

Not All Technostress Is Equal: Techno-Complexity, Perceived Ease of Use, and Developers' Acceptance of Emerging Development Technologies

Marija Boban

University of Split, Faculty of Law
Department of Information Science and Statistics
Domovinskog rata 8, 21000 Split, Croatia
marija.boban@pravst.hr

Marija Gombar

General Staff of the Armed Forces of the Republic of
Croatia
Sarajevska 7, 10000 Zagreb, Croatia
gombar.ma@gmail.com

Abstract. *This study examines techno-complexity as a usability-related barrier to developers' acceptance of emerging development technologies. Survey data from 237 Croatian software-development professionals show that techno-complexity reduces perceived ease of use, while perceived ease of use increases intention to use. The negative indirect effect confirms that complexity weakens acceptance through a usability pathway rather than as a uniformly negative direct predictor. The contribution lies in isolating techno-complexity from techno-insecurity and techno-uncertainty in a developer-specific context. Respondents evaluated development tools, programming languages, databases, and operating systems, not a specific AI tool; AI-related conclusions are therefore presented only as implications.*

Keywords. technostress, techno-complexity, developers, technology acceptance, perceived ease of use, emerging development technologies

1 Introduction

Software development work is increasingly shaped by new or substantially changing technologies, including development tools, programming languages, databases, and primary operating systems. More recently, generative AI and AI-assisted coding tools have intensified this broader transformation by changing how developers search for solutions, write code, evaluate outputs, and interact with development environments (Barke et al., 2023; Nguyen-Duc et al., 2025; Russo et al., 2024; Weisz et al., 2025). However, the empirical scope of this study is broader than AI-assisted software engineering and does not concern a named tool, platform, or AI system. Respondents evaluated new technologies used in their own areas of software-development work. Their acceptance therefore depends not only on expected technical benefits, but also on whether such technologies are perceived as understandable, manageable, and

sufficiently easy to integrate into existing work practices.

Technology acceptance research has long emphasised the importance of perceived ease of use and perceived usefulness in explaining users' intention to adopt information technologies (Davis, 1989; Venkatesh & Davis, 2000). In this paper, the focus is placed on perceived ease of use because techno-complexity is expected to operate primarily as a usability-related barrier. In software development contexts, this issue becomes particularly relevant because new tools often require continuous learning, adaptation to changing workflows, and evaluation of outputs whose correctness, security, or reliability may not be immediately transparent. Studies on AI-assisted programming show that developers may benefit from code generation and automation, but they also face usability, trust, security, and validation challenges (Pearce et al., 2025; Perry et al., 2023; Prather et al., 2024; Vaithilingam et al., 2022). Therefore, understanding technology acceptance among developers requires attention not only to perceived benefits, but also to the cognitive and practical burden created by technological complexity. Technostress provides a useful lens for analysing this burden. Earlier research conceptualised technostress as a multidimensional phenomenon associated with overload, invasion, complexity, insecurity, and uncertainty in technology-intensive work (Ragu-Nathan et al., 2008; Tarafdar et al., 2019). More recent studies suggest that technostress should not be treated as a uniform barrier, especially in contexts shaped by AI and rapidly evolving digital tools (Chang et al., 2024; Högemann et al., 2025; Nastjuk et al., 2024). Some forms of technological pressure may primarily affect usability perceptions, while others may relate to professional insecurity, uncertainty, or adaptation. This distinction is important for responsible digital transformation because organisations cannot support technology adoption effectively if all forms of technostress are addressed in the same way.

This paper focuses on techno-complexity as a specific and actionable barrier to developers'

acceptance of emerging development technologies. Techno-complexity refers to the extent to which users perceive technologies as difficult to understand, learn, or use. In development work, such complexity may reduce perceived ease of use and indirectly weaken intention to use new tools. At the same time, the growing role of intelligent and AI-assisted development environments makes this issue increasingly relevant for organisations that seek to introduce new technologies responsibly, without overlooking developers' competencies, workload, and need for human-centred support. This paper aims to examine the relationship between techno-complexity, perceived ease of use, and intention to use emerging development technologies among developers and programmers. The main research question is: How does techno-complexity influence developers' intention to use emerging development technologies, and is this relationship mediated by perceived ease of use? In addition, the paper considers techno-insecurity and techno-uncertainty as exploratory extensions to examine whether different technostress dimensions provide additional explanatory value beyond techno-complexity.

The novelty of this paper lies in the combination of a developer-specific context, the isolation of techno-complexity from other technostress dimensions, and the testing of perceived ease of use as the mechanism linking complexity to acceptance. Rather than treating technostress as a single aggregate barrier or presenting its dimensions only as parallel predictors, the study conceptualises techno-complexity as a distinct usability-related mechanism and contrasts it with techno-insecurity and techno-uncertainty through an exploratory extension. The contribution is therefore not the proposal of a new general technology-acceptance model, but a more precise explanation of how one specific technostress dimension operates among software-development professionals.

2 Theoretical Background and Research Model

Technology acceptance research explains technology adoption primarily through users' beliefs about whether a technology is useful and easy to use. In the Technology Acceptance Model, perceived ease of use refers to the degree to which a person believes that using a technology would be free of effort, while intention to use reflects behavioural readiness to adopt the technology (Davis, 1989; Venkatesh & Davis, 2000). In software development work, perceived ease of use is particularly relevant because new development tools often require continuous learning, adaptation to new workflows, and evaluation of complex or automatically generated outputs. Recent research on AI-assisted and generative development tools shows that developers do not interact with such

technologies only as passive users. They evaluate suggestions, validate outputs, manage uncertainty, and adjust their programming strategies (Barke et al., 2023; Prather et al., 2024; Vaithilingam et al., 2022; Weisz et al., 2025). These tools may improve productivity and support complex development tasks, but they may also introduce new usability, security, and verification challenges (Pearce et al., 2025; Perry et al., 2023; Russo et al., 2024). Therefore, developers' acceptance of emerging development technologies should be examined not only through their expected benefits, but also through the perceived complexity of using them.

Technostress research provides a useful framework for understanding technology-related strain in digitally intensive work. Technostress creators include several dimensions, such as techno-overload, techno-invasion, techno-complexity, techno-insecurity, and techno-uncertainty (Ragu-Nathan et al., 2008; Tarafdar et al., 2019). However, recent work suggests that these dimensions should not be treated as identical barriers to technology adoption, especially in contexts shaped by artificial intelligence and rapidly changing digital tools (Chang et al., 2024; Högemann et al., 2025; Nastjuk et al., 2024). Among these dimensions, techno-complexity is especially relevant for software developers because it captures the perceived difficulty of understanding, learning, and using new technologies. In this paper, techno-complexity is conceptualised as a specific barrier to developers' acceptance of emerging development technologies. When developers perceive a technology as complex, they may also perceive it as less easy to use. Lower perceived ease of use can then reduce their intention to use the technology. This mechanism is important for responsible digital transformation because organisations introducing increasingly intelligent development environments need to support developers not only with access to tools, but also with training, integration support, and human-centred implementation practices (Gombar & Boban, 2026; Russo et al., 2024).

Based on this reasoning, the following hypotheses are proposed:

H1. Techno-complexity negatively influences perceived ease of use of emerging development technologies.

H2. Perceived ease of use positively influences intention to use emerging development technologies.

H3. Perceived ease of use mediates the relationship between techno-complexity and intention to use emerging development technologies.

In addition to the confirmatory model, the study considers two further technostress dimensions: techno-insecurity and techno-uncertainty. Techno-insecurity refers to the perceived threat that technological change may weaken one's professional position or competence, while techno-uncertainty reflects the pressure of frequent technological change. Since these dimensions may influence acceptance differently from

techno-complexity, they are included as exploratory extensions.

ERQ1. Do techno-insecurity and techno-uncertainty provide additional explanatory value beyond techno-complexity in explaining developers' intention to use emerging development technologies?

As shown in Fig. 1, the proposed research model centres on techno-complexity as the main predictor, perceived ease of use as the mediator, and intention to use as the outcome variable, with techno-insecurity and techno-uncertainty examined as exploratory technostress extensions.

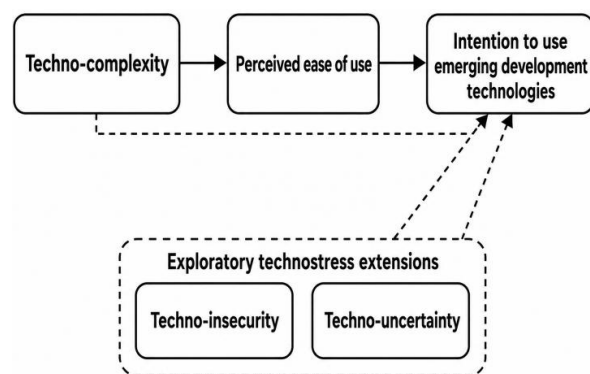


Figure 1. Proposed research model

Note. Solid arrows indicate hypothesised main relationships. Dashed arrows indicate direct or exploratory paths.

3 Methodology

3.1 Sample and Data Collection

The study is based on survey data collected from 237 software-development professionals in the Croatian IT sector. Data were collected during the first quarter of 2025 through an online questionnaire administered using Google Forms. The questionnaire link was distributed via email and messaging applications, specifically WhatsApp and Viber. Before accessing the questionnaire, all participants were informed about the purpose of the study, the voluntary nature of participation, the confidentiality and protection of their data, and that responses would be used exclusively for research purposes in aggregated form. Respondents were included in the analytical sample when their reported occupational title identified them as developers, programmers, or professionals performing an equivalent software-development role. Responses were anonymised, and the individual respondent was used as the unit of analysis. Regarding project roles, 219 respondents (92.4%) were developers, nine (3.8%) were software architects, six (2.5%) were project leaders, and three (1.3%) were test engineers. Most respondents were aged 21–30 years (68.4%), followed by 31–40 years (21.5%) and 41–50 years (8.0%). Men represented 69.6% of the sample, women 29.1%, while

1.3% preferred not to report gender. Regarding education, 59.9% held a graduate degree, 20.3% an undergraduate degree, 9.3% vocational secondary education, 6.8% general secondary education, and 3.4% a postgraduate degree. Information on respondents' years of professional experience was not available in the analytical dataset.

Respondents did not evaluate one named technology, platform, or AI tool. The questionnaire referred more broadly to new technologies encountered in respondents' own software development work, encompassing categories such as development tools, programming languages, databases, and primary operating systems. Accordingly, the term "emerging development technologies" is used as an umbrella category rather than as a reference to a single homogeneous technology. Implications concerning AI-assisted software engineering are therefore interpretative extensions of the findings and not direct empirical evidence about the use of specific AI systems.

3.2 Measures

The study included five constructs measured with 20 items: techno-complexity (five items), techno-insecurity (four items), techno-uncertainty (five items), perceived ease of use (three items), and intention to use (three items). Techno-complexity captures the perceived difficulty of understanding, learning, and using new technologies. Techno-insecurity reflects concerns that technological change may weaken one's professional competence or position, while techno-uncertainty captures the pressure associated with frequent technological changes. Perceived ease of use refers to the extent to which respondents perceive new technologies as easy to learn and use, while intention to use reflects their readiness to use such technologies. The questionnaire was developed by adapting established technostress and technology-acceptance scales to the context of emerging development technologies. The technostress items were based on established technostress-creator dimensions, particularly techno-complexity, techno-insecurity, and techno-uncertainty (Ragu-Nathan et al., 2008; Tarafdar et al., 2019). The perceived ease-of-use and intention-to-use items were grounded in the Technology Acceptance Model (Davis, 1989; Venkatesh & Davis, 2000). Respondents indicated their agreement on a seven-point Likert scale ranging from 1 (strongly disagree) to 7 (strongly agree). Composite scores were calculated as the mean of the respective items. Table 1 presents the English wording of the items included in the analysis.

Table 1. Measurement items

Construct	Items
Techno-complexity (TC)	<p>TC1: I do not know enough about new technologies in my area of work to perform my job satisfactorily.</p> <p>TC2: It takes me a long time to understand and use new technologies in my area of work.</p> <p>TC3: I cannot find enough time to learn and improve my technological skills.</p> <p>TC4: I believe that new employees in this organisation know more about computer technology than I do.</p> <p>TC5: I often find it too complex to understand and use new technologies in my area of work.</p>
Techno-insecurity (TINS)	<p>TINS1: I feel a constant threat to my job security because of new technology in my area of work.</p> <p>TINS2: I have to continually upgrade my technological skills to avoid being replaced.</p> <p>TINS3: I feel threatened by colleagues with advanced knowledge of new technologies in my area of work.</p> <p>TINS4: I feel that less knowledge about new technologies in my area of work is shared among colleagues because of fear of replacement.</p>
Techno-uncertainty (TU)	<p>TU1: There are always new developments in the technologies used in our organisation.</p> <p>TU2: There are constant changes in the programming languages used in our organisation.</p> <p>TU3: There are constant changes in the development tools used in our organisation.</p> <p>TU4: There are constant changes in the databases used in our organisation.</p> <p>TU5: There are constant changes in the primary operating systems used in our organisation.</p>
Perceived ease of use (PEOU)	<p>PEOU1: I find it easy to learn and master new technologies in my area of work.</p> <p>PEOU2: I easily find the information I need about new technologies in my area of work.</p> <p>PEOU3: I consider new technologies in my area of work easy to use.</p>
Intention to use (ITU)	<p>ITU1: I am inclined to use new technologies in my area of work.</p> <p>ITU2: Assuming that I have access to new technologies in my area of work, I intend to use them.</p> <p>ITU3: Given that I have access to new technologies in my area of work, I expect that I will use them.</p>

Note. All items were measured on a seven-point Likert scale ranging from 1 (*strongly disagree*) to 7 (*strongly agree*). TC = techno-complexity; TINS = techno-insecurity; TU = techno-uncertainty; PEOU = perceived ease of use; ITU = intention to use.

3.3 Data Analysis

The analysis was conducted in several steps. First, the analytical dataset was screened for missing values, and

descriptive statistics were calculated for all constructs. Second, internal consistency was assessed using Cronbach's alpha and McDonald's omega. Third, Pearson correlations were used to examine bivariate relationships among the constructs. To test the research model, robust observed-variable path analysis was conducted using standardized composite scores. The main model examined the effect of techno-complexity on perceived ease of use and the effect of perceived ease of use on intention to use. The mediating role of perceived ease of use was assessed using bootstrap confidence intervals with 5,000 resamples (Hayes, 2022). Techno-insecurity and techno-uncertainty were analysed as exploratory extensions to assess whether they provided additional explanatory value beyond techno-complexity. These exploratory paths were not treated as confirmatory hypothesis tests but as diagnostic extensions used to examine whether other technostress dimensions changed the interpretation of techno-complexity. Discriminant validity was additionally assessed using the heterotrait–monotrait ratio of correlations (HTMT), calculated from the item-level correlations among the five constructs. HTMT values below 0.90 were considered evidence that the constructs were empirically distinguishable. The results of this assessment are reported together with the descriptive statistics and reliability indicators.

4 Results

4.1 Descriptive Statistics, Reliability, and Discriminant Validity

Table 2 presents descriptive statistics and reliability indicators for the constructs included in the analysis. All scales showed acceptable internal consistency, with Cronbach's alpha values ranging from 0.730 to 0.879. The highest mean value was observed for intention to use, indicating that respondents generally expressed a relatively high readiness to use emerging development technologies. Techno-complexity, techno-insecurity, and techno-uncertainty showed lower mean values, suggesting that technostress-related perceptions were present but not dominant in the sample.

Table 2. Descriptive statistics and reliability of constructs

Construct	k	M	SD	α	ω
TC	5	2.775	1.184	0.785	0.860
TINS	4	2.310	1.123	0.733	0.846
TU	5	2.430	1.000	0.730	0.829
PEOU	3	4.602	1.322	0.832	0.902
ITU	3	5.167	1.387	0.879	0.926

Note. Constructs were measured using multi-item Likert-type scales. Composite scores were calculated as mean values of their respective items.

Discriminant validity was supported by the HTMT assessment. All ten pairwise HTMT values were below the 0.90 threshold, ranging from 0.106 to 0.863. The highest value was observed between techno-complexity and techno-insecurity (HTMT = 0.863), followed by techno-insecurity and techno-uncertainty (HTMT = 0.819) and perceived ease of use and intention to use (HTMT = 0.787). These results indicate that the five constructs were empirically distinguishable. The correlation analysis showed a moderate negative relationship between techno-complexity and perceived ease of use ($r = -0.341$), and a strong positive relationship between perceived ease of use and intention to use ($r = 0.675$). The bivariate relationship between techno-complexity and intention to use was weak and not statistically significant ($r = -0.072$), indicating that the role of techno-complexity should be interpreted primarily through the proposed usability pathway rather than as a simple direct effect.

4.2 Test of the Main Model

The main model tested the relationship between techno-complexity, perceived ease of use, and intention to use. Robust path analysis was conducted using standardized composite scores. The results are presented in Table 3.

Table 3. Main path and mediation results

Path	β	Evidence/decision
H1: TC $\rightarrow$ PEOU	-0.341	$p < .001$; supported
H2: PEOU $\rightarrow$ ITU	0.735	$p < .001$; supported
Direct: TC $\rightarrow$ ITU	0.179	$p = .002$; mediation test
H3: TC $\rightarrow$ PEOU $\rightarrow$ ITU	-0.249	95% CI [-0.345, -0.150]; supported

Note. Standardized coefficients are reported. The direct path was included to test mediation and should not be interpreted as evidence that techno-complexity has a simple negative total effect on intention to use. The indirect effect was estimated using 5,000 bootstrap resamples.

Fig. 2 visualises the tested mediation mechanism. The results show that techno-complexity significantly reduces perceived ease of use, while perceived ease of use significantly increases intention to use emerging development technologies. The indirect effect is negative and statistically significant, indicating that techno-complexity weakens intention to use specifically through the perceived ease-of-use pathway.

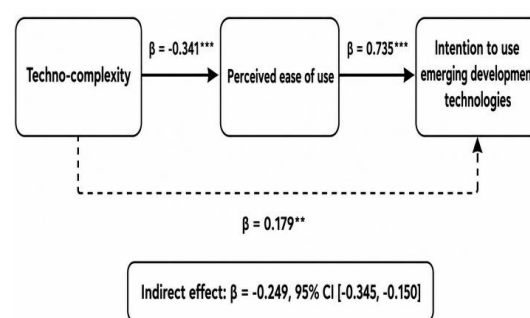


Figure 2. Empirical model with standardised coefficients

Note. ** $p < .01$; *** $p < .001$.

The results support H1. Techno-complexity had a significant negative effect on perceived ease of use ($\beta = -0.341$, $p < .001$), indicating that developers who perceive emerging development technologies as more complex also perceive them as less easy to use. The model explained 11.6% of the variance in perceived ease of use. H2 was also supported. Perceived ease of use had a strong positive effect on intention to use ($\beta = 0.735$, $p < .001$) when techno-complexity was included in the model. Together, techno-complexity and perceived ease of use explained 48.3% of the variance in intention to use. The mediation analysis supported H3. The indirect effect of techno-complexity on intention to use through perceived ease of use was negative and statistically significant ($\beta = -0.249$, 95% CI [-0.345, -0.150]). The results suggest that techno-complexity weakens developers' intention to use emerging development technologies indirectly, through its negative effect on perceived ease of use.

The total bivariate relationship between techno-complexity and intention to use was weak and not statistically significant, while the direct path became positive after perceived ease of use was controlled. This pattern indicates that techno-complexity should not be interpreted as a uniform negative predictor of intention to use. Rather, its negative role appears through the usability pathway: when complexity reduces perceived ease of use, it indirectly weakens acceptance readiness. The positive direct path may suggest a suppression pattern, or that developers who perceive technologies as complex may still recognise their professional relevance once ease of use is accounted for.

4.3 Exploratory Extension

The exploratory analysis examined whether techno-insecurity and techno-uncertainty provided additional explanatory value beyond techno-complexity. When techno-complexity, techno-insecurity, techno-uncertainty, and perceived ease of use were included as predictors of intention to use, perceived ease of use remained the strongest predictor ($\beta = 0.735$, $p < .001$). Techno-insecurity had a significant negative effect on

intention to use ($\beta = -0.215$, $p = 0.035$), while techno-uncertainty had a small positive effect ($\beta = 0.116$, $p = 0.042$). Techno-complexity also showed a positive direct effect in this extended model ($\beta = 0.267$, $p = 0.003$), confirming that its role is not adequately captured by a simple negative direct relationship.

5 Discussion

The results support the proposed mediation mechanism and clarify the role of techno-complexity in developers' acceptance of emerging development technologies. Techno-complexity had a significant negative effect on perceived ease of use, perceived ease of use had a significant positive effect on intention to use, and the indirect effect of techno-complexity on intention to use through perceived ease of use was statistically significant. These findings show that techno-complexity is not simply a general negative attitude toward technology, but a specific usability-related barrier. In other words, complexity becomes problematic when it makes technologies harder to understand, learn, or integrate into development work. At the same time, the positive direct path from techno-complexity to intention to use suggests a possible dual role of complexity in developer contexts. However, this interpretation should be treated cautiously. For developers, complex technologies may also signal advanced functionality, professional relevance, or opportunities for competence development. The negative indirect effect shows that complexity weakens acceptance when it reduces perceived ease of use. The main contribution of the study is therefore not the claim that techno-complexity uniformly decreases intention to use, but that it operates through a specific usability pathway.

This finding is relevant in the context of increasingly AI-assisted software engineering. Recent work shows that generative AI is becoming embedded in programming, requirements engineering, and broader software engineering tasks, but also that its value depends on developers' ability to understand, evaluate, and integrate AI-generated outputs into existing work practices (Cheng et al., 2025; Feuerriegel et al., 2024; Russo, 2024; Sengar et al., 2025). The present study does not directly measure generative AI adoption, and its empirical focus is broader than a specific AI coding assistant or generative AI system. Nevertheless, the results are relevant for such environments because AI-assisted tools may intensify the same usability, learning, and verification demands that are captured by techno-complexity.

The findings also contribute to the discussion on organisational AI capability and responsible digital transformation. Organisations do not build digital capability only by acquiring advanced technologies; they also need human competencies, learning routines, and implementation practices that allow employees to

use such technologies effectively and critically (Mikalef & Gupta, 2021). In software development, this is especially important because emerging tools may influence not only productivity, but also code quality, security, and professional responsibility. When automated outputs are used in knowledge-intensive tasks, issues such as reliability, interpretation, and responsible use become part of the adoption process, particularly in algorithmic environments where unreliable or incorrect outputs may affect user decisions and professional outcomes (Ji et al., 2023). From this perspective, techno-complexity should be understood as a practical signal: it indicates where developers may need clearer explanations, better integration support, and more structured learning before a tool can be responsibly adopted. The exploratory results further support the central claim that not all technostress dimensions are equal. Techno-insecurity and techno-uncertainty did not operate in the same way as techno-complexity. While techno-complexity primarily acted as a usability-related barrier, techno-insecurity appeared to reduce willingness to use emerging technologies, and techno-uncertainty showed a small positive effect that may reflect adaptation pressure in fast-changing technological environments rather than simple resistance. This distinction is important because generic technology adoption interventions may miss the real source of strain. If the problem is complexity, organisations should reduce cognitive and usability barriers. If the problem is insecurity, they should address professional confidence, role clarity, and competence development. If the problem is uncertainty, they should provide clearer communication about technological change and support employees in managing continuous adaptation.

The practical implication is that organisations should assess the "complexity budget" of emerging development technologies. This means evaluating how much new knowledge, workflow change, verification effort, and cognitive load a tool requires from developers. Such an assessment is particularly important when introducing highly automated or intelligent development tools, in which developers must not only operate the system but also assess the quality, relevance, and risks of its outputs. Responsible implementation should therefore include gradual introduction, targeted training, peer learning, feedback mechanisms, and explicit guidance on the limits of automated suggestions. In this sense, the complexity budget is not only a usability issue, but also part of responsible digital behaviour in algorithmic environments. When development tools become more automated, developers need not only technical skills but also awareness, trust calibration, and critical engagement with automated outputs. Prior work on algorithmic environments shows that user awareness, trust, and digital responsibility shape how individuals interpret personalised or automated content (Gombar & Boban, 2025; Gombar et al., 2026). Similarly,

concerns about generative AI implementation highlight that uncritical adoption may create organisational and cybersecurity risks when responsibility for automated outputs is not clearly understood (Humphreys et al., 2024). This aligns with human-centred approaches to generative AI in software engineering, which emphasise that productivity gains should not be separated from developer experience, accountability, and responsible professional practice (Russo et al., 2024). The study has several limitations. First, it relies on self-reported survey data, which capture developers' perceptions rather than observed tool use. Second, the cross-sectional design does not allow causal conclusions. Third, the study examines emerging development technologies broadly and does not directly measure the adoption of a specific AI coding assistant or generative AI system. The results should therefore not be interpreted as direct evidence about generative AI adoption, but they still identify techno-complexity as a specific usability-related barrier in developers' acceptance of emerging development technologies.

Future research should examine how this mechanism operates in concrete AI-assisted development settings, such as code generation, requirements engineering, testing, or software architecture. Longitudinal studies could investigate whether techno-complexity decreases as developers gain experience with a tool, or whether it remains a persistent barrier when tools evolve rapidly. Experimental or mixed-method research could also test whether training, transparency, peer support, or better workflow integration reduces the negative effect of techno-complexity on perceived ease of use. In this way, future studies can move beyond asking whether developers adopt emerging technologies and examine how organisations can make adoption more understandable, responsible, and sustainable.

6 Conclusion

This paper examined techno-complexity as a specific barrier to developers' acceptance of emerging development technologies. The findings show that higher techno-complexity is associated with lower perceived ease of use and that, through this usability pathway, it weakens intention to use. The study, therefore, supports the argument that technostress should not be treated as a single uniform obstacle to technology adoption.

The paper's contribution is its focused explanation of how complexity affects developer readiness. Rather than suggesting that emerging technologies are rejected simply because they are new or demanding, the findings indicate that complexity becomes problematic when it makes tools feel harder to understand and use. This distinction is important for organisations introducing increasingly intelligent development environments. For practice, the key

message is that responsible technology adoption requires more than access to advanced tools. Organisations should manage the complexity budget of new development technologies by supporting developers with training, gradual implementation, clear communication, and human-centred integration into work practices. As software engineering environments become more automated and AI-assisted, developer readiness will depend not only on technological capability but also on whether complexity is translated into usable, understandable, and responsible professional practice.

References

- Barke, S., James, M. B., & Polikarpova, N. (2023). Grounded Copilot: How programmers interact with code-generating models. *Proceedings of the ACM on Programming Languages*, 7(OOPSLA1), Article 78, 85–111. <https://doi.org/10.1145/3586030>
- Chang, P.-C., Zhang, W., Cai, Q., & Guo, H. (2024). Does AI-driven technostress promote or hinder employees' artificial intelligence adoption intention? A moderated mediation model of affective reactions and technical self-efficacy. *Psychology Research and Behavior Management*, 17, 413–427. <https://doi.org/10.2147/PRBM.S441444>
- Cheng, H., Husen, J. H., Lu, Y., Racharak, T., Yoshioka, N., Ubayashi, N., & Washizaki, H. (2025). Generative AI for requirements engineering: A systematic literature review. *Software: Practice and Experience*. <https://doi.org/10.1002/spe.70029>
- Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, 13(3), 319–340. <https://doi.org/10.2307/249008>
- Feuerriegel, S., Hartmann, J., Janiesch, C., & Zschech, P. (2024). Generative AI. *Business & Information Systems Engineering*, 66(1), 111–126. <https://doi.org/10.1007/s12599-023-00834-7>
- Gombar, M., & Boban, M. (2025). Research on the impact of algorithmic echo chambers on perceptions and attitudes of social network users in a digital society. In 2025 MIPRO 48th ICT and Electronics Convention (MIPRO) (pp. 1026–1033). IEEE. DOI: 10.1109/MIPRO65660.2025.11131918.
- Gombar, M., & Boban, M. (2026). Algorithmic awareness and digital responsibility: The role of platform trust and digital literacy. *Business Systems Research*, 17(1), 179–203. <https://doi.org/10.2478/bsrj-2026-0009>
- Gombar, M., Boban, M., & Pejić Bach, M. (2026). Digital trust and phygital responsibility: A user-centered model for sustainable consumer behavior in algorithmic environments. *World*, 7, Article 86. <https://doi.org/10.3390/world7050086>
- Hayes, A. F. (2022). *Introduction to mediation, moderation, and conditional process analysis: A*

- regression-based approach* (3rd ed.). The Guilford Press.
- Högemann, M., Hein, L., Britsche, J.-O., & Thomas, O. (2025). Technostress and generative AI in the workplace: A qualitative analysis of young professionals. *Frontiers in Artificial Intelligence*, 8, Article 1728881. <https://doi.org/10.3389/frai.2025.1728881>
- Humphreys, D., Koay, A., Desmond, D., & Mealy, E. (2024). AI hype as a cyber security risk: The moral responsibility of implementing generative AI in business. *AI and Ethics*, 4, 791–804. <https://doi.org/10.1007/s43681-024-00443-4>
- Ji, Z., Lee, N., Frieske, R. M., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., & Fung, P. (2023). Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12), Article 248. <https://doi.org/10.1145/3571730>
- Mikalef, P., & Gupta, M. (2021). Artificial intelligence capability: Conceptualization, measurement calibration, and empirical study on its impact on organizational creativity and firm performance. *Information & Management*, 58(3), Article 103434. <https://doi.org/10.1016/j.im.2021.103434>
- Nastjuk, I., Trang, S., Grummeck-Braamt, J.-V., Adam, M. T. P., & Tarafdar, M. (2024). Integrating and synthesising technostress research: A meta-analysis on technostress creators, outcomes, and IS usage contexts. *European Journal of Information Systems*, 33(3), 361–382. <https://doi.org/10.1080/0960085X.2022.2154712>
- Nguyen-Duc, A., Cabrero-Daniel, B., Przybylek, A., Arora, C., Khanna, D., Herda, T., Rafiq, U., Melegati, J., Guerra, E., Kemell, K.-K., Saari, M., Zhang, Z., Le, H., Quan, T., & Abrahamsson, P. (2025). Generative artificial intelligence for software engineering—A research agenda. *Software: Practice and Experience*. <https://doi.org/10.1002/spe.70005>
- Pearce, H., Ahmad, B., Tan, B., Dolan-Gavitt, B., & Karri, R. (2025). Asleep at the keyboard? Assessing the security of GitHub Copilot's code contributions. *Communications of the ACM*, 68(2), 96–105. <https://doi.org/10.1145/3610721>
- Perry, N., Srivastava, M., Kumar, D., & Boneh, D. (2023). Do users write more insecure code with AI assistants? In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security* (pp. 2785–2799). Association for Computing Machinery. <https://doi.org/10.1145/3576915.3623157>
- Prather, J., Reeves, B. N., Denny, P., Becker, B. A., Leinonen, J., Luxton-Reilly, A., Powell, G., Finnie-Ansley, J., & Santos, E. A. (2024). “It’s weird that it knows what I want”: Usability and interactions with Copilot for novice programmers. *ACM Transactions on Computer-Human Interaction*, 31(1), Article 4. <https://doi.org/10.1145/3617367>
- Ragu-Nathan, T. S., Tarafdar, M., Ragu-Nathan, B. S., & Tu, Q. (2008). The consequences of technostress for end users in organizations: Conceptual development and empirical validation. *Information Systems Research*, 19(4), 417–433. <https://doi.org/10.1287/isre.1070.0165>
- Russo, D. (2024). Navigating the complexity of generative AI adoption in software engineering. *ACM Transactions on Software Engineering and Methodology*, 33(5), Article 135, 1–50. <https://doi.org/10.1145/3652154>
- Russo, D., Baltes, S., van Berkel, N., Avgeriou, P., Calefato, F., Cabrero-Daniel, B., Catolino, G., Cito, J., Ernst, N., Fritz, T., Hata, H., Holmes, R., Izadi, M., Khomh, F., Kjærgaard, M. B., Liebel, G., Lafuente, A. L., Lambiase, S., Maalej, W., ... Vasilescu, B. (2024). Generative AI in software engineering must be human-centered: The Copenhagen Manifesto. *Journal of Systems and Software*, 216, Article 112115. <https://doi.org/10.1016/j.jss.2024.112115>
- Sengar, S. S., Hasan, A. B., Kumar, S., & Carroll, F. (2025). Generative artificial intelligence: A systematic review and applications. *Multimedia Tools and Applications*, 84, 23661–23700. <https://doi.org/10.1007/s11042-024-20016-1>
- Tarafdar, M., Cooper, C. L., & Stich, J. F. (2019). The technostress trifecta—Techno eustress, techno distress and design: Theoretical directions and an agenda for research. *Information Systems Journal*, 29(1), 6–42. <https://doi.org/10.1111/isj.12169>
- Vaithilingam, P., Zhang, T., & Glassman, E. L. (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. <https://doi.org/10.1145/3491101.3519665>
- Venkatesh, V., & Davis, F. D. (2000). A theoretical extension of the Technology Acceptance Model: Four longitudinal field studies. *Management Science*, 46(2), 186–204. <https://doi.org/10.1287/mnsc.46.2.186.11926>
- Weisz, J. D., Kumar, S. V., Muller, M., He, J., Houde, S., Ross, S. I., Martinez, F., Agarwal, M., Talamadupula, K., & Richards, J. T. (2025). Examining the use and impact of an AI code assistant on developer productivity and experience in the enterprise. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery. <https://doi.org/10.1145/3706599.3706670>