Human Systems Engineering and Design (IHSED2025), Vol. 198, 2025, 309–317
https://doi.org/10.54941/ahfe1006784

Enhancing Programming Task Performance With LLMs: The Role of Query Formulation and Task-Technology Fit

Darko Etinger and Lucija Josipović Deranja

Faculty of Informatics, Juraj Dobrila University of Pula, 52100 Pula, Croatia

ABSTRACT

This study investigates the effectiveness of large language models (LLMs) in solving programming tasks, with a particular focus on how query formulation influences response quality. Using the Task-Technology Fit (TTF) framework, the research explores the alignment between task requirements and LLM capabilities, and how this alignment impacts student performance. An experiment was conducted with 60 students from the Faculty of Informatics and the Faculty of Engineering at the Juraj Dobrila University of Pula. Participants were asked to solve the "8 Queens" problem in two 30-minute phases: first independently, and then with assistance from the ChatGPT-4 model, during which they could issue 5–10 iterative queries. This setup enabled a comparative analysis of student performance in traditional versus LLM-assisted conditions. Data collection included a structured questionnaire aligned with TTF constructs. The data was analyzed using Partial Least Squares Structural Equation Modeling (PLS-SEM), which assessed the relationships between task characteristics, technology characteristics, task-technology fit, and user performance. The results indicate that both task complexity and technology capabilities significantly contribute to TTF. In turn, TTF strongly influences user performance with the model explaining 47.3% of the variance in TTF and 33.2% in performance. Statistical analysis confirmed a significant improvement in solution accuracy when tasks were completed with LLM support. Furthermore, qualitative analysis showed that well-structured, context-rich queries led to more accurate and relevant model responses. The findings underscore the pivotal role of query formulation in optimizing the use of LLMs for programming tasks. Developing effective query strategies enhances task-technology alignment and ultimately improves performance. This study contributes to a growing understanding of human–AI interaction and highlights the importance of integrating query design skills into educational and professional programming contexts.

Keywords: Large language models (LLMS), Task-technology fit (TTF), Programming performance, PLS-SEM, Human–AI interaction

INTRODUCTION

The rapid advancement of large language models (LLMs), such as ChatGPT and GitHub Copilot, has introduced new possibilities in the automation of programming, code optimization, and intelligent user assistance. Their

ability to comprehend and generate programming code in real time positions them as potentially disruptive tools within both higher education and professional software development (Chen et al., 2024; Amaratunga, 2023). However, despite their technical sophistication, the empirical validation of their effectiveness in educational settings remains limited and often methodologically inconsistent.

In computer science education, LLMs are increasingly integrated into programming courses as tools for student support during problem-solving, algorithm analysis, and independent learning (Sarsa et al., 2022; Sheese et al., 2024). Nevertheless, the quality and usefulness of the responses generated by LLMs largely depend on how users formulate their prompts. Recent studies emphasize the growing importance of *prompt engineering*—the discipline of designing semantically rich, context-aware, and precise queries—as a key factor in improving the reliability and relevance of LLM outputs (Bansal, 2024; Denny et al., 2024). Yet, the role of query formulation in educational contexts remains insufficiently explored and rarely subjected to systematic evaluation.

This study addresses this research gap by combining quantitative and qualitative analyses of LLM-assisted programming task performance in a university setting. The research is grounded in the *Task-Technology Fit* (TTF) framework, which posits that technology effectiveness stems from its alignment with task requirements (Goodhue & Thompson, 1995). Within this study, TTF is operationalized through the interaction between task characteristics, model capabilities, and user performance. This conceptual model provides a foundation for quantifying and interpreting differences in task outcomes under independent versus LLM-assisted conditions.

The objectives of the study are to:

1. Evaluate the extent to which LLM support enhances the fit between task and technology, as defined by the TTF model
2. Assess the impact of LLM usage on task accuracy and efficiency
3. Analyze how different query formulation strategies influence the quality of generated responses.

Based on these objectives, the following hypotheses are proposed:

H1: LLMs demonstrate a high degree of task–technology fit in programming tasks.

H2: LLM usage significantly improves user performance.

H3: There is a statistically significant difference in accuracy between independently and LLM-assisted task completion.

H4: Structured, context-rich prompts yield more accurate and functionally relevant outputs. The scientific contribution of this study lies in applying the TTF framework to the domain of LLM-assisted programming, thereby expanding the current body of knowledge. Additionally, the findings highlight the need to integrate prompt formulation skills into educational programs as a component of digital literacy in AI-supported learning environments.

METHODOLOGY

The study was conducted with a sample of 62 students from the Faculty of Informatics and the Faculty of Engineering (Computer Science program) at Juraj Dobrila University of Pula. Due to incomplete participation, two responses were excluded from the final dataset, resulting in a total of 60 valid participants (74.2% male, 25.8% female), evenly distributed across the two faculties. Participation was voluntary, and all submissions were collected anonymously.

The experiment followed a two-phase design. In the first phase, participants independently solved a programming task without any external assistance. In the second phase, they were asked to solve the same task using the ChatGPT-4 model, with the ability to submit between five and ten prompts. Each phase was limited to 15 to 30 minutes, and incomplete solutions were accepted after the time limit expired. The task required the implementation of a Python function that places the maximum number of queens on a chessboard such that no two queens attack each other, with certain cells blocked. The structure of the experimental procedure and data collection process is illustrated in Figure 1.

Data were collected from three primary sources: task completion time, solution accuracy, and responses to a structured post-task questionnaire. Completion time was recorded manually by participants for each phase. Solution accuracy was evaluated by comparing the output to a predefined optimal solution and recorded as a binary variable (1 = correct, 0 = incorrect). After completing both phases, participants completed a structured questionnaire based on the Task-Technology Fit (TTF) framework, which included constructs related to task characteristics, technology characteristics, task–technology fit, performance, and user experience. Responses were rated using a five-point Likert scale.

In addition to these quantitative measures, the prompts submitted in the second phase were manually evaluated and classified into three categories (good, medium, poor) based on predefined criteria, including clarity, specificity, and technical accuracy. This classification was used for qualitative analysis but was not included in the structural model.

Several statistical methods were employed to analyze the data. McNemar's test was used to assess differences in task accuracy between the two phases, while a paired-samples t-test compared task completion time. Spearman's rank correlation was calculated to explore the relationship between time and accuracy. Structural relationships among the TTF constructs were examined using Partial Least Squares Structural Equation Modeling (PLS-SEM) via SmartPLS version 4.

This methodological approach enabled a comprehensive analysis of the effects of LLM assistance on problem-solving performance, the role of prompt formulation in shaping model output, and the influence of task–technology alignment on user outcomes.

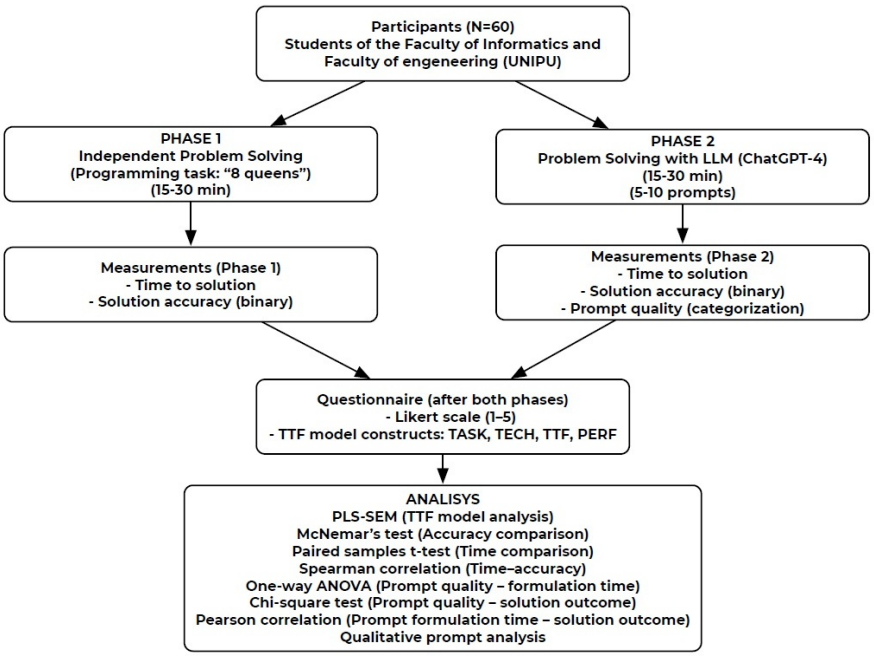

Figure 1: Overview of the experimental design and data collection process.

RESULTS AND DISCUSSION

The results of the structural model, analyzed using Partial Least Squares Structural Equation Modeling (PLS-SEM), indicate a strong relationship between task and technology characteristics and their combined role in predicting user performance when working with large language models (LLMs). Task characteristics ($\beta = 0.366$; $p < 0.001$) and technology characteristics ($\beta = 0.577$; $p = 0.001$) both showed a statistically significant positive impact on task–technology fit (TTF). Furthermore, TTF had a strong positive effect on user performance ($\beta = 0.576$; $p < 0.001$). These findings provide empirical support for H1, confirming that LLMs demonstrate a high degree of task–technology fit in programming tasks, and for H2, showing that TTF significantly contributes to improved user performance. The detailed coefficients and explanatory power of the model are presented in Table 1.

Table 1: Structural model results: path coefficients, significance, and explained variance.

Relationship	Path Coefficient (β)	p-Value	R^2 of Dependent Variable
TASK TTF	0.366	< 0.001	
TECH TTF	0.577	< 0.001	47.3% (TTF)
TTF PERF	0.576	< 0.001	33.2% (PERF)

The structural relationships between the latent constructs were also examined using Partial Least Squares Structural Equation Modeling (PLS-SEM). The resulting model, illustrated in Figure 2, displays the standardized path coefficients between constructs as well as the coefficient of determination (R^2) for each endogenous variable.

This visual representation highlights the statistically significant relationships between task characteristics, technology characteristics, task–technology fit (TTF), and performance (PERF). Specifically, both task and technology characteristics contribute substantially to explaining TTF ($R^2 = 0.473$), while TTF itself explains 33.2% of the variance in performance ($R^2 = 0.332$). These R^2 values suggest a moderate to strong explanatory power of the model, consistent with accepted thresholds in the literature (Hair et al., 2019).

The model confirms all hypothesized relationships with high statistical significance ($p < 0.001$), as previously summarized in Table 1. These outcomes align with H1 and H2, reinforcing the theoretical structure proposed by the Task-Technology Fit framework. The visual layout in Figure 2 enables an intuitive understanding of how input constructs affect the outcome, reinforcing the theoretical structure proposed by the Task-Technology Fit framework. Moreover, it reflects the model's overall robustness, based on the strength of path coefficients and the proportion of variance explained.

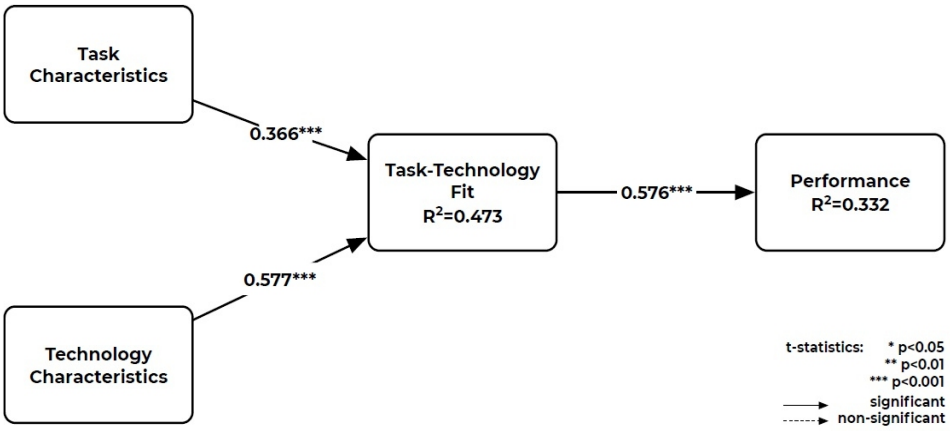

Figure 2: PLS-SEM model results.

In addition to the structural path analysis, internal consistency and construct reliability were examined to assess the quality and robustness of the measurement model. Three key indicators were used for this purpose: Cronbach's Alpha (CA), Composite Reliability (CR), and Average Variance Extracted (AVE).

Cronbach's Alpha is a commonly used measure of internal consistency, indicating the extent to which items within a construct are correlated and therefore measure the same underlying concept. All constructs in the model demonstrated CA values above the recommended threshold of 0.70, with the lowest being 0.768, which suggests a high degree of internal coherence among the measurement items.

Composite Reliability (CR) provides a more refined estimate of internal consistency by accounting for the different loadings of items within a construct. CR values above 0.80 are considered indicative of good reliability,

and in this study, all constructs exceeded this benchmark. Notably, the CR value for the TECH construct reached 0.917, highlighting particularly strong reliability.

Average Variance Extracted (AVE) measures the proportion of variance captured by the construct in relation to the variance due to measurement error. An AVE value of 0.50 or higher is generally taken as evidence of convergent validity, meaning that the indicators effectively reflect the latent construct they are intended to measure. In this study, AVE values ranged from 0.588 to 0.786, confirming that each construct captures a sufficient amount of its associated item variance.

As presented in Table 2, these results collectively indicate that the constructs used in the model are both reliable and valid. The measurement model thus meets established psychometric criteria and provides a solid foundation for interpreting the structural relationships in the subsequent analysis.

Table 2: Internal consistency and construct reliability.

Construct	PERF	TASK	TECH	TTF
Cronbach's Alpha (CA)	0.824	0.768	0.865	0.868
Composite Reliability (CR)	0.883	0.805	0.917	0.910
Average Variance Extracted (AVE)	0.654	0.588	0.786	0.716
PERF	**0.808**			
TASK	0.155	**0.767**		
TECH	0.535	0.014	**0.887**	
TTF	0.576	0.374	0.583	**0.846**

In addition to the structural model, quantitative indicators further support the findings. Solution accuracy increased from 46.7% in the first phase (without assistance) to 66.7% in the second phase (with LLM support), with the difference confirmed as statistically significant using McNemar's test ($p = 0.021$). This statistically significant difference supports H3, confirming that LLM-assisted task completion leads to higher solution accuracy. The average task completion time decreased from 26.7 to 20.9 minutes, and a paired-samples t-test indicated a significant difference ($t = 4.72$; $p < 0.001$). A weak but significant negative correlation was also found between time and accuracy ($\rho = -0.285$; $p = 0.029$). These metrics are summarized in Table 3.

Table 3: Descriptive metrics and significance tests.

Measured variable	Phase 1	Phase 2	Statistical test
Accuracy	46.7% (28/60)	68.3%	McNemar, $p = 0.021$
Avg. time (minutes)	26.7	20.9	$t = 4.72$, $p < 0.001$
Time–accuracy correlation	-	-	$\rho = -0.285$, $p = 0.029$

Prompt quality analysis showed that 38.3% of participants submitted high-quality prompts, 35% medium-quality, and 26.7% low-quality prompts. Although not part of the structural model, this analysis provided valuable qualitative insight into user interaction patterns that correlated with

solution accuracy. While no statistically significant association was found between prompt quality and solution accuracy, the timing data suggest that higher-quality prompts required more time to formulate, possibly indicating greater cognitive engagement. These observations provide partial support for H4, suggesting that structured, context-rich prompts contribute to more accurate and relevant model responses, albeit with varying degrees of statistical confirmation.

These results support the assumptions of the TTF theory and emphasize the importance of aligning technology with task nature to improve efficiency and accuracy in problem-solving. They also raise important questions for future research, particularly in the field of prompt engineering and the ways in which various prompting strategies influence the quality of model-generated outputs.

Part of these findings align with other studies that emphasize the importance of prompt formulation quality in shaping LLM performance. For example, Zhou et al. (2024) found that prompts with contextual information and explicit instructions significantly improved the accuracy and relevance of generated outputs. Similarly, Jošt et al. (2024) observed that reliance on structured prompts can lead to more effective task guidance, although it may also reduce opportunities for deeper individual reasoning.

Beyond prompt quality, this study also highlights prompt formulation time as an important indicator. These aspects point to the need for educational strategies that not only promote students' technical understanding of LLMs but also support the cognitive alignment between problem-solving strategies and AI interaction patterns (Rodríguez-Serrano, 2024).

Altogether, the integration of LLM systems in education should not be considered solely in terms of their technical capabilities, but also through the lens of user interaction dynamics, which directly influence outcomes. This opens space for the design of learning tools that include prompt support, interaction monitoring, and adaptive learning environments.

CONCLUSION

This study examined the effectiveness of large language models (LLMs) in supporting students with programming tasks, focusing on the role of user query formulation and the theoretical framework of Task-Technology Fit (TTF). Results from an experimental study involving 60 students demonstrate that the application of LLM tools significantly improves task accuracy and reduces completion time, especially when the alignment between task characteristics and technology capabilities is high.

The structural model (PLS-SEM) confirmed that both task and technology characteristics significantly influence the perception of TTF, which in turn acts as a strong predictor of user performance. Quantitative findings further revealed a statistically significant increase in solution accuracy and a reduction in solving time with LLM support. Although the quality of prompts was not directly associated with solution accuracy, higher-quality prompts required more time to formulate, suggesting greater cognitive engagement.

The scientific contribution of this study lies in the empirical application of the TTF model in the context of LLM use, while its practical relevance highlights the importance of developing prompt engineering skills within educational settings. Integrating LLM tools into programming education can offer considerable benefits, but it requires a balanced approach that fosters critical thinking and independent problem-solving.

Limitations of the study include the specificity of the sample and the nature of the programming task, which may affect the generalizability of the findings. Future research should explore a wider range of tasks, compare different LLM systems, and investigate prompt formulation strategies in more detail.

Ultimately, the results confirm that the ability to formulate effective prompts is becoming a critical competence in interactions with LLM technologies. The effectiveness of these tools depends not only on their underlying capabilities but also on the user's ability to engage them through precise, goal-oriented, and contextually rich prompts.

ACKNOWLEDGMENT

The authors would like to thank the Faculty of Informatics and the Faculty of Engineering at the Juraj Dobrila University of Pula for their support in conducting the research. The authors are especially grateful to the students who participated in the experiment, without whom this study would not have been possible.

REFERENCES

Amaratunga, T. (2023). *Understanding Large Language Models*. Apress. https://doi.org/10.1007/979-8-8688-0017-7

Bansal, P. (2024). Prompt engineering importance and applicability with generative AI. *Journal of Computer and Communications*, 12, 14–23. https://doi.org/10.4236/jcc.2024.121002

Chen, A., Wei, Y., Le, H., & Zhang, Y. (2024). Learning-by-teaching with ChatGPT: The effect of teachable ChatGPT agent on programming education. https://arxiv.org/pdf/2412.15226

Denny, P., Leinonen, J., Prather, J., Luxton-Reilly, A., Amarouche, T., Becker, B. A., & Reeves, B. N. (2024). *Prompt problems: A new programming exercise for the generative AI era*. In *Proceedings of the ACM Conference on International Computing Education Research* (pp. pages). Publisher. https://doi.org/10.1145/3626252.3630909

Goodhue, D. L., & Thompson, R. L. (1995). Task-technology fit and individual performance. *MIS Quarterly, 19*(2), 213–236.

Hair, J. F., Hult, G. T. M., Ringle, C. M., & Sarstedt, M. (2019). *A Primer on Partial Least Squares Structural Equation Modeling (PLS-SEM)* (2nd ed.). Thousand Oaks, CA: SAGE Publications. https://doi.org/10.4135/9781076218956

Jošt, G., Taneski, V., & Karakatič, S. (2024). The impact of large language models on programming education and student learning outcomes. *Applied Sciences, 14*(10), 4115. https://doi.org/10.3390/app14104115

Rodríguez-Serrano, J. A. (2024). *What is the future of programming with large language models?* ESADE. https://dobetter.esade.edu/en/programming-LLM

Sarsa, S., Denny, P., Hellas, A., & Leinonen, J. (2022). Automatic generation of programming exercises and code explanations using large language models. In *Proceedings of the 2022 ACM Conference on International Computing Education Research, V. 1* (pp. 531–537). ACM. https://doi.org/10.1145/3501385.3543957

Sheese, B. E., Liffiton, M., Šavelka, J., & Denny, P. (2024). *Patterns of student help-seeking when using a large language model-powered programming assistant*. In *Proceedings of the 2024 ACM Conference on International Computing Education Research (ICER '24)* (pp. [pages]). ACM. https://doi.org/10.1145/3636243.3636249

Zhou, K. Z., Kilhoffer, Z., Sanfilippo, M. R., Underwood, T., Gumusel, E., Wei, M., Choudhry, A., & Xiong, J. (2024). *"The teachers are confused as well": A multiple-stakeholder ethics discussion on large language models in computing education*. ArXiv (Cornell University). arXiv:2401.12453.