Research Statement

Christoph Treude
School of Computing and Information Systems, Singapore Management University
Tel: (65) 6808 5438; Email: ctreude@smu.edu.sg
03 (Day) 12 (Month) 2023 (Year)

Background

Software has become a critical element in our daily activities, significantly influencing everything from mobile apps to car navigation systems. In our digital age, software is fundamental to numerous systems and services that affect our ways of communication, work, and leisure. Its widespread use and influence are evident in various sectors such as healthcare, education, finance, and entertainment, marking it as a key component of today's society.

The digital technologies sector, highly dependent on software development, is essential for global market competitiveness. High-quality software development is vital not only for the success of individual businesses but also for the overall health of the digital technology sector. As companies increasingly rely on software for their operations, the demand for reliable, efficient and secure software has increased. This trend underscores the growing need for skilled software developers and innovative software engineering methods.

However, the software industry faces challenges, including instances of poor software quality that lead to major business and safety issues. These problems range from software errors causing financial losses and security breaches resulting in data theft, to failures in critical systems causing safety risks. Such incidents underscore the critical importance of quality in software development and the need for comprehensive development and testing processes.

Software faults often cause frequent outages, disrupting businesses, and inconveniencing consumers. These outages not only cause immediate problems but can also erode consumer trust over time. In a market where user experience is crucial, the dependability of software systems is a major factor.

Human and social elements play an important role in successful software development. The process involves extensive collaboration and communication among different stakeholders, including developers, project managers, quality assurance teams, and end users. Effective knowledge sharing and collaboration among these groups are essential to create high-quality software. The balance of user needs, team collaboration, and efficient project management is as important as technical skill in software development.

The evolution from traditional waterfall models to agile practices in software development shows the industry's efforts to meet these challenges. This change acknowledges that software development is a complex, dynamic process that requires flexibility, ongoing learning, and teamwork.

My research focuses on addressing these complex challenges, with the aim of improving software quality, boosting developer productivity, and ensuring that software meets the needs of its users and stakeholders. Recognizing the crucial role of software in modern society and the complexities of its development, my research seeks to contribute to the advancement of the field of software engineering.

Research Areas

My research is organized into three main streams:

Empirical Studies of Software Development

My research focuses on the diverse information needs within software development, taking into account the varying needs, perspectives, and expertise involved. This field is dynamic and constantly evolving with new technologies and trends, creating both challenges and opportunities. Empirical methods play a key role in this context.

For instance, I have employed Grounded Theory, known for its iterative and exploratory approach, to understand patterns and themes in stakeholder interactions and their information needs. This method has been applied through interviews, where I gather direct insights from individuals involved in the software development lifecycle, from developers to project managers. I have also analyzed software repositories to provide a data-driven perspective on trends and irregularities in development practices. These approaches are examples of my research methods and areas of focus.

Another interesting part of my research includes studying the impact of new technologies like bots [1] on software development. These bots, which handle tasks ranging from code integration to issue tracking, are changing the way developers work. My research in this area explores their impact on productivity and software quality, as well as the ethical issues they present. Additionally, I have examined sponsorware [2], where developers are sponsored for their open source contributions, investigating its effect on software quality and innovation. Furthermore, my research has studied protestware [3], or software altered for political statements, highlighting the intersection of software development with social and political issues.

This research has been recognized for its contributions, as evidenced by a Best Paper Award [4]. It has been presented at prestigious institutions such as the National Institute of Informatics in Japan and the University College London, reflecting its global relevance. My ICSE paper [5] on awareness tools was recently ranked as the third most relevant Software Engineering paper from the last five years by program managers at Microsoft, out of a total of 571 papers considered in a study titled “How Practitioners Perceive the Relevance of Software Engineering Research”.

Through this research, I aim to uncover the complex interplay of human interaction, technological advancement, and their collective impact on software development, contributing to a deeper understanding of this dynamic field and guiding more informed, efficient, and responsible development practices.

Innovation and Development of Software Engineering Techniques

This research stream focuses on connecting empirical insights with practical applications in software engineering. It is about understanding and actively influencing the future of software engineering. The process involves turning data and observations from empirical studies into practical tools and techniques for those involved in software engineering.

The aim is to create a world where human-computer interaction is so intuitive that it eases the cognitive burden on developers. Imagine software development where natural language processing accurately understands and interprets human communication and machine learning algorithms preemptively address complex issues. This research leads to the integration of these advanced technologies with the real-life challenges of software development.

Collaborating with industry partners is essential for this work, not just an addition. This partnership ensures that research remains relevant and practical and that the solutions developed meet the actual needs of software engineers. This synergy between academia and industry is crucial because it helps create practical tools and techniques that address the real challenges in software engineering.

The significance of this research extends beyond academia and is recognized within the wider software engineering community. For example, the awards I received at ASE 2019 [6], ICSE 2021 [7], and ESEM 2019 [8] are a testament to the practical impact of my work. These accolades serve as examples, highlighting the innovation and relevance of my research in the advancement of software engineering.

This stream goes beyond just creating new tools and techniques; it aims to expand the limits of what is possible in software engineering, using advanced technology to make the software development process more efficient and user-friendly. Through this research, we are not just observing the evolution of software engineering, but are actively driving it forward, shaping a future where technology amplifies human capability and where the process of software creation is as refined as the software itself.

Open Source Ecosystems

Open source software represents a paradigm shift in the way digital solutions are created and distributed. At its core, open source is about collaborative software development, where the source code is freely available for anyone to use, modify, and distribute. This approach contrasts with traditional proprietary software, where the source code is closely guarded and licensed under restrictive terms. Open source projects foster a community-driven development process, encouraging participation from a diverse group of individuals and organizations. This has led to significant innovations and the development of robust, widely-used software platforms and tools. The importance of open source software extends beyond its technical contributions; it also plays a vital role in promoting transparency, collaboration, and shared learning in the tech community. It acts as a catalyst for democratizing software development, enabling broader access and participation in the creation of technology that significantly impacts various aspects of our lives.

My research in this stream targets the unique dynamics of open source software ecosystems, likening them to an ecological system where interactions have widespread effects, similar to how a butterfly's small actions can have large-scale impacts. This metaphor underlines the difference in scale and complexity compared to traditional software development. Open source projects are more than code; they are vibrant communities of innovation and interaction. The aim is to understand how the interactions of developers within these ecosystems influence productivity and software quality.

I investigate individual contributions and their interactions in these projects, using the butterfly effect as a metaphor to illustrate the interconnectedness of actions and outcomes. The focus is on social dynamics, communication patterns, and collaboration in successful open source projects. The research questions how these interactions drive software evolution and the social dynamics that lead to innovation and efficiency.

Using both quantitative and qualitative methods, this research provides a thorough analysis of open source development. Surveys explore developers' motivations, challenges, and experiences. In parallel, software repository analysis reveals patterns and trends, shedding light on unique aspects of open source development.

The findings have a significant academic and industry impact, as evidenced by presentations at leading conferences [9] and publications in renowned journals [10]. This research goes beyond simple observation; it aims to decode the essence of collaboration in open source ecosystems. It brings to light the human aspect of software development, the blend of personalities, skills, and ideas, and how this influences the software that shapes our world. The unique approach of this research lies in its focus on the vast scale and intricate interdependencies within software ecosystems, similar to the delicate balance of a biological ecosystem.

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] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.