



SOFTWARE ENGINEERING AND MACHINE LEARNING

Sultanmuradov Biloliddin

*Student at the Academic Lyceum of the Tashkent University of Information
Technologies*

sultonovbilol67@gmail.com

Introduction: Software engineering has always been one of the most important fields in technology because it provides the structure and logic behind almost every digital system we use today. Over time, the discipline has evolved from manual coding and strict procedures to more flexible and automated development environments. Recently, the rise of artificial intelligence (AI) and machine learning (ML) has brought one of the biggest shifts the industry has ever seen. AI techniques now support many stages of the software life cycle, from planning and coding to testing and system maintenance. Researchers note that this shift is not temporary but part of a long-term transformation affecting how developers work and how software is produced (Amershi et al., 2019). As machine learning becomes more deeply integrated into everyday tools, future software engineers must learn how to use these systems effectively. ML does not replace human programmers, but it changes the expectations placed on them. According to studies, the combination of software engineering skills and AI literacy will be essential for professionals who want to stay competitive in an industry that now relies on automation and intelligent optimisation (Zhang et al., 2020). The following sections discuss how software engineering developed, how AI influences it today, and what challenges may appear in the future as technology continues to expand.

Software Engineering Initial Development

In its early years, software engineering relied completely on human effort. Programmers wrote every line of code manually, often in low-level languages, and every component required careful planning and long hours. Teams were large, and even simple applications took significant time to complete. There was no advanced automation, no AI assistance, and no high-level frameworks to simplify the work. As a result, quality control and error detection were more difficult, and performance improvements depended mainly on human skill.

These early practices built the foundations of modern development. Concepts such as modular design, structured programming, and later object-oriented methods helped developers manage the increasing complexity of software. Although technology has changed dramatically since then, many of



these core ideas still shape how engineers think about building systems.

Software Engineering in the AI Era

At the moment, machine learning and AI are playing a pivotal role in the IT industry. It is a subjective behaviour that boosts many aspects of information technology systems while also bringing challenges for many specialists. Before, when there was no AI or peak-level automation, everything was done by hand by large teams of programmers. What about now? It is crucial to have foundational AI skills to adapt to current flows. Especially when many businesses that rely on IT are now adopting automation across their operations, they always need well-experienced specialists, rather than just old software engineers.

Additionally, software engineering remains one of the main jobs, as it does not matter whether AI will take over positions; every new project from scratch still needs human capital, only adaptation to AI and its features. The main components needed for such adaptation are the inclusion of machine learning foundations in the code processing. If the backstage is already set up, everything is ready for the machine to be included, and it will not blow up the workflow. Later, it will need updates, but eventually it will help everyone by boosting operations. It will not be a big deal either if the programmer automates some basics across every part of a new engagement, whether with a system update or a machine expansion.

Upcoming Changes and Challenges

Though specialists might assume that the inclusion of AI automation or its features in projects might make them ready for newer adaptations, this method might still face capability limitations due to other problems. For example, in the near future, scientists claim that qubits will replace bits, so software engineering must adapt to them as well. Then, what about their security? Will software be able to operate in such an environment without additional load on security algorithms?

Every new adoption of a cryptographic algorithm involves a trade-off between operation performance and speed. It is difficult to predict the future of software development, but every transition should be considered before it happens, as it can completely change the environment. However, machine learning cases and their relationship with programming experience might help create a way out of all this news. As some researchers state, human oversight and decision-making will still be essential even when advanced automation is used (Raisamo et al., 2023).

Conclusion

Software engineering continues to evolve as automation and machine learning become standard parts of technological systems. Although AI can



simplify work, optimise performance, and support large-scale operations, human creativity and problem-solving remain at the centre of development. Engineers will need to understand both traditional coding methods and modern AI-supported tools to stay effective in the future. Challenges such as quantum computing, new security demands, and the rapid pace of technological change will require careful adaptation. Still, with proper learning and integration, AI and software engineering can develop together, strengthening both fields.

REFERENCES

1. Amershi, S., et al. (2019). Software Engineering for Machine Learning: A Case Study. ICSE-SEIP. <https://doi.org/10.1109/ICSE-SEIP.2019.00042>
2. Zhang, D., Wang, H., & Chen, X. (2020). Bridging Software Engineering and Machine Learning: A Systematic Review of Challenges. *Journal of Systems and Software*, 169, 110697. <https://doi.org/10.1016/j.jss.2020.110697>
3. Raisamo, R., Colley, A., & Häkkinen, J. (2023). Human Roles in AI-Assisted Systems. *International Journal of Human-Computer Studies*, 175, 103046. <https://doi.org/10.1016/j.ijhcs.2023.103046>