

Insights from the Frontline: GenAI Utilization Among Software Engineering Students

Rudrajit Choudhuri*, Ambareesh Ramakrishnan*, Amreeta Chatterjee*, Bianca Trinkenreich*, Igor Steinmacher†, Marco Gerosa†, Anita Sarma*

*Oregon State University, United States, {choudhru, ramakria, chattera, bianca.trinkenreich, anita.sarma}@oregonstate.edu

†Northern Arizona University, United States, {igor.steinmacher, marco.gerosa}@nau.edu

Abstract—Generative AI (genAI) tools (e.g., ChatGPT, Copilot) have become ubiquitous in software engineering (SE). As SE educators, it behooves us to understand the consequences of genAI usage among SE students and to create a holistic view of where these tools can be successfully used. Through 16 reflective interviews with SE students, we explored their academic experiences of using genAI tools to complement SE learning and implementations. We uncover the contexts where these tools are helpful and where they pose challenges, along with examining why these challenges arise and how they impact students. We validated our findings through member checking and triangulation with instructors. Our findings provide practical considerations of where and why genAI should (not) be used in the context of supporting SE students.

Index Terms—Empirical Study, Generative AI, Challenges and Benefits, Software Engineering Education, Human-AI Interaction

I. INTRODUCTION

Generative AI (genAI) tools (e.g., ChatGPT [1], Gemini [2], Copilot [3]) have asserted their prevalence over a couple of years. These tools are widely used in software development as well as education [4, 5, 6]. Thus, it behooves us as educators to understand how and where to incorporate genAI to train the next generation of AI-savvy SE workforce [7, 8].

However, how to seamlessly incorporate genAI use by SE students without long-lasting adverse effects is a non-trivial question. There are intense debates about the appropriateness of students using genAI in their learning processes [9]. Some argue that this marks the end of traditional education [10, 11] and fear that the future workforce will lack essential skills, such as problem-solving, due to over-reliance on AI tools [12, 13]. The counterargument is that it is not all negative; there are opportunities despite the risks [14, 15, 9, 16] and “it depends” on how these tools are being used.

The concerns raised by the former group are serious. First, genAI systems are not infallible. They are trained on available (sometimes biased or incomplete) data, and function by predicting the most likely sequence of symbols, without guaranteeing the veracity of information [17, 18]. The convincing nature of genAI responses, even when it is hallucinating [19], can foster over-reliance among novices [12, 13], who might not have the relevant knowledge to assess the accuracy of its responses. Second, a deeper concern is that genAI will stymie learning: not only for the ones who want to bypass learning, seeking ready-made answers instead of engaging with the material; but also for the ones who actually want to learn. This is because of the fundamental difference between

comprehending a solution provided by genAI and creating a solution through personal, mental effort. Using genAI reduces students’ problem-solving and critical thinking into a mere reading comprehension exercise [20], undermining metacognition, the very essence of the learning process.

Yet, considering a complete “ban” or forbidding students from using genAI tools is impractical [21], as these tools are already widely adopted [6, 22]. Further, with the software industry’s exuberant adoption of genAI, proficiency with genAI has become a necessary job skill for the future workforce [23, 24]. Acknowledging the inevitability of some instances of academic misconduct [25], the emphasis should thus shift towards guiding students on the effective use of genAI. Educators and policymakers aligned with this thought are speculating on the incorporation of genAI into curricula and pedagogy in the long term [21, 26, 7], seeking a clear landscape of where these tools can help or hinder. Prior research has explored the potential benefits and challenges of genAI usage from both student [27, 28, 29] and educator [21, 27, 30, 31] perspectives.

However, there is a dearth in understanding the current state of genAI usage among SE students: *when* and *how* they use genAI, *where* they are finding benefits or facing challenges, and *why* these challenges exist. Without answers to these questions, we will not know how to effectively integrate genAI into (SE) education without risking adverse impacts.

In this paper, we investigate: **(RQ1): What is the current state of genAI usage among SE students?** Through reflective interviews with 16 SE students, validated through member checking and interviews with two instructors, we elicit students’ experiences of genAI usage to complement learning of SE concepts and their implementations in SE courses. We explored the specific contexts when students turn to genAI tools, examining how they incorporate these tools in their work, and where they face benefits and challenges.

Furthermore, to gain a deeper understanding of the appropriateness of genAI tools for supporting SE students, it is essential to identify why students face challenges when using these tools. This led to our second research question: **(RQ2): What are the causes and consequences of the challenges faced by SE students?** Qualitative analysis of student experiences revealed six intrinsic categories of issues (faults and gaps) within genAI itself, which contributed to five categories of student challenges. These challenges ultimately impacted students’ learning and task outcomes, self-perception, and willingness to adopt genAI technology.

The primary contributions of this paper are twofold: (1) It assesses the use of genAI among SE students, detailing the circumstances (“when”) and methods (“how”) of their genAI usage. This evaluation further explores the contexts (“where”) students perceive benefits and challenges, offering practical insights for educators interested in incorporating genAI to enhance student learning experiences. (2) It identifies the causes and consequences of the challenges (“why’s”) faced by students in using genAI. These findings offer practical insights for educators to guide students on where and why genAI can (or cannot) be effectively used in the context of SE education.

II. RELATED WORK

User studies with genAI: Research on genAI-based programming tools [32, 33, 34, 35, 36, 37, 38, 39] has shown their usefulness and limitations. Vaithilingam et al. [36] found programmers faced more difficulties completing tasks with GitHub Copilot compared to traditional autocomplete, although with no significant effects on task completion time. Barke et al. [32] investigated programmers’ interaction with Copilot identifying exploration (using genAI for planning) and acceleration (using genAI to speed up code authoring) interaction modes. They observed that over-reliance on Copilot can lead to increased cognitive load. Bird et al. [40] studied how developers who were first-time Copilot users, engaged with it. They found that developers accepted generated suggestions for efficiency, trading autonomy and control over code. Other studies [22, 38] focused on programmers’ perceptions, with Liang et al. [22] finding appreciation for autocomplete features but concerns over code quality, control, and IP risks. While these studies offer insights into user perception of genAI tools, their findings primarily apply to professional programmers. Closest to our work is the study by Choudhuri et al. [19]. They investigated the effectiveness and pitfalls of ChatGPT in supporting students in SE tasks. Their findings highlight that using ChatGPT had no significant improvements in productivity, although it significantly increased participants’ frustration. Further, they investigated the causes and consequences of AI faults arising from human-AI interaction guideline [41] violations, leading to negative consequences for students using ChatGPT for SE implementations.

(+) In contrast to these studies, our study did not control or focus on specific tasks or genAI tools. We complement their findings by concentrating on student challenges and presenting associations between their causes and implications across the SE learning and implementation phases.

GenAI in education: Previous research has examined the use of genAI tools in education, with research investigating both the opportunities and challenges these AI tools present [14, 15, 9, 21, 30, 42, 43]. This included advantages, impact on teaching methods due to readily available solutions, and concerns over plagiarism, biases, and fostering detrimental habits among students. In work exploring the opportunities and risks presented by these tools, Bommasani et al. [13] explicitly list Copilot as a challenge for educators, stating students’ over-reliance may negatively impact their learning. How students

should adopt genAI tools remains unclear [44], but their increasing role inside and outside the classroom seems certain. Beyond theoretical discussion, practical research efforts have aimed at enhancing student assistance in code generation on introductory programming problems [45], providing learning resources [46, 47, 48], explanations [49, 50, 47, 51], and finding issues in problematic code [52]. While suggesting significant aid, these studies highlight varying extents of help based on task complexity and prompt quality, emphasizing the need for clarification on where genAI can be effectively used.

(+) Our study adds to this body of work by presenting the current state of genAI usage among SE students (when and how), alongside where students perceive benefits and challenges, giving practical considerations for educators to integrate genAI in SE education.

Student and instructor perspectives: Recent studies have examined student and instructor perceptions of genAI tools in broad educational settings, revealing a generation gap in perceptions [53, 54]. Students are generally optimistic, but are self-aware of the negative impacts of AI on learning [54], while instructors express concerns about over-reliance, ethics, and skepticism regarding genAI’s abilities [21, 27, 29, 30, 31]. Instructors emphasize the urgent need for clear policies and guidelines on where and to what extent genAI should be integrated to maintain academic integrity and equitable learning. Lau and Guo [21] interviewed instructors about adapting genAI tools like ChatGPT and Copilot in courseware, framed around a hypothetical futuristic scenario. Instructors’ short-term concerns centered on cheating, plagiarism, and reliance on invigilated exams, while long-term perspectives varied between resisting AI tools and integrating them into the curriculum. Zastudil et al.’s [31] comparative analysis showed aligned concerns between students and instructors regarding over-reliance, trustworthiness, and plagiarism, but divergent preferences for addressing these issues. Students focused on the quality of genAI responses, whereas instructors worried about students’ ability to identify incorrect or misleading responses. Other student-centric investigations [28, 55] have explored students’ querying behavior, types of assistance sought, as well as benefits and challenges, emphasizing concerns related to usability, reliability, engagement, learning, and ethics.

(+) Prior work has identified the “what’s” — benefits, challenges, and perspectives — from both student and instructor viewpoints. Our work analyzes the “where” and “why”, contributing an understanding of the current genAI usage in SE education, alongside explaining the causes and consequences of challenges faced by SE students in using these tools.

III. METHOD

We conducted semi-structured interviews with software engineering (SE) students to elicit their academic experiences with genAI. See supplemental [56] for the interview script.

A. Interview Planning

Script design: Learning is a continuous process that begins with acquiring and understanding knowledge and progresses to applying and synthesizing that knowledge in new contexts

[57, 58]. In the context of SE education, this spans from learning SE concepts to their implementations [59]. To understand how SE students use genAI to complement their SE learning, we structured our interviews into three parts. The first two parts focused on students’ experiences in using genAI to complement their (1) learning of SE concepts and (2) implementation of SE tasks. The final part focused on their (3) overall experiences of using genAI tools.

In the first two parts, we asked questions about (1) their reasons for using genAI tools, (2) how and in what contexts they leveraged these tools, and (3) the benefits and challenges they perceived, along with the specific contexts and reasons for these perceptions. We anchored these discussions in tangible artifacts, as suggested by Lau and Guo [21], by requesting participants to examine their conversation history with genAI tools (e.g., ChatGPT) and reflect on their experiences through these concrete examples. Participants’ previous conversation threads with genAI acted as a foundation to support deeper recollections of their experiences. A potential risk of focusing on specific artifacts is that participants can fixate on specific conversations or experiences. Therefore, to balance detail-oriented and holistic perspectives, we structured the final part of our script to gather participants’ overall experiences with genAI. In the end, a wrap-up question invited any additional insights and feedback not covered by the structured questions.

To reduce cognitive biases such as priming [60] and anchoring [61], we intentionally avoided mentioning specific tools, such as ChatGPT, in our interview protocol. Everything was worded as “genAI tools”. However, if participants initiated discussions about particular tools, we allowed the conversation to naturally shift toward those tools. Before the main interviews, we conducted seven sandbox sessions and a pilot to test and refine our interview scripts. This led to simplifying our script for clearer, more straightforward questions. The results reported in this paper are based on the actual 16 interviews, excluding these preliminary sessions. We did not segregate the interviews and data analysis into distinct phases. Instead, we continuously refined our understanding, adjusting our interview script and code set as necessary, in line with recommended practices in qualitative research [62].

Recruitment: Our goal was to elicit students’ experiences using genAI to complement their SE learning and implementations. So, we conferred with SE instructors at our university, who recommended senior SE courses and SE capstone courses, as students from these classes would have had the opportunity to use genAI to learn concepts and help implement solutions. Our selection criteria were students who had: (a) familiarity with genAI and experience with it in SE, (b) completed both introductory and intermediate SE courses in the previous year (2023), and (c) enrolled in senior SE or capstone courses. Instructors facilitated recruitment by announcing the study in both in-person and e-campus classes. Interested students completed a questionnaire about their demographics (age, gender, and academic level), familiarity with genAI, and experience with genAI in SE. We received responses from 33 students, of which 28 satisfied the selection criteria.

B. Data Collection and Analysis

Interviews were conducted remotely through Zoom, adhering to the university’s IRB protocol. Before their participation, participants agreed to an IRB-approved informed consent. The sessions were audio-recorded and lasted around an hour each. The first author transcribed the interviews.

Participants: We initially invited 28 students who met our selection criteria, inquiring about their availability for participation. Out of these, 19 confirmed their availability. However, three students withdrew, leading to a final count of 16 participants: 10 self-identified as men, five as women, and one as non-binary or gender diverse. In subsequent sections, participants are denoted as P1-P16. Their demographics are summarized in Table I. As a token of appreciation, students received a \$20 Amazon gift card.

TABLE I
PARTICIPANT DEMOGRAPHICS (N=16)

<i>Attribute</i>	<i>N</i>	<i>Percentage</i>
Gender		
Men	10	62.5%
Women	5	31.3%
Non-Binary or Gender Diverse	1	6.3%
Academic Level		
Junior	4	25.0%
Senior	12	75.0%
GenAI Used		
ChatGPT	16	100.0%
Gemini/Bard	5	31.2%
Copilot	4	25.0%

Qualitative analysis: We analyzed the data using *reflexive thematic analysis* [63, 64] to discern patterns and meanings within it. We used Atlas.ti [65] to facilitate this process, which involved an iterative method of adjusting codes, e.g., merging and splitting them, as our understanding of the data deepened and themes began to emerge. To ensure reliability in our analysis, we held multiple team meetings over seven weeks. During these sessions, the authors critically compared and contrasted the codes and discussed the differences, as advocated for in thematic analysis [62, 66]. Our themes are an *output* of the analysis process rather than an input, as occurs in other forms of qualitative analysis. In more detail, we proceeded as follows:

Two authors inductively open-coded the data, identifying preliminary codes. Subsequent team meetings were dedicated to refining these codes as a shared understanding of the data developed. As the analysis progressed, the authors built post-formed codes and associated them with respective parts of the interview transcripts, following a negotiated agreement. Iterative adjustments to the code set and the interview script were made as necessary. Next, we compared and contrasted the codes, merging (or splitting) them as required. A key aspect of this step was determining the granularity of the codes, which was achieved by analyzing co-occurrences: codes that frequently appeared together were merged; otherwise, they remained separate. During this process, codes with logical connections were grouped into higher-level categories.

We reached saturation after conducting 10 interviews, beyond which subsequent data did not yield new insights [67]. Nonetheless, we proceeded with six additional interviews to ensure that no other insights emerged, as well as to focus on achieving a balanced representation of participant demographics and incorporating diverse perspectives.

We validated our findings through member checking with interviewees and then triangulated the findings with (1) social science theories and (2) interviews with instructors from the courses where we recruited participants.

1) **Member checking with students:** Following data analysis, we emailed a questionnaire to our participants, presenting each of our findings with a Likert scale for agreement and an open-ended text option for feedback (or corrections). All interviewees except P10 and P14 responded. Participant feedback confirmed our findings, with some providing additional clarifications, but no new insights or disagreements emerged.

2) **Instructor interviews:** We interviewed SE instructors, presenting to them where students reported the benefits and challenges of using genAI along with the consequences of these challenges. The instructors confirmed all findings, with one exception, as noted in the results (see supplemental for detailed examples). The interview script is available in the supplemental material [56].

IV. GENERATIVE AI USAGE AMONG SE STUDENTS (RQ1)

In this section, we map the *current usage of genAI amongst SE students*, identifying WHEN and HOW students use genAI to complement their SE learning and implementations, along with WHERE they perceive benefits and challenges.

A. When do SE students use genAI?

We explicitly inquired about students' use of genAI for learning and implementation phases. When analyzing their answers, we identified that each phase can be divided into two, as presented below.

Learning (Initial - L1) corresponded to situations where participants used genAI to learn SE concepts from scratch, without any prior knowledge. Participants mentioned using genAI tools to get definitions and a basic understanding of SE concepts. For example, P9 *“asked genAI to tell what fuzzing is and what its purpose was.”*

Learning (Incremental - L2) is characterized by situations when participants, after having some background knowledge, used genAI to *“refine understanding and tie up loose ends”* (P5). P2 mentioned that genAI was useful for *“clarifying sprint and agile concepts that were confusing in the lecture”* (P2).

Implementation (Initial - I1) involved participants' engagement with genAI during the initial stages of software implementations. They reported instances of using genAI to *“whip up boilerplates”* (P15), set up frameworks, and assess code rundowns. P10 elaborated, *“I used ChatGPT for setting up a quick react framework and got rundowns of structure and content of the code”*.

Implementation (Advanced - I2) corresponded to scenarios when participants used genAI to complement contributions

to existing code bases or their own term/capstone projects. P1 used genAI for: *“optimizing the solutions that I have built. I ask it for the recommended best practices or industry standard approach for the type of problem I am dealing with”*.

Table II presents exemplary concepts and course contents corresponding to each of these phases, as provided by the course instructors. This can help readers relate our subsequent findings from an SE curriculum perspective. For clarity and ease of reference throughout this paper, we will use the abbreviations L1, L2, I1, and I2 to refer to these phases.

TABLE II
SE COURSE CONTENTS CORRESPONDING TO DIFFERENT PHASES OF SE LEARNING AND IMPLEMENTATIONS

	Introductory/Intermediate	Senior/Capstone
L1	Version control systems, APIs and microservices, UI design principles, User stories, Scrum practices (e.g. sprints, backlogs), Code review	Project requirement analysis, Architectural patterns (MVC, MVVM), System design and scaling, SE best practices (code, documentation, review)
L2	Advanced Git features (e.g. merging, rebasing, resolving conflicts), RESTful services, HTTP methods, Code smells, Software design, Software process models (principles, roles)	Designing API gateways, Linters, Application performance monitoring, Software security, OAuth, Code performance optimization, System integration testing
I1	Setting up frameworks and environments, Dependency management, UI development (HTML, CSS, JS), Third party API integration, Unit tests	Software prototyping (design choices, core functionalities), Environment configuration for multiple service support, Setting automated test frameworks
I2	Building microservices, Testing frameworks, Integration and end2end tests, Inter-process communication, Term project, Code reviews, PRs and issues	Software deployment (GH pages, AWS), Maintenance routines, CI/CD, Manage code conflicts across git branches, Term project

B. How do they use it?

During the **learning** phase, participants described four ways in which they used genAI to support their learning.

1) **Role-based prompting** involves instructing genAI systems to assume a specific role (e.g., a teaching assistant) or to explain concepts to a particular type of audience (e.g., a child). Participants mentioned that they prompted genAI tools to simplify complex concepts by requesting explanations in layman's terms or asking it to *“explain things to a child”* (P8). Further, participants gave tutoring roles to genAI to get deeper explanations on topics. For example, P2 shared, *“I used to often ask ChatGPT and Gemini to pretend to be my TA for the class, pretending I'm a complete novice”* (P2).

2) **Using AI as an advanced search engine** allowed participants to filter and *“narrow down the search space”* (P12), pinpointing information for the topic of interest. For example, P9 mentioned, *“I used to ask ChatGPT to get a basic idea of what all I needed to learn. If it gave me 3 things, I would then look up and read about those things”*.

3) **Combining AI responses with other sources** allowed for a more complete understanding. Participants considered genAI to be *“not really good at giving the complete answer”* (P16). They mentioned combining AI responses with other sources such as Youtube (P16), Stack Overflow (P9), for a better understanding of SE concepts, instead of solely relying on genAI-provided explanations.

Initial Learning (L1) ⚠		Incremental Learning (L2)	
C1. Unclear understanding of genAI & its use • Ethical dilemma • Setting realistic expectations	C2. Difficulty in effectively communicating needs or context • Framing the right questions or prompts • Communicating context ★ • Need for domain knowledge to query AI ★ • Multiple back and forth exchanges	Clarifying concepts and code	Getting practice problems ★
C3. Difficulty in aligning AI to process or preferences • Hard to align with learning style ★	C5. Difficulty in appropriately using genAI responses • Hard to verify AI responses • Misleading or confusing responses • Hard to adapt AI to needs	Getting structured outlines and roadmaps	Getting surface level understanding
C4. Issues in obtaining proper rationales behind genAI's responses		Learning syntax ★	Obtaining descriptive examples ★
		Personalized support	Revising concepts ★
Initial Implementation (I1)		Advanced Implementation (I2) ⚠	
Brainstorming ideas ★	Clarifying concepts and code	C1. Unclear understanding of genAI & its use • Ethical dilemma • Setting realistic expectations • Gauging genAI's limitations ★	C2. Difficulty in effectively communicating needs or context • Framing the right questions or prompts • Multiple back and forth exchanges
Comprehending code snippets ★	Finding useful resources ★	C3. Difficulty in aligning AI to process or preferences • Hard to align AI to preferences ★ • Interrupts thought processes ★	C5. Difficulty in appropriately using genAI responses • Hard to verify AI responses • Misleading or confusing responses • Hard to adapt AI to needs • Hard to modify or generalize AI responses ★
Getting structured outlines and roadmaps	Getting surface level understanding	C4. Issues in obtaining proper rationales behind genAI's responses	
Obtaining initial guidance ★	Personalized support		

Fig. 1. Perceived benefits and challenges of genAI usage among students: Benefits were in incremental learning (L2) & initial implementations (I1), with challenges in initial learning (L1) & advanced implementations (I2). Stars indicate perceptions unique to the quadrant.

4) **Using course materials to query AI** enabled participants to ask focused questions to refine or expand on their knowledge. For instance, P4 described, “I get familiarized with the jargon using class resources, and then come back to GPT and ask specific questions. I then use that knowledge to learn the concept and then apply it for my problem”.

Participants reported using genAI tools for supporting **implementations** in three distinct ways.

1) **Using AI to improve own or obtained code:** Participants used genAI tools to refine their code (or the ones retrieved from other sources), for better performance and compliance with best practices. P1 explained, “When dealing with a problem, I use generative AI to optimize the solutions I have built. I ask it for the recommended best practices or industry-standard approach for this type of problem”.

2) **Using AI for code snippets and organizing it on their own:** Participants highlighted their use of genAI to acquire code components or “pieces of the puzzle”, as P10 puts it, which they then organize into cohesive solutions for their projects. P3 shared, “I lean towards using genAI for individual parts that I planned dealing with in my project, that’s where I think its value comes from...I can take those pieces and organize it to fit them into my project”.

3) **Combining AI-generated code with traditional sources:** Participants blended genAI-generated code with insights from traditional sources, like online documentation (P3), Google searches (P7), and forums such as Stack Overflow (P15), to craft effective solutions. This approach mirrors the *combination of AI responses with other sources* for learning concepts. P15 described this approach: “for me, the development cycle is to start with ChatGPT, refine it with documentation or specific Stack Overflow examples to fill the portions it missed”.

C. Perceived benefits and challenges

Figure 1 presents an overview of the benefits and challenges perceived by SE students in using genAI tools, highlighting the

specific phases in which these perceptions arise. Participants perceived the benefits of using genAI only for incremental learning (L2) and initial implementation tasks (I1). Conversely, they encountered challenges when using genAI for initial learning (L1) and advanced implementations (I2).

BENEFITS: Interviewees valued genAI tools both in incremental learning (L2) and initial implementations (I1) for its ability to provide *personalized support* and *clarify concepts and code*. One participant mentioned that “if I need a clearer explanation of a topic from class or if I am stuck with something in my code, I definitely turn to generative AI for a personalized explanation” (P12). Moreover, genAI was valued for *providing surface level understanding* and *structured outlines or roadmaps* which aided in identifying key topics or areas of focus. P3 shared, “I find myself leaning more and more on genAI tools to get structured outlines and to know what information I need to gather”.

In the incremental learning phase (L2), genAI proved beneficial for *revising concepts*: “ChatGPT can help in brushing up or polishing concepts if you have a basic understanding of it” (P14) and *obtaining descriptive examples*: “you can ask generative AI to give you an example or demonstrate what you’re asking about in practice, which is helpful” (P3). Participants also perceived these tools to be useful for *getting practice problems* to enhance their understanding of a topic. P1 highlighted, “I would also ask for incrementally challenging practice problems with a topic. It helped me have a practical understanding”. Moreover, these tools were useful for *learning syntax*, when participants had a programming background: “GenAI was useful when I needed to learn a particular JS syntax. I could understand what it said because I already knew how to program in JS” (P13).

Interviewees also valued genAI tools during the initial implementations (I1) phase, particularly for *comprehending code snippets*, *brainstorming ideas*, and *finding useful resources*. P4

stated that “*given a problem or an assignment, the hardest part is to get started. AI is really helpful in that, it is useful in brainstorming, and it points you to sources that can lead you to the goal*”. Participants noted that they turned to genAI for *initial guidance*, seeking instructions to follow and a code structure to work with. P9 shared, “*I was clueless at the start of a game development project. I asked ChatGPT how to go through developing that and it gave me instructions on steps I needed to follow tailored to my project*”.

Instructors largely confirmed these benefits, with the exception of the value of “personalized support”, where they emphasized that “*AI doesn’t always get the context right or understand learning goals, making personalized [support] double-edged, it is risky when students build too much on flawed help*” (Capstone Instructor).

CHALLENGES: Interviewees highlighted various challenges in using genAI tools as complete beginners learning SE concepts (L1) and in tackling complex implementations (I2). These challenges can be organized into five main categories. First, participants reported challenges in understanding genAI’s limitations and identifying its appropriate and ethical applications (C1). They found it difficult to effectively communicate their needs or context to the AI (C2) and to align it with their process or preferences (C3). They also faced obstacles in obtaining clear rationales for the genAI responses (C4) and applying them appropriately (C5).

These categories were identified in both the L1 (initial learning) and I2 (advanced implementation) phases, with specific challenges occurring in either phase or spanning both. For clarity, we outline these distinctions in Figure 1 and proceed to explain the categories in the text:

C1) Unclear understanding of genAI and its use: Participants reported difficulties understanding genAI’s limitations and gauging its appropriate applications in software engineering. For example, they struggled to *determine the ethical use of genAI in SE*: “*I don’t know whether it is ethical to use generative AI for actual implementations*” (P10), indicating moral ambiguity surrounding the tool’s use. *Setting realistic expectations* is another hurdle reported by our sample. Participants’ high expectations often clashed with genAI’s actual capabilities, leading to dissatisfaction. P14’s experience with genAI illustrates this gap: “*I know for a fact that it wasn’t exactly as helpful as I thought it would be...because it did not quite match my expectation*”. Lastly, *gauging genAI’s limitations* proved difficult for participants, particularly at the onset of use. P14 commented on their initial skepticism, “*Initially, when I first started using it, I would trust it less because I wasn’t quite exposed yet to how they work and how effective they actually were*”.

C2) Difficulty in effectively communicating needs or context: Participants highlighted several challenges in effectively conveying their needs and contexts to genAI. *Framing the right questions or prompts* emerged as an obstacle for those seeking useful responses. P6 highlighted, “*My biggest challenge is asking it the right questions, to get a useful response out of it*”. Participants reported that *communicating context* to genAI

was challenging. P3 described the struggle: “*The biggest challenge was providing enough context to [ChatGPT] and communicate my needs about what I’m trying to do*”. Furthermore, interviewees emphasized that achieving appropriate responses required engaging in iterative dialogues with genAI involving *multiple back-and-forth exchanges*. P7 explained, “*Sometimes, I just go back and forth with the context till it fixes its answers*”. Effective use of genAI also *requires domain knowledge*, complicating its utility for those without it: “*The difficulty is when there is a knowledge gap and you are using it for things without having solid fundamentals about it. If you don’t understand the concept, you can’t use AI for it*” (P4).

C3) Difficulty in aligning AI to process or preferences: Interviewees reported difficulties in *aligning genAI to their preferences*. P16 articulated this issue, stating that genAI “*does not take into account my preferences or style or what I want to fix in the codebase*”. Moreover, code-generating AI tools, such as Copilot, can *interrupt thought processes* when it “*gives huge blocks of things that you don’t want, it breaks the flow and thought process*” (P7). Further, participants mentioned facing significant challenges in learning when genAI does not *align with their learning style*. P12 reported that “*generative AI was not aligned to my style of learning...the way it talks to me is not the way I want to learn*”.

C4) Issues in obtaining proper rationales behind genAI responses: Interviewees noted their struggle to secure rationales for AI-suggested procedures and solutions. P1 stated, “*If I want to know why a procedure operates the way it does, ChatGPT doesn’t give me a full answer...when I am seeking a deeper understanding of what’s going on*”. Participants shared that it was hard to discern errors with genAI and “*to understand what was going wrong and why*” (P12).

C5) Difficulty in appropriately using genAI responses: Participants highlighted their struggles for appropriately using genAI responses in their work. *Misleading or confusing responses* troubled them, as described by P5: “*It outputted sensible looking code but it didn’t match my instructions, leading to confusion*”. They found it hard to *adapt genAI solutions to their context* or “*make it generate content that fits all the needs*” (P3). Lastly, participants reported challenges in *verifying genAI responses alongside modifying or generalizing them for wider use*. P3 mentioned, “*the responses were very hard to generalize or modify to include all cases*”.

Instructors validated these challenges without revealing any new ones. They also noted that students still remain hesitant to discuss challenges due to unclear academic AI policies.

V. ANTECEDENTS & IMPACTS OF CHALLENGES (RQ2)

In this section, we answer RQ2, detailing the causes and consequences of the challenges encountered by SE students using genAI tools. Figure 2 illustrates the associations between genAI intrinsic issues (causes), challenges, and subsequent impacts (consequences), which we detail next:

A. Causes

Interviewees identified several intrinsic issues with genAI that contributed to their challenges, which we classified into

faults and gaps. Faults refer to flaws within genAI systems, while gaps highlight areas where genAI is currently lacking.

FAULTS: Participants highlighted 13 different AI faults, which fall under four different categories:

1) **Reasoning flaws** in genAI manifest notably through *logic loops* and *self-contradiction*. Logic loops occur when genAI is stuck in repetitive cycles of responses, failing to advance toward a solution that is appropriate. Self-contradiction involves genAI contradicting its statements or presenting “*conflicting information*” (P4) with its earlier conditions or premises. P7 illustrated a typical scenario: “*sometimes AI will give you the wrong piece of code, and then you’ll try it. It doesn’t work...oh, I see the mistake and then will give you the same piece of code back...I just go back and forth with it and that doesn’t lead me anywhere*”. These reasoning flaws created significant hurdles for participants in communicating their needs effectively to genAI (C2). These issues also led to difficulties in aligning genAI to match personal preferences (C3): “*It ignores constraints and gives the same answer again and again*” (P16), obtaining clear rationale behind responses (C4) and appropriately using genAI outputs (C5): “*GPT kept going back and back...then I was confused about how to use what it said and couldn’t understand why that worked*” (P12).

2) **Response quality issues** encompass several problems with genAI outputs, including *incomplete assistance*, *inconsistent responses*, *incompatibility*, *suboptimal responses*, *limited help on specifics*, and *non-generalizability* of solutions.

GenAI’s incomplete assistance resulted in challenges related to setting appropriate expectations (C1). For instance, P14 mentioned, “*ChatGPT was not as helpful as I expected it to be...I needed a precise explanation for each step and it wasn’t providing me with that, as it would skip through certain steps*”. Moreover, genAI’s limited help on specifics made it difficult for participants to obtain proper rationales (C4). P1 shared, “*if I wanted to know why a procedure operates the way it does, rather than what I would envision in my head to be a better way, generative AI didn’t give me a full answer which was annoying as I wanted a deeper understanding of what was going on*”. Additionally, other issues like incompatibility and non-generalizability posed extra challenges in aligning AI outputs with personal processes (C3) and appropriately using AI responses (C5), as P3 pointed out, “*It was hard to build general purpose software with GPT...it provided code that was not really reusable*”.

3) **Deceptive behavior** in genAI is characterized by its *hallucination* and *confirmation bias* tendencies. These issues differ from mere response quality problems—such as incomplete or incorrect information—due to genAI’s active efforts to persuade or convince the user. GenAI often generates plausible yet entirely fabricated information, leading to situations where “*it gives baseless answers with convincing explanations...I never end up learning from it because there’s a debate between what the AI said and the original source, and I don’t know why something is correct*” (P3). Further, genAI often affirms user beliefs regardless of their validity. P4 mentioned, “*AI won’t take any accountability and would just apologize and agree to*

whatever I am saying”, highlighting difficulties in discerning reliable assistance from misleading confirmations (C5).

4) **GenAI neglects students’ context and preferences**, which manifests in various forms. It often *neglects preferences*, *misinterprets students’ problem*, and enforces *misguided guardrails*, which, ultimately, leads to responses that ignore unique contexts. This was what P5 reported: “*It doesn’t want to give me output because it thinks it is a terms of use issue without evaluating my context*”. This kind of faults lead to challenges in communicating needs (C2), and aligning AI to preferences (C3), as P16 illustrated, “*when you ask ChatGPT or any other genAI to not do something a particular way, it does it anyways*”. Further, participants reported that these issues led to challenges in appropriately using AI responses (C5) in their work. P9 shared: “*after I explained repeatedly that I needed to optimize my script without changing external dependencies, the solutions suggested by Gemini and ChatGPT still involved alternate libraries*”.

GAPS: Interviewees highlighted six different gaps (areas where genAI lacks offerings) categorized as follows:

1) **Scaffolding gaps** refer to the current shortcomings of genAI tools (as of 2024) in supporting learning. Participants reported that these tools fell short in *providing an in-depth understanding of concepts* leading to challenges in obtaining proper rationales (C4): “*It was in [SE2] class...GPT or Bard didn’t provide a specific reason that I felt was good enough or believable on why [a test case] should be designed that way*” (P2). Some participants noted that these tools *lack visualization support for explaining concepts* (in free versions (e.g., ChatGPT) as of 2024) and considered these to be “*not very helpful for visual learners*” (P16). Moreover, participants highlighted that genAI’s tendency of *giving out direct answers* often conflicted with their learning preferences (C3), “*ChatGPT does take away from my learning, just because it gives me that immediate answer without me having to find it. I wait before I ask a super big question just so I can think about it before I see a direct answer*” (P12).

2) **Gaps in programming support:** Participants pointed out gaps in genAI’s ability to assist with programming tasks, notably its *inadequate debugging support*, *gaps in orchestrating code*, and its *unsuitability for complex programming*, often attributed to genAI’s limited working memory or outdated training information [68]. Participants noted that genAI frequently faltered in “*debugging functions or programs in general*” (P10) and struggled to “*orchestrate code from different functions, files, or services together*” (P7), complicating the effective use of its responses (C5). For complex programming tasks, genAI’s support falls short, as described by P4: “*working with ChatGPT or using GPT generated code for bigger projects or large apps is hard, it can break the codebase*”.

B. Consequences

Interviewees highlighted several consequences as a result of facing challenges in using genAI within the SE context:

1) **Impact on learning:** Challenges in effectively communicating needs or context (C2) resulted in participants *learning*

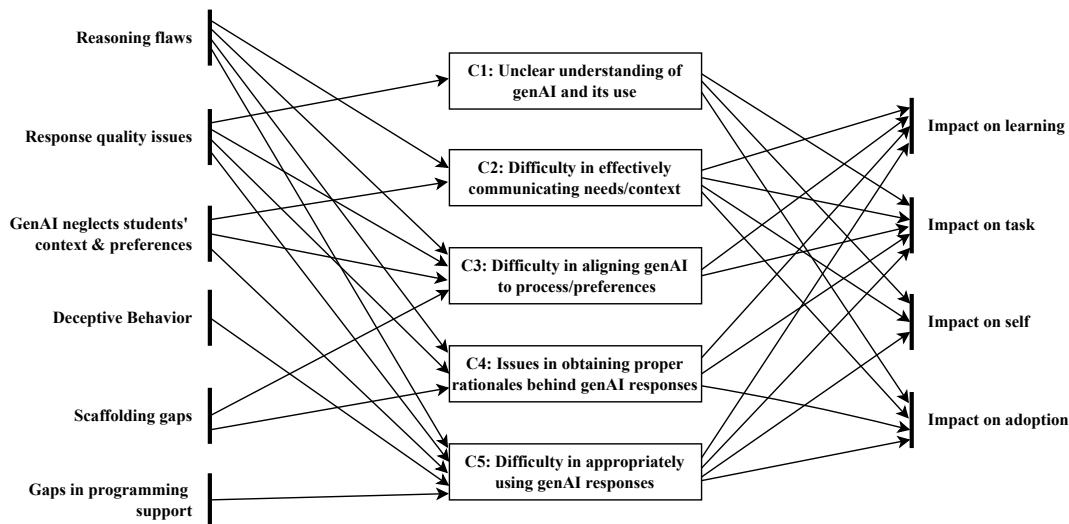


Fig. 2. Associations between genAI's intrinsic issues (faults and gaps), challenges (C1-C5), and the resulting impacts. Issues in genAI contributed to various challenges encountered by participants, which subsequently had impacts on learning, task, self, and genAI adoption.

wrong information. P1 reflected on this, stating, “Sometimes, genAI gives wrong information due to mistakes in how I’ve inquired as I didn’t know enough about the problem to ask the question properly. I had to unlearn what AI said and relearn it using YouTube”. Still, participants reported that difficulties in aligning genAI with learning preferences (C3) and obtaining proper rationales (C4) led to an incomplete understanding of concepts, hindering in-depth comprehension: “I am not able to use AI to wrestle with the ideas in the way I like to learn and understand the reasoning behind things. Using genAI is just like being told what to do, I don’t like that” (P1). The participants also reported that lack of clear explanations from genAI impedes grasping the underlying principles of concepts, as on P6 words: “It didn’t really seem to help me understand the critical components of why something might work and a lot of things went over my head”. Moreover, participants indicated difficulties in verifying AI responses (C5), particularly as beginners (L1), leading to uncertainty about the correctness of the learned material. P8’s experience underlined this: “I ended up learning the wrong things often in the past...so when it gives me responses that are not there on other platforms, I am skeptical about whether I learned the right thing.”

These impacts on learning can be viewed through the lens of Cognitive Load Theory (CLT) [69]. Essentially, difficulties in communicating needs (C2), aligning AI with preferences (C3), issues in obtaining clear rationales (C4), and verifying AI responses (C5) all add to extraneous cognitive load, where students must expend additional effort to use genAI and interpret its responses, detracting from their learning process. Moreover, challenges in obtaining proper rationales (C4) make it harder to grasp underlying principles (or reasoning), thus hindering in-depth comprehension.

2) **Impact on task:** Challenges in obtaining proper rationales (C4) and effectively using AI responses (C5) forced participants to *figure things out on their own*, leading to time-consuming setbacks: “GPT struggled to give an answer that worked or explain why it didn’t work, it kept going back

and back ... I ended up doing it again from scratch using a different resource” (P12). Difficulty in aligning genAI to processes or preferences (C3) not only slowed them down but also led to abandonment of tasks in certain situations (I2). P16 shared, “ChatGPT didn’t really give me a good answer. It went on for a long period of time, suggesting things that were similar but not useful to my project. I gave up and had to downsize that project”, illustrating how genAI derailed their project goals and affected their timeline. Effectively communicating needs (C2) affected participants who spent substantial time refining prompts for actionable advice. P8 said that they “spent at least 30 minutes fixing prompts and going back and forth to make it work”. Lastly, an unclear understanding of genAI’s limitations (C1) often led students, particularly the beginners (L1), to give up on their task, when they found obstacles, due to lack of clear expectations about genAI’s applicability. P12 clearly walked us through that: “For someone new, AI won’t work, it would cause roadblocks and they would give up lacking an assumption of what it can be used for and what not.” Extraneous load from ineffective AI responses (e.g., P12’s need to redo tasks from scratch due to unsatisfactory explanations) and continuous prompt refinement (e.g., P8’s time spent refining prompts) distract students from focusing on the task at hand. Further, task abandonment by P12 and P16 may be attributed to the combined intrinsic load (inherent difficulty of the task) and extraneous load (setting proper expectations about genAI’s applicability and managing its responses), both of which impeded task completion.

3) **Impact on self:** Participants shared how they were personally impacted by using genAI, highlighting instances of *over-reliance*, *self-doubt*, and *frustration*. Over-reliance is common among beginners (L1) [12] due to a lack of understanding of genAI’s limitations or its use (C1), leading to over-dependence and initial blind trust in genAI. This was pointed out by P7, who mentioned that “Initially, I would just prompt it and trust whatever came back...I still find it confusing to realize whether GPT can actually provide a solution or just is

claiming to do so”. P15 added: “I think that new people should kind of stay away from AI as it is tempting to trust it blindly. If this black box is infallible to you, what happens when it’s wrong?”. Challenges in effectively communicating needs (C2) and using AI appropriately (C5) contributed to self-doubt and frustration, as expressed by P15, “It wasn’t helpful because it made false assumptions about the codebase...the provided code was profoundly unhelpful, I couldn’t modify it. It was very frustrating and just hopeless”. These experiences align with the concepts highlighted in the Social Cognitive Theory (SCT) [70], where repeated failures erode users’ confidence and self-efficacy. When students repeatedly experience errors or incorrect outputs from genAI, it contributes to a cycle of frustration and self-doubt. Finally, P4’s comparison with peers underlines the impact on their self-esteem: “my friends can use it well, but I can’t...I am worried I would be left behind when it replaces Google search and I don’t know how to talk to it.” This aligns with Self-Determination Theory (SDT) [71], which emphasizes the need for competence and relatedness. When users feel less competent than their peers, their intrinsic motivation and self-esteem suffer, leading to reduced engagement and satisfaction.

4) **Impact on genAI adoption:** Participants’ willingness to adopt genAI was mainly influenced by challenges in appropriately using its responses (C5). Yet, specific challenges impacting their willingness varied depending on the context of use. For learners (L1), difficulties in effectively communicating needs (C2) and obtaining proper rationales (C4) led to a sense of *distrust*. P2 noted, “I don’t trust generative AI for any critical work, it doesn’t provide proper justifications; it’s more like a predictive text answer that is hard to verify”. P7 added: “It apologizes when you confront it, but gives you the same answer again and again. It’s better to trust yourself and not the tool”. These experiences revealed a broader skepticism towards genAI, deterring its adoption among SE students. Further, participants’ concerns over ethical considerations (C1) steered them away from using genAI in advanced implementations (I2). P10 articulated this concern: “I don’t use it for any implementations because I don’t consider it to be something I have written...and I think until the boundaries become clearer on whether it’s right to use it or not, I am not comfortable using it for work that I have to claim as my own”.

The impact of these challenges on genAI adoption can be understood through the Technology Acceptance Model (TAM) [72], primarily through two constructs: perceived ease of use (PEOU) and perceived usefulness (PU). Both PEOU and PU influence users’ intentions to use software tools. In this context, challenges in using genAI responses (C5) diminish PEOU, particularly for learners (L1), who struggle with communicating needs (C2) and receiving proper rationales (C4). Moreover, ethical considerations (C1) negatively influence both PU and PEOU, steering students away from adopting genAI for advanced implementations (I2).

As with RQ1, we validated these findings with instructors, who confirmed the patterns without proposing any changes. The codebook is available in the supplemental material [56].

We identified different ways SE students make use of genAI, alongside where they perceive benefits and hurdles when using it to complement SE education. It is important to carefully consider how to incorporate genAI into the curriculum and its impact on students; as an instructor reflected:

“Maybe I thought of my students to be confident in figuring out the AI tools, I didn’t predict they would internalize these struggles and get frustrated...It’s more serious than I predicted and can be a very stressful experience for them.”

Backed by our results, we present a set of recommendations for integrating genAI in the SE education context.

Considerations before adopting genAI in SE education. Educators aiming to incorporate genAI into their SE curricula should be aware of the issues that students might encounter when using these tools for initial learning (L1) and advanced implementations (I2), as depicted in Figure 1 and detailed in Section IV-C. A significant obstacle in adopting this rapidly developing technology is understanding and creating awareness of its limitations and applicability (C1). This can be particularly problematic for novice learners, who might rely too heavily on genAI, viewing its outputs as final or unquestionable truths [12]. Educators must, thus, educate themselves and the students about the workings and faults of genAI, specifically that these tools are language models and not knowledge models (i.e., they generate a sequence of symbols and not necessarily factual information). Several universities have put forth resources to help instructors understand how to manage/identify genAI usage in the classroom [73, 74], but these largely explain how to interact with the tool (C2) and not how to incorporate its responses (C5). Educators must emphasize the need to verify genAI’s outputs. This will help especially those starting in the course by (1) setting clear expectations regarding genAI’s assistance, and (2) scoping their trust in AI [75, 76], instead of treating it like an expert.

Further, instead of penalizing students for using genAI, which would only lead to more creative ways of academic misconduct [9], what is needed is to educate them on ethical and appropriate genAI usage and encourage them to acknowledge the extent of AI’s role in their work. Students should also recognize that they are ultimately responsible for the code they generate—with (or without) genAI help. Thus, if security flaws or faults exist in the code, they need to take ownership. This ethos is particularly important to instill in senior SE students as they work with external clients (capstone projects), open source projects, or internships.

Equip students with the right skills. Educators must focus on equipping SE students to be savvy-AI users, given the software industry’s growing interest in integrating genAI into its practices [77]. While AI will not (potentially) supplant software engineers, individuals proficient in these technologies will have more job opportunities and career advancement. This leads to two guiding questions:

How to teach SE students to be savvy-AI users? Prior research has frequently discussed the significance of prompt

engineering [78] in leveraging genAI. Although it helps (instances being role-based prompting), it alone can't completely solve the problem. One of our participants (P3) highlighted struggles with prompt engineering: *"even after spending time learning prompt engineering I couldn't make it listen to me"*. To become savvy-AI users, students need to be taught the nuances of interacting with genAI: how to create different types of queries, how to effectively articulate the problem context to the AI, and that an effective prompt might require fundamental knowledge of SE concepts. Additionally, students need to be taught how to perform critical thinking to evaluate genAI outputs. They should be taught how to iteratively interact with the tool asking it further details or specific questions, and how to triangulate the AI responses with verifiable sources.

How to adapt SE courses to genAI? In hindsight, we now know the concerns in the 1940's about calculators undermining math education were not a reality. Instead, calculators became a useful tool in solving higher-order math problems. Today, genAI tools raise similar yet serious concerns, involving education and practices that require problem-solving [20]. This is problematic in SE education, which involves understanding the problem space, trade-offs, and contextual decision-making. So, *what can we do about this?* First, just like students in primary school still need to learn to do basic math without calculators, complete novices need to be encouraged to learn SE topics without the aid of genAI tools. Second, educators must scaffold the introduction of genAI usage in their curricula, redesigning their assignments that place higher emphasis on reflection and problem-solving skills. For example, in addition to submitting code, students can be asked to reflect on their design decisions/program rationale/alternate choices that are connected back to the lectures. Asking for explanations for why the AI tool gave a particular solution and why that solution is a right choice (or not)—an area where genAI is "stubbornly opaque" [75]—will shift the focus from "obtaining just a solution" to critical thinking about the solution. Such reflections on alternate solutions are skills that SE students need to learn to be ready for an AI-infused workplace.

VII. THREATS TO VALIDITY

One important threat in interview studies is obtaining accurate constructs through questions and avoid asking ambiguous ones. To minimize this threat, we thoroughly evaluated our script internally and assessed its validity through sandboxing and pilot sessions until a team consensus was reached.

Another threat could arise from the absence of concrete tasks conducted by the students during the study, limiting our results to our participants' recollections. We reduced this by focusing discussions on concrete artifacts to elicit participant experiences. Despite these measures, we recognize the potential for measurement inaccuracies.

It is difficult to replicate qualitative research since human behaviors, feelings, and perceptions change over time. Thus, in the reliability thread, Merriam and Tisdell [79] suggest checking the consistency of the inferences. To increase consistency, the analysis was conducted by two authors, followed

by meetings with the team to refine and discuss the themes and categories until we reached a consensus. We also performed member checking and triangulated with instructors to validate the authenticity of our data interpretations and findings.

Our study, like others, was susceptible to self-selection bias, where individuals interested in genAI might be more inclined to participate. Desirability bias might also have impacted our study, where participants might favor genAI due to its current popularity. To mitigate this, we framed our questions in neutral, unbiased language, encouraging participants to draw from actual experiences by providing concrete examples. Still, we analyzed the data cautiously acknowledging the potential influence of desirability bias while interpreting our results. Finally, despite our explicit efforts to assure participants about the confidentiality of the study, it is plausible that concerns over possible repercussions regarding genAI's academic use might have deterred them from fully disclosing their thoughts.

Our study included participants from various courses and settings (in-person and online classes), still it is limited to a single university. Additionally, all participants interacted with the genAI tools in English and faced no language barriers. Our findings, thus, may not generalize to other contexts or domains. Lastly, we report qualitative findings aggregated in emergent categories. While quantitative insights provide useful information, prior work has cautioned against using quantitative research methods on qualitative data [80]. Therefore, we eschewed from discussing frequency or percentage of occurrences of categories.

VIII. CONCLUSION

Our study explored the current state of genAI usage among SE students to understand how they complement their SE learning and implementation tasks. Through 16 reflective interviews, we uncovered the contexts where these tools are helpful and where they pose challenges, examining why these challenges arise and how they impact students. Participants reported that they perceived benefits when using genAI for incremental learning (L2) or initial implementation tasks (I1). Conversely, challenges arose in using these tools as beginners (L1) or in advanced implementations (I2). Several challenges occurred due to intrinsic issues in genAI itself, leading to impacts on learning, tasks, self-perception, and genAI adoption.

A key call to action is on how to instill critical thinking, problem-solving abilities, and AI literacy among SE students both within traditional academic pathways and those aspiring to enter the field of SE. Without clear guidelines on how to effectively use genAI, we may have a scenario akin to *"learning to play the piano—once you hit the wrong key, it becomes hard to stop, as what you've learned in that moment is what you learn best."* (SE-Intermediate Instructor)

Data Availability: The research artifacts for this study are publicly available on the companion website [56].

Acknowledgements: We thank all the study participants. This work was partially supported by the National Science Foundation under Grant Numbers 2235601, 2236198, 2247929, 2303042, and 2303043.

REFERENCES

- [1] OpenAI, “Gpt-4,” <https://openai.com/product/gpt-4>, 2024.
- [2] Google, “Gemini,” <https://gemini.google.com>, 2024.
- [3] Microsoft, “Copilot,” <https://copilot.microsoft.com>, 2024.
- [4] A. Fan, B. Gokkaya, M. Harman, M. Lyubarskiy, S. Sengupta, S. Yoo, and J. M. Zhang, “Large language models for software engineering: Survey and open problems,” in *2023 IEEE/ACM International Conference on Software Engineering: Future of Software Engineering (ICSE-FoSE)*. IEEE, 2023, pp. 31–53.
- [5] C. Ebert and P. Louridas, “Generative AI for software practitioners,” *IEEE Software*, vol. 40, no. 4, pp. 30–38, 2023.
- [6] D. Russo, “Navigating the complexity of generative AI adoption in software engineering,” *ACM Transactions on Software Engineering and Methodology*, 2024.
- [7] C. Bull and A. Kharrufa, “Generative AI assistants in software development education: A vision for integrating generative AI into educational practice, not instinctively defending against it,” *IEEE Software*, 2023.
- [8] A. Nguyen-Duc, B. Cabrero-Daniel, A. Przybylek, C. Arora, D. Khanna, T. Herda, U. Rafiq, J. Melegati, E. Guerra, K.-K. Kemell *et al.*, “Generative artificial intelligence for software engineering—a research agenda,” *arXiv preprint arXiv:2310.18648*, 2023.
- [9] P. Denny, J. Prather, B. A. Becker, J. Finnie-Ansley, A. Hellas, J. Leinonen, A. Luxton-Reilly, B. N. Reeves, E. A. Santos, and S. Sarsa, “Computing education in the era of generative AI,” *Communications of the ACM*, vol. 67, no. 2, pp. 56–67, 2024.
- [10] M. Welsh, “The end of programming,” *Communications of the ACM*, vol. 66, no. 1, pp. 34–35, 2022.
- [11] D. M. Yellin, “The premature obituary of programming,” *Communications of the ACM*, vol. 66, no. 2, pp. 41–44, 2023.
- [12] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. d. O. Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman *et al.*, “Evaluating large language models trained on code,” *arXiv preprint arXiv:2107.03374*, 2021.
- [13] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill *et al.*, “On the opportunities and risks of foundation models,” *arXiv preprint arXiv:2108.07258*, 2021.
- [14] M. Amoozadeh, D. Daniels, S. Chen, D. Nam, A. Kumar, M. Hilton, M. A. Alipour, and S. S. Ragavan, “Towards characterizing trust in generative artificial intelligence among students,” in *2023 ACM Conference on International Computing Education Research-Volume 2*, 2023, pp. 3–4.
- [15] B. A. Becker, P. Denny, J. Finnie-Ansley, A. Luxton-Reilly, J. Prather, and E. A. Santos, “Programming is hard-or at least it used to be: Educational opportunities and challenges of AI code generation,” in *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*, 2023, pp. 500–506.
- [16] J. Penney, J. F. Pimentel, I. Steinmacher, and M. A. Gerosa, “Anticipating user needs: Insights from design fiction on conversational agents for computational thinking,” in *Chatbot Research and Design*, A. Følstad, T. Araujo, S. Papadopoulos, E. L.-C. Law, E. Luger, M. Goodwin, S. Hobert, and P. B. Brandtzaeg, Eds. Cham: Springer Nature Switzerland, 2024, pp. 204–219.
- [17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, E. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.
- [18] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altschmidt, S. Altman, S. Anadkat *et al.*, “Gpt-4 technical report,” *arXiv preprint arXiv:2303.08774*, 2023.
- [19] R. Choudhuri, D. Liu, I. Steinmacher, M. Gerosa, and A. Sarma, “How Far Are We? The Triumphs and Trials of Generative AI in Learning Software Engineering,” in *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, 2024, pp. 1–13.
- [20] A. J. Ko, “More than calculators: Why large language models threaten public education,” Jan 2024.
- [21] S. Lau and P. Guo, “From ‘ban it till we understand it’ to ‘resistance is futile’: How university programming instructors plan to adapt as chatgpt and github copilot,” in *2023 ACM Conference on International Computing Education Research-Volume 1*, 2023, pp. 106–121.
- [22] J. T. Liang, C. Yang, and B. A. Myers, “A large-scale survey on the usability of AI programming assistants: Successes and challenges,” in *Proceedings of the 46th IEEE/ACM International Conference on Software Engineering*, 2024, pp. 1–13.
- [23] K. Whiting, “How AI is helping to identify skills gaps and future jobs,” May 2023. [Online]. Available: <https://www.weforum.org/agenda/2023/05/ai-skills-gaps-future-jobs/>
- [24] A. Woodruff, R. Shelby, P. G. Kelley, S. Rouso-Schindler, J. Smith-Loud, and L. Wilcox, “How knowledge workers think generative AI will (not) transform their industries,” in *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 2024, pp. 1–26.
- [25] J. Prather, P. Denny, J. Leinonen, B. A. Becker, I. Albluwi, M. Craig, H. Keuning, N. Kiesler, T. Kohn, A. Luxton-Reilly *et al.*, “The robots are here: Navigating the generative AI revolution in computing education,” in *Proceedings of the 2023 Working Group Reports on Innovation and Technology in Computer Science Education*, 2023, pp. 108–159.
- [26] T. Malik, Y. Dwivedi, N. Kshetri, L. Hughes, E. L. Slade, A. Jeyaraj, A. K. Kar, A. M. Baabdullah, A. Koohang, V. Raghavan *et al.*, “‘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*, vol. 71, p. 102642, 2023.
- [27] P. Rajabi, P. Taghipour, D. Cukierman, and T. Doleck, “Exploring chatgpt’s impact on post-secondary education: A qualitative study,” in *Proceedings of the 25th Western Canadian Conference on Computing Education*, 2023, pp. 1–6.
- [28] I. Joshi, R. Budhiraja, P. D. Tanna, L. Jain, M. Deshpande, A. Srivastava, S. Rallapalli, H. D. Akolekar, J. S. Challa, and D. Kumar, “From ‘let’s google’ to ‘let’s chatgpt’: Student and instructor perspectives on the influence of llms on undergraduate engineering education,” *arXiv preprint arXiv:2309.10694*, 2023.
- [29] R. Raman, S. Mandal, P. Das, T. Kaur, J. Sanjanasri, and P. Nedungadi, “University students as early adopters of chatgpt: Innovation diffusion study,” 2023.
- [30] T. Wang, D. Vargas-Diaz, C. Brown, and Y. Chen, “Towards adapting computer science courses to AI assistants’ capabilities,” *arXiv preprint arXiv:2306.03289*, 2023.
- [31] C. Zastudil, M. Rogalska, C. Kapp, J. Vaughn, and S. MacNeil, “Generative ai in computing education: Perspectives of students and instructors,” in *2023 IEEE Frontiers in Education Conference (FIE)*. IEEE, 2023, pp. 1–9.
- [32] S. Barke, M. B. James, and N. Polikarpova, “Grounded copilot: How programmers interact with code-generating models,” *Proceedings of the ACM on Programming Languages*, vol. 7, no. OOPSLA1, pp. 85–111, 2023.
- [33] M. Kazemitabaar, J. Chow, C. K. T. Ma, B. J. Ericson, D. Weintrop, and T. Grossman, “Studying the effect of AI code generators on supporting novice learners in introductory programming,” in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 2023, pp. 1–23.
- [34] H. Mozannar, G. Bansal, A. Fourney, and E. Horvitz, “Reading between the lines: Modeling user behavior and costs in ai-assisted programming,” *arXiv preprint arXiv:2210.14306*, 2022.
- [35] S. I. Ross, F. Martinez, S. Houde, M. Muller, and J. D. Weisz, “The programmer’s assistant: Conversational interaction with a large language model for software development,” in *Proceedings of the 28th International Conference on Intelligent User Interfaces*, 2023, pp. 491–514.
- [36] P. Vaithilingam, T. Zhang, and E. L. Glassman, “Expectation vs. experience: Evaluating the usability of code generation tools powered by large language models,” in *Chi conference on human factors in computing systems extended abstracts*, 2022, pp. 1–7.
- [37] F. F. Xu, B. Vasilescu, and G. Neubig, “In-ide code generation from natural language: Promise and challenges,” *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 31, no. 2, pp. 1–47, 2022.
- [38] A. Ziegler, E. Kalliamvakou, X. A. Li, A. Rice, D. Rifkin, S. Simister, G. Sittampalam, and E. Aftandilian, “Productivity assessment of neural code completion,” in *Proceedings of the 6th ACM SIGPLAN International Symposium on Machine Programming*, 2022, pp. 21–29.
- [39] J. Prather, B. N. Reeves, P. Denny, B. A. Becker, J. Leinonen, A. Luxton-Reilly, G. Powell, J. Finnie-Ansley, and E. A. Santos, “‘it’s weird that it knows what i want’: Usability and interactions with copilot for novice programmers,” *ACM Transactions on Computer-Human Interaction*, vol. 31, no. 1, pp. 1–31, 2023.
- [40] C. Bird, D. Ford, T. Zimmermann, N. Forsgren, E. Kalliamvakou, T. Lowdermilk, and I. Gazit, “Taking flight with copilot,” *Communications of the ACM*, vol. 66, no. 6, pp. 56–62, 2023.

- [41] S. Amershi, D. Weld, M. Vorvoreanu, A. Fournery, B. Nushi, P. Collisson, J. Suh, S. Iqbal, P. N. Bennett, K. Inkpen *et al.*, “Guidelines for human-AI interaction,” in *Proceedings of the 2019 chi conference on human factors in computing systems*, 2019, pp. 1–13.
- [42] J. Prather, B. N. Reeves, J. Leinonen, S. MacNeil, A. S. Randrianasolo, B. A. Becker, B. Kimmel, J. Wright, and B. Briggs, “The widening gap: The benefits and harms of generative AI for novice programmers,” in *Proceedings of the 2024 ACM Conference on International Computing Education Research-Volume 1*, 2024, pp. 469–486.
- [43] J. Skripchuk, J. Bacher, and T. Price, “An investigation of the drivers of novice programmers’ intentions to use web search and GenAI,” in *Proceedings of the 2024 ACM Conference on International Computing Education Research-Volume 1*, 2024, pp. 487–501.
- [44] N. A. Ernst and G. Bavota, “AI-driven development is here: Should you worry?” *IEEE Software*, vol. 39, no. 2, pp. 106–110, 2022.
- [45] B. A. Becker, P. Denny, R. Pettit, D. Bouchard, D. J. Bouvier, B. Harrington, A. Kamil, A. Karkare, C. McDonald, P.-M. Osera *et al.*, “Compiler error messages considered unhelpful: The landscape of text-based programming error message research,” *Proceedings of the working group reports on innovation and technology in computer science education*, pp. 177–210, 2019.
- [46] J. Finnie-Ansley, P. Denny, B. A. Becker, A. Luxton-Reilly, and J. Prather, “The robots are coming: Exploring the implications of openai codex on introductory programming,” in *Proceedings of the 24th Australasian Computing Education Conference*, 2022, pp. 10–19.
- [47] S. Sarsa, P. Denny, A. Hellas, and J. Leinonen, “Automatic generation of programming exercises and code explanations using large language models,” in *Proceedings of the 2022 ACM Conference on International Computing Education Research-Volume 1*, 2022, pp. 27–43.
- [48] J. Savelka, A. Agarwal, C. Bogart, Y. Song, and M. Sakr, “Can generative pre-trained transformers (gpt) pass assessments in higher education programming courses?” in *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1*, 2023, pp. 117–123.
- [49] J. Leinonen, P. Denny, S. MacNeil, S. Sarsa, S. Bernstein, J. Kim, A. Tran, and A. Hellas, “Comparing code explanations created by students and large language models,” in *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1*, 2023, pp. 124–130.
- [50] S. MacNeil, A. Tran, A. Hellas, J. Kim, S. Sarsa, P. Denny, S. Bernstein, and J. Leinonen, “Experiences from using code explanations generated by large language models in a web software development e-book,” in *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*, 2023, pp. 931–937.
- [51] M. Wermelinger, “Using GitHub copilot to solve simple programming problems,” in *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*, 2023, pp. 172–178.
- [52] J. Leinonen, A. Hellas, S. Sarsa, B. Reeves, P. Denny, J. Prather, and B. A. Becker, “Using large language models to enhance programming error messages,” in *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*, 2023, pp. 563–569.
- [53] C. K. Y. Chan and K. K. Lee, “The AI generation gap: Are Gen Z students more interested in adopting generative AI such as chatgpt in teaching and learning than their Gen X and millennial generation teachers?” *Smart Learning Environments*, vol. 10, no. 1, p. 60, 2023.
- [54] A. Padiyath, X. Hou, A. Pang, D. Viramontes Vargas, X. Gu, T. Nelson-Fromm, Z. Wu, M. Guzdial, and B. Ericson, “Insights from social shaping theory: The appropriation of large language models in an undergraduate programming course,” in *Proceedings of the 2024 ACM Conference on International Computing Education Research-Volume 1*, 2024, pp. 114–130.
- [55] N. W. Kim, H.-K. Ko, G. Myers, and B. Bach, “Chatgpt in data visualization education: A student perspective,” *arXiv preprint arXiv:2405.00748*, 2024.
- [56] Anonymous, “Supplemental Material for GenAI Utilization Among SE Students,” Anonymous, October 2024. [Online]. Available: <https://doi.org/10.5281/zenodo.13882829>
- [57] D. R. Krathwohl, “A revision of bloom’s taxonomy: An overview,” *Theory into practice*, vol. 41, no. 4, pp. 212–218, 2002.
- [58] M. Forehand *et al.*, “Bloom’s taxonomy: Original and revised,” *Emerging perspectives on learning, teaching, and technology*, vol. 8, pp. 41–44, 2005.
- [59] I. Cabezas, R. Segovia, P. Caratozzolo, and E. Webb, “Using software engineering design principles as tools for freshman students learning,” in *2020 IEEE Frontiers in Education Conference (FIE)*. IEEE, 2020, pp. 1–5.
- [60] J. A. Bargh and T. L. Chartrand, “The mind in the middle,” *Handbook of research methods in social and personality psychology*, vol. 2, pp. 253–285, 2000.
- [61] A. Furnham and H. C. Boo, “A literature review of the anchoring effect,” *The journal of socio-economics*, vol. 40, no. 1, pp. 35–42, 2011.
- [62] J. W. Creswell and C. N. Poth, *Qualitative inquiry and research design: Choosing among five approaches*. Sage publications, 2016.
- [63] V. Braun and V. Clark, “Using thematic analysis in psychology,” *Qualitative research in psychology*, vol. 3, no. 2, pp. 77–101, 2006.
- [64] V. Braun and V. Clarke, “Conceptual and design thinking for thematic analysis,” *Qualitative Psychology*, vol. 9, no. 1, p. 3, 2022.
- [65] “Atlas.ti (version 3) [computer software],” 2024, accessed: 2024-09-07. [Online]. Available: <https://atlasti.com/>
- [66] N. McDonald, S. Schoenebeck, and A. Forte, “Reliability and inter-rater reliability in qualitative research: Norms and guidelines for CSCW and HCI practice,” *Proceedings of the ACM on human-computer interaction*, vol. 3, no. CSCW, pp. 1–23, 2019.
- [67] J. J. Francis, M. Johnston, C. Robertson, L. Glidewell, V. Entwistle, M. P. Eccles, and J. M. Grimshaw, “What is an adequate sample size? operationalising data saturation for theory-based interview studies,” *Psychology and health*, vol. 25, no. 10, pp. 1229–1245, 2010.
- [68] S. Biswas, “Evaluating errors and improving performance of chatgpt,” *International Journal of Clinical and Medical Education Research*, vol. 2, no. 6, pp. 182–188, 2023.
- [69] J. L. Plass, R. Moreno, and R. Brünken, “Cognitive load theory,” 2010.
- [70] A. Bandura *et al.*, “Social foundations of thought and action,” *Englewood Cliffs, NJ*, vol. 1986, no. 23-28, p. 2, 1986.
- [71] E. L. Deci and R. M. Ryan, “Self-determination theory,” *Handbook of theories of social psychology*, vol. 1, no. 20, pp. 416–436, 2012.
- [72] F. D. Davis, R. Bagozzi, and P. Warshaw, “Technology acceptance model,” *J Manag Sci*, vol. 35, no. 8, pp. 982–1003, 1989.
- [73] Harvard Office of Undergraduate Education, “AI Guidance & FAQs,” <https://oue.fas.harvard.edu/ai-guidance-faqs>, 2023.
- [74] UCLA Center for the Advancement of Teaching, “Guidance for the use of Generative AI,” <https://teaching.ucla.edu/resources/guidance-for-the-use-of-generative-ai/>, 2023.
- [75] M. Amoozadeh, D. Daniels, D. Nam, A. Kumar, S. Chen, M. Hilton, S. Srinivasa Ragavan, and M. A. Alipour, “Trust in Generative AI among students: An exploratory study,” in *55th ACM Technical Symposium on Computer Science Education V. 1*, 2024, pp. 67–73.
- [76] R. Choudhuri, B. Trinkenreich, R. Pandita, E. Kalliamvakou, I. Steinmacher, M. Gerosa, C. Sanchez, and A. Sarma, “What Guides Our Choices? Modeling Developers’ Trust and Behavioral Intentions Towards GenAI,” *arXiv preprint arXiv:2409.04099*, 2024.
- [77] C. Bird, D. Ford, T. Zimmermann, N. Forsgren, E. Kalliamvakou, T. Lowdermilk, and I. Gazit, “Taking flight with copilot: Early insights and opportunities of ai-powered pair-programming tools,” *Queue*, vol. 20, no. 6, pp. 35–57, 2022.
- [78] J. White, Q. Fu, S. Hays, M. Sandborn, C. Olea, H. Gilbert, A. El-nashar, J. Spencer-Smith, and D. C. Schmidt, “A prompt pattern catalog to enhance prompt engineering with chatgpt,” *arXiv preprint arXiv:2302.11382*, 2023.
- [79] S. B. Merriam and E. J. Tisdell, *Qualitative research: A guide to design and implementation*. John Wiley & Sons, 2015.
- [80] N. K. Denzin and Y. S. Lincoln, *The Sage handbook of qualitative research*. sage, 2011.