Exploring the use of generative artificial intelligence in systems development

Matt Hyde

Abstract

The use of generative artificial intelligence (AI) to generate data requested by a human provides advantages and disadvantages for systems development. The main advantage is its potential to assist developers, leading to an increase in their productivity and work quality. The disadvantages include the risk of generating incorrect work, legal issues of the infringement of copyrighted work, environmental concerns and the potential for an over-reliance on AI leading to a long-term decrease in developer competency. Overall, this paper finds that the benefits of generative AI are too important for the systems development industry to ignore and so should be implemented cautiously, with considerations for mitigating potential issues. It is also found that there is a research gap in AI-assisted and manual systems development approaches that should be filled by a comparative experiment into the development time and quality of a system created with the two approaches.

1 Introduction

Sætra (2023) finds generative artificial intelligence (AI) to be the increasingly popular use of machine learning algorithms to generate data relevant to a request from a user. Feuerriegel et al. (2024) explains that this is achieved by providing an AI model with large amounts of data, training it to identify patterns within the data and using those patterns as a basis for generating data such as images, audio, text and programming code. The objectives of this paper are to investigate the role of generative AI within systems development, discuss and analyse its advantages and disadvantages and identify potential areas of future work that would benefit the industry. As for the advantages, research such as Hourani et al. (2019) finds that generative AI can improve system quality by creating tests that ensure the system meets its requirements and that generated tests can outperform those of human testers. Research by Soni et al. (2023), Damyanov et al. (2024) and Russo (2024) shows that by allowing generative AI to complete repetitive tasks, a human developer is free to work on more complex issues, increasing their productivity and resulting in an increased development speed. There are numerous disadvantages that must be considered. Ji et al. (2023) finds that there is a risk of incorrect outputs being generated which appear to be correct and as a result, Ebert and Louridas (2023) find that developers will need to spend time verifying outputs. Schwartz et al. (2020) finds that high-performance AI models may require a large amount of energy and as a result, cause significant damage to the environment. Feuerriegel et al. (2024) and Fui-Hoon Nah et al. (2023) find that there are potential legal concerns of generative AIs use of copyrighted material within its training data as there is a risk that protected works could be illegally replicated. Sætra (2019) finds that there is a danger of people becoming over-reliant on AI and losing the skills needed to complete certain tasks unassisted. This could lead to a long-term decrease in developer competency across the industry, increasing the risk of system failures when developers are required to refactor code generated by AI or work without AI assistance. Despite these issues, this paper argues that generative AI's impact on productivity and quality is too important for it to be overlooked and so it should be cautiously implemented within the systems development industry with considerations for potential long-term issues.

2 Related Work

Research into the use of generative AI within systems development by Rajbhoj et al. (2024) found that it can decrease the time it takes to complete stages of the software development lifecycle (SDLC). They demonstrated this with an experiment where the generative AI platform ChatGPT was used to produce a system following the SDLC. The researchers estimated that developing the system manually would take 75 person days and found that developing the system with ChatGPT took only 22 person days. Whilst this demonstrates

its potential to improve systems development productivity, a limitation of the study is that it relies on a prediction of the manual development time and so the results may not reflect real-world systems development. The following subsections will investigate in more detail how the application of generative AI across SDLC stages such as design, development and testing can support systems development.

2.1 Generative AI in the design stage

In an investigation of the use of generative AI in development, Coutinho et al. (2024) found that generative AI can be used to identify requirements and complete design tasks. They explain that productivity is increased by reducing the time it takes to gather research and create design plans to guide the development of a system that meets its requirements. Coutinho et al. (2024) explains that a limitation of this paper is that it is a pilot study with a small sample size and therefore, the results may not be applicable to the systems development industry. Despite this, the paper has demonstrated the potential of generative AI to support the design stage which is supported by research such as Pothukuchi et al. (2023) and Feuerriegel et al. (2024).

2.2 Generative AI in the development stage

In a survey of developers conducted by Damyanov et al. (2024) it was found that the majority of the participants use generative AI and that it can increase their productivity by completing repetitive tasks whilst they work on more important tasks. They also found that it can support developers by making their code more efficient and providing answers to their questions. A limitation of this study is that all the participants work within the same country and so the results may not be applicable to the rest of the world. Despite this, the study highlights the potential for generative AI to support the development stage by assisting developers in efficiently creating code. This is supported by Soni et al. (2023) where it is found that the quality and efficiency of systems development is increased by using generative AI to complete repetitive tasks and optimise code. Generative AI's positive impact on coding efficiency is also supported by an experiment by Peng et al. (2023) where a group of developers with access to the generative AI tool GitHub Copilot where found to be able to create a web server faster than a group that were restricted to only using internet searches for support.

2.3 Generative AI in the testing stage

In an experiment by Pereira et al. (2024), generative AI was used to quickly and efficiently create scripts for testing an application programming interface (API). They found that the tests are valid the majority of the time and that the chance of generating valid tests increases with successive attempts. A limitation of this study is that it focusses on testing for an API and so may not be relevant to other types of systems. Despite this, the study demonstrates generative AI's potential to support development by quickly creating test scripts to ensure that the system meets its requirements, which is supported by research such as Hourani et al. (2019) and Rajbhoj et al. (2024). According to Bajaj and Samal (2023), Soni et al. (2023) and Ebert et al. (2022), generative AI can also improve system testing by generating a large number of tests which cover cases that may be missed by a human developer, resulting in secure, high-quality systems which meet their requirements.

3 Discussion

Whilst generative AI has the potential to increase the efficiency and quality of a systems development project, there are numerous drawbacks to its use. These include incorrect outputs, legal issues, environmental concerns and a possible negative long-term effect on the systems development industry. The following subsections will discuss the advantages and disadvantages of generative AI in more detail and provide an analysis of the findings and potential future work.

3.1 Findings

3.1.1 Advantages of Generative AI in Systems Development

The main advantage of generative AI in systems development is its potential to increase developer productivity. Soni et al. (2023), Damyanov et al. (2024) and Russo (2024) find generative AIs ability to quickly complete repetitive tasks across the SDLC whilst the developers spend their time on more complicated tasks decreases the time to deliver the system. An increase in productivity is also supported by results of experiments by Rajbhoj et al. (2024) and Peng et al. (2023). According to Damyanov et al. (2024), Russo (2024) and Ebert and Louridas (2023), generative AI can provide accurate answers to developer's questions faster than normal search engines. This further increases developer productivity as they can quickly access the information they require. Bajaj and Samal (2023), Soni et al. (2023) and Ebert et al. (2022) find that generative AI can increase system quality by identifying test cases that human developers may overlook and generating tests that ensure the system meets its requirements. Hourani et al. (2019) finds that generative AI further increases system quality by creating tests that outperform those of human developers. Overall, there is potential for generative AI to enhance the systems development industry by assisting developers in producing high-quality systems faster than traditional, manual approaches.

3.1.2 Disadvantages of Generative AI in Systems Development

A disadvantage is the potential for generative AI to produce hallucinations. Ji et al. (2023) explains that hallucinations are incorrect outputs that appear to be correct and are caused by errors within the model's training data. Because of this, Ebert and Louridas (2023) finds that human developers will need to spend time reviewing and correcting generated work to avoid errors and system failures. This is supported by Coutinho et al. (2024) where it was found that developers often need to adjust generated work to fix inaccuracies. There are also legal issues with creating systems with generative AI. Feuerriegel et al. (2024) and Fui-Hoon Nah et al. (2023) note that as a model's training data can contain copyrighted work, data generated by a model may infringe on copyrights, raising concerns over the legality of using generative AI to professionally develop systems. Sætra (2019) explains there is a danger of an over-reliance on AI, which may result in people no longer developing the skills needed to complete certain tasks without assistance. This is supported by Russo (2024) where it was found that developers with limited technical knowledge that rely on generative AI for coding may create errors that need to be resolved by more experienced developers. Because of this, there is a risk of the systems development industry becoming too heavily reliant on AI, resulting in a lack of developers with the ability to maintain systems or fix issues that were generated by AI. Environmental concerns must also be considered, Schwartz et al. (2020) finds that generative AI models are often developed with a priority of performance over efficiency and as a result, require a large amount of energy, causing significant damage to the environment. If the industry increasingly utilises AI for systems development, there is a danger of a long-term, negative impact on the environment.

3.2 Analysis

3.2.1 The potential of generative AI for enhancing systems development

Despite the concerns of using generative AI within systems development, its benefits are too important for it to be excluded from the industry. Its ability to assist developers and enhance their productivity and work quality will appeal to organisations and allow them to produce bigger and better systems faster than purely manual development can. There are mitigations for the disadvantages discussed. Ooi et al. (2025) and Schwartz et al. (2020) finds that the environmental issues of generative AI can be mitigated with models that focus on efficiency. As the efficiency of AI models continues to improve, there is potential for the systems development industry to progress with generative AI whilst causing reduced damage to the environment. Ooi et al. (2025) also finds that generative AI can minimise the long-term environmental impact of a system by gathering research to help developers create more environmentally friendly system designs. As for the concerns of copyright infringement, Quang (2021) argues that the use of copyrighted works for generative AI model training does not infringe on copyrights. Sag (2023) finds that techniques can be used to decrease the risk

of a model reproducing parts of its training data, for example, removing duplicated samples within the training data. Because of this, organisations using quality models should be able to develop systems without fearing copyright infringement. The concerns of hallucinations and an over-reliance on AI remain. It will be the responsibility of developers to avoid trusting generative AI completely and to ensure they maintain the skills to review and refactor generated works. Ultimately, whilst there are risks that need to be managed, generative AI can assist developers in producing high quality systems at a much faster rate than manual development and should therefore be cautiously integrated within the systems development industry.

3.2.2 Future Work

There is limited research comparing manual systems development approaches with AI-assisted approaches. The experiment conducted by Peng et al. (2023) focusses solely on the speed of coding of a web server, overlooking other systems development activities that can be assisted by AI and the quality of the systems developed with the manual and AI-assisted approaches. The study by Rajbhoj et al. (2024) considers the full SDLC but only compared the AI-assisted development time to an estimation of the manual development time. As they did not produce the system manually, the estimated results may not reflect the real development and they cannot compare the quality of the systems produced by the different approaches. I believe that the industry would benefit from further work comparing the efficiency and quality of a system developed with the assistance of generative AI with a system developed manually. It would allow us to clearly observe any differences in speed and quality across the SDLC.

4 Conclusion

Generative AI is becoming increasingly popular and its ability to support various SDLC tasks will likely see it play an important role in the future of systems development. Research has shown that it can enhance the productivity of developers, supporting them in quickly delivering critical work within the SDLC. Despite this there are concerns that will need to be addressed, generative AI models can produce incorrect outputs, produce outputs that infringe on copyrighted work, damage the environment and an over-reliance on AI may decrease the competency of developers in the long-term. The systems development industry should integrate generative AI for its potential to develop high quality systems, faster than manual approaches but it should be done cautiously. Developers should utilise reliable and efficient models to mitigate the legal and environmental concerns and must understand the importance of reviewing generated work and maintaining their skills to avoid system failures and long-term damage to the industry. There is also a research gap that needs to be addressed. The industry would greatly benefit from a comparative investigation of the productivity and quality of the SDLC stages with manual and AI-assisted approaches.

References

Bajaj, Y. and Samal, M. K. (2023). Accelerating software quality: Unleashing the power of generative ai for automated test-case generation and bug identification. *International Journal for Research in Applied Science and Engineering Technology*, 11(7):345–350.

Coutinho, M., Marques, L., Santos, A., Dahia, M., França, C., and de Souza Santos, R. (2024). The role of generative ai in software development productivity: A pilot case study. In *Proceedings of the 1st ACM International Conference on AI-Powered Software*, pages 131–138.

Damyanov, I., Tsankov, N., and Nedyalkov, I. (2024). Applications of generative artificial intelligence in the software industry. *TEM Journal*, 13(4).

Ebert, C., Bajaj, D., and Weyrich, M. (2022). Testing software systems. *IEEE Software*, 39(4):8–17.

Ebert, C. and Louridas, P. (2023). Generative ai for software practitioners. *IEEE Software*, 40(4):30–38.

Feuerriegel, S., Hartmann, J., Janiesch, C., and Zschech, P. (2024). Generative ai. *Business & Information Systems Engineering*, 66(1):111–126.

Fui-Hoon Nah, F., Zheng, R., Cai, J., Siau, K., and Chen, L. (2023). Generative ai and chatgpt: Applications, challenges, and ai-human collaboration. *Journal of information technology case and application research*, 25(3):277–304.

Hourani, H., Hammad, A., and Lafi, M. (2019). The impact of artificial intelligence on software testing. In *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, pages 565–570. IEEE.

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., and Fung, P. (2023). Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38.

Ooi, K.-B., Tan, G. W.-H., Al-Emran, M., Al-Sharafi, M. A., Capatina, A., Chakraborty, A., Dwivedi, Y. K., Huang, T.-L., Kar, A. K., Lee, V.-H., et al. (2025). The potential of generative artificial intelligence across disciplines: Perspectives and future directions. *Journal of Computer Information Systems*, 65(1):76–107.

Peng, S., Kalliamvakou, E., Cihon, P., and Demirer, M. (2023). The impact of ai on developer productivity: Evidence from github copilot. *arXiv preprint arXiv:2302.06590*.

Pereira, A., Lima, B., and Faria, J. P. (2024). Apitestgenie: Automated api test generation through generative ai. *arXiv preprint arXiv:2409.03838*.

Pothukuchi, A. S., Kota, L. V., and Mallikarjunaradhya, V. (2023). Impact of generative ai on the software development lifecycle (sdlc). *International Journal of Creative Research Thoughts*, 11(8).

Quang, J. (2021). Does training ai violate copyright law? *Berkeley Tech. LJ*, 36:1407.

Rajbhoj, A., Somase, A., Kulkarni, P., and Kulkarni, V. (2024). Accelerating software development using generative ai: Chatgpt case study. In *Proceedings of the 17th innovations in software engineering conference*, pages 1–11.

Russo, D. (2024). Navigating the complexity of generative ai adoption in software engineering. *ACM Transactions on Software Engineering and Methodology*, 33(5):1–50.

Sætra, H. S. (2019). The ghost in the machine: being human in the age of ai and machine learning. *Human Arenas*, 2(1):60–78.

Sætra, H. S. (2023). Generative ai: Here to stay, but for good? *Technology in Society*, 75:102372.

Sag, M. (2023). Copyright safety for generative ai. *Hous. L. Rev.*, 61:295.

Schwartz, R., Dodge, J., Smith, N. A., and Etzioni, O. (2020). Green ai. *Communications of the ACM*, 63(12):54–63.

Soni, A., Kumar, A., Arora, R., and Garine, R. (2023). Integrating ai into the software development life cycle: Best practices, tools, and impact analysis. *Tools, and Impact Analysis (June 10, 2023)*.