

Xin PENG, 2023. Software development in the age of intelligence: embracing large language models with the right approach. *Frontiers of Information Technology & Electronic Engineering*, 24(11):1513-1519.

<https://doi.org/10.1631/FITEE.2300537>

Software development in the age of intelligence: embracing large language models with the right approach

Key words: Large language model (LLM); ChatGPT; Software engineering; Software development

Corresponding author: Xin PENG

E-mail: pengxin@fudan.edu.cn

 ORCID: <https://orcid.org/0000-0003-3376-2581>

Motivation

1. In the field of software engineering, both the academic and industrial communities have been astounded by the all-encompassing capabilities of large language models (LLMs) in software development.
2. The powerful capabilities of LLMs in software development are undoubtedly undeniable, and their disruptive impact on the field can be foreseen. However, it is unclear what kind of changes can be expected in the foreseeable future regarding software development approaches.
3. To see a right way through the fog, I write this article to disseminate information and foster an open discussion within the academic community.

Calm reflection

Possible limitations of the current application of LLMs in the field of software development, considering their training approach and the essence of software development:

1. Scale and complexity of software development may limit the capabilities of LLMs from both human and machine perspectives.
2. LLMs may lack abstract thinking capabilities and may exhibit limitations in precision.
3. In software requirements and design, there exists a significant amount of tacit knowledge that is difficult to capture.
4. There are doubts about the long-term maintenance and support capabilities of LLMs for complex software systems.

Embracing LLMs with the right approach

To achieve comprehensive and systematic software intelligent development, there is still a lot of foundational work to do, and there are several key issues that need to be explored. Here are some related considerations and suggestions corresponding to the four limitations:

1. Solidly building a digitized and knowledge-driven foundation for software development;
2. Placing greater emphasis on fundamental software engineering capabilities such as requirements, design, and validation;
3. Exploring an intelligent interactive engine that effectively integrates LLMs, developers, and various tool capabilities.

Conclusions

1. Embracing LLMs is definitely a correct and even necessary direction for software enterprises to improve quality and efficiency. However, achieving systematic and comprehensive intelligent software development still requires careful consideration and there is much fundamental work to do.
2. For enterprises, solidifying the digitization and knowledge accumulation of software development, as well as the fundamental capabilities of software engineering such as requirement analysis, design, and validation, remains crucial and is also a basic condition for achieving higher levels of intelligent development.
3. For academic research, there is still much work to do in the direction of systematic and comprehensive intelligent software development. This also requires us to have a deeper understanding of the complexity of software systems and software requirements and design, based on the understanding the capabilities of LLMs.



Xin PENG is professor and Deputy Dean of the School of Computer Science at Fudan University, China. He received his PhD in computer science from Fudan University in 2006. He is Deputy Director of CCF (China Computer Federation) Technical Committee on Software Engineering. He is Co-Editor-in-Chief of *Journal of Software: Evolution and Process* and serves on the editorial boards of reputable journals, such as *ACM Transactions on Software Engineering and Methodology*, *Empirical Software Engineering*, and *Chinese Journal of Software*. His research interests include intelligent software development, cloud native and artificial intelligence for IT operations (AIOps), and software development and testing for smart vehicle. His work won the Best Paper Award of ICSM 2011, the ACM SIGSOFT Distinguished Paper Award of ASE 2018/2021 and ICPC 2022, the IEEE TCSE Distinguished Paper Award of ICSME 2018/2019/2020 and SANER 2023, and *IEEE Transactions on Software Engineering* Best Paper award for 2018.