Motivational letter

I'm a second year computer science student nearing the completion of my bachelors degree. As a field, computer science is dominated by English, and as such it has played a large role in my studies. If i'd have to guess i'd say that almost half of my courses have been fully in English, and for the rest of the courses almost all of the course material has been in English, including Books or articles or other material we might have been provided. As most of my studies and materials have been in English, and we are encouraged to use English naming conventions in our code, it's quite natural that we get quite accustomed to the language, and it almost becomes a second nature. As my studies have shaped and expanded my academic and professional language skills, I'm also an avid reader, and tend to read most of my books in English, further enhancing my knowledge.

Concerning the academic text I followed the same process that I've gotten used to on my other courses, I spent some time researching and finding good and informative sources, and then got straight to work forming the essay. After I had completed my first draft, I rewrote it a few times, iterating through the text to clarify the structure and language.

The most difficult part of the process was finding appropriate sources to cite. As the internet is filled with articles and blog posts about the subjects I was writing about, it was hard to find longer, more comprehensive texts, while I was bombarded with short, simplified blog posts. Eventually I succeeded by just looking and looking until I found articles that were satisfactory to my requirements.

Concerning my weaknesses in academic writing, I tend to struggle with structure. I have ideas and I can write a lot of text, but forming it into a cohesive essay, with a logical structure is something I tend to struggle with. When it comes to strengths, I feel like I have the ability to look outside the box, and find new perspectives to the subjects I'm writing about.

How Generative AI is changing the software industry, and the ethics we must confront

Axel Lindroos

015464712

Bachelor's programme in Computer Science

+358 45 3222 960

Axel.lindroos1@gmail.com

In the past few years the rise of generative artificial intelligence has introduced a massive change in the way software is designed, written, and maintained. Tools like OpenAI's Codex, GitHub Copilot, and ChatGPT have redefined what it means to be a software developer. No longer restricted to manual coding and designing systems from scratch, developers are increasingly turning to AI as a second brain in creating software. While this transformation has brought remarkable increases in productivity and innovation, it also raises ethical questions that challenge existing norms in the software industry. Issues such as intellectual property rights, algorithmic bias, workforce displacement, cyber security, and accountability are now central to conversations about the future of programming. As generative AI becomes more deeply embedded in development workflows, understanding its impact, both good and bad, is essential.

Software development has traditionally been a human-centered endeavor, driven by logic, creativity, and problem-solving. However, with the advances of large language models trained on millions of lines of code, AI is no longer just a passive tool, but is becoming an active participant. Generative AI systems can write entire functions, generate UI components, debug programs and even help design application architecture. Developers no longer need to start from scratch, instead they can describe what they want in natural language, and AI can offer functional code and solutions almost instantly.

This shift has the potential to significantly accelerate the software development life cycle. According to Gnanasambandam et al. (2025), many organizations are already experiencing this speed-up, not just in coding, but across the entire product management and development pipeline. Much of this acceleration occurs from shortening the journey from strategy and vision to late-stage deployment and scaling. Generative AI can automate time-consuming tasks such as project management, market analysis and performance testing, enabling engineers, designers and product managers to focus on higher-value, more creative work that can't be offloaded to AI. These time savings can allow for more frequent product iterations, faster market testing, and quicker response to shifting consumer demands.

One of the most immediate benefits of generative AI is making it easier for beginners to start writing and shipping code. Individuals without a formal programming background can now design and build their own software almost entirely with the help of AI. This has opened new pathways for creativity and entrepreneurship. In the same boat AI also presents significant

opportunities for small business owners, who may lack the time or resources to hire developers or business analysts. AI can for example provide real-time insights into sales trends, cash flow, and other key financial data, helping small business owners to better understand their business without needing to become financial experts or spend excessive time on manual bookkeeping (Pazzanese, 2023).

Yet these advancements don't come without significant ethical concerns. One of the most debated issues involves the question of intellectual property. Generative AI models are trained on vast amounts of existing code, much of which originates from open-source repositories. While open-source software is generally free to use, it is often governed by specific licenses that dictate how the code can be reused, modified, or attributed. When an AI model reproduces code it encountered during training, does that constitute original output or derivative work? And if it is derivative, who owns it, the developer who prompted the AI, the original author of the code, or the creators of the AI model itself? These questions remain unresolved, and they pose a serious challenge to the norms of software licensing and attribution.

Already AI code generators are known to occasionally recommend exact copies of code used to train the underlying model. For instance, GitHub has acknowledged that Copilot may at times recite publicly available open-source code on which it was trained. Although the probability is low (about 1%), Copilot may generate code that exactly matches its training data. This becomes particularly problematic when the reproduced code falls under a copyleft license, such as GPL V3. In such cases, inclusion of this code may "taint" the entire body of the developers code, requiring the entire work to be distributed under the same open-source license. This could override any proprietary licensing, and impose ongoing obligations, including the requirement to disclose source code (Jandhyala et al., 2023). Thus the legal ramifications of Ai generated code can extend far beyond the act of generation itself, potentially binding developers to licences they neither intended nor fully understood

In addition to ownership, the issues of trust in AI generated code cannot be overlooked. Despite their impressive capabilities, generative modern are prone to errors. They can produce insecure code, suggest deprecated practices, or perpetuate bugs present in their training data. Developers who rely too heavily on AI without thoroughly reviewing the output may accidentally introduce vulnerabilities into their applications. Moreover AI models can

also reflect the biases present in their training data. If the training set includes code or data that lacks inclusivity or adheres to biased assumptions, those biases can be amplified in AI generated suggestions. The result is a subtle, but problematic reinforcement of systemic issues that still pervade a lot of the data sets used in training AI.

Privacy and data security further complicates the picture. Generative AI tools like Copilot typically operate via cloud based services, which means code input by users is sent to external servers for processing. For companies with sensitive or proprietary codebases, this represents a serious risk. Some organizations, including Apple, have warned employees not to use tools like Copilot due to fears that internal source code, and by extension their intellectual property, could be exposed (Heaven, 2023). As Justim Gottscilch, CEO of Merly put it. “If I'm Google or Intel and my IP is my source code, I'm never going to use it. Why don’t I just send you all my trade secrets too”. These concerns have prompted AI companies to begin developing in-house versions of their generative tools, allowing companies to retain full control of their code, and ensure it never leaves secure environments. But until these private deployments are widely available, some companies remain reluctant to adopt these tools on a very large scale.

Beyond company hesitance, there’s also concern about how these tools are shaping the next generation of software developers. While AI generated code can be a tremendous timesaver, it also risks encouraging poor software engineering habits, particularly among inexperienced programmers. Because AI tools can produce functional code without explanation, some developers may begin to rely on AI suggestions without truly understanding what the code does or why it works. This leads to a superficial kind of coding that resembles copying and pasting from the internet, but on an even larger scale. If junior developers accept AI outputs blindly, the resulting software may be fragile, unoptimized, and even insecure

The potential for harm is especially significant when AI generated code makes its way into more critical systems. Consider a developer who unknowingly integrated AI suggested functions that mishandle encryption or user authentication. The application may appear to work, but under the surface it could be riddled with vulnerabilities. These risks highlight the importance of human oversight, not only to review code quality, but to foster a culture where understanding and intent remain central to programming. If generative AI becomes a

substitute for foundational knowledge, the long- or even short-term consequences could be dire.

Another major consideration is the impact of AI on the job market. As AI becomes more capable of handling routine coding tasks, concerns have grown about the displacement of junior developers and software engineers, whose roles usually center around this type of repetitive work. While AI is unlikely to replace seasoned professionals who design more complex systems, the demand for entry level positions may decline, limiting opportunities for aspiring developers to gain experience. Eventually this could lead to a shortage of senior developers, as companies don't hire enough juniors to grow into more seasoned professionals. On the other hand, many think that AI will augment, rather than replace human developers, and that by offloading repetitive tasks, AI can allow developers to focus on harder, more complex problems, and thus boost overall productivity.

In conclusion, the rise of generative AI marks a profound shift in how software is built, who builds it, and the ethical frameworks that guide the process. While the benefits in productivity, accessibility, and innovation are undeniable, they come with complex challenges that the industry must confront. As AI becomes a more integral part of the developer's toolkit, balancing its capabilities with human judgment, responsibility, and a commitment to ethical standards will be crucial. The future of programming may be increasingly AI-assisted, but ensuring it remains thoughtful, secure, and inclusive is a responsibility we all share.

Works Cited

[1] Heaven, W.D., 2023. *AI assistants like Copilot are changing how we code*. MIT Technology Review, 6 December. Available at: https://www.technologyreview.com/2023/12/06/1084457/ai-assistants-copilot-changing-code-software-development-github-openai/ [Accessed 7 Apr. 2025].

[2] Gnanasambandam, C., Harrysson, M., Singh, R. & Chawla, A., 2025. *How an AI-enabled software product development life cycle will fuel innovation*. McKinsey & Company, 10 February. Available at: https://www.mckinsey.com/industries/technology-media-and-telecommunications/our-insights/how-an-ai-enabled-software-product-development-life-cycle-will-fuel-innovation [Accessed 7 Apr. 2025].

[3] Jandhyala, S., Kim, J., Bhattacharyya, A., 2023. *Copyrights: A professional perspective on IP issues with AI-generated code*, 18 September. Bloomberg Law. Available at: https://www.bloomberglaw.com/external/document/X4H9CFB4000000/copyrights-professional-perspective-ip-issues-with-ai-code-gener [Accessed 7 Apr. 2025].

[4] Pazzanese, C., 2020. *Ethical concerns mount as AI takes bigger decision-making role*. Harvard Gazette, 26 October. Available at: https://news.harvard.edu/gazette/story/2020/10/ethical-concerns-mount-as-ai-takes-bigger-decision-making-role/ [Accessed 7 Apr. 2025].