ChatGPT as a Software Development Bot: A Project-Based Study

Muhammad Waseem[1], Teerath Das[1], Aakash Ahmad[2], Peng Liang[3], Mahdi Fahmideh[4] and Tommi Mikkonen[1]

[1]*Faculty of Information Technology, University of Jyväskylä, Jyväskylä, Finland*
[2]*School of Computing and Communications, Lancaster University Leipzig, Leipzig, Germany*
[3]*School of Computer Science, Wuhan University, Wuhan, China*
[4]

Keywords: ChatGPT, AI for SE, Automated Software Engineering, Learning Impact, Empirical SE.

Abstract: Artificial Intelligence has demonstrated its significance in software engineering through notable improvements in productivity, accuracy, collaboration, and learning outcomes.This study examines the impact of generative AI tools, specifically ChatGPT, on the software development experiences of undergraduate students. Over a three-month project with seven students, ChatGPT was used as a support tool. The research focused on assessing ChatGPT's effectiveness, benefits, limitations, and its influence on learning. Results showed that ChatGPT significantly addresses skill gaps in software development education, enhancing efficiency, accuracy, and collaboration. It also improved participants' fundamental understanding and soft skills. The study highlights the importance of incorporating AI tools like ChatGPT in education to bridge skill gaps and increase productivity, but stresses the need for a balanced approach to technology use. Future research should focus on optimizing ChatGPT's application in various development contexts to maximize learning and address specific challenges.

1 INTRODUCTION

Artificial Intelligence (AI) and especially Machine Learning (ML) is increasingly being integrated into software development processes (Barenkamp et al., 2020), leveraging ML algorithms and intelligent systems to automate code generation, bug fixing, and testing, thereby enhancing efficiency and reducing manual effort, (Shehab et al., 2020). This technological progression paves the way towards Large Language Models (LLMs) (Kim et al., 2021), which have revolutionized textual and coding tasks by understanding and generating human-like text. Notable developments within LLMs include BERT (Devlin et al., 2019), GPT, and LLaMA, each having distinct functionalities. Among these, the Generative Pre-trained Transformer model, for example GPT-3 has garnered substantial attention due to its ability to train on extensive human data and generate coherent text, both labelled and unlabelled datasets.

Context and Motivation: GPTs have pervaded numerous applications, notably AI chatbots, content creation, and diverse areas within software development, as demonstrated by multiple studies and practical applications (e.g., (Ahmad et al., 2023)). Nevertheless, empirical reports specifically detailing the application of GPTs, including GPT-3 and the more recent ChatGPT, in the training of undergraduate students across various software development phases remain notably limited. We will use 'students' for short in the rest of the paper. For instance, GPT-3 has been investigated for its capabilities in automatic code generation and automated code documentation, and utilized to develop tools like *Codex*, which converts natural language instructions into code and aids in automatically generating code documentation (Khan and Uddin, 2022). Simultaneously, students consistently face challenges such as understanding fundamental architectural concepts, managing subsystem integration, and confronting debugging issues. A survey involving both educators and students highlighted common struggles that students face, including program designing, procedure segmentation, and error identification within their code, while additional research has shown the occurrence of recurrent code quality issues in novice-authored programs (Techapalokul and Tilevich, 2017) and difficulties in utilizing tools like version control systems, performing infrastructure testing, and managing bug tracking and resolution.

Waseem, M., Das, T., Ahmad, A., Liang, P., Fahmideh, M. and Mikkonen, T.
ChatGPT as a Software Development Bot: A Project-Based Study.
DOI: 10.5220/0012631600003687

In *Proceedings of the 19th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2024)*, pages 406-413
ISBN: 978-989-758-696-5; ISSN: 2184-4895

Objective of the Study: The objective of this study is to explore the effectiveness of ChatGPT as software development bot within different phases of the software development life cycle, as undertaken by students. This includes evaluating ChatGPT's impact on requirements analysis, design, architecture, development, testing, and deployment. The study also aims to identify ChatGPT's advantages and limitations in each phase, understand its influence on students' learning curves and skill development, assess improvements in software development proficiency, and pinpoint challenges faced by students. By considering objective we formulated following four Research Questions (RQs):

- **RQ1:** How did the utilization of ChatGPT as development bot impact the requirements analysis, design, architecture, development, testing, and deployment phases of software development, and what are the advantages and limitations of its use at each phase of the software development life cycle?
- **RQ2:** How did the involvement in software development projects that incorporated ChatGPT influence the learning curve and skill development of the students?
- **RQ3:** To what extent did the students' proficiency in software development concepts improve through their participation in these projects with the help of ChatGPT as development bot?
- **RQ4:** What challenges did the students face when using ChatGPT as development bot?

The **contributions** of this study are (i) assessing ChatGPT's impact on students' software development lifecycle, with a focus on advantages and limitations; (ii) identifying challenges and evaluating ChatGPT's educational value in student software projects; and (iii) presenting evidence of ChatGPT's integration and effectiveness in software development, contributing to discussions on human-AI collaboration.

2 RELATED WORK

2.1 ChatGPT-Assisted Software Design

Bencheikh et al.(Bencheikh and Höglund, 2023) conducted a study that explores the efficacy of AI tools, particularly ChatGPT, in generating software requirements efficiently. They highlighted the ability of ChatGPT to emulate human expertise, though emphasizing the indispensable nature of human feedback to enhance requirement quality. Moreover, the time efficiency of ChatGPT was acknowledged, albeit with a recognition that experienced human participants tend to produce more comprehensive requirements. The study also differentiated between the premium and free versions of ChatGPT, showing a superior consistency and overall quality in the former. On a related note, White et al.(White et al., 2023) proposed several ChatGPT prompt patterns to elevate software requirements elicitation. Transitioning to the architectural phase of software development, a case study by Ahmad et al. (Ahmad et al., 2023) shed light on the collaborative efforts between a novice software architect and ChatGPT.

2.2 Code Generation and Testing with ChatGPT

In the coding and implementation, several researchers have used ChatGPT. For instance, Al-Khiami et al.(Al-Khiami and Jaeger, 2023) conducted a case study to explore the feasibility of using ChatGPT for generating JavaScript code suitable for Android Studio, aimed at creating a functional app. Concurrently, Bera et al.(Bera et al., 2023) employed ChatGPT to support agile software development. Additionally, the significance of ChatGPT prompt patterns in improving code quality and facilitating refactoring was discussed by White et al. (White et al., 2023). Beyond these, the GPT-3 model has been utilized for automatic code generation (Narasimhan et al., 2021) and automation of code documentation (Khan and Uddin, 2022), with Tian et al.(Tian et al., 2023) employing ChatGPT as a programming assistant, albeit noting limitations regarding ChatGPT's attention span.

3 RESEARCH METHOD

The research method of this study consists of four phases, illustrated in Figure 1 and detailed as follows.

3.1 Developers and Projects Selection

Developers Selection: Seven undergraduates from IT, computer science, and software engineering programs were selected to evaluate their software development experiences with ChatGPT (versions GPT-3.5 and GPT-4), recruited through university's online bulletin board and social media, followed by interviews. These first or second-year students were chosen for their limited familiarity with software development practices, including Scrum, CI/CD, UML, Python programming, automated testing, and deployment.

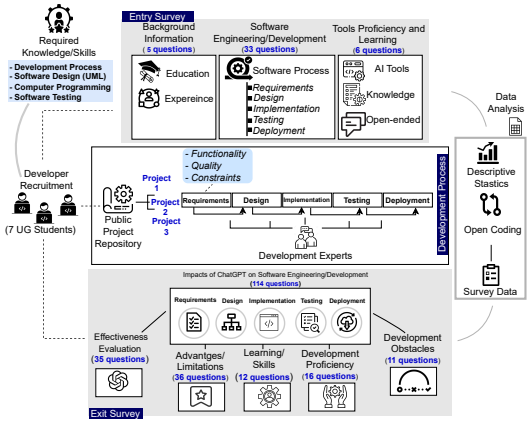

Figure 1: Overview of the Research Method.

Project Selection: This study focus on three publicly announced software projects assigned to a group of seven students, divided into two teams. ChatGPT, an AI-powered tool, was utilized extensively to support students throughout the software development process, spanning from understanding project requirements to deployment.

We developed three systems to enhance procurement processes, foster educational technology collaboration, and assess student performance. (i) For Solita Ltd, the AI Procurement Assistant (AIPA) utilized AI to streamline identifying procurement opportunities, with ChatGPT simplifying procurement option identification. (ii) The AI-based Teacher-Tech Forum, created for Jyväskylä University of Applied Sciences, established an online platform for teachers and tech companies to exchange ideas for using technology in education. (iii) Lastly, the AI-based Skills Assessment SaaS, aimed at city governments through the AXZ public procurement system, assessed junior-level students' performance in subjects like mathematics and English, providing municipalities and schools with valuable insights into student achievements.

3.2 Conducting Entry Survey

In this study, we adopted a cross-sectional survey design, recognized for its effectiveness in capturing data at a specific point in time across a targeted population (Kitchenham and Pfleeger, 2008). Opting for a web-based approach facilitated an efficient, cost-effective method of data collection in diverse formats (Lethbridge et al., 2005). The survey's design was informed by both grey and peer-reviewed literature, alongside consultations with software engineering professors possessing industrial experience. Our team, leveraging academic and industrial expertise, refined the survey through focus group discussions, ensuring its alignment with the study's objectives. Notably, a pilot survey was conducted with two students external to the development team to validate the survey instrument's relevance and effectiveness. The survey instrument, detailed in the Entry Survey Questionnaire sheet (Waseem et al., 2024), comprised three sections with a total of 53 questions, designed to explore the participants' backgrounds, software engineering experiences, and proficiency with AI tools, including their readiness to learn these tools.

3.3 Development Process

The development process began with identifying projects from Finland's public procurement system. One project was chosen from Solita Oy software development company, one from Jyväskylä University of Applied Sciences (JAMK), and one from the procurement system, following discussions between the study authors and developers. The students were distributed into two teams, working on their projects in parallel. Team roles, like front-end and back-end developers, were predefined. After selecting the projects, the developers engaged in iterative activities, such as collecting requirements, setting up CI/CD infrastructure, software design, project development, system testing, and system deployment, in collaboration with client representatives from Solita Oy and JAMK University.

3.4 Conducting Exit Survey

The exit survey, similar to the entry survey, used a cross-sectional approach (Kitchenham and Pfleeger, 2008) (detailed in the Exit Survey Questionnaire sheet (Waseem et al., 2024)) with 114 questions. It aimed to thoroughly understand ChatGPT's impact on software development, assessing its effectiveness, advantages, limitations, educational value, challenges, and its potential to improve project outcomes and developer skills. Tailored to address the study's Research Questions, it used a 5-level Likert scale for responses. The survey is organized into five sections, each focusing on different aspects of ChatGPT's impact in software development projects.

3.5 Analysis

We employed descriptive statistics to analyze the quantitative (i.e., closed-ended questions). To evaluate the responses to few open-ended questions, we employed open coding from grounded theory (Glaser and Strauss, 2017) to segment and label the survey

data, thereby identifying and emphasizing the text fragments in each student's response that could be treated as distinct data points regarding ChatGPT.

4 RESULTS

4.1 Background, Experience, and Readiness

We present below the key results from the survey.

Developers' Background: The survey data revealed that the participants mostly had high school education and limited software development experience (see Figure 2): four with 1-2 years and three with less than a year, related to undergraduate computing studies. Most self-assessed their software engineering skills as 'Intermediate', indicating a lack of advanced expertise. There was also a noticeable unfamiliarity with agile methodologies and limited proficiency in professional collaboration tools like Jira, suggesting the participants lacked the typical educational and practical skills for advanced software development roles.

Experience in Software Engineering: The entry survey with 35 questions on the software development life cycle showed a consistent trend (Figure 2). Participants were mostly 'Somewhat familiar' with Software Requirements Engineering tasks and 'Somewhat comfortable' in Development and Design phases. Confidence in Implementation, Testing, and Quality Assurance ranged from 'Somewhat' to 'Moderately confident', with varying proficiency in coding and test strategy development. The largest skill gap was in Deployment and Release Management, where most had 'No experience' or were 'Not proficient at all', underscoring significant skill gaps, particularly in Deployment and Release Management.

Readiness to Learn About AI Tools: Figure 2 outlines participants' AI tool proficiency and learning readiness in software development. Of seven respondents, five had minimal AI tool usage, with some familiarity through ChatGPT. They showed higher proficiency in Machine Learning for Predictive Analytics and Deep Learning for Image/Audio Processing. The preferred learning methods for AI skills were hands-on experimentation and projects, followed by online tutorials or video courses. Collaboration on AI projects was also favored, indicating a preference for collaborative learning. Less preferred were reading research papers and attending workshops or seminars on AI tools.

Perception and Potential of AI Tools: Open-ended questions explored students' views on AI tools in

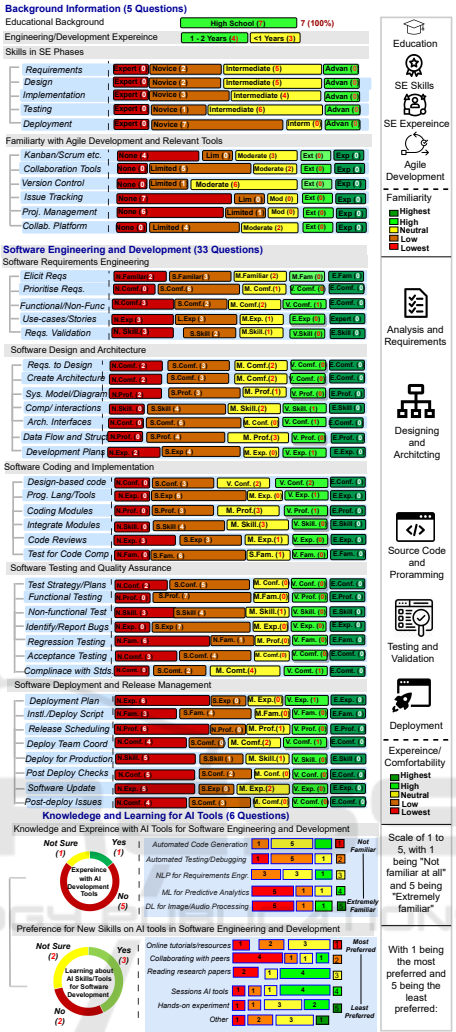

Figure 2: Results Overview: Background, Experience, Readiness.

software development, covering aspects like enhancement, limitations, and future prospects. We summarize participants' responses (P1 to P7) as follows:

(i) Students perceived AI tools like ChatGPT as beneficial, especially for complex projects with limited experience. They appreciated AI's quick code generation and efficiency in finding tools and algorithms, reducing manual search time (P3, P4, P5). However, concerns about the quality of rapid outputs and integration difficulties were noted (P1, P2). While AI offers productivity and efficiency, quality assurance and system integration remain challenges.

(ii) Concerns included AI's impact on privacy, ethics, developer education, and reliability. Risks of privacy breaches and sensitive information leaks in AI-generated code were mentioned (P1, P2), as

well as ethical concerns over code ownership (P4). Over-reliance on AI possibly hindering skill development (P3) and AI code's reliability (P7) were other concerns.

(iii) Mixed feedback was given on AI's integration in software development. Concerns about AI's long-term sustainability (P1), the development of AI tools for larger project assistance (P2), trust issues due to past inaccuracies (P5), and AI's potential in streamlining workflows for new developers (P6) were discussed. Feedback ranged from acknowledging AI's benefits to addressing challenges in trust and effective utilization.

Takeaways

Significant Skill Gaps: The surveyed developers showed notable gaps in their software engineering skills, especially in deployment and release management stages.

Eagerness to Learn AI: Despite minimal previous experience with AI tools, participants demonstrated a strong desire to learn, particularly through hands-on experiences and projects.

4.2 Effectiveness, Advantages, and Limitations (RQ1)

To answer the RQ1, we asked 71 survey questions to gather answers from students who extensively used ChatGPT for three months to develop three systems as desribed in Section 3.

Effectiveness: Developers reported (see Figure 3) that ChatGPT's positive impact on software development's early stages. In requirements analysis, 7 respondents agreed ChatGPT significantly reduced time for defining requirements and 6 agreed it helped accurately identify needs. In design, 5 respondents agreed ChatGPT led to clearer, scalable architectures and 7 agreed it aligned architectural decisions with project constraints. ChatGPT also streamlined the development phase, improving coding process and quality as agreed by 6 students. In testing, it optimized processes, enabled early defect detection, and improved test coverage, with 3 to 6 respondents agreeing on its effectiveness in bug identification, risk management, and software reliability.

Takeaways

Proficiency in Software Development: ChatGPT consistently enhanced foundational understanding and soft skills in software development among participants, but experiences varied notably in its ability to provide unique insights during the software development process.

Advantages: Developer responses (see Figure 3) highlighted the perceived benefits of using ChatGPT in software development. Seven students agreed that ChatGPT enhanced requirement clarity. Its role in stakeholder communication was mixed, with 4 neutral and 3 in agreement. Opinions on its impact in conflict resolution, innovative design formulation, and efficient design validation were diverse. However, promoting user-centered design, robust software architecture, and informed architectural decision-making received positive feedback, with most students seeing advantages. Six students strongly agreed on the benefits of clear architecture representation.

Takeaways

Positive Impact: ChatGPT was generally perceived to positively impact various facets of software development learning and skill development, despite some isolated instances of dissent or neutral perspectives from participants.

Limitations: The survey addressed ChatGPT's limitations in software development. Opinions on complex requirement limitations were mixed, with 3 students disagreeing and 4 neutral. Analysis dependency risk saw varied responses (4 agreeing, 2 neutral). Objective misalignment in design was mostly neutral among 5 students. Design oversimplification and impractical recommendations received divided opinions, leaning slightly towards agreement. Creativity hindrance concerns were notable, with a significant portion disagreeing. Mixed responses were seen in architectural expertise neglect and constraint conflict.

4.3 Learning Curves and Skill Development (RQ2)

To get the answer of RQ2, we asked 11 survey questions to gather students' opinions on the influence of ChatGPT on their learning and skill development. The survey results highlight the participants' experiences and perceptions regarding several aspects of using ChatGPT in their software development projects (see Figure 3). For instance, when asked about "Reducing the time needed to understand software development concepts", the feedback was mostly positive: 3 participants agreed, and 3 strongly agreed. Similarly, "Enhancing adaptability to new technology" was positively received with 5 strongly agreeing and 1 agreeing.

Takeaways

Valuable Learning Despite Challenges: Participants unanimously agreed that they got valuable learning and insights from these experiences, suggesting that obstacles faced were constructively impactful on their developmental journey.

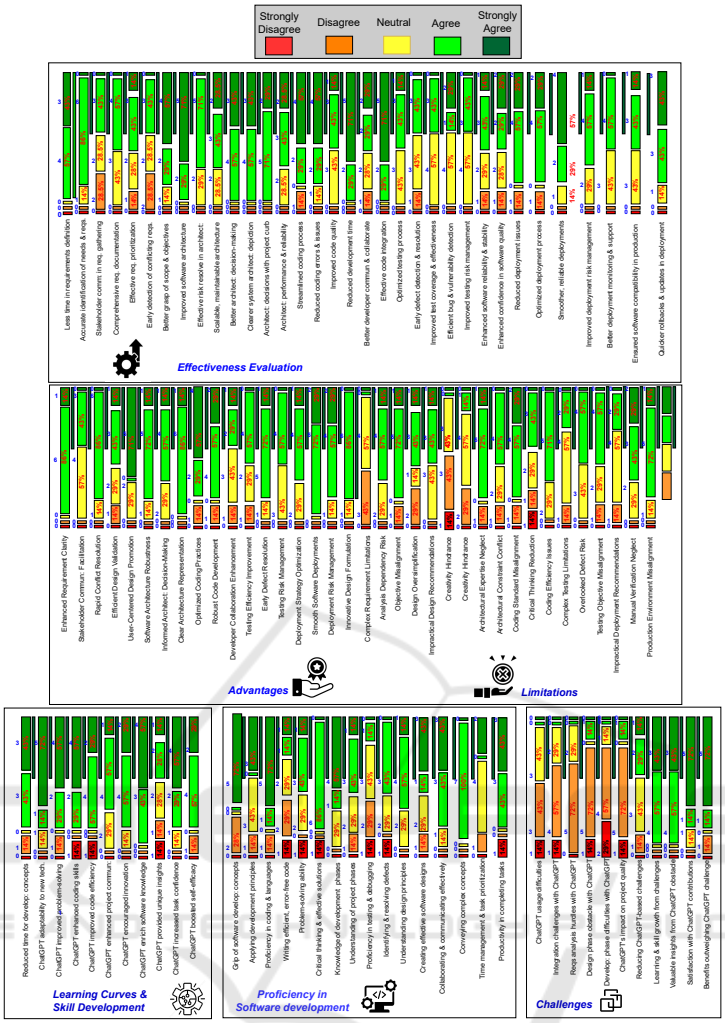

Figure 3: Results Overview: ChatGPT's effectiveness, Advantages, Limitations, Skill development, Proficiency.

4.4 Proficiency in Software Development (RQ3)

An agreed sentiment of approval was observed concerning ChatGPT's role in enriching software knowledge (see Figure 3), with all 7 respondents either agreeing or strongly agreeing. This indicates a consistent acknowledgment of ChatGPT's capacity to enhance foundational understanding and proficiency in software development among the participants. Further, the efficacy of ChatGPT was evident in several other facets of the software development learning process. For example, 6 out of the 7 respondents concurred (agree or strongly agree) that the tool positively impacted their adaptability to new technologies, boosted practical problem-solving skills, and encouraged innovative thinking during their development.

On the other hand, it is noteworthy that certain aspects yielded more varied perceptions, providing a view of the participants' experiences. Specifically, regarding the statement, "ChatGPT provided unique insights," there was an almost equal distribution of responses across all available options (strongly disagree to strongly agree), showcasing that the participants had diverse experiences and perceptions concerning ChatGPT's capability in offering unique insights during the software development process.

4.5 Challenges (RQ4)

In addressing RQ4, we asked 11 survey questions to understand the difficulties of students might have experienced while using ChatGPT in their projects. When it came to difficulties in various phases of project development with ChatGPT, the students generally did not find using ChatGPT hard or tricky. For

example, in dealing with ChatGPT usage difficulties, 4 out of 7 respondents disagreed, suggesting that most found it user-friendly.

Even with the overall positive feedback regarding usability, there were areas where participants felt they learned and improved their skills from the challenges they faced. A key observation was that all participants (7 out of 7) agreed or strongly agreed that they experienced learning and skill growth from the challenges faced while using ChatGPT. In the same way, all respondents agreed or strongly agreed that they gained valuable insights from the challenges faced with ChatGPT.

5 DISCUSSION

❶ **Positive Impact on Software Development Phases:** This study reported ChatGPT support for various software development phases, from requirements analysis to deployment, aligns with existing literature highlighting the beneficial role of AI in streamlining development workflows. For example, AI-driven tools have been substantiated for enhancing requirement gathering through natural language processing capabilities and facilitating design phases (Ahmad et al., 2023). The positive impact across development phases prompts deeper explorations into specifying contexts, projects, or phases where ChatGPT's application could be maximized.

❷ **Diverse Opinions on Advantages and Limitations:** The presence of both significant advantages and noticeable limitations in using ChatGPT for software development echoes previous literature (e.g., (Kalla and Smith, 2023)) on AI tools in development environments. In terms of advantages, the recognized enhancement in requirement clarity and promotion of user-centered design correlate with the narrative of AI-driven tools being useful in translating user needs into technical requirements efficiently. The support in clear architecture representation and testing efficiency is coherent with studies that underscore automated testing and model-driven development (e.g., (Planas et al., 2021)) facilitated by AI. In contrast, the limitations, particularly around coding efficiency issues and design recommendation impracticalities, mirror concerns found in both in peered review and gray literature (e.g., (CodeLogic, 2023)) that cautions against over-reliance on AI for complex decision-making in software development. ChatGPT shows promise for enhancing parts of the Software Development Life Cycle (SDLC) but must be used wisely due to its limitations, like possibly oversimplifying complex tasks or suggesting unhelpful designs due to its partial understanding of context and software tasks.

❸ **Positive Impact on Learning and Skill Development:** The majority of students found ChatGPT beneficial in aiding their understanding and skill enhancement in various facets of software development, which aligns with literaturesuggesting that AI can expedite the learning curve by providing instant, context-aware assistance (e.g., (Al-Khiami and Jaeger, 2023)). However, the dissenting responses, particularly the one participant who strongly disagreed about enhancing coding skills, indicating that personal experiences with ChatGPT varied. ChatGPT, acting as a supplementary tool, seems to align with these findings by providing support and instant feedback, enhancing both the theoretical and practical understanding of the students. However, it is crucial to consider the small sample size of 7 students, which may limit the generalizability of these findings. Positive impacts on skill development might be subjected to the initial proficiency level of the participants or other unaccounted contextual factors.

❹ **Varied Experiences About Software Knowledge and Soft Skills Development:** Even though ChatGPT positively impacted software knowledge and soft skills development, it provided varied experiences regarding unique insights into the software development process among participants. The varied experiences with ChatGPT mirror the existing literature on AI in education, which suggests that while AI can offer valuable support, the utility can differ based on users' expectations, existing skills, and the nature of tasks (Fraiwan and Khasawneh, 2023). The disparity in how ChatGPT's capability to provide unique insights was perceived implies that the tool might be interpreted or utilized differently among users. The limitation pertains to understanding the depth and nature of these varied experiences, given that the reasoning behind such divergence is not explored in detail.

❺ **Valuable Learning Despite Challenges:** Challenges faced when using ChatGPT were not seen as drawbacks but rather as valuable learning experiences, aiding skill development and insight generation among students. This aligns with the pedagogical perspective that views challenges and obstacles as crucial learning elements that facilitate deep understanding and skills mastery (Ohlsson, 2014). Challenges faced, while initially perceived as hurdles, become opportunities for active learning and skill enhancement. Given the unanimously positive feedback regarding learning from challenges, there may be potential bias in the responses, or there might be a limitation in exploring the specific nature and impact of these challenges due to the small participant group and the quantitative approach of a survey.

6 CONCLUSIONS

This study explored ChatGPT's impact on students in software projects as development bot, revealing skill gaps and a keen interest in AI. ChatGPT proved to be a beneficial support tool, enhancing efficiency and collaboration, but opinions varied on its advantages and limitations in design and coding. It improved participants' software development skills and soft skills, though its contribution to unique insights was inconsistent. Despite challenges, the experience was viewed positively for learning, underscoring AI tools' educational value. Future research will explore into diverse experiences with ChatGPT as development bot, optimize its use with multi-agents, and create guidelines for effectively integrating AI in software engineering education at both undergraduate and graduate levels.

REFERENCES

Ahmad, A., Waseem, M., Liang, P., Fahmideh, M., Aktar, M. S., and Mikkonen, T. (2023). Towards human-bot collaborative software architecting with chatgpt. In *Proceedings of the 27th International Conference on Evaluation and Assessment in Software Engineering (EASE)*, pages 279–285.

Al-Khiami, M. I. and Jaeger, M. (2023). Leveraging chatgpt in android app development: A case study on supporting novice developers in creating successful apps. *Preprints*.

Barenkamp, M., Rebstadt, J., and Thomas, O. (2020). Applications of ai in classical software engineering. *AI Perspectives*, 2(1):1.

Bencheikh, L. and Höglund, N. (2023). Exploring the efficacy of chatgpt in generating requirements: An experimental study. Bachelor's Thesis, Chalmers University of Technology.

Bera, P., Wautelet, Y., and Poels, G. (2023). On the use of chatgpt to support agile software development. In *Short Paper Proceedings of the 2nd International Workshop on Agile Methods for Information Systems Engineering (Agil-ISE) co-located with the 35th International Conference on Advanced Information Systems Engineering (CAiSE)*, volume 3414, pages 1–9. CEUR-WS.org.

CodeLogic (2023). The future of software development: Maximizing benefits and minimizing risks of ai-assisted coding. Accessed: September 12, 2023.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL), Volume 1 (Long and Short Papers)*, pages 4171–4186. ACL.

Fraiwan, M. and Khasawneh, N. (2023). A review of chatgpt applications in education, marketing, software engineering, and healthcare: Benefits, drawbacks, and research directions. *arXiv:2305.00237*.

Glaser, B. G. and Strauss, A. L. (2017). *Discovery of grounded theory: Strategies for qualitative research*. Routledge.

Kalla, D. and Smith, N. (2023). Study and analysis of chat gpt and its impact on different fields of study. *International Journal of Innovative Science and Research Technology*, 8(3).

Khan, J. Y. and Uddin, G. (2022). Automatic code documentation generation using gpt-3. In *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 1–6. IEEE.

Kim, Y. J., Awan, A. A., Muzio, A., Salinas, A. F. C., Lu, L., Hendy, A., Rajbhandari, S., He, Y., and Awadalla, H. H. (2021). Scalable and efficient moe training for multitask multilingual models.

Kitchenham, B. A. and Pfleeger, S. L. (2008). Personal opinion surveys. In *Guide to Advanced Empirical Software Engineering*, pages 63–92. Springer.

Lethbridge, T. C., Sim, S. E., and Singer, J. (2005). Studying software engineers: Data collection techniques for software field studies. *Empirical Software Engineering*, 10(3):311–341.

Narasimhan, A., Rao, K. P. A. V., et al. (2021). Cgems: A metric model for automatic code generation using gpt-3. *arXiv:2108.10168*.

Ohlsson, J. (2014). Pedagogic challenges in the learning organization. *The Learning Organization*, 21(3):162–174.

Planas, E., Daniel, G., Brambilla, M., and Cabot, J. (2021). Towards a model-driven approach for multiexperience ai-based user interfaces. *Software and Systems Modeling*, 20:997–1009.

Shehab, M., Abualigah, L., Jarrah, M. I., Alomari, O. A., and Daoud, M. S. (2020). Artificial intelligence in software engineering and inverse: Review. *International Journal of Computer Integrated Manufacturing*, 33(10-11):1129–1144.

Techapalokul, P. and Tilevich, E. (2017). Understanding recurring quality problems and their impact on code sharing in block-based software. In *Proceedings of the 2017 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 43–51.

Tian, H., Lu, W., Li, T. O., Tang, X., Cheung, S.-C., Klein, J., and Bissyandé, T. F. (2023). Is chatgpt the ultimate programming assistant–how far is it? *arXiv:2304.11938*.

Waseem, M., Das, T., Ahmad, A., Liang, P., Fahmideh, M., and Mikkonen, T. (2024). Dataset for the Paper: ChatGPT as a Software Development Bot: A Project-based Study. https://zenodo.org/doi/10.5281/zenodo.10457831.

White, J., Hays, S., Fu, Q., Spencer-Smith, J., and Schmidt, D. C. (2023). Chatgpt prompt patterns for improving code quality, refactoring, requirements elicitation, and software design. *arXiv:2303.07839*.