Tapajit Dey

RESEARCH STATEMENT

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

Overview

My research focuses on applying innovative quantitative and qualitative methods, ranging from statistical analysis and machine learning techniques to Bayesian Network-based causal inference and deep learning methods to solve existing and emerging software development challenges in Open Source, Inner Source, and proprietary software development. I have published my work in a range of venues, including conferences like ICSE (Technical Track), ESEC/FSE (Journal-First Track), MSR, ESEM, PROMISE, and journals like the Empirical Software Engineering journal and IEEE Software, on topics related to open source and InnerSource software development, empirical software engineering, mining software repositories, software ecosystems/software supply chain, data analysis, human aspects of software engineering, and the role of bots/automation in software development.

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]. The two major types of work I have conducted during my Ph.D. were related to 1. Development of Methodologies/Measures and 2. Large-scale studies of software supply chains, specifically, the NPM ecosystem (JavaScript).

For the **first type** of work, I have had **three** major contributions. *Firstly*, I have worked on developing a systematic method for identifying code-commit bots named **BIMAN** - the first one of its kind in literature [10]. Identifying, and often removing bots from a dataset is important for many empirical studies, especially the ones looking into developer productivity, culture, collaboration, etc. and all of the previous bot identification methods were manual or based on simple heuristics. Our method paved the first stone in systematically addressing this problem and have inspired more research into this topic since. The *second* major contribution I had was on developing a better measure for the perceived quality of a software. This work was important because the traditional measure of quality in industry was to count failures or faults, but since more faults typically indicate more usage (due to better software), using that measure provided a conflicting signal to software management teams. I addressed this issue by identifying empirical evidence of the effect of usage on the perceived quality of a software using Bayesian Network analysis and proposing a usage-independent measure of quality [7, 8] for objectively comparing the qualities of different software products/releases. My *third* and final major contribution in this area was establishing a method for representing OSS developers, projects, APIs, and languages in a 200-dimensional vector space called "Skill Space" [3] by considering the APIs associated with each code blob change made by the developers. This enabled having a representation of the domain-level expertise of developers, which was found to be consistent with their self-reported expertise, and the alignment between developers and projects was found to be significant in determining which projects a developer joins and also whether a pull request created by them is accepted.

For the **second** type of work, I have studied the NPM ecosystem using the World of Code[13] dataset, and discovered the effect of the software supply chain on software popularity [6], revealed the limited visibility in the supply chain [4] by identifying that the developers interact (contribute issues or patches) primarily with their direct dependencies but the interaction drops drastically for transitive dependencies, and have also identified the effects of various social and technical factors on Pull Request acceptance in the ecosystem [9]. In addition to having led to deeper insights into the ecosystem, these works have identified pain points (e.g. limited visibility) that are crucial for ensuring its long-term survivability, as evidenced by the fact that there have been several recent supply chain attacks on the NPM ecosystem (e.g.[14]), which could have been avoided if the visibility issues were addressed early-on.

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 [12], worked on developing the World of Code [13] infrastructure and helped incorporate author identity disambiguation [11] 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 would like to continue on the same path of addressing the existing and emerging technical and community-related challenges to software engineering. As a direct extension of my past and current work, I would like to work on supporting OSS communities for better manage the projects and help the OSS newcomers integrate into the ecosystem.

I would also like to continue interacting with the InnerSource community and support the work they do in promoting and spreading the open source development methodology. InnerSource is a critical area of interest of many companies around the globe, with major technology companies like Microsoft, Paypal, Tencent, and Huawei, and even non-IT companies such as Bloomberg, Bosch, and Nike embracing the InnerSource ideology. However, InnerSource has arguably not received the attention it deserves from the academic communities, owing to various problems, e.g., publicly disclosing potentially sensitive data, or admitting to less than stellar success in implementing InnerSource initiatives. This is unfortunate because several critical problems in InnerSource could benefit from the expertise of academic researchers, for instance, understanding the long-term benefits of InnerSource, the economic benefits to the company brought in by InnerSource adoption, understanding why some InnerSource projects succeed and others fail, etc. This is another major area of research I would like to work on in the future.

At the same time, I would also like broaden my horizons to address challenges related to social issues like fairness, equality, diversity, and inclusion in software developer communities both in the context of open source and industry. I also want to study the environmental and economic effects of software (*sustainable software engineering*), and work on designing responsible software by leveraging modern ML & AI techniques. I believe these are some of the most important emerging issues which will be in the forefront of software research in the coming days.

I would also strive to promote Open Science, one of my biggest passions, through my research and facilitate Open Science adoption for other practitioners in the field.

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] 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.

[12] 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.

[13] 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.

[14] Andrey Polkovnychenko and Shachar Menashe. Npm supply chain attack targets germany-based companies with dangerous backdoor malware. `https://jfrog.com/blog/npm-supply-chain-attack-targets-german-based-companies/`, 2022. [Online] Accessed: 04-Oct-2022.