excluding the 11 participants who wrote "Not Applicable", 17 (53.13%) of the participants never or rarely asked candidates about their experience in AI code generation tools, and 2 (6.25%) asked at every interview. In Figure 4, excluding the 10 who answered "Not Applicable," more than 50% have a moderate or strong preference for candidates who demonstrate AI code generation skills.

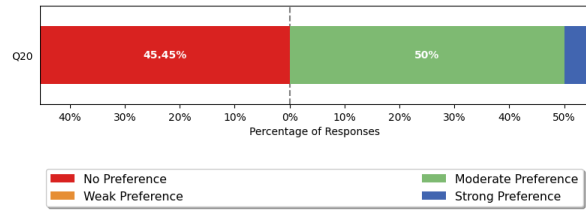


Figure 4: Extent of preference given to candidates who demonstrate skills in using AI code generation tools

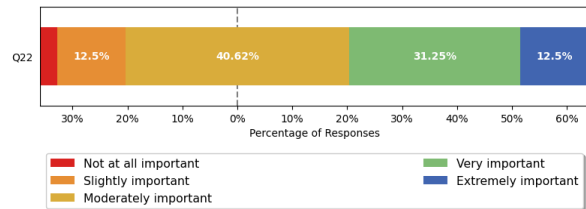


Figure 5: Importance of integrating AI code generation tools into computer science curricula.

Next, when asked to describe the critical skills candidates should have to effectively use AI code generation tools (survey question #21), we identified five categories:

- Critical Thinking & Problem-Solving Skills:** Five participants mentioned the importance of candidates being able to analyze problems, understand fundamental concepts, and general problem-solving skills (e.g., "*Critical Thinking and Problem Solving Skills*").
- Prompt Engineering:** Five participants highlighted that candidates should be able to ask the right questions and provide clear, context-rich prompts to the AI tool to obtain the most relevant and accurate results (e.g., "*The ability to properly prompt and provide accurate and detailed context to the tool to get the best possible outcomes*").
- Evaluate AI Response:** Six participants reported that knowing the limitations and shortcomings of responses from AI code generation tools is an essential skill that candidates should possess (e.g., "*They need to be able to tell when the generated code is not just correct technically but achieves the requirements of the software*").
- Familiarity of AI Tools:** Two participants mentioned the importance of knowing how to effectively use the tools/technologies as an important skill for candidates (e.g., "*Should be able to navigate platforms and provide demonstration to audiences*").
- Fundamental Computer Science Knowledge:** One participant mentioned the importance of core computer science courses (e.g., "*applied statistics,*

data structures”).

Two participants did not answer the question, and 11 stated that they were unsure about the required skills.

Summary for RQ3. Most participants never inquire about candidates’ AI code generation tool experience during interviews. However, around half of the participants give candidates a moderate to strong preference if they demonstrate skills in using these tools. Key skills deemed necessary for effective utilization of these tools include critical thinking and problem-solving, prompt engineering, evaluating AI outputs, familiarity with the tools, and fundamental computer science knowledge. Additionally, some participants were uncertain about the necessary skills.

RQ4: What is the perception within the software engineering industry on the importance of integrating GenAI code generation tools into the computer science (or related) curricula?

In this RQ, participants were asked about the importance of integrating AI code generation tools into computer science curricula (question #22) and to outline the benefits (question #23) and risks (question #24) they foresee in doing so.

In terms of importance, as shown in Figure 5, only a small percentage believe it is not at all important to have educational institutes integrate AI code generation tools into the course curricula, and almost 45% believe it is very important or extremely important.

We next report on the responses participants provided for the benefits and risks survey question. Three of the authors reviewed the 32 free-text responses to identify common themes. Below we describe the categories our analysis yields:

Benefits:

- **Improved Productivity:** Five participants mentioned that AI code generation tools can help students and educators save time on repetitive tasks, such as writing boilerplate code and grading. Instead, they can have more time to focus on problem-solving skills or other high-level concepts (e.g., *“More time to focus on unique problem sets and overall quality”*).
- **Equitable Access:** Four participants touched on that AI code generation tools have the potential to support students who may not have had prior exposure to programming or those who are transitioning from other fields (e.g., *“Easier time obtaining knowledge before going into the field”*).
- **General Awareness & Preparedness:** Ten participants mentioned that incorporating AI code

generation tools into courses can potentially help students stay ahead of the curve, better prepare for the future of work, and enhance their understanding of how AI can be utilized in various contexts (e.g., *“AI is inevitable, if anything, it will be developed so it should be incorporated into a academic courses”*).

- **Industry Relevance and Preparedness:** Three participants specifically mentioned how exposure to AI code generation tools in academia will better prepare students for the real world, where they will encounter such tools (e.g., *“College should be teaching students real world applications. Using AI code generation tools is a real world use case”*).

Ten participants either did not provide a response or stated that they were unsure about the benefits.

Risks:

- **Academic Dishonesty:** Six participants had concerns about academic integrity, including instances of cheating and plagiarism due to the use of these tools (e.g., *“Potential for cheating and plagiarism”*).
- **Challenges for Educators:** Four participants mentioned that these tools will negatively impact educators as they would need to spend time restructuring their courses and face difficulty in evaluating students’ skills (e.g., *“Basically professors are going to have a harder time determining what students actually learned”*).
- **Student Overreliance:** Six participants highlight the risk of students becoming overly reliant on these tools, potentially lacking essential computing and problem-solving skills. (e.g., *“Dependency on AI for all tasks resulting in lack of problem solving skills”*).
- **Lack of Responsible Use:** Four participants were concerned about the potential misuse of these tools (e.g., *“Responsible use and malicious code”*).

Three participants indicated that there are no risks, while nine either did not provide a response or stated that they were unsure about the risks.

Summary for RQ4. Most participants are of the opinion that integrating education on AI code generation tools into computer science curricula is important. The benefits identified include improved productivity, equitable access, general awareness and preparedness, as well as industry relevance and preparedness. However, there are also perceived risks such as academic dishonesty, challenges for educators, student overreliance, and lack of responsible use. Some participants were unsure or did not provide a response regarding both the benefits and risks.

5. Discussion & Takeaways

With the recent emergence of GenAI-based code generation tools and the limited research on their impact on hiring software engineers, our exploratory study offers valuable insights that can benefit various stakeholders, including practitioners, educators, researchers, and students.

Our RQ1 findings indicate that most practitioners are familiar with and have experience in using these tools, especially ChatGPT. However, the relative newness of these technologies presents a challenge for organizations and recruiters in effectively evaluating candidates' true skills and abilities, considering the potential influence of these tools, as revealed by our RQ2 results. Further, the lack of industry standards and guidelines on assessing candidates' actual capabilities in light of the increasing use of AI code generation tools makes it harder for recruiters to make informed decisions. Moving on, insights from RQ3 highlight the importance practitioners place on a candidate's cognitive skills, including critically analyzing the tool's output (Wang et al., 2024). Finally, based on RQ4, it is encouraging to see that practitioners, educators, and students share similar views on the risks and benefits of these tools (Sheard et al., 2024; Zastudil et al., 2023).

Takeaways

- There is a need for *close collaboration between industry and academia to develop curricula that effectively prepare students for the AI-augmented future of software engineering*. This collaboration should focus on designing learning experiences that go beyond simply introducing and teaching students about GenAI tools. It's important for students to understand the challenges of GenAI tools and work on assignments resembling real-world software engineering scenarios to develop the skills needed to use GenAI tools effectively.
- It is crucial for *organizations to continually update their candidate hiring and evaluation techniques to effectively gauge candidates' skills and abilities in the current AI-augmented era*. This includes training their hiring team to understand the capabilities, limitations, and potential impact of GenAI tools on the software development process. Organizations should also focus on evaluating candidates' cognitive abilities, such as their ability to engage in chain-of-thought reasoning, prompt engineering, and code review skills, among others.

6. Threats To Validity

The generalizability of the findings may be affected by factors such as the number of participants, geographical location, and self-selection bias. However, as per the demographic data, these participants come from various industries and have been verified to be involved in hiring software engineers. Additionally, the anonymous nature of our survey encouraged participants to provide honest and unbiased responses. Although the study includes open-ended questions, the depth of qualitative data may be limited. Conducting in-depth interviews or focus groups could provide more comprehensive insights. Moreover, there is a possibility that participants may have interpreted the survey questions differently, resulting in variations in responses. Lastly, given the rapidly evolving nature of the AI field, the survey responses offer valuable insights at this specific point in time, serving as a platform for future studies to examine the evolving hiring practices.

7. Conclusion & Future Work

This study provides early insights into the extent to which organizations and recruiters are adapting their evaluation techniques to assess the true abilities of candidates and their perceptions of these tools. Furthermore, our findings show a consensus among industry professionals on the importance of integrating AI code generation tools into university curricula, recognizing the need to prepare students for the AI-augmented future of software engineering. Our future work involves conducting in-depth interviews and focus groups with recruiters, educators, and software engineers to gain a more comprehensive understanding.

References

- Adnin, R., Afroz, S., Ulfat, M., & Iqbal, A. (2022). A hiring story: Experiences of employers in hiring cs graduates in software startups. *Companion Publication of the 2022 Conference on Computer Supported Cooperative Work and Social Computing*, 126–129.
- Akdur, D. (2022). Analysis of software engineering skills gap in the industry. *ACM Trans. Comput. Educ.*, 23(1). <https://doi.org/10.1145/3567837>
- Alekseeva, L., Azar, J., Gine, M., Samila, S., & Taska, B. (2021). The demand for ai skills in the labor market. *Labour economics*, 71, 102002.
- Bull, C., & Kharrufa, A. (2024). Generative artificial intelligence assistants in software development education: A vision for integrating generative

- artificial intelligence into educational practice, not instinctively defending against it. *IEEE Software*, 41(2). <https://doi.org/10.1109/MS.2023.3300574>
- Chen, A., Huo, T., Nam, Y., Port, D., & Peruma, A. (2024). Project Artifacts. <https://doi.org/10.5281/zenodo.13623897>
- Chinn, D., & Vandegrift, T. (2008). Uncovering student values for hiring in the software industry. *J. Educ. Resour. Comput.* <https://doi.org/10.1145/1316450.1316454>
- Ebert, C., & Louridas, P. (2023). Generative ai for software practitioners. *IEEE Software*. <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. *2023 IEEE/ACM International Conference on Software Engineering: Future of Software Engineering (ICSE-FoSE)*, 31–53. <https://doi.org/10.1109/ICSE-FoSE59343.2023.00008>
- Garousi, V., Giray, G., Tüzün, E., Catal, C., & Felderer, M. (2019). Aligning software engineering education with industrial needs: A meta-analysis. *Journal of Systems and Software*, 156, 65–83.
- Liang, J. T., Yang, C., & Myers, B. A. (2024). A large-scale survey on the usability of ai programming assistants: Successes and challenges. *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*. <https://doi.org/10.1145/3597503.3608128>
- Linåker, J., Sulaman, S. M., de Mello, R. M., & Höst, M. (2015). Guidelines for conducting surveys in software engineering.
- Lunn, S. J., Zerbe, E., & Ross, M. (2024). You're hired! a phenomenographic study of undergraduate students' pathways to job attainment in computing. *ACM Trans. Comput. Educ.* <https://doi.org/10.1145/3636514>
- Ozkaya, I. (2023). The next frontier in software development: Ai-augmented software development processes. *IEEE Software*. <https://doi.org/10.1109/MS.2023.3278056>
- Petersheim, C., Lahey, J., Cherian, J., Pina, A., Alexander, G., & Hammond, T. (2023). Comparing student and recruiter evaluations of computer science resumes. *IEEE Transactions on Education*, 66(2), 130–138. <https://doi.org/10.1109/TE.2022.3199685>
- Russo, D. (2024). Navigating the complexity of generative ai adoption in software engineering. *ACM Trans. Softw. Eng. Methodol.* <https://doi.org/10.1145/3652154>
- Sauvola, J., Tarkoma, S., Klemettinen, M., Riekkki, J., & Doermann, D. (2024). Future of software development with generative ai. *Automated Software Engineering*, 31. <https://doi.org/10.1007/s10515-024-00426-z>
- Scaffidi, C. (2018). Employers' needs for computer science, information technology and software engineering skills among new graduates. *International Journal of Computer Science, Engineering and Information Technology*, 8(1).
- Sheard, J., Denny, P., Hellas, A., Leinonen, J., Malmi, L., & Simon. (2024). Instructor perceptions of ai code generation tools - a multi-institutional interview study. *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*. <https://doi.org/10.1145/3626252.3630880>
- Singh, P., & Singh, L. K. (2021). Reliability and safety engineering for safety-critical systems in computer science: A study into the mismatch between higher education and employment in brazil and india. *IEEE Transactions on Education*, 64(4). <https://doi.org/10.1109/TE.2021.3057611>
- Singla, A., Sukharevsky, A., Yee, L., Chui, M., & Hall, B. (2024). The state of ai in early 2024: Gen ai adoption spikes and starts to generate value.
- Stepanova, A., Weaver, A., Lahey, J., Alexander, G., & Hammond, T. (2021). Hiring cs graduates: What we learned from employers. *ACM Trans. Comput. Educ.*, 22(1). <https://doi.org/10.1145/3474623>
- Wagner, S., Mendez, D., Felderer, M., Graziotin, D., & Kalinowski, M. (2020). Challenges in survey research. In M. Felderer & G. H. Travassos (Eds.), *Contemporary empirical methods in software engineering*. Springer International Publishing. https://doi.org/10.1007/978-3-030-32489-6_4
- Wang, R., Cheng, R., Ford, D., & Zimmermann, T. (2024). Investigating and designing for trust in ai-powered code generation tools. <https://doi.org/10.1145/3630106.3658984>
- Zastudil, C., Rogalska, M., Kapp, C., Vaughn, J., & MacNeil, S. (2023). Generative ai in computing education: Perspectives of students and instructors. *2023 IEEE Frontiers in Education Conference (FIE)*. <https://doi.org/10.1109/FIE58773.2023.10343467>