Tapajit Dey

Limerick,Ireland
✉ *tapjdey@gmail.com*
https://tapjdey.github.io

RESEARCH STATEMENT

Overview

My research primarily lies in the domain of **Empirical Software Engineering**, covering the topics of Open Source and InnerSource software development, Human Aspects of Software Engineering, Mining Software Repositories, Data Analysis, and Role of Bots/Automation in Software Development. I have used both frequentist and Bayesian statistical techniques in my research, including casual inference using Bayesian Networks, and I am familiar with various quantitative and qualitative methods, including various deep learning models, as well as standard data mining techniques like CRISP-DM.

So far, my research has focused on studying the technical and community aspects of software development with an emphasis on open source software. I have used quantitative and qualitative methods for collecting and analyzing data on various software artifacts and development traces and conducted surveys and interviews with developers as needed to gain a deeper understanding of the interaction between software developers, code, and users. Thus far, my research has resulted in 10 peer-reviewed publications in various conferences, including one in the ICSE-2021 Technical Track, 5 journal articles, 2 workshop papers, and a co-authored book chapter.

In the future, I would like to continue addressing the technical and community-related challenges to software engineering and also broaden my horizons to address other challenges related to the topic of **Sustainable Software Engineering**, e.g., social issues like fairness, equality, diversity, and inclusion, environmental effects of software, and designing responsible software by leveraging modern ML & AI techniques. At the same time, I would promote Open Science, one of my biggest passions, through my research, adhering to the 8 pillars of Open Science and facilitating Open Science adoption for other practitioners in the field.

Past & Current Research

In my **Ph.D. research**, I adopted the socio-technical view to understand how software users and the interdependencies among software components (the software supply chain) affect its success by leveraging the Information System Success Model [1]. A few highlights of my Ph.D. research are as follows: 1. Identified empirical evidence of the effect of usage on the perceived quality of a software using Bayesian Network analysis and proposed a usage-independent quality measure [7, 8]. 2. Provided an understanding of the effect of the software supply chain on software popularity [6]. 3. Revealed the limited visibility in the supply chain of the NPM ecosystem [4] by identifying that the developers interact (contribute issues or patches) primarily with their direct dependencies but the interaction drops drastically for transitive dependencies. 4. Identified the effects of various social and technical factors on Pull Request acceptance in the NPM ecosystem [9]. 5. Developed **BIMAN**, a systematic method for identifying code-commit bots - the first one of its kind in literature [10] and also studied the types of commits made by bots [11]. 6. Established a method of representing the domain expertise of OSS developers, projects, APIs, and languages by creating a vector space called “Skill Space” [3]. In the course of these projects, I have collaborated with other researchers at the University of Tennessee, including Russell Zaretzki, Randy V. Bradley, and Bogdan Bichescu from the Haslam College of Business, and James D. Herbsleb, Bogdan Vasilescu, and Chris Bogart from Carnegie Mellon University.

Other than these primary works, I have collaborated with Peter Rigby from Concordia University to investigate the effect of code review variables on software defects using Bayesian Networks [13], worked on developing the World of Code [14] infrastructure and helped incorporate author identity disambiguation [12] into it, and lately, worked with Alexander Nolte from the University of Tartu and Carnegie Mellon University to identify the origin of code used in hackathon projects and the reuse of code developed during hackathons, which revealed that many hackathons are not “one-off” events as many tend to think. I have also worked briefly in the fields of video game development [5] and causal analysis.

During my **postdoctoral research**, I have focused more on the community side of software development. I have worked on facilitating the adoption of **InnerSource**, the use of OSS development best practices and the establishment of an open source culture within an organization for developing their in-house software, at Huawei as a part of my postdoctoral project. I have conducted a state-of-InnerSource survey with members from InnerSource Commons, surveyed the InnerSource project owners, managers, and prospective developers at Huawei to understand the unique obstacles and cultural challenges faced by them, helped create an InnerSource project fitness assessment tool and an incentive framework [2] to aid in InnerSource adoption, and have also conducted several workshops and arranged invited talks by veterans from various corporations (e.g., Microsoft) and academic institutions.

My current research focuses on supporting the new OSS developers and projects in finding suitable projects and developers respectively, and also in helping the developers progress to more significant roles in their projects, and helping projects retain contributors. I have developed a tool to assist OSS newcomers find suitable projects based on their expertise and am working on identifying how they can best manage the image they portray to the OSS communities.

Future Research

In the future, I plan to both continue my current research trajectory of understanding the socio-technical dynamics of software ecosystems and to broaden my research to address the manifold challenges related to **Sustainable Software Engineering**. Sustainability, defined as *the ability to maintain or support a process over time*, can be operationalized in the software engineering context as identifying and mitigating the potential technical, societal, and other (e.g., environmental) threats to the software engineering process to ensure its continued success. To that end, I want to explore challenges related to fairness, equality, diversity, and inclusion (EDI), environmental effects of software, and designing responsible software by leveraging modern ML & AI techniques. I would also continue to interact with and support the InnerSource community and promote Open Science through my research.

I foresee several funding and industry collaboration opportunities arising from my research, including supporting nascent Open Source or InnerSource initiatives at companies by providing evidence-based decision support and consulting by conducting archival analyses and surveys/interviews, supporting EDI initiatives by OSS communities/industries, etc.

References

[1] William H DeLone and Ephraim R McLean. Information systems success: The quest for the dependent variable. *Information systems research*, 3(1):60–95, 1992.

[2] Tapajit Dey, Willem Jiang, and Brian Fitzgerald. Knights and gold stars: A tale of innersource incentivization. *arXiv preprint arXiv:2207.08475*, 2022.

[3] Tapajit Dey, Andrey Karnauch, and Audris Mockus. Representation of developer expertise in open source software. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, pages 995–1007. IEEE, 2021.

[4] Tapajit Dey, Yuxing Ma, and Audris Mockus. Patterns of effort contribution and demand and user classification based on participation patterns in npm ecosystem. In *Proceedings of the fifteenth international conference on predictive models and data analytics in software engineering*, pages 36–45, 2019.

[5] Tapajit Dey, Jacob Logan Massengill, and Audris Mockus. Analysis of popularity of game mods: A case study. In *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts*, pages 133–139, 2016.

[6] Tapajit Dey and Audris Mockus. Are software dependency supply chain metrics useful in predicting change of popularity of npm packages? In *Proceedings of the 14th International Conference on Predictive Models and Data Analytics in Software Engineering*, pages 66–69, 2018.

[7] Tapajit Dey and Audris Mockus. Modeling relationship between post-release faults and usage in mobile software. In *Proceedings of the 14th International Conference on Predictive Models and Data Analytics in Software Engineering*, pages 56–65, 2018.

[8] Tapajit Dey and Audris Mockus. Deriving a usage-independent software quality metric. *Empirical Software Engineering*, 25(2):1596–1641, 2020.

[9] Tapajit Dey and Audris Mockus. Effect of technical and social factors on pull request quality for the npm ecosystem. In *Proceedings of the 14th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pages 1–11, 2020.

[10] Tapajit Dey, Sara Mousavi, Eduardo Ponce, Tanner Fry, Bogdan Vasilescu, Anna Filippova, and Audris Mockus. Detecting and characterizing bots that commit code. In *Proceedings of the 17th International Conference on Mining Software Repositories*, pages 209–219, 2020.

[11] Tapajit Dey, Bogdan Vasilescu, and Audris Mockus. An exploratory study of bot commits. In *ICSEW'20: Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops*, pages 61–65, 2020.

[12] Tanner Fry, Tapajit Dey, Andrey Karnauch, and Audris Mockus. A dataset and an approach for identity resolution of 38 million author ids extracted from 2b git commits. In *IEEE International Working Conference on Mining Software Repositories*, pages 518–522. ACM, 2020.

[13] Andrey Krutauz, Tapajit Dey, Peter C Rigby, and Audris Mockus. Do code review measures explain the incidence of post-release defects? *Empirical Software Engineering*, 25(5):3323–3356, 2020.

[14] Yuxing Ma, Tapajit Dey, Chris Bogart, Sadika Amreen, Marat Valiev, Adam Tutko, David Kennard, Russell Zaretzki, and Audris Mockus. World of code: enabling a research workflow for mining and analyzing the universe of open source vcs data. *Empirical Software Engineering*, 26(2):1–42, 2021.