Economic Equilibrium: The Case for State-Level Protection of Software Engineering Jobs and University Enrolments

Basit Shahzad
Department of Software Engineering, College of Computer and Cyber Sciences
University of Prince Mugrin, Kingdom of Saudi Arabia.
Email: b.shahzad@upm.edu.sa

Abstract: The rapid integration of Generative AI (GenAI) into the Software Development brings staggering productivity gains and cost efficiencies but creating a profound dichotomy between organizational success and professional risk. While AI automates core activities like requirements engineering, coding, and testing, it simultaneously threatens the job market for new graduates and the long-term viability of computing programs. To mitigate this, the paper proposes a proactive regulatory framework, including human labor quotas and "human-in-the-loop" mandates to balance industrial advancement with the preservation of the software engineering profession, ensuring that both human talent and artificial intelligence can coexist sustainably.

1. Introduction:

Software engineering was once a field characterized by "softening without discipline," but it has matured significantly since its formal origins. Emerged in 1970s and 1980s, discipline has continuously evolved through both methodological and technological advancements. Over the past two decades, we have witnessed comprehensive improvements in tools and processes, particularly with the rise of Agile methods. These iterative approaches have become the standard for modern development, enabling systems to grow through incremental progress.

Parallel to the growth of software engineering, the field of Artificial Intelligence (AI) has undergone its own rapid evolution. As algorithmic complexity increases, the capabilities of Generative AI (GenAI) have expanded, leading to a prominent intersection between the two fields: Intelligent Software Engineering. In this new era, GenAI serves as a catalyst, allowing for the rapid production of software in diversified domains while utilizing significantly fewer resources (1).

2. Transformation of SDLC

When examining the software process in detail, the productivity gains offered by GenAI are staggering in the following activities:

- **Requirements Engineering:** Traditionally, gathering requirements for a year-long project could take a few months. With AI, this phase can be compressed into a few weeks (if not days), allowing humans to shift their focus to confirming and validating these findings with stakeholders to ensure quality, and thus completing the process at a rapid pace by saving time (and thus cost).
- **Modeling and Design:** AI's utility has been tested and increasingly trusted in the realms of software modeling, architecture and design.

- **Programming and Coding:** Large Language Models (LLMs) now generate effective code across diverse programming languages. This enables organizations to develop components at a rapid pace with fewer errors and decreased time and cost. This makes the use of GenAI tools quite a choice for the organizations as it helps them in decreasing the cost of development, time, and increases the possibility of successfully completing the projects. Thus, the use of GenAI, is in the interest of the organizations to achieve their business goals.
- **Testing:** Software testing is no longer an isolated human task; AI can now automate test case generation and component testing without significant manual intervention.

The integration (or assistance) of the AI tools in performing software engineering activities has reduced the effort required to develop the software. This, as a result, decreases the time and resources, and increases the pace of development for a project.

3. Organizational Success vs. Professional Risk

The ability to have software largely developed by AI, with humans only managing integration and release, presents a dichotomy (2). For organizations and clients, this is a success story characterized by lower costs and faster time-to-market (3). However, for the next generation of graduates, the outlook is less promising. While students study to become architects and programmers, the very jobs they seek are being significantly absorbed by AI automation. This creates a critical challenge: how can future software engineers remain vital in a shrinking job market? This is regardless of the fact that the magnitude of software development worldwide is increasing yet the work that can be done by humans is decreasing.

4. Proposed Regulatory Framework

The threat is not the total elimination of software engineering jobs, but a drastic reduction in opportunities relative to the increasing number of global graduates. As organizations prioritize the efficiency of GenAI, new graduates may find themselves pushed toward freelancing or facing long-term unemployment. This may eventually make them dependent on state resources, in case they don't get an opportunity to positively contribute to society (through their role as software engineers).

To prevent this situation, governments and society must act as regulators. My reflection suggests a futuristic policy where governments enforce human-in-the-loop kind of arrangements. For example:

- **Human Labor Quotas:** Regulators could mandate that 50% of software development activities must be performed by humans for the product to be eligible for purchase by government agencies or NGOs, or for that purpose for any acquisition of the software to be done by any entity in the country.
- **Economic Balance:** Such measures would ensure graduates with the right skills remain employed while still allowing industries to benefit from AI advancements.
- **Employment Cycle:** The adoption of this regulatory measure is becoming an imminent necessity, requiring a gradual and incremental implementation strategy. Without such a policy framework, computing programs face a significant existential risk; declining employment prospects may deter prospective students, leading to a sharp drop in

enrolment. This downturn would directly impact university revenues and institutional development, potentially forcing faculty layoffs that further exacerbate the unemployment cycle within the sector.

5. Conclusion

By adopting a proactive and slightly more protective stance, society can balance rapid industrial development with the preservation of the software engineering profession. This ensures that the discipline conventionally known as software engineering remains intact while fostering an environment where both human talent and artificial intelligence can thrive.

References

1. Sauvola, J., Tarkoma, S., Klemettinen, M., Riekki, J., & Doermann, D. (2024). Future of software development with generative AI. Automated Software Engineering, 31(1),
2. Hultin, J., Trudell, C., Vashistha, A., & Glover, T. (2017). Implications of technology on the future workforce (No. 17S2249).
3. Amershi, S., Weld, D., Vorvoreanu, M., Fourney, A., Nushi, B., Collisson, P., ... & Horvitz, E. (2019, May). Guidelines for human-AI interaction. In Proceedings of the 2019 chi conference on human factors in computing systems (pp. 1-13).

Disclosure: The author used Gemini 3.0 to refine the clarity, grammar, and flow of this paper. All original ideas, reflections, and final edits remain the sole responsibility of the author.

The article has been improved by Gemini 3.0 using this prompt. “By taking the following file improve the flow and sentences to make it a good reflection paper. The outcome should only be different in sentence formation, continuity, otherwise the concept alignment and length of paper should remain same.”

Note: The original draft is intact with the author.