Research Statement

Christoph Treude
School of Computing and Information Systems, Singapore Management University
Tel: (65) 6808 5438; Email: ctreude@smu.edu.sg
20 (Day) 12 (Month) 2024 (Year)

Background

Software has become integral to our daily activities, shaping everything from mobile apps to car navigation systems. In the digital age, software underpins countless systems and services that influence how we communicate, work, and engage in leisure. Its pervasive impact spans sectors such as healthcare, education, finance, and entertainment, establishing software as a cornerstone of modern society.

The digital technologies sector, which relies heavily on software development, is critical for global market competitiveness. High-quality software development is essential not only for the success of individual businesses but also for the vitality of the entire digital technology sector. As companies increasingly depend on software for their operations, the demand for reliable, efficient, and secure software continues to grow, highlighting the need for skilled developers and innovative engineering methods.

Despite its significance, the software industry faces persistent challenges, including poor software quality, which can lead to serious business and safety consequences. Issues such as software errors causing financial losses, security breaches resulting in data theft, and failures in critical systems posing safety risks emphasize the necessity of stringent development and testing processes. These challenges underscore the pivotal role of quality in software engineering.

Frequent software faults often result in outages that disrupt businesses and inconvenience consumers. Beyond immediate impacts, such incidents can erode consumer trust, which is especially damaging in a market where user experience is paramount. The reliability of software systems, therefore, becomes a key determinant of success.

Human and social factors are fundamental to effective software development. The process demands extensive collaboration and communication among diverse stakeholders, including developers, project managers, quality assurance teams, and end-users. Successful software development relies on effective knowledge sharing, balancing user needs, team collaboration, and efficient project management alongside technical expertise.

The industry's shift from traditional waterfall models to agile practices reflects efforts to address these challenges. This transition recognizes software development as a complex, dynamic process that requires flexibility, ongoing learning, and teamwork.

Adding to this evolving landscape, artificial intelligence has emerged as a transformative force, disrupting traditional software development practices and methodologies. AI-powered tools, such as generative models and code completion assistants, are redefining how developers write, test, and debug software. These

technologies promise increased efficiency and productivity but also raise critical questions about trust, security, and ethical considerations. The integration of AI into software systems has sparked new opportunities for innovation while simultaneously creating challenges, such as biases in AI-generated outputs, the need for human oversight, and the implications for the software workforce. Addressing these disruptions requires a careful balance of leveraging AI's potential while mitigating its risks.

My research addresses these multifaceted challenges by focusing on improving software quality, enhancing developer productivity, and ensuring software aligns with the needs of its users and stakeholders. Acknowledging the critical role of software in modern society, the complexities of its development, and the disruptions caused by AI, my work aims to advance the field of software engineering.

Research Areas

My research is organized into three main streams:

1. Empirical Studies of Software Development

This stream focuses on the diverse information needs within software development, considering the varying perspectives, expertise, and challenges arising from new technologies and trends. Empirical methods are central to this work.

For example, I have used Grounded Theory, known for its iterative and exploratory approach, to identify patterns and themes in stakeholder interactions and their information needs. This approach has been applied through interviews with individuals across the software development lifecycle, including developers and project managers. Additionally, I have analyzed software repositories to provide data-driven insights into trends and irregularities in development practices.

A significant focus of this research includes studying the impact of emerging technologies like bots [1], which automate tasks such as code integration and issue tracking, transforming developer workflows. My research examines how bots influence productivity and software quality while addressing the ethical challenges they present. Other notable areas include investigations into sponsorware [2], where developers receive sponsorship for open-source contributions, exploring its effects on software quality and innovation, and protestware [3], where software is altered for political statements, highlighting the intersection of software development with broader societal issues.

Additionally, I have explored how developers use AI in their workflows, such as in the context of generative AI for pull request descriptions [4]. This work investigates the adoption of generative AI tools, their impact on developer productivity and collaboration, and the interventions developers employ to integrate these tools into existing practices. Complementing this, I have also studied ethical considerations in AI models [5] which highlights the importance of transparency and responsibility in the development and deployment of AI systems.

Research in this stream has garnered recognition, including a Best Paper Award [6]. It has been presented at prestigious institutions like the National Institute of Informatics in Japan and University College London, demonstrating its global

relevance. Furthermore, my ICSE paper [7] on awareness tools was recently ranked as the third most relevant Software Engineering paper in the last five years by Microsoft program managers, based on a study of 571 papers titled "How Practitioners Perceive the Relevance of Software Engineering Research."

Through this work, I aim to uncover the intricate interplay between human interactions and technological advancements, contributing to a deeper understanding of software development and informing more efficient, ethical, and innovative practices.

2. Innovation and Development of Software Engineering Techniques

This stream connects empirical insights with practical applications, aiming to influence the future of software engineering. The goal is to translate data and observations from empirical studies into actionable tools and techniques for developers.

This research envisions a future where human-computer interaction is intuitive, reducing the cognitive burden on developers. For instance, leveraging natural language processing to interpret human communication accurately and employing machine learning to address complex issues proactively. By integrating these advanced technologies with the real-world challenges of software development, this work seeks to transform the field.

Collaboration with industry partners is not just a supplementary aspect but a cornerstone of this research. These partnerships ensure relevance and practicality, aligning academic research with the needs of software engineers. Such synergies have led to practical outcomes, as reflected in awards from ASE 2019 [8], ICSE 2021 [9], and ESEM 2019 [10], which recognize the innovative and impactful nature of my work.

This research goes beyond creating tools and techniques; it seeks to redefine the boundaries of software engineering, making development processes more efficient and user-friendly. By driving technological progress, this work shapes a future where software development is both accessible and amplified by cutting-edge advancements.

3. Open Source Ecosystems

Open source software represents a paradigm shift in how digital solutions are created and distributed. Unlike proprietary software, open source fosters a community-driven development model, encouraging broad participation. This approach has catalyzed significant innovation, resulting in robust, widely-used platforms and tools.

Open source also promotes transparency, collaboration, and shared learning, democratizing software development by enabling wider access and participation. Its impact extends beyond technical achievements, influencing how technology shapes various aspects of society.

My research in this area examines the unique dynamics of open source ecosystems, likening them to ecological systems where small actions can have large-scale impacts—a phenomenon captured by the "butterfly effect" metaphor. Open source projects are not merely codebases but vibrant communities of innovation. I explore

how developer interactions within these ecosystems influence productivity, software quality, and innovation.

This includes analyzing the interplay between social media and open source sponsorships [11]. This research highlights how external platforms like Twitter/X amplify engagement and sponsorships, shaping funding dynamics and collaboration in open source communities.

Using both quantitative and qualitative methods, I analyze the motivations, challenges, and experiences of developers through surveys while examining repository data for patterns and trends. This dual approach sheds light on the social and collaborative aspects of open source development.

The academic and industry relevance of this research is reflected in presentations at leading conferences [12] and publications in renowned journals [13]. By decoding the essence of collaboration within open source ecosystems, this work highlights the interplay of personalities, skills, and ideas that influence software evolution. It uniquely emphasizes the vast scale and interdependencies within these ecosystems, mirroring the delicate balance of biological systems.

Selected Publications and Outputs

[1] Wessel, Mairieli, Joseph Vargovich, Marco A. Gerosa, and Christoph Treude. "GitHub Actions: the impact on the pull request process." Empirical Software Engineering 28, no. 6 (2023): 1-35.

[2] Shimada, Naomichi, Tao Xiao, Hideaki Hata, Christoph Treude, and Kenichi Matsumoto. "GitHub sponsors: exploring a new way to contribute to open source." In Proceedings of the 44th International Conference on Software Engineering, pp. 1058-1069. 2022.

[3] Kula, Raula Gaikovina, and Christoph Treude. "In war and peace: the impact of world politics on software ecosystems." In Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, pp. 1600-1604. 2022.

[4] Xiao, Tao, Hideaki Hata, Christoph Treude, and Kenichi Matsumoto. "Generative AI for pull request descriptions: Adoption, impact, and developer interventions." Proceedings of the ACM on Software Engineering 1, no. FSE (2024): 1043-1065.

[5] Gao, Haoyu, Mansooreh Zahedi, Christoph Treude, Sarita Rosenstock, and Marc Cheong. "Documenting ethical considerations in open source ai models." In Proceedings of the 18th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, pp. 177-188. 2024.

[6] Reboucas De Almeida, Rodrigo, Rafael do Nascimento Ribeiro, Christoph Treude, and Uirá Kulesza. "Business-driven technical debt prioritization: An industrial case study." In 2021 IEEE/ACM International Conference on Technical Debt (TechDebt), pp. 74-83. IEEE, 2021.

[7] Treude, Christoph, and Margaret-Anne Storey. "Awareness 2.0: staying aware of projects, developers and tasks using dashboards and feeds." In Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 1, pp. 365-374. 2010.

[8] Liu, Zhongxin, Xin Xia, Christoph Treude, David Lo, and Shanping Li. "Automatic generation of pull request descriptions." In 2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE), pp. 176-188. IEEE, 2019.

[9] Cao, Kaibo, Chunyang Chen, Sebastian Baltes, Christoph Treude, and Xiang Chen. "Automated query reformulation for efficient search based on query logs from stack overflow." In 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE), pp. 1273-1285. IEEE, 2021.

[10] Thiselton, Emillie, and Christoph Treude. "Enhancing python compiler error messages via stack overflow." In 2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), pp. 1-12. IEEE, 2019.

[11] Fan, Youmei, Tao Xiao, Hideaki Hata, Christoph Treude, and Kenichi Matsumoto. ""My GitHub Sponsors profile is live!" Investigating the Impact of Twitter/X Mentions on GitHub Sponsors." In Proceedings of the IEEE/ACM 46th International Conference on Software Engineering, pp. 1-12. 2024.

[12] Gerosa, Marco, Igor Wiese, Bianca Trinkenreich, Georg Link, Gregorio Robles, Christoph Treude, Igor Steinmacher, and Anita Sarma. "The shifting sands of motivation: Revisiting what drives contributors in open source." In 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE), pp. 1046-1058. IEEE, 2021.

[13] Wattanakriengkrai, Supatsara, Dong Wang, Raula Gaikovina Kula, Christoph Treude, Patanamon Thongtanunam, Takashi Ishio, and Kenichi Matsumoto. "Giving back: Contributions congruent to library dependency changes in a software ecosystem." IEEE Transactions on Software Engineering 49, no. 4 (2022): 2566-2579.