Shh! Don't Tell Them I Use GenAI: Exploring How Participants Use Generative AI in Hackathons

Pawan Acharya
Northern Arizona University
Flagstaff, USA
pa577@nau.edu

Tayana Conte
UFAM - Federal University of Amazonas
Manaus, Amazonas, Brazil
tayanaconte@gmail.com

Jacob Penney
Northern Arizona University
Flagstaff, USA
jmp458@nau.edu

Pedro Oliveira
Northern Arizona University
Flagstaff, Arizona, USA
pro32@nau.edu

Misan Paul Etchie
Northern Arizona University
Flagstaff, Arizona, USA
mpe45@nau.edu

Jared Duval
Northern Arizona University
Flagstaff, Arizona, USA
jared.duval@nau.edu

Bruno Gadelha
Federal University of Amazonas
Manaus, Amazonas, Brazil
bruno@icomp.ufam.edu.br

Marco Aurélio Gerosa
Northern Arizona University
Flagstaff, Arizona, USA
marco.gerosa@nau.edu

Igor Steinmacher
Northern Arizona University
Flagstaff, Arizona, USA
igor.steinmacher@nau.edu

Abstract

Generative AI (GenAI) tools are increasingly used in software development, yet little is known about how teams use them collaboratively in real-world, time-constrained settings. We studied a 2-day hackathon where participants were encouraged to use GenAI, combining 16 observations with 9 follow-up interviews. We found that GenAI use was selective, often focused on lower-risk tasks such as simple coding and idea refinement. Participants described using GenAI to gather and refine ideas, sometimes turning them into workable solutions, but were often cautious due to concerns about accuracy, difficulty crafting prompts, and discomfort with openly using AI. These findings suggest that GenAI use in collaborative settings is shaped by technical capabilities, social expectations, and user confidence. We discuss implications for designing environments that support transparent and effective AI use.

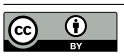
CCS Concepts

• **Software and its engineering** → **Software development process management**; • **Human-centered computing** → *Collaborative and social computing*; *Empirical studies in collaborative and social computing*.

Keywords

Hackathons; Generative AI; Human–AI Collaboration; Social Stigma

ACM Reference Format:
Pawan Acharya, Tayana Conte, Jacob Penney, Pedro Oliveira, Misan Paul Etchie, Jared Duval, Bruno Gadelha, Marco Aurélio Gerosa, and Igor Steinmacher. 2026. Shh! Don't Tell Them I Use GenAI: Exploring How Participants Use Generative AI in Hackathons. In *34th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (FSE Companion '26), July 5–9, 2026, Montreal, QC, Canada.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3803437.3806711



FSE Companion '26, Montreal, QC, Canada

ACM ISBN 979-8-4007-2636-1/2026/07
https://doi.org/10.1145/3803437.3806711

1 Introduction

Generative AI (GenAI) tools such as ChatGPT and GitHub Copilot have quickly become part of everyday software development practice. Multiple developer surveys suggest that these tools are widely adopted: for example, StackOverflow survey 2025 shows that 84% of developers report using at least one AI coding assistant, and over half say they have tried them in professional contexts [16]. Rather than replacing developers, GenAI typically acts as a "co-pilot" assisting with routine tasks like boilerplate generation, debugging, and syntax recall [17]. Recent industry studies also highlight growing expectations around team productivity [32]

However, this rapid adoption has provoked debate. Supporters highlight productivity gains and even new creative possibilities, for example. Studies report that AI pair programmers can significantly boost developer productivity and even improve code quality [5]. On the other hand, critics point to significant risks like unreliable or wrong outputs [13], over-reliance on unverified suggestions [7, 41, 43], and the erosion of learning opportunities. Researchers have cautioned that heavy reliance on code assistants might hinder skill development [26], negatively impacting learning and retention over time [14]. Human–AI interaction research underscores that responsible use of such AI assistants depends on transparency, user control, and calibrated trust in the system's outputs [2, 33].

The challenges of GenAI adoption are not only technical. They also raise pressing social questions: "How do individuals and teams decide when to trust GenAI output? When is it verified or ignored? And under what circumstances is its use concealed from peers? These questions matter because real-world adoption of GenAI will depend not just on its capabilities, but on how people perceive its legitimacy, fairness, and risks. Without understanding how developers make these choices under time pressure and social visibility,

we risk designing tools and guidelines that fail to support actual team practices.

As GenAI tools become more prevalent in software development, it is important to understand how humans decide when to use them, when to trust them, and when to disclose or hide such use from others. These decisions are not only happening at the individual level, but are also often contingent on teams collectively deciding. Although recent research focuses on independent individuals, we still lack an understanding of how collaborative and social aspects affect GenAI use in real-world settings.

Hackathons offer a way to study how developers collaborate under social and temporal constraints. Such events have been adopted in various domains to generate innovative solutions, foster learning, build and expand communities [20]. These events typically require participants to rapidly generate ideas, solutions, and build functioning prototypes in a short time frame [15, 23, 38]. During these events, teams collaboratively negotiate design decisions and prepare their projects for ev aluation. Beyond technical production, hackathons involve socio-technical dynamics such as informal learning, peer collaboration, and social signaling, which influence how participants present themselves and engage with tools [11, 23].

For Hackathons, GenAI offers several well-recognized benefits, like quickly generating boilerplate code, debugging, ideating, or polishing presentations. However, hackathons also introduce significant challenges. Verification is difficult under time pressure, participants may lack the domain knowledge to craft effective prompts, and the competitive nature may influence whether GenAI use is openly disclosed. In short, hackathons serve as a "stress test" for how GenAI is applied in collaborative software work.

Despite growing enthusiasm, little empirical research has examined how GenAI is used in fast-paced, collaborative settings. Prior work has primarily focused on individual developers, either in controlled environments or through usage logs from systems such as Copilot [39]. While these studies offer insights into productivity and usability, they provide a limited understanding of how GenAI shapes team dynamics, the negotiation of shared norms, and situated collaboration practices [28]. This study addresses the following research question: *How do hackathon participants use GenAI in collaborative, time-constrained settings, and what factors shape their adoption, trust, and disclosure decisions?*

To address this question, we conducted a study of a two-day university hackathon. Our team engaged in participant observation throughout the event and conducted post-event interviews. This paper makes the following contributions:

(1) An empirical account of GenAI use in a real hackathon;
(2) Evidence of adoption barriers rooted in trust, prompt literacy, and peer norms;
(3) Evidence that adoption is shaped by trust, prompt literacy, and social visibility;
(4) A compact model of an AI-supported creative interaction pattern (*gather* → *synthesize* → *reformulate* → *operationalize*);
(5) A characterization of recurring patterns in how participants engaged with GenAI during problem solving;
(6) Practical implications for designing learning environments and events that support transparent and effective AI use.

2 Related Work

In the following, we discuss (i) hackathons, (ii) creativity support tools that facilitate team ideation, and (iii) generative AI in software engineering, particularly its role in coding and teamwork.

2.1 Hackathons and Software Development

Hackathons are short, intensive events where ad hoc teams build prototypes under severe time constraints [9]. They now span corporate, academic, and civic settings, functioning as time-bounded "coopetition" spaces that blend collaboration and competition [12]. Prior research characterizes hackathons as environments where participants balance exploration and delivery [38] and as innovation contests that support networking and experimentation [15].

Beyond the event itself, studies show that hackathons can generate longer-term outcomes. Some projects continue after the event, particularly when aligned with organizational needs [23]. A systematic review reports outcomes ranging from innovation and learning to community building, while noting sustainability challenges [20]. Because of their compressed timelines and collaborative intensity, hackathons have also been framed as experiential learning environments [9, 12]. These characteristics make them a useful "living lab" for studying how emerging technologies, such as GenAI, are adopted under time and social pressure.

2.2 Creativity Support Tools

Hackathons are not only coding sprints but also sites of collective creativity. Research on creativity support tools examines how technology scaffolds ideation, exploration, and production [27, 35]. Effective tools broaden the search space, support rapid iteration, and make alternatives visible [34]. Design principles such as exploratory play, collaboration, and reflection are especially salient under time constraints [27].

In hackathons, tools such as frameworks, APIs, and platforms shape what teams can accomplish [9]. More broadly, creativity support tools can structure idea generation, coordination, and convergence under pressure [27, 34]. GenAI represents a new class of creativity support tools that can generate code, text, or design suggestions. While such tools may expand teams' search space and accelerate prototyping, they also raise questions about transparency, authorship, trust, and shared understanding within teams.

2.3 Generative AI in Software Engineering

In software engineering, Generative AI assistants are commonly studied as tools for code generation, debugging, and documentation. Prior work shows that systems such as Copilot can accelerate routine programming tasks and provide useful starting points [30, 39]. However, usability challenges and difficulties in understanding or verifying outputs may reduce efficiency when suggestions are trusted uncritically [30]. Other studies highlight correctness and security concerns, showing that AI-generated code can introduce subtle bugs or vulnerabilities [24, 25]. These findings emphasize the need for calibrated trust.

Beyond code generation, GenAI broadens human–AI collaboration. Large language models have been studied as creative co-writers in domains such as scriptwriting, illustrating AI as a collaborative partner rather than merely a coding assistant [21]. This

raises issues of authorship, credit, and ethical concerns such as bias. Human–AI interaction research calls for transparency, user control, and feedback mechanisms to support appropriate reliance [2, 40].

Research also shows that effective prompting requires articulation and domain understanding [19, 42]. At the same time, social context influences adoption. Drawing on impression management theory [11], visible AI use may threaten perceived competence. Shifts in help-seeking behavior, including reduced reliance on public Q&A forums, suggest that GenAI's privacy and immediacy may reshape norms of assistance [39].

3 Method

This study used a qualitative approach to understand how hackathon teams engage with generative AI.

We studied a two-day game development hackathon hosted at Northern Arizona University in January 2025 as part of a global game development event. The event took place in a computing lab and included access to shared workstations, whiteboards, and presentation space. Sixteen participants registered and were grouped into five self-organized teams (3–4 members each). Participants were primarily undergraduate students in computer science and software engineering. Teams were free to select their own engines, frameworks, and libraries (e.g., Unity, Godot, C#, Python). Prizes were awarded for creativity and technical execution, but the judging criteria did not explicitly address GenAI use.

The organizers mentioned several times in the opening session that tools such as ChatGPT or Copilot could be used if teams wished, but they requested disclosure during presentations. This created a permissive but lightly structured environment, where participants had to decide whether and how to use AI tools.

3.1 Participants

Table 1 summarizes the teams' identifications, sizes, primary platforms, and whether GenAI use was visible during the event. Participants ranged in age from 19 to 26 years (median = 21), included 9 undergraduates, and had 0 to 4 prior hackathon experiences (median = 1). All participants reported some prior exposure to GenAI tools, though experience levels varied: two reported daily use of ChatGPT, three described occasional use, and four had only experimented briefly before the hackathon.

Participants reported varying levels of prior experience with GenAI tools. Two participants reported daily use (primarily ChatGPT), three described occasional use, and four had only experimented briefly before the hackathon. None reported consistent use of IDE-integrated assistants (e.g., Copilot), and no participants reported paying for premium access. This variation in prior experience is important for interpreting observed differences in trust, prompting behavior, and reliance on AI outputs.

Table 1: Summary of the hackathon participants/teams

Team	Size	Primary Platform	GenAI Use	GenAI Experience
Team A	4	Unity	Yes (debugging, code generation)	High
Team B	3	Twine	Yes (ideas and suggestion)	Moderate
Team C	2	Godot, GDScript	Minimal (AI Overview)	No report
Team D	5	Godot, GDScript	Minimal (concealed use)	low
Team E	2	Board games	No visible use	Rare

3.2 Researcher Role

The observation team was composed of four graduate students (two master's students and two Ph.D. students) and one faculty member. All five attended the hackathon as participant observers. None of the researchers joined a team or contributed to project development; instead, they circulated among groups, observed discussions, and documented practices in field notes. The observers did not hold formal positions of authority in the event itself. To minimize bias, the team adopted a descriptive stance [8, 37] during observations, recording what participants said and did rather than making evaluative judgments.

3.3 Data Collection

We collected data through two complementary strategies:

Observation. The team observed the full two-day hackathon, totaling approximately 16 hours of observation. Before the event, the lead researcher developed an observation guide that included a structured checklist of focal points such as team composition, idea generation, visible GenAI use, prompt formulation, error-handling, discussion patterns, disagreements, and role assignments. This guide was shared and discussed among all observers to ensure consistency. During the hackathon, the observers took notes, focusing on capturing descriptive accounts of interactions, behaviors, and contexts rather than evaluative judgments. After each session, notes were expanded into detailed narratives and compared across observers. The combined field notes added up to more than 30 pages, giving a detailed picture of how the teams interacted.

Interviews. We conducted semi-structured interviews within one week of the hackathon to capture participants' reflections while their experiences were still fresh. This was based on the Critical Incident Technique [10], a method used to collect observations of behavior by interviewing people about specific events (in our case, those we observed during the Hackathon). Each interview lasted up to 40 minutes (average 32 minutes) and was conducted in person, audio-recorded with consent, and transcribed verbatim. The interview explored participants' prior experience with GenAI, moments of use during the hackathon, perceived benefits and frustrations, strategies for formulating prompts, collaboration dynamics, and decisions about disclosure. To minimize recall bias, we provided contextual cues and examples from our observations to help participants reconstruct specific moments and reasoning.

3.4 Data Corpus

In total, the corpus comprised more than 30 pages of field notes and nine conducted interviews, of which four were fully transcribed and included in the analysis (approximately 62,000 words). The field notes capture real-time team interactions and observable GenAI use during the hackathon, while the interviews provide participants' reflective accounts of those practices. These are available as part of our supplemental material [3].

3.5 Analysis

We analyzed the data qualitatively through an iterative, inductive process [29]. Our goal was to identify recurring practices and patterns that characterized participants' engagement with GenAI

during the hackathon. The analysis was interpretive, seeking to understand how participants made sense of their experiences, how their interactions happened, and how things were built through collaborative work. Rather than applying a predefined theoretical framework, we allowed categories to emerge from the data. We detail the steps followed in the process in the following.

Open coding. We started the analysis by open-coding the field notes and interview transcripts to generate labels for concrete actions, experiences, and perceptions. Codes were inductively derived and anchored in participants' words and behaviors, coded from both the interview answers and interactional cues captured in observation notes. This produced a set of codes that represented the diverse ways participants engaged with GenAI and with one another. Following initial coding, we conducted multiple team-based review sessions to refine the code set. These meetings focused on examining overlaps and inconsistencies, questioning interpretations, and revisiting the data to ensure that codes accurately reflected participants' practices. Where appropriate, we merged codes (for example, "gathering diverse ideas" and "synthesizing different approaches") to consolidate meaning. New codes were also introduced based on a collective rereading of the transcripts.

Category consolidation. After defining and refining the initial codeset, we examined relationships between codes and grouped them into broader conceptual categories. Eight categories emerged from this analysis: (1) ideation and creativity processes, (2) team dynamics and role assignments, (3) communication patterns and conflict management, (4) participant expertise and knowledge sharing, (5) decision-making and problem-solving methods, (6) perceptions and utilization of LLMs, (7) external resource and platform usage, and (8) emotional and psychological experiences. These categories were not mutually exclusive; rather, they represented recurring clusters of activity that cut across different teams and moments. For example, "debugging with LLMs" was simultaneously coded as a technical problem-solving practice and as part of a broader category about perceptions of reliability.

Themes Definition. Finally, through multiple rounds of collaborative review, we integrated the axial categories into a smaller set of cross-cutting themes that synthesized the most salient insights about GenAI-supported collaboration in hackathons. Selective coding emphasized themes that spanned both the real-time field notes and retrospective interview accounts. These themes included how participants enlisted GenAI as a *coding partner*, how trust and prior experience shaped adoption, how prompt formulation acted as a barrier, how social stigma influenced disclosure, and how a recognizable GenAI-supported interaction pattern emerged (gathering → synthesizing → reformulating → operationalizing). These themes provide the structure for our Results section.

The analysis process was conducted collaboratively. While the first author conducted the initial open coding, the team met regularly to review code lists, discuss disagreements, and refine categories. This iterative sense-making process functioned as a check on individual interpretations and added reflexivity, given the researchers' varying levels of familiarity with hackathons and GenAI tools. Reflexive memos written after each coding session further documented emerging insights and potential biases, ensuring that the final themes were grounded in the data.

3.6 Ethical Considerations

This study was approved by the university's institutional review board (IRB). All participants provided informed consent. Participation was voluntary, and students could decline to be interviewed without consequence. To protect anonymity, each interview participant is assigned a unique identifier (P1–P9), which is used throughout the paper. Because several participants reported concealing GenAI use from peers due to perceived stigma, we took particular care in anonymizing quotes and descriptions to prevent unintended disclosure of individual practices.

The research team consisted of graduate students and a faculty member from the same institution, but none held evaluative authority over participants in the context of the hackathon. Reflexive memos were used to monitor potential bias arising from institutional proximity. We provided a $20 gift card to each interview participant as a token of appreciation.

3.7 Researcher Positionality

As researchers, we acknowledge that our own backgrounds may have influenced both data collection and interpretation. The observation team consisted of four graduate students, and one faculty member, all affiliated with the same university as the participants. Several members had prior experience participating in or mentoring hackathons, and all were familiar with GenAI tools. This insider perspective helped us quickly understand team interaction patterns and technical references, but it also carried the risk of assuming shared knowledge or overlooking unfamiliar practices.

To mitigate these risks, we used a structured observation guide, maintained descriptive field notes, and wrote reflexive memos after each observation session to surface potential biases. Coding was conducted iteratively, with discussions across the research team to challenge assumptions and incorporate multiple perspectives. These steps helped ensure that our findings remained grounded in participants' own accounts rather than our preconceptions.

3.8 Supplementary Material

We provide a supplementary package with the materials used in this study [3], including interview scripts, observation guides, field notes, processed transcripts, and coded qualitative analysis.

4 Results

In the following, we present the results of our participant observation and interviews, organized around the main themes that emerged from our qualitative approach. Our goal was to illustrate how participants used GenAI during the hackathon and show how technical and social forces shaped these practices.

The most pervasive theme we discovered was *social stigma*. Although organizers allowed open use of GenAI, participants feared negative peer judgment. This stigma triggered a cascade of behavioral adaptations. Participants managed the visibility of their AI use, recalibrated trust under peer scrutiny, relegated GenAI to "safe" marginal tasks, avoided prompting in front of teammates (which increased the cost of seeking help), and confined GenAI to the ideation periphery. Figure 1 synthesizes these cascading effects and provides a roadmap for the subsections that follow.

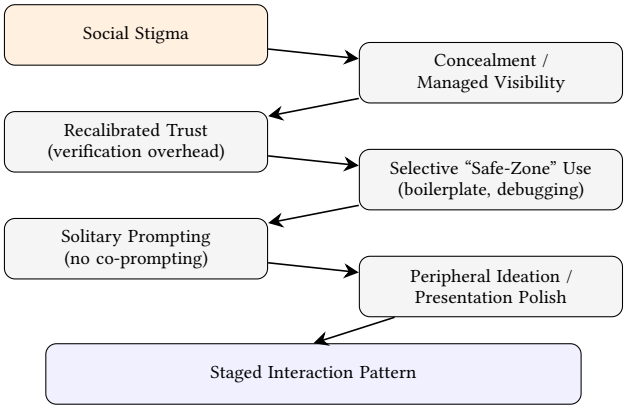


Figure 1: Stigma cascade: linear sequence from stigma through concealment, trust recalibration, selective "safe-zone" use, prompting, and peripheral ideation, culminating in a staged GenAI-supported recurring interaction pattern.

4.1 Visibility, Reputation, and Social Stigma

Many participants said they worried that using GenAI openly would make them look less competent. Even though the organizers allowed GenAI use, some participants felt that judges or teammates might see it as "cheating" or not doing the real work. *"when I was doing the digital stuff, I started by just looking for ideas online. It kind of feels still because of how kind of new chat GPT is, like it's kind of like cheating."* (P5, Team B) During the event, we noticed a few people trying to hide their use, for example, opening ChatGPT on a personal laptop away from others or quickly minimizing the window when someone walked by. Unlike documentation searches, which people often talked about out loud, AI prompts were usually typed quietly and kept private.

During interviews, participants mentioned weighing the benefits of GenAI against the risk of being judged, for example: *"I tried to solve it in my head, and if I couldn't do it in a couple of seconds, then I went to ChatGPT because it was faster"* (P3, Team A). Another participant shared a similar view about efficiency: *"It is hard to ask specific bug questions to each other since we're working on different games... ChatGPT was more efficient"* (P4, Team A).

Our observations matched what people said in interviews. For example, we saw that one participant used ChatGPT privately for debugging and did not tell their teammates, while another pair looked frustrated when the model's answers did not help. This shows that while GenAI helped people work faster, it brought social pressure. Participants worried that relying on AI might make them look less capable, so many chose to hide their use.

Key Insight.

These concerns about visibility and reputation did not end at the level of perception. They also shaped participants' concrete behaviors, influencing when and how GenAI was used during collaboration.

4.2 Managing Visibility: Disclosure, Delay, and Private Prompting

Because of the reputational concerns described earlier, many participants tried to make their GenAI use less noticeable. From our observations, participants often chose when to open ChatGPT carefully, waiting until they were working alone or when others weren't paying attention. One participant explained that they only used it when they lost key members: *"The only time we really pulled up ChatGPT was when our two other coders had left, and [P6] was the only one trying to figure out how to use Godot."* (P7, Team D)

We noticed participants quietly switching between tabs or copying and pasting text so it looked like it came from a tutorial or something they had found online. Prompting usually happened off-screen and was not shared during group discussions.

These patterns show that participants did not avoid GenAI completely. However, they just used it in ways that felt safer and less public.

Key Insight.

This shows that worries about being judged made people careful not just about when they used AI, but also about how openly they talked about it with others.

4.3 Calibrated Trust Under Scrutiny

Participants' decisions about using GenAI were shaped mainly by their confidence in the technology. Across interviews and observations, they described or demonstrated limited trust in model accuracy. P8, for instance, made their skepticism explicit: *"I heard other teams talking about using ChatGPT stuff. Personally, I don't trust a lot of that. Google Gemini has consistently proven me wrong. I don't trust programs like that to give me the answers and support I'm looking for."* (P8, Team E).

This lack of confidence was also reflected in practice. During the hackathon, [P1, Team A] stated that even when ChatGPT's instructions appeared correct, they still had to *"go into YouTube or Google"* to understand how to follow them. This behavior, observed repeatedly, showed that GenAI outputs were cross-checked against other sources before being used.

Observed interactions further showed that GenAI use was occasional and often individual rather than collaborative. [Participant from Team D] consulted ChatGPT privately while working on a coding problem, but did not discuss or share the output with teammates. [P7, Team D] also tried to use ChatGPT for animation code but expressed visible frustration when the suggestions failed to solve the issue. Afterward, they turned to YouTube tutorials and mentor guidance instead. These instances illustrated that participants preferred to verify or supplement GenAI advice with traditional resources such as documentation, videos, and human help.

Under hackathon time pressure, participants generally relied on these familiar resources rather than GenAI tools.

> *[Observational note, Team D] "P7 is looking for some(Bubble) designs for the Game background, and she switched to Pinterest as a guide."*

Their choices appeared motivated by efficiency and prior experience rather than by explicit concerns about reputation or visibility.

Key Insight.

Trust in GenAI was limited and conditional: participants used the tools experimentally, verified the results through other means, and reverted to conventional supports when accuracy or clarity was uncertain.

4.4 Safe-Zone Uses: GenAI as a Coding Partner

From our observations, participants created what could be called "safe zones," where using GenAI felt acceptable and low-risk. In these situations, ChatGPT was mainly used like a coding helper for writing small pieces of code, fixing errors, or explaining what went wrong. These tasks were easy to check and didn't feel like they replaced the real work of building the game. One participant explained their approach: *"I tried to solve it in my head; if I couldn't in a couple of seconds, then I went to ChatGPT. For generating scripts, it was faster. It is hard to ask specific bug questions to each other because we're working on different games, so you'd have to learn the context. ChatGPT was more efficient"* (P4, Team A).

Others talked about using ChatGPT for setting things up or dealing with common bugs at the start of the projects: *"Just bug fixing, saying, 'hey, what if you made a script that did this¿ I used ChatGPT to answer errors, understand what they meant, and where to look."* (P1, Team A).

However, when the task involved something new or complicated, ChatGPT often struggled. One participant shared that: *"it wasn't effective in generating code. The dancing game was supposed to be like Dance Dance Revolution, where the arrows would come down on the screen, but ChatGPT had a hard time explaining it and generating code that actually worked"* (P3, Team A).

Key Insight.

From what we observed, GenAI helped most with smaller coding tasks like debugging or setting up things that could be checked quickly or replaced if needed. When it came to bigger or more creative features, many participants were more careful, and some avoided them completely.

4.5 The Cost of Asking: Prompt Formulation Without Co-Prompting

Across teams, GenAI use tended to be an individual rather than a shared activity. No team was seen co-prompting on a shared screen, and most prompts were entered privately on personal devices. This pattern suggests that working together on prompts might have felt awkward or uncomfortable, even though using GenAI was allowed. Because of this, participants couldn't rely on teammates to help refine questions, choose better technical terms, or build on each other's ideas. When domain knowledge was limited (for example, about Godot's interface), this made it harder to get useful results:

"I use AI pretty frequently . . . but I only ask AI questions if I know how. With my limited knowledge of Godot, I wouldn't be able to ask it in a way to get what I needed immediately. Even perfect instructions would still send me to YouTube or Google to figure out the menus" (P6, Team D).

Key Insight.

In practice, participants often tried a single prompt and then gave up, turning to videos or documentation instead. The same hesitation that made participants avoid showing GenAI use in public also meant that prompt writing became a private, trial-and-error process.

4.6 Peripheral Ideation and Presentation Polish

In our observations, most of the creative work, like discussing game mechanics, sketching interfaces, and referencing other games, was done directly between teammates. No teams used GenAI tools during brainstorming or early design discussions. Participants focused on talking through ideas together or testing them out, without turning to AI for suggestions or examples.

GenAI mostly appeared near the end of the process, when teams were wrapping up or polishing their projects. These uses were simple and low stakes, like checking for title ideas. One participant explained: *"I mean, at the very end, when I had to be like the end screen. I didn't really know what to write for the title, because, like, we didn't have a victory screen, and there was no like you lose screen. So I was like, what is a creative way to like make an end screen? So I asked Chat GPT like, Oh, give me some suggestions to like as add a title"* (P4, Team A).

> *[Observational note, Team A] "During their final preparation, one participant opened ChatGPT to ask for title ideas but closed it soon after getting a few suggestions."*

From what we observed, GenAI was mostly absent during the creative stages and only appeared near the end of the process for small finishing tasks. It stayed on the edges of the creative work, something that participants used quietly, when it felt safe or convenient, rather than as part of their main design process.

Key Insight.

This kind of limited and cautious use was different from what we noticed in technical work, where GenAI was more actively used for fixing problems or debugging, but still kept separate from the team's creative and collaborative decisions.

4.7 A Staged Interaction Pattern, Shaped by Social Norms

From our interviews and observations, we identified a four-stage recurring interaction pattern that shows how participants used GenAI tools like ChatGPT for problem-solving during the hackathon. These stages were: *gathering*, *synthesizing*, *reformulating*, and *operationalizing*. While this recurring interaction pattern appeared technical, it was also shaped by social norms about what kinds of AI use were acceptable, visible, or worth mentioning. Some participants chose to keep their AI use private or limited to low-stakes tasks, reflecting uncertainty about how others would perceive it: *"We really didn't think about using ChatGPT at all... other than that, I really don't think we used ChatGPT or any AI for anything"* (P4, Team A). This suggests that, beyond technical efficiency, participants also managed the social visibility of AI in their recurring interaction patterns.

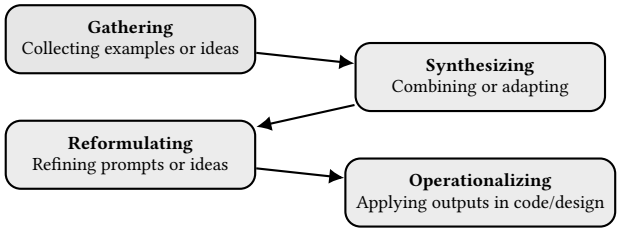


Figure 2: Staged interaction pattern: gathering, synthesizing, reformulating, and operationalizing.

In the first stage, *gathering*, participants used ChatGPT to collect examples, tutorials, and ideas. They often treated it like a large searchable database that could quickly bring together many options at once: *"I went to ChatGPT to see if it could help me since it has like the whole internet as a database... it could give examples and stuff, and like how to set up that table"* (P5, Team B).

This helped participants understand what resources existed and what was missing, such as specific game tutorials or examples relevant to their needs.

In the second stage, *synthesizing*, participants combined ideas or adapted partial examples that the model provided: *"Even though this isn't what you are looking for, it has some good basic techniques... I ended up using a combination of the different things it told me"* (P5, Team B). Here, ChatGPT acted as a bridge between ideas, helping them integrate multiple resources instead of replacing their own reasoning.

In the third stage, *reformulating*, participants iteratively refined prompts or ideas until the output matched what they wanted. This process often included giving feedback to the model to correct or clarify its responses: *"I had to ask it a bunch of times and give feedback about my code-block setup... eventually I got what I wanted"* (P5, Team B). Participants described this stage as helpful for rearticulating ideas and clarifying next steps, though it often took several tries.

Finally, in the *operationalizing* stage, participants translated AI suggestions into practical actions, writing or fixing code, setting up animations, or refining UI elements. For instance, one participant noted: *"It was helpful for like getting what I wanted, like from an art standpoint for our game... reformulating ideas and getting what we thought might be useful"* (P5, Team B).

Key Insight.

Participants' recurring interaction patterns showed that GenAI supported technical progress in multiple ways, but participants still managed its visibility carefully. They were more comfortable using it privately for exploration or refinement, and less so for visible or creative tasks, reflecting the social uncertainty around acceptable AI use.

5 Discussion

Our findings reveal that participants' use of GenAI in hackathons was not only a matter of technical utility but also of social negotiation. The discussion below unpacks how issues of trust, literacy, and legitimacy are intertwined with existing social norms to shape when and how GenAI was used. We connect these observations to prior work in human–AI interaction, educational practice, and collaborative software development, outlining broader implications for designing and evaluating AI-supported teamwork.

Calibration and Trust. Prior work in human–AI interaction emphasizes the need for mechanisms that help users judge when to trust model outputs [2, 40]. Our findings extend this by showing that, in collaborative and time-pressured settings, calibration is not only technical but also reputational. Participants weighed not just whether an output was correct, but whether being seen relying on a wrong suggestion might harm their credibility with teammates. This suggests that design cues such as provenance indicators, confidence levels, or modular, testable outputs should aim to reduce both the cognitive effort of verifying accuracy and the social risk of being visibly mistaken. In short, trust calibration is entangled with impression management.

Prompt Literacy as Situated Know-How. Previous research shows that effective prompting relies on articulation, iteration, and repair [2, 40, 42]. Our study adds that prompt literacy is also shaped by social context. Because participants avoided co-prompting, they missed opportunities to share domain vocabulary or collectively refine phrasing. This individualization made it harder to "ask the right question," especially when their knowledge of the target tool, like Godot UI terminology, was partial. Support for prompting should therefore go beyond individual skill-building. Context-aware templates or collaborative prompting interfaces could lower articulation costs and normalize shared prompting—similar to how teams already co-debug or co-search in conventional practice.

Social Norms and Impression Management. Even though organizers permitted GenAI use with disclosure, participants managed it carefully to avoid negative peer judgment. Goffman's theory of impression management [11] helps explain this tension between formal rules and informal norms: GenAI use became part of the "presentation of self," often concealed to maintain an image of competence. This complicates current disclosure policies. Simply requiring participants to report AI use may intensify stigma if disclosure signals suspicion. Instead, judging rubrics could reward transparent, well-documented AI use as a form of reflective practice. This would make visible GenAI use as a professional skill rather than a shortcut, reducing the social penalties participants feared.

Educational Context and Legitimacy. Our study took place in a university setting, participants' attitudes toward GenAI were shaped by academic norms that often frame AI assistance as questionable or even as academic misconduct [1, 6]. Even though the hackathon permitted GenAI use, several participants acted as if disclosure might still be risky. This helps explain why prompting remained private and why GenAI was positioned in "safe" parts of the recurring interaction pattern. Bridging this gap between educational norms and evolving professional practice may require explicit guidance on when and how AI use is acceptable, and reflective activities that reward transparent, critical engagement rather than silent efficiency.

Interaction Patterns Shaped by Stigma. Participants did not reject GenAI outright but incorporated it into a careful, staged interaction pattern: gather, synthesize, reformulate, operationalize, which was cautious, incremental, and often private. This adds nuance to existing HAI perspectives that treat recurring interaction integration mainly as a cognitive process [36]. Our findings show that social stigma shaped where in the interaction pattern GenAI was acceptable: it was welcomed for peripheral or verifiable tasks but avoided in public or central ones [42]. For researchers, this highlights that adoption is socially regulated as much as technically constrained; for practitioners, it underscores that designing AI for team settings requires anticipating how visibility, authorship, and credit influence appropriate use.

Implications Beyond the Hackathon. Our study focused on a hackathon, similar patterns such as trust shaped by social risk, individualized prompting, and cautious integration likely appear in classrooms, design sprints, and workplaces. Adoption depends on accuracy or literacy and the social legitimacy [4, 22]. For educators, organizers, and tool designers, this means that effective integration requires both technical scaffolds and cultural change [36]. Encouraging visible, critical use of GenAI may be as important as improving model performance if teams are to move from hidden, peripheral reliance toward open, confident collaboration with AI.

Broader Reflections on Human–AI Collaboration. Our findings show that using GenAI is not only about how well the tools work or how useful they are. It is also about how social norms and visibility shape what people see as appropriate or fair collaboration [2, 18]. During the hackathon, participants constantly balanced being efficient with looking competent and being fair to others. Future research on human-AI collaboration should look beyond individual skills or system design, and also consider how people negotiate whether AI use is acceptable in social settings [31]. These dynamics can help us understand how AI tools are changing what it means to be creative, knowledgeable, and to collaborate.

6 Limitations

As in any scientific work, this study has a few limitations that should be kept in mind when interpreting the findings. Since the study was conducted in one university hackathon with small number of teams, the findings cannot be generalized. They represent what happened within this setting. Social and learning conditions affected how participants explained and used GenAI, so things might appear reasonably different in industry or larger hackathons.

Only nine participants took part in the follow-up interviews, which is 54% of the total participants, which means the data may reflect the views of people who were more comfortable talking about their experiences.

Although interviews were conducted shortly after the hackathon, participants' recollections may still be influenced by recall bias. We sought to mitigate this by providing examples and observational prompts during interviews.

Our observations focused on what could be seen. Therefore, there might be a visibility bias in the data. Some participants may have used GenAI privately or tried to hide it, so subtle uses might be missing. The low amount of visible GenAI use we observed could therefore reflect concealment rather than a lack of interest. The interview data may also include recall or self-presentation bias, as participants might not remember everything accurately or may have described their behavior in ways that sound more acceptable.

Because the event was university-based, participants were also shaped by academic cultures where GenAI use is sometimes discouraged or treated as academic misconduct. This context may have influenced how they perceived the legitimacy of GenAI use and their willingness to disclose it. Their caution may therefore reflect broader institutional norms rather than purely personal hesitation.

We used open coding to organize the data and identify patterns, but these decisions were informed by our own reading of the notes and by conversations with my advisor and colleague. Another researcher might have coded or explained things differently. The idea of "stigma," for instance, came from how people acted or spoke rather than something that would be measurable directly.

Also, our analysis was influenced by the method we applied and by our own expectations as researchers interested in GenAI. Using Goffman's ideas about performance and visibility helped us understand why participants hid their AI use, but it might have led us to pay less attention to other factors, like usability or teamwork.

7 Conclusion

This paper presented a study of GenAI use in a game development hackathon. While participants appropriated GenAI as scaffolding for coding, debugging, and occasional ideation, their practices were shaped as much by social dynamics as by technical affordances. We observed a staged interaction pattern: gathering, synthesizing, reformulating, and operationalizing through which GenAI became a pragmatic, if peripheral, collaborator. Yet three barriers consistently moderated adoption: calibrated trust in output quality, challenges in prompt formulation, and social stigma that made visible use feel risky even when formally allowed.

These findings extend current understandings of human–AI collaboration by showing how impression management and peer norms constrain AI appropriation in time-pressured, collective settings. For designers and organizers, they point toward concrete directions: transparency cues and verifiable steps to support calibrated trust; context-aware prompt scaffolds aligned with specific stacks and tools; and evaluation rubrics that normalize, rather than penalize, disclosed use. More broadly, our study suggests that the future of GenAI in collaborative creative work will hinge not only on improving model performance but also on fostering cultures where visible, critical, and reflective use is socially legitimate.

Acknowledgements

We acknowledge the use of GenAI tools (ChatGPT, Claude AI) for polishing the text and helping proofread the whole paper. We thank all the participants of the Hackathon for supporting our study. Authors of this work were partially supported by Northern Arizona University, the National Science Foundation grant #2236198 and CNPq grants #314797/2023-8 and #443934/2023-1.

References

[1] Muhammad Abbas, Farooq Ahmed Jam, and Tariq Iqbal Khan. 2024. Is it harmful or helpful? Examining the causes and consequences of generative AI usage

among university students. *International journal of educational technology in higher education* 21, 1 (2024), 10.

[2] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N Bennett, Kori Inkpen, et al. 2019. Guidelines for Human-AI interaction. In *Proceedings of the 2019 CHI conference on human factors in computing systems*. 1–13.

[3] Anonymous. 2025. Replication Package. https://doi.org/10.5281/zenodo.17430278

[4] Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big?. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*. 610–623.

[5] Sergio Cavalcante, Erick Ribeiro, and Ana Oran. 2025. The Impact of AI Tools on Software Development: A Case Study with GitHub Copilot and Other AI Assistants. In *Proceedings of the 27th International Conference on Enterprise Information Systems - Volume 2: ICEIS*. INSTICC, SciTePress, 245–252. doi:10.5220/0013294700003929

[6] Debby RE Cotton, Peter A Cotton, and J Reuben Shipway. 2024. Chatting and cheating: Ensuring academic integrity in the era of ChatGPT. *Innovations in education and teaching international* 61, 2 (2024), 228–239.

[7] Otávio Cury and Guilherme Avelino. 2025. The Impact of Generative AI on Code Expertise Models: An Exploratory Study. *arXiv preprint arXiv:2507.08160* (2025).

[8] Robert M Emerson, Rachel I Fretz, and Linda L Shaw. 2011. *Writing ethnographic fieldnotes*. University of Chicago Press.

[9] Jeanette Falk, Yiyi Chen, Janet Rafner, Mike Zhang, Johannes Bjerva, and Alexander Nolte. 2025. How Do Hackathons Foster Creativity? Towards Automated Evaluation of Creativity at Scale. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–23.

[10] John C Flanagan. 1954. The critical incident technique. *Psychological bulletin* 51, 4 (1954), 327.

[11] Erving Goffman. 1959. The presentation of self in everyday life. In *Social theory re-wired*. Routledge, 450–459.

[12] Seija Halvari, Anu Suominen, Jari Jussila, Vilho Jonsson, and Johan Bäckman. 2019. Conceptualization of hackathon for innovation management. In *ISPIM The International Society for Professional Innovation Management*.

[13] Samia Kabir, David N Udo-Imeh, Bonan Kou, and Tianyi Zhang. 2024. Is Stack Overflow obsolete? An empirical study of the characteristics of ChatGPT answers to Stack Overflow questions. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–17.

[14] Majeed Kazemitabaar, Justin Chow, Carl Ka To Ma, Barbara J Ericson, David Weintrop, and Tovi Grossman. 2023. 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*. 1–23.

[15] Marko Komssi, Danielle Pichlis, Mikko Raatikainen, Klas Kindström, and Janne Järvinen. 2014. What are hackathons for? *IEEE Software* 32, 5 (2014), 60–67.

[16] Stack Overflow Labs. 2023. Developer sentiment around AI/ML. https://stackoverflow.blog/2023/06/12/developer-survey-sentiment-ai-ml/ Accessed in Feb. 2026.

[17] Jenny T Liang, Chenyang Yang, and Brad A Myers. 2024. 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*. 1–13.

[18] Ying Liu and Lei Shen. 2025. Consolidating Human-AI Collaboration Research in Organizations: A Literature Review. *Journal of Computer, Signal, and System Research* 2, 1 (2025), 131–151.

[19] Atefeh Mahdavi Goloujeh, Anne Sullivan, and Brian Magerko. 2024. Is it AI or is it me? Understanding users' prompt journey with text-to-image generative AI tools. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–13.

[20] Maria Angelica Medina Angarita and Alexander Nolte. 2020. What do we know about hackathon outcomes and how to support them?–A systematic literature review. In *International conference on collaboration technologies and social computing*. Springer, 50–64.

[21] Piotr Mirowski, Kory W Mathewson, Jaylen Pittman, and Richard Evans. 2023. Co-writing screenplays and theatre scripts with language models: Evaluation by industry professionals. In *Proceedings of the 2023 CHI conference on human factors in computing systems*. 1–34.

[22] Khoa Viet Nguyen. 2025. The use of generative AI tools in higher education: Ethical and pedagogical principles. *Journal of Academic Ethics* (2025), 1–21.

[23] Alexander Nolte, Ei Pa Pa Pe-Than, Anna Filippova, Christian Bird, Steve Scallen, and James D Herbsleb. 2018. You Hacked and Now What? -Exploring Outcomes of a Corporate Hackathon. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (2018), 1–23.

[24] Hammond Pearce, Baleegh Ahmad, Benjamin Tan, Brendan Dolan-Gavitt, and Ramesh Karri. 2025. Asleep at the keyboard? assessing the security of GitHub CoPilot's code contributions. *Commun. ACM* 68, 2 (2025), 96–105.

[25] Neil Perry, Megha Srivastava, Deepak Kumar, and Dan Boneh. 2023. Do users write more insecure code with AI assistants?. In *Proceedings of the 2023 ACM SIGSAC conference on computer and communications security*. 2785–2799.

[26] James Prather, Brent N Reeves, Juho Leinonen, Stephen MacNeil, Arisoa S Randrianasolo, Brett A Becker, Bailey Kimmel, Jared Wright, and Ben Briggs. 2024. 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*. 469–486.

[27] Mitchel Resnick, Brad Myers, Kumiyo Nakakoji, Ben Shneiderman, Randy Pausch, Ted Selker, and Mike Eisenberg. 2005. Design principles for tools to support creative thinking. (2005).

[28] Ramteja Sajja, Carlos Erazo Ramirez, Zhouyayan Li, Bekir Z Demiray, Yusuf Sermet, and Ibrahim Demir. 2024. Integrating generative AI in hackathons: Opportunities, challenges, and educational implications. *Big Data and Cognitive Computing* 8, 12 (2024), 188.

[29] Johnny Saldaña. 2023. Developing a Codebook for a Case Participant. In *The Handbook of Teaching Qualitative and Mixed Research Methods*. Routledge, 219–222.

[30] Gustavo Sandoval, Hammond Pearce, Teo Nys, Ramesh Karri, Siddharth Garg, and Brendan Dolan-Gavitt. 2023. Lost at c: A user study on the security implications of large language model code assistants. In *32nd USENIX Security Symposium (USENIX Security 23)*. 2205–2222.

[31] Isabella Seeber, Eva Bittner, Robert O Briggs, Triparna De Vreede, Gert-Jan De Vreede, Aaron Elkins, Ronald Maier, Alexander B Merz, Sarah Oeste-Reiß, Nils Randrup, et al. 2020. Machines as teammates: A research agenda on AI in team collaboration. *Information & management* 57, 2 (2020), 103174.

[32] Inbal Shani and GitHub Staff. 2023. Survey reveals AI's impact on the developer experience. *GitHub. blog* 13 (2023).

[33] Esther Shein. 2023. Stack Overflow's 2023 Developer Survey: Are Developers Using AI? https://www.techrepublic.com/article/stack-overflow-2023-developer-survey-ai/

[34] Ben Shneiderman. 2007. Creativity support tools: accelerating discovery and innovation. *Commun. ACM* 50, 12 (2007), 20–32.

[35] Ben Shneiderman. 2008. Creativity support tools: A grand challenge for HCI researchers. In *Engineering the user interface: From research to practice*. Springer, 1–9.

[36] Ben Shneiderman. 2020. Bridging the gap between ethics and practice: guidelines for reliable, safe, and trustworthy human-centered AI systems. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 10, 4 (2020), 1–31.

[37] James P Spradley. 1980. *Participant Observation*. Holt, Rinehart and Winston.

[38] Erik H Trainer, Arun Kalyanasundaram, Chalalai Chaihirunkarn, and James D Herbsleb. 2016. How to hackathon: Socio-technical tradeoffs in brief, intensive collocation. In *proceedings of the 19th ACM conference on computer-supported cooperative work & social computing*. 1118–1130.

[39] Priyan Vaithilingam, Tianyi Zhang, and Elena L Glassman. 2022. 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*. Association for Computing Machinery, New York, NY, USA, 1–7. doi:10.1145/3491101.3519665

[40] Qian Yang, Aaron Steinfeld, Carolyn Rosé, and John Zimmerman. 2020. Re-examining whether, why, and how human-AI interaction is uniquely difficult to design. In *Proceedings of the 2020 CHI conference on human factors in computing systems*. 1–13.

[41] Ramazan Yilmaz and Fatma Gizem Karaoglan Yilmaz. 2023. Augmented intelligence in programming learning: Examining student views on the use of ChatGPT for programming learning. *Computers in Human Behavior: Artificial Humans* 1, 2 (2023), 100005.

[42] J Diego Zamfirescu-Pereira, Richmond Y Wong, Bjoern Hartmann, and Qian Yang. 2023. Why Johnny can't prompt: how non-AI experts try (and fail) to design LLM prompts. In *Proceedings of the 2023 CHI conference on human factors in computing systems*. 1–21.

[43] Beiqi Zhang, Peng Liang, Xiyu Zhou, Aakash Ahmad, and Muhammad Waseem. 2023. Demystifying practices, challenges and expected features of using GitHub Copilot. *International Journal of Software Engineering and Knowledge Engineering* 33, 11n12 (2023), 1653–1672.