"Will I Be Replaced?" Revisited: An Exploratory Replication on AI Tool Perceptions with Software Professionals in Brazil

Icaro Santos de Oliveira
State University of Ceara (UECE)
Fortaleza, CE, Brazil
icaro.santos@aluno.uece.br

Felipe Rodrigues de Oliveira
State University of Ceara (UECE)
Fortaleza, CE, Brazil
felipeo.rodrigues@aluno.uece.br

Paulo Henrique Mendes Maia
State University of Ceara (UECE)
Fortaleza, CE, Brazil
pauloh.maia@uece.br

Ismayle S. Santos
State University of Ceara (UECE)
Fortaleza, CE, Brazil
ismayle.santos@uece.br

Abstract

Generative Artificial Intelligence (AI) tools have rapidly entered professional software development, yet evidence on developers' perceptions remains concentrated in international samples and was largely collected in the early, single-tool era of ChatGPT 3.5, before multi-tool and agent-assisted workflows became commonplace. This paper presents an exploratory conceptual replication of the survey by Kuhail et al. on programmers' perceptions of AI tools, with the instrument translated into Brazilian Portuguese, updated to the current tool landscape, and administered to software professionals working in Brazil (18 valid responses). Combining descriptive statistics, exploratory Spearman correlations, and thematic analysis, we find frequent use and high perceived usefulness (4.50/5) under a deliberately assistive mode: reliability received the lowest trust rating and stood apart from an otherwise coherent positive-perception cluster. Perceived job-security threat reproduces the present-versus-future asymmetry of the original study (2.33 now vs. 3.06 in the future), and four themes converge on supervised assistive use, reshaping of the profession rather than replacement, and fundamentals-first education with AI literacy.

Keywords

Human Factors, Artificial Intelligence, Software Engineering, Developer Perceptions, LLMs, Survey, Replication Study

1 Introduction

The integration of generative Artificial Intelligence (AI) tools into software engineering practice is one of the most significant recent transformations in the field. In less than three years, tools such as GitHub Copilot, ChatGPT, Claude, and Cursor have become active components of professional development workflows, documented across code generation, debugging, documentation, testing, and code comprehension [5–7]—often before organizations establish formal policies or quality assurance practices for them. Perception research converges on tangible value—increased productivity, reduced cognitive load, faster onboarding [7, 9]—alongside persistent concerns about output reliability, limited context awareness, security risks, and ethical implications [5, 6]. A recurring finding is a *productivity-anxiety paradox*: developers reporting higher productivity gains and stronger trust also tend to report higher perceived threat to their job security [5].

Two gaps motivate this study. The first is **geographic and cultural**: the empirical record is dominated by North American, European, and Asian contexts. Brazil hosts one of the largest software workforces in Latin America, yet the study with a Brazilian-majority sample [9] treats the country as an undifferentiated context rather than an analytical lens, even though Brazilian research on mental health [1] and soft skills [3] shows that national specificities produce distinct findings. The second is **temporal and instrumental**: influential perception studies, including the reference work of Kuhail et al. [5], were conducted in the early era of ChatGPT 3.5, when a single conversational tool dominated. The ecosystem has since diversified into IDE-integrated assistants, competing model families, and agent-based workflows. Whether the perceptions documented then—in particular, the asymmetry between low *current* and higher *future* perceived job-security threat—persist under the current landscape, and whether they hold in underrepresented contexts, are open questions well suited to replication.

We present an exploratory *conceptual replication* of the survey component of Kuhail et al. [5], with the instrument translated into Brazilian Portuguese, updated to the current catalog of LLM-based development tools, and administered to software professionals working in Brazil. Three research questions, detailed in Section 3.1, guide the study: how these professionals assess the current landscape of AI tools (RQ1); what concerns they associate with the growing presence of AI (RQ2); and what perspectives they project for its impact on their job security and on the formation of future programmers (RQ3).

The study contributes *empirically* a structured account of AI tool perceptions in an underrepresented national context; *methodologically* a conceptual replication testing whether the reference findings—notably the present-versus-future threat asymmetry—hold under a different culture and an updated tool landscape, with instrument and data publicly available; and *practically* implications for responsible AI adoption in engineering teams and for software engineering education in Brazil.

2 Background and Related Work

This section establishes the conceptual ground of the study and positions our contribution against prior empirical work on developer perceptions of AI tools.

2.1 Foundational Concepts

We adopt the term *LLM-based development tools* for the family of software development assistants powered by Large Language Models (e.g., GitHub Copilot, ChatGPT, Cursor, Claude). Three characteristics distinguish them from earlier productivity aids: probabilistic outputs whose reliability cannot be guaranteed in advance, operation across multiple stages of the software lifecycle, and natural language interfaces that blur the boundary between tool use and collaboration [7, 10]—a property that underlies mental models of AI as collaborator, mentor, or junior peer [11].

2.2 Developer Perceptions and Positioning

Across this body of work, only the reference study [5] explicitly frames present-versus-future expectations, yet it drew on an international sample; conversely, the studies closest to the Brazilian setting engage it only partially, as Vaillant et al. [9] treated country as a descriptive variable and the work of Aires et al. [1] predates the AI-tool question, while the remaining studies [6, 7, 10, 11] rely on international samples. No prior study has replicated the perception instrument of the reference work in the Brazilian context, in Portuguese, and under the current multi-tool landscape—the combination that defines the present contribution.

Kuhail et al. [5] surveyed 99 programmers and surfaced the counterintuitive finding that higher trust and perceived productivity correlate with stronger job-security threat, alongside a clear asymmetry between current and future perceived threat. Vaillant et al. [9] examined the impact of ChatGPT on productivity, quality, and job satisfaction; although Brazilians dominated the sample, country was treated as a descriptive variable, while the study documented concerns about regulatory absence and entry barriers for junior professionals. Santos, Magalhães and de Souza Santos [7] found that most practitioners treat LLMs as assistive rather than autonomous solutions, verifying or modifying generated outputs, with trust tied to perceptions of accountability. Liang, Yang and Myers [6] found adoption driven by productivity perceptions, with limitations clustered around control over outputs and non-functional requirements. Zakharov et al. [10] identified two dominant mental models—AI as inanimate tool and AI as humanlike teammate—and showed that coding experience shapes which model practitioners adopt [11], evidence that AI perceptions are not uniform across the profession and that contextual variables deserve targeted investigation.

3 Methodology

This study is an exploratory conceptual replication of the survey component (Study 2) of Kuhail et al. [5], which assessed programmers' perceptions of ChatGPT and other AI tools. Whereas the original instrument was applied in English to an international audience and centered on ChatGPT 3.5, the present study adapts it to the current landscape of LLM-based development tools and administers it in Brazilian Portuguese to professionals working in Brazil. We adopted a mixed-methods design, combining descriptive analysis of closed-ended items with thematic analysis of open-ended responses.

3.1 Research Questions

The study is guided by three research questions:

- **RQ1.** How do software engineering professionals working in Brazil assess the current landscape of AI tools, in terms of capabilities, limitations, perceived productivity, trust, and integration into their work?
- **RQ2.** What concerns do these professionals associate with the growing presence of AI in software development, spanning technical, professional, and societal dimensions?
- **RQ3.** What perspectives do they project for the impact of AI on the profession, both for their own job security and for the formation of future programmers?

RQ1 and RQ3 directly replicate constructs of the reference study (usage, usefulness, productivity, trust, and current versus future job-security threat), enabling cross-context comparison; RQ2 extends the analysis through thematic examination of concerns expressed in open-ended responses.

3.2 Survey Instrument

The instrument was implemented as an online form with seven blocks: informed consent and demographics; programming profile; AI tool usage; productivity; limitations in coding and non-coding tasks; job security (current and future threat); and five trust items. Closed-ended items used 1–5 scales, and five open-ended items covered additional benefits, the future of the tools, how aspiring programmers should learn to code, and the reasoning behind job-security perceptions; the full instrument is in the replication package. Mirroring the reference instrument, demographics capture age range, role, and self-rated expertise, but not years of professional experience (Section 6).

3.3 Recruitment and Data Collection

The questionnaire was distributed over two weeks through professional research groups on WhatsApp and posts on LinkedIn, targeting practitioners and researchers active in software development and related technology roles in Brazil. Participation was voluntary and anonymous, and no personally identifying information was collected. Respondents were required to agree to the informed consent statement and to confirm that they either work professionally in a technology role or study/research within computing. Eligibility was defined by professional activity in the Brazilian software industry rather than by physical residence. Because the call was open rather than addressed to a closed list, an exact response rate cannot be computed, yielding a final dataset of 18 valid responses. Given this sample size, the study is positioned as exploratory: it aims to surface patterns within the Brazilian context and to verify whether the central tendencies of the reference study reappear, rather than to produce statistically generalizable estimates.

3.4 Data Analysis

Closed-ended responses were analyzed using descriptive statistics—frequencies, proportions, and, where appropriate, means and standard deviations—following the analytical structure of the reference study [5]. Consistent with the exploratory sample, we treat any inferential statistics as exploratory, not confirmatory; we report distributions and note observed tendencies, such as the contrast

between current and future job-security threat. Open-ended responses were examined through thematic analysis [2]: familiarization, initial coding, grouping into candidate themes, and iterative review and definition of final themes, supported by the open-source tool QualCoder[1]. Coding was performed by the first author from an initial codebook that was iteratively refined, with a generative AI tool assisting the first pass and every code reviewed and confirmed by the author (see Acknowledgments). Candidate themes were subsequently reviewed and validated with the co-authors until agreement. Because the final coding was produced by a single researcher, no inter-rater agreement statistic is reported; the initial and refined codebooks and the complete coding matrix are available in the replication package to permit independent audit. Themes are reported in Section 4 with anonymized illustrative quotations translated from the original Portuguese.

4 Results

Given the exploratory sample size, all quantitative results are reported descriptively. Table 1 summarizes the main Likert items: responses cluster in the upper range for usefulness, productivity, competence, effectiveness, and recommendation, while reliability and present job-security threat are notably lower.

4.1 Participant Profile

All 18 respondents agreed to the informed consent statement. The sample was predominantly male (16 of 18; two female) and concentrated in early-to-mid career age ranges, with the 26–30 bracket most frequent (7 respondents), followed by 21–25 (4). The remaining respondents were distributed across the 15–20, 31–35, 36–40, and over-41 brackets. Most participants reside in the Northeast region of Brazil, with the state of Ceará being the most common location; two respondents participate in international work arrangements (one residing in the United Kingdom and one employed by a company based in Germany); excluding them shifts the means of the closed-ended items by less than **0.15** points on the 1–5 scale, so both are retained in the analysis.

Reported roles spanned the software engineering spectrum, including systems analysts, software engineers (junior, mid-level, and senior), QA engineers and leads, a platform engineer, an AI engineer, a technology director, and a computing student. Self-rated programming expertise (1–5) averaged 3.83 (SD = 1.04), with most respondents rating themselves at 4 or 5 (11 of 18), indicating that participants perceive themselves as proficient programmers, although the sample spans early-career to leadership roles. Regarding programming languages, Python was the most common both for expertise (12 of 18) and daily use (11), followed by Java (9 expertise / 7 daily) and JavaScript (7 / 6); TypeScript also featured prominently in daily use (5).

4.2 Use of AI Tools (RQ1)

Tools and frequency. The most used AI tool was Google Gemini (12 of 18), followed by ChatGPT (8), GitHub Copilot (7), and Claude (6); OpenAI Codex and a few others appeared marginally. This distribution itself illustrates how the landscape has shifted since the reference study, in which ChatGPT was the focal tool: respondents now combine multiple assistants in their workflows. Usage was frequent: on a 1–5 scale, the average frequency of use was 3.94 (SD = 0.87), and no respondent reported low usage (all answers fell at 3 or above, with 6 reporting daily use).

Perceived usefulness. Participants rated the tools as highly useful, with an average of 4.50 (SD = 0.62) on the 1–5 scale; 10 of 18 gave the maximum rating and only one rated below 4. This is among the strongest signals in the dataset.

Activities supported. The tools were applied across a wide range of tasks. The most frequent were research and proof-of-concept exploration (14 of 18), writing documentation (12), and—tied at 11—generating boilerplate code, finding errors in code, and explaining code. Rewriting code to follow specific conventions (10) and writing test cases (9) were also common. Notably, full or near-full code generation was rare: only 3 respondents reported using AI to generate most of the code, and just 1 to generate all of it. This indicates a predominantly *assistive*, rather than autonomous, mode of use, consistent with the international pattern reported by Santos et al. [7].

4.3 Productivity (RQ1)

Respondents reported a clear productivity benefit, averaging 4.17 (SD = 0.71) on the 1–5 scale, with 15 of 18 rating the gain at 4 or 5. The most cited mechanism was completing programming tasks faster (14 of 18). Beyond speed, respondents reported writing more concise code, producing code that better follows standards, and understanding colleagues' code more quickly (each cited by 10 respondents), followed by writing code with fewer bugs (9). In free-text comments, several participants described the tools as handling repetitive or "manual" work so they could focus on analysis and design; one participant reported using AI agents to execute defined workflows, such as writing a function, writing its test, and pushing the change (P15)—a usage mode that did not exist at the time of the reference study—a signal echoed in the open-ended responses, where productivity gain was the most frequently assigned code (7 of 18).

4.4 Limitations of AI Tools (RQ2)

Participants identified consistent limitations. For coding specifically, the most cited issues were errors in the generated code (10 of 18) and code with security vulnerabilities (10), followed by inefficient code (8), code not meeting requirements (6), and code failing some test cases (6). For non-coding uses, the dominant limitation was loss of requirement context (14 of 18), followed by incorrect solutions (8) and misinterpretation of the problem's requirements (7). Together, these point to a shared perception that the tools are productive on well-scoped tasks but unreliable when broader context, correctness, or security is at stake—a concern reinforced by several open-ended comments stressing the need to always review generated code (e.g., P18).

4.5 Trust in AI Tools (RQ1)

Trust was measured through five agreement-scale items (1 = strongly disagree to 5 = strongly agree). Respondents agreed most strongly that they would recommend the tools to other programmers (4.28,

[1]Open-source, freely available CAQDAS tool: https://github.com/ccbogel/QualCoder

Table 1: Statistical Results for Survey Questions on AI Tool Perceptions

Question	Mean	Median	Max	Min	StdDev
10. How often do you use AI tools to assist you with programming?	3.94	4.00	5.00	3.00	0.87
11. How useful are AI tools for programming?	4.50	5.00	5.00	3.00	0.62
15. How much more productive have AI tools made you as a programmer?	4.17	4.00	5.00	3.00	0.71
20. Do you think AI tools currently affect your job security as a programmer?	2.33	2.50	4.00	1.00	1.14
21. Do you think AI tools will affect your job security as a programmer in the future?	3.06	3.00	5.00	1.00	1.30
23. I feel that AI tools are competent in the assistance they provide.	3.94	4.00	5.00	2.00	0.80
24. I feel that AI tools are effective in the assistance they provide.	4.06	4.00	5.00	1.00	0.94
25. I feel that AI tools provide assistance that serves my best interests.	3.89	4.00	5.00	3.00	0.58
26. I feel that AI tools provide reliable assistance.	3.11	3.00	4.00	2.00	0.58
27. I would recommend AI tools to other programmers.	4.28	4.00	5.00	3.00	0.67

SD = 0.67) and that the tools are effective (4.06, SD = 0.94) and competent (3.94, SD = 0.80) in the help they provide. Agreement was more moderate on whether the help serves the user's best interest (3.89, SD = 0.58) and notably lowest on reliability (3.11, SD = 0.58), where 12 of 18 neither agreed nor disagreed.

Answer to RQ1. Professionals in our sample use AI tools frequently and rate them as highly useful (4.50) and productivity-enhancing (4.17), but integrate them in a deliberately assistive mode: use concentrates on well-scoped support tasks, full code generation is rare, and trust is lowest precisely on the reliability of the generated output (3.11).

4.6 Job Security: Present vs. Future (RQ3)

When asked whether AI tools threaten their job security *today*, respondents reported a low average threat of 2.33 (SD = 1.14), with half (9 of 18) selecting the two lowest levels. Looking to the *future*, the perceived threat rose to 3.06 (SD = 1.30), and 12 of 18 selected a mid-to-high level. Comparing the two answers for each respondent, 9 rated their future threat higher than their current one, 8 rated it the same, and only 1 rated it lower. This upward shift—an increase of 0.72 in the scale mean—provides evidence consistent with the present-versus-future asymmetry reported in the reference study [5]. Practitioners in our sample perceived AI tools as posing limited threat to their job security today, but expressed greater concern about their future impact.

4.7 Thematic Analysis of Open-Ended Responses (RQ2, RQ3)

The open-ended responses on the reasoning behind job-security perceptions, the future evolution of the tools, and the education of new programmers were analyzed thematically. Four recurring themes emerged. For each, we report the number of distinct supporting participants (out of N = 18); because the themes are not mutually exclusive, a participant may contribute to more than one.

Theme 1: AI as an assistant under human supervision (12 of 18). The most prevalent view was that AI tools are support instruments that still require human judgment and oversight. Participants framed the human role as supervisory and validating: "the programmer as supervisor must master everything being done in order to correct and prevent problems" (P7). One participant captured the sentiment with a metaphor, arguing that AI models "should not be a pair of crutches; they should be an exoskeleton" (P3).

Theme 2: Concerns about reliability, context, and security (10 of 18). Participants repeatedly tied their reservations to the tools' weaknesses on correctness and context. Several stressed that the quality of AI output depends on the user's own knowledge and on clear requirements, and that the tools can confidently produce flawed solutions. One participant warned of the risk that "people learn that AI solves everything, so they don't check the information they receive or send" (P14). Another, working directly with LLM-based applications, noted that the tools handle simple tasks, maintenance, and medium-complexity bugs well but falter on long architectural decisions (P15).

Theme 3: Reshaping of the role rather than replacement (7 of 18). When projecting the future, most participants anticipated transformation of the profession rather than its disappearance. They expected the work to move up the abstraction ladder: "writing code should become less relevant; the focus will be on a higher level—system choices, vendors, business rules, infrastructure; the programmer will be a less manual worker" (P7). Several judged full replacement unlikely without a major technological leap, with one explicitly noting that "a complete replacement of the developer profession would require another leap (in the style of *attention is all you need*) in capability" (P15). A recurring concern, however, was the effect on entry-level positions: one participant noted that with AI tools in a senior team, "you wouldn't need interns/juniors," while qualifying that this depends on company maturity, product, and architecture (P10).

Theme 4: Education must preserve fundamentals while integrating AI (14 of 18). On how aspiring programmers should learn, participants were nearly unanimous that foundations come first and AI second. Many recommended an initial period of learning without AI to build reasoning, followed by guided, critical use of the tools: "have a brief experience without AI first … learning to program with AI but without programming knowledge is not advisable" (P4). Others advocated integrating AI literacy from the start alongside traditional concepts (P15), and using the tools to explain and consolidate base concepts rather than to produce finished code (P13, P17). The shared message was that the tools should be auxiliary—"a tool, not a decision-maker" (P5).

Answer to RQ2. Concerns concentrate on three technical fronts—incorrect code, security vulnerabilities, and loss of requirement

Table 2: Exploratory Spearman correlations ($N = 18$). Reliability (Q26) does not join the positive-perception cluster.

Variable Pair	ρ	p
Positive-perception cluster		
Q23 Competence × Q24 Effectiveness	0.850	< 0.001
Q24 Effectiveness × Q25 Best interest	0.694	0.001
Q24 Effectiveness × Q27 Recommendation	0.684	0.002
Q23 Competence × Q27 Recommendation	0.633	0.005
Q23 Competence × Q25 Best interest	0.598	0.009
Q25 Best interest × Q27 Recommendation	0.518	0.028
Reliability (Q26)		
Q26 Reliability × Q23 Competence	0.040	0.875
Q26 Reliability × Q24 Effectiveness	0.086	0.735
Q26 Reliability × Q27 Recommendation	−0.171	0.497
Job security		
Q20 Threat now × Q21 Threat future	0.449	0.062

context— and extend to professional and societal ones: over-reliance without verification, erosion of fundamentals among newcomers, and pressure on entry-level positions.

Answer to RQ3. Participants project a reshaping rather than a disappearance of the profession: perceived threat rises from 2.33 today to 3.06 in the future, work is expected to shift toward supervisory and higher-abstraction activities, and the near-unanimous prescription for training new programmers is fundamentals first, with AI introduced as an object of critical use.

4.8 Exploratory Correlation Analysis

To complement the descriptive results, we computed Spearman rank correlations [8] among the main Likert-scale variables (usage, usefulness, productivity, job-security threat, and trust), with values cross-checked in Orange Data Mining [4]. With $N = 18$, coefficients carry wide confidence intervals and are sensitive to individual responses: we therefore read only associations above $\rho = 0.60$ as robust signals and treat the remaining coefficients—including the $p = 0.062$ tendency between current and future threat—as directions for larger studies rather than as findings.

Table 2 lists the strongest associations. A coherent positive cluster links competence, effectiveness, best-interest alignment, and recommendation, led by competence × effectiveness ($\rho = 0.850$). Usage frequency, usefulness, and productivity relate only weakly to these items ($\rho < 0.40$, $p > 0.10$). Reliability behaves differently: it correlates negligibly with competence ($\rho = 0.040$), effectiveness ($\rho = 0.086$), and recommendation ($\rho = -0.171$; all $p > 0.49$), standing apart from the positive-perception cluster. Current and future job-security threat are moderately associated ($\rho = 0.449$, $p = 0.062$).

5 Discussion

This section interprets the findings in light of the reference study and the broader literature, and derives implications for research, industry, and education.

5.1 What the Replication Suggests—and What May Have Changed

The most direct contribution of this study is to provide evidence, in the Brazilian context, that is consistent with the present-versus-future asymmetry in perceived job-security threat first reported by Kuhail et al. [5]. Despite differences in language, country, and tool landscape, participants in our sample reported lower perceived threat in the present than when projecting into the future. This pattern suggests that future-oriented concerns about AI remain relevant even as practitioners incorporate these tools into everyday development workflows.

The tool landscape also differs: whereas the original study was centered on the early ChatGPT era, our respondents combine Gemini, ChatGPT, Copilot, Claude, and agent-assisted workflows, suggesting that perception studies should now treat the assistant ecosystem, rather than a single tool, as the unit of analysis.

Two departures from the reference study deserve emphasis. First, Kuhail et al. [5] report that higher perceived productivity and trust accompany a stronger sense of job-security threat; no such association emerges in our data—productivity relates only weakly and non-significantly to the trust items ($\rho < 0.40$, $p > 0.10$) and shows no relevant association with perceived threat—so the productivity-anxiety paradox is not reproduced here. Second, the original study reports trust as a composite score, which renders internal divergence invisible; examining the five trust items separately shows that reliability (3.11) sits well below the remaining items (3.89–4.28) and is statistically detached from them. Whether these differences stem from the national context, from a more mature multi-tool landscape, or from the exploratory sample cannot be settled here: our comparison with the reference study is descriptive, not inferential.

5.2 The Reliability–Recommendation Paradox

The correlation analysis offers one explanation for the tension between strong recommendation and weak perceived reliability: competence, effectiveness, best-interest alignment, and recommendation form a coherent cluster from which reliability remains detached (Section 4.8). Participants recognize the practical value of the tools without extending that judgment to the dependability of their outputs.

Far from being contradictory, this selective trust is consistent with the cautious, accountability-driven usage documented by Santos et al. [7]. Practitioners seem to treat AI output as a useful *draft*, but not as a final artifact. The thematic analysis reinforces this interpretation: the most prevalent theme frames the practitioner as a supervisor, and the "exoskeleton, not crutches" metaphor (P3) captures a deliberate stance of augmentation under human control. For the Intelligent Software Engineering community, this finding suggests that tooling research should invest not only in generation capability, but also in *verifiability*, including provenance, test scaffolding, and explanations of generated changes.

5.3 Implications for Practice and Education

A concern cutting across Themes 3 and 4 is the effect of AI on entry-level positions: if AI absorbs the well-scoped, repetitive tasks traditionally assigned to interns and juniors (P10), organizations may reduce entry-level hiring, echoing the entry-barrier concerns

documented internationally [9]. In Brazil, where internships and junior roles are the dominant pipeline from undergraduate programs into industry, this produces a double burden on newcomers: fewer entry routes and higher expectations of supervisory competence. For industry, the results support mandatory review of AI-generated code—with explicit attention to the security vulnerabilities participants ranked among the top limitations—explicit organizational policies for tool use, and preserving structured entry-level roles that pair juniors with AI-supervision duties. For education, they translate into a fundamentals-first curriculum in which AI is introduced progressively as an object of critical use, explaining and consolidating concepts rather than producing finished artifacts (P4, P13, P17).

6 Threats to Validity

Several threats to validity should be considered. Regarding *construct validity*, the instrument was translated by the authors and piloted with a single graduate student, leading to minor wording adjustments but without formal back-translation, so subtle shifts in item meaning relative to the original cannot be ruled out; moreover, mirroring the reference instrument, the demographic section did not capture years of professional experience, limiting our ability to characterize seniority precisely. Regarding *internal validity*, recruitment through WhatsApp research groups and LinkedIn introduces self-selection bias toward practitioners already engaged with AI tools, which plausibly inflates the observed frequency-of-use (3.94) and usefulness (4.50) means, and more broadly the positive tone of the closed-ended items, and the cross-sectional design means the present-versus-future contrast reflects simultaneous projections by the same respondent rather than observed change over time. Regarding *external validity*, the sample is small (N = 18), predominantly male (16 of 18), and geographically concentrated in the Northeast region (14 of 18, mostly in Ceará); the findings should therefore not be generalized to the Brazilian software industry, although this concentration documents a regional context rarely visible in the literature, and the study is explicitly exploratory, surfacing patterns that warrant confirmation with larger, stratified samples. Finally, regarding *conclusion validity*, inferential statistics (Spearman correlations) were treated as exploratory rather than confirmatory, with no hypothesis-testing claims; the qualitative coding was produced by a single researcher, so no inter-rater agreement statistic (e.g., Cohen's κ) could be computed; individual coding bias was mitigated through an a priori codebook, iterative refinement, validation of the candidate themes with the co-authors, and public release of the codebooks and the full coding matrix.

7 Conclusion and Future Work

This paper presented an exploratory conceptual replication of the survey by Kuhail et al. [5] on programmers' perceptions of AI tools, administered in Brazilian Portuguese to 18 software professionals working in Brazil and updated to the current multi-tool landscape. Its main contribution is exploratory evidence that central perceptual patterns of the reference study—notably the present-versus-future asymmetry in perceived job-security threat—are also observable in this context, alongside two divergences: the productivity-anxiety paradox does not reappear, and reliability separates from the remaining trust items. As future work, we plan to extend data collection with a larger sample stratified by region, gender, and years of professional experience; to conduct follow-up interviews on the tension between recommendation and reliability; and to reapply the survey periodically with explicit short-, medium-, and long-term horizon items, tracking whether the projected threat materializes, dissipates, or is continually deferred.

Artifact Availability

All scripts, datasets, and methodological artifacts used in this research are publicly available to ensure transparency and reproducibility. These resources are hosted in an open-access repository and can be accessed via the following link[2].

Acknowledgements

We thank the National Council for Scientific and Technological Development (CNPq) and the Ceará Foundation for Scientific and Technological Development (Funcap) for their support. We also thank CNPq for the productivity grants awarded to Prof. Dr. Ismayle S. Santos (312173/2025-3).

In accordance with the ISE'26 policy, the authors disclose that Claude (Anthropic) was used to generate initial drafts of Sections 5–7 and to assist the initial qualitative coding; all AI-assisted content was revised, validated, and consolidated by the authors, who assume full responsibility for the final work.

References

[1] Maria Eulália Aires, Lucas Migge, Kiev Gama, and Aline Lacerda. 2024. Exploring the Role of Job Satisfaction and Work-Life Balance: A Study on the Mental Health Among Software Engineers. In *Simpósio Brasileiro de Engenharia de Software (SBES)*. SBC, Porto Alegre, RS, Brazil, 269–279.

[2] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative research in psychology* 3, 2 (2006), 77–101.

[3] Murilo Coelho, Allysson Allex Araújo, Sávio Freire, and Matheus Paixao. 2024. Um Estudo Comparativo entre a Visão de Líderes e Liderados sobre a Importância de Soft Skills em Desenvolvimento de Software. In *Workshop sobre Aspectos Sociais, Humanos e Econômicos de Software (WASHES)*. SBC, Porto Alegre, RS, Brazil, 130–140.

[4] Janez Demšar, Tomaž Curk, Aleš Erjavec, et al. 2013. Orange: Data Mining Toolbox in Python. *Journal of Machine Learning Research* 14 (2013), 2349–2353.

[5] Mohammad Amin Kuhail, Sujith Samuel Mathew, Ashraf Khalil, Jose Berengueres, and Syed Jawad Hussain Shah. 2024. "Will I be replaced?" Assessing ChatGPT's effect on software development and programmer perceptions of AI tools. *Science of Computer Programming* 235 (2024), 103111.

[6] 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 (ICSE)*. Association for Computing Machinery, New York, NY, USA, 1–13.

[7] Italo Santos, Cleyton Magalhaes, and Ronnie De Souza Santos. 2025. Model-Assisted and Human-Guided: Perceptions and Practices of Software Professionals Using LLMs for Coding. In *2025 2nd IEEE/ACM International Conference on AI-powered Software (AIware)*. IEEE, Piscataway, NJ, USA, 105–112.

[8] Charles Spearman. 1904. The Proof and Measurement of Association between Two Things. *The American Journal of Psychology* 15, 1 (1904), 72–101.

[9] Thiago S. Vaillant, Felipe Deveza de Almeida, Paulo Anselmo Neto, Cuiyun Gao, Jan Bosch, and Eduardo Santana de Almeida. 2024. Developers' Perceptions on the Impact of ChatGPT in Software Development: A Survey. arXiv:2405.12195.

[10] Ilya Zakharov, Ekaterina Koshchenko, and Agnia Sergeyuk. 2025. AI in Software Engineering: Perceived Roles and Their Impact on Adoption. In *Proceedings of the 33rd ACM International Conference on the Foundations of Software Engineering (FSE)*. Association for Computing Machinery, New York, NY, USA, 1305–1309.

[11] Ilya Zakharov, Ekaterina Koshchenko, and Agnia Sergeyuk. 2025. From Teacher to Colleague: How Coding Experience Shapes Developer Perceptions of AI Tools. arXiv:2504.13903.

[2] https://doi.org/10.5281/zenodo.20947923