Poster: How Has Forking Changed in the Last 20 Years? A Study of Hard Forks on GitHub

Shurui Zhou
Carnegie Mellon University, USA

Bogdan Vasilescu
Carnegie Mellon University, USA

Christian Kästner
Carnegie Mellon University, USA

ABSTRACT

The notion of forking has changed with the rise of distributed version control systems and social coding environments, like GitHub. Traditionally forking refers to splitting off an independent development branch (which we call *hard forks*); research on hard forks, conducted mostly in pre-GitHub days showed that hard forks were often seen critical as they may fragment a community. Today, in social coding environments, open-source developers are encouraged to fork a project in order to contribute to the community (which we call *social forks*), which may have also influenced perceptions and practices around hard forks. To revisit hard forks, we identify, study, and classify 15,306 hard forks on GitHub and interview 18 owners of hard forks or forked repositories. We find that, among others, hard forks often evolve out of social forks rather than being planned deliberately and that perception about hard forks have indeed changed dramatically, seeing them often as a positive non-competitive alternative to the original project.

ACM Reference Format:
Shurui Zhou, Bogdan Vasilescu, and Christian Kästner. 2020. Poster: How Has Forking Changed in the Last 20 Years? A Study of Hard Forks on GitHub. In *42nd International Conference on Software Engineering Companion (ICSE '20 Companion), October 5–11, 2020, Seoul, Republic of Korea.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3377812.3390911

1 INTRODUCTION

The notion of *forking* in open-source has evolved: Traditionally, forking was the practice of copying a repository and splitting off new independent development, often under a new name; forking was rare and was typically intended to compete with or supersede the original project [5, 7]. Nowadays, with the rise of social coding and explicit support in (distributed) version control systems, forks are public copies of repositories in which developers can make changes, potentially, with the intention of integrating those changes back into the original repository. Forking of repositories has become very popular [6].

However, most of these modern forks are not forks in the traditional sense. We distinguish between **social forks**, referring to creating a public copy of a repository on a social coding site like GitHub, often with the goal of contributing to the original project, and **hard forks**, referring to the traditional notion of splitting off a new development branch [10, 11]. Traditionally, hard forks were frequently considered as risky to fragment a community and lead to confusion for both developers and users [7], and there was a strong norm against forking. In this paper, we argue that perceptions and practices around forking could have changed significantly since the advances in distributed version control systems and transparency mechanisms on social coding sites [4], may have enabled new opportunities and changed common practices and perceptions. Therefore, we ask the research question: **How have perceptions and practices around hard forks changed?** Furthermore, we automate the process of identifying hard forks among social forks and quantifying how frequent hard forks are across GitHub.

Using a mixed-method *exploratory* research strategy [3], combining repository mining with 18 developer interviews, we investigate: (1) Frequency of hard forks; (2) Common evolution patterns of hard forks; and (3) Perceptions of hard forks.

2 RESEARCH METHODS

Instrument for Visualizing Fork Activities. We created *commit history graphs*, a custom visualization of commit activities in forks and corresponding upstream repositories, as illustrated in Tab. 1.

Identifying Hard Forks. We proceeded iteratively validating and combining various heuristics. Our final classifier proceeds in two steps: first, we use multiple simple heuristics to identify candidate hard forks; second, we use a more detailed and more expensive analysis to decide which of those candidates are actual hard forks. Our classifier identifies a total of 15,306 hard forks across GitHub.

Classifying Evolution Patterns. We identified different evolution patterns among the analyzed forks using card sorting [8]. Evolution patterns describe how a hard fork and the corresponding upstream project coevolve and can help to characterize forking outcomes. After several iterations, we arrived at a stable list of 15 patterns with which we could classify 97.7 % of all hard forks.

Interviews. We selected potential interviewees among the maintainers of the 15,306 identified hard forks and corresponding upstream repositories. Overall, 18 maintainers volunteered to participate in our study (7 % response rate). More information about our interviewees are in the paper [11]).

3 RESULTS

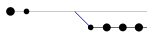
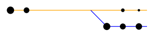
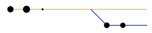
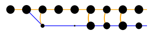
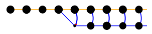
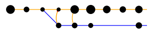
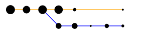
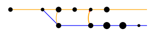
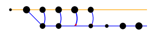
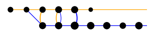
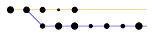
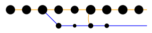
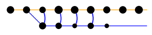
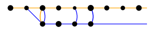
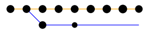
3.1 Frequency of Hard Forks

Hard forks are generally a rare phenomenon. As our analysis of evolution patterns reveals, cases where both the upstream repository and the hard fork remain active for extended periods of time are not common (patterns 1, 2, and 4–7). Most hard forks actually survive the upstream project, if the upstream project was active when the fork was created (patterns 8–11), but many also run out of steam eventually (patterns 3 and 12–15), While most hard forks are created as forks of active projects (patterns 4–15), there are a substantial number of cases where hard fork are created to revive a

ICSE '20 Companion, October 5–11, 2020, Seoul, Republic of Korea

ACM ISBN 978-1-4503-7122-3/20/05.
https://doi.org/10.1145/3377812.3390911

Table 1: Evolution patterns of hard forks

Id	Category	Total	Sub-category	Example	Count
1	Success (F. active > 2 Qt.)	632	U remains inactive		576
2			U active again		56
3	Not success (F active <= 2 Qt)	420			420
4	Both Alive	723	only M		26
5			only S		107
6			M & S		28
7			No Int		562
8	F Lived Longer	7280	only M		174
9			only S		686
10			M & S		107
11			No Int		6313
12	Fork does not out live upstream	6251	only M		388
13			only S		762
14			M & S		199
15			No Int		4902

1-3: Revive Dead Project; 4-15: Fork Active Project
U – Upstream, F – Fork, M – Merge, S – Synchronize, Int – Interaction

dead project (pattern 1–3), in some cases even triggering or coinciding with a revival of the upstream project (pattern 2), but also here not all hard fork sustain activity (pattern 3).

Discussion and implications. Even though the percentage of hard forks is low, the total number of attempted and sustained hard forks is not. Considering the significant cost a hard fork can put on a community through fragmentation, but also the potential power a community has through hard forks, we argue that hard forks are an important phenomenon to study even when they are comparably rare. We release the dataset of all hard forks with corresponding visualizations as dataset [1]. In addition, hard forks are not likely to be avoidable in general, because of a project's *tension between being specific and begin general*. One could argue that hard forks are a good test bed for contributions that diverge from the original project. Therefore, a family of related projects that serve slightly different needs but still coordinate may be a way to overcome this specificity-generality dilemma.

3.2 Interactions between Fork and Upstream Repository

Many interviewees indicate that they are interested in coordinating across repositories, either for merging changes back upstream or to monitor activity in the upstream repository to incorporate select or all changes. Some hard fork owners did not see themselves competing with the upstream project, but rather being part of a larger project. Also upstream maintainers tend to be usually interested in what happens in their forks. However, we see little evidence of actual synchronization or merging across forks in the repositories.

Discussion and Implications. We see new opportunities in coordinating and considering multiple forked projects as part of a larger community. Recent academic tools for improved monitoring [9] are potentially promising. Also more experimental ideas about virtual product-line platforms that unify development of multiple variants of a project [2] may provide inspiration for maintaining and coordinating hard forks. We believe that our dataset can be useful to develop and evaluate such tools in a realistic setting.

3.3 Perceptions of Hard Forking

The line between hard forks and social forks is subjective, but, when prompted, participants could draw distinctions that largely mirror our definition (long-term focus, extensive changes, fork with own community). For most interviewees, the dominant meaning of a fork is social fork. When asked specifically about hard forks, interviewees raised concerns about potential community fragmentation, confusing end users, and would have preferred to see hard-fork owners to contribute to the upstream instead.

Discussion and Implications. Overall, we see that the perception of forking has significantly changed compared to perceptions reported in earlier work. While there is still some concern about community fragmentation, it is rarely a concrete concern if there are actual reasons behind a hard fork. Transparent tooling seems to help with acceptance and with considering multiple hard forks as part of a larger community that can mutually benefit from each other. With tooling for coordination and merging, we think hard forks can be a powerful tool for exploring new ideas or testing whether there is sufficient support for features and ports for niche requirements or new target audiences (e.g., solving the specificity-generality dilemma with a deliberate process).

REFERENCES

[1] 2020. Appendix. https://github.com/shuiblue/ICSE20-hardfork-appendix.

[2] Michał Antkiewicz, Wenbin Ji, Thorsten Berger, Krzysztof Czarnecki, Thomas Schmorleiz, Ralf Lämmel, Ștefan Stănciulescu, Andrzej Wąsowski, and Ina Schaefer. 2014. Flexible Product Line Engineering with a Virtual Platform. In *Comp. Int'l Conf. Software Engineering (ICSE)*. ACM, 532–535.

[3] John W Creswell and J David Creswell. 2017. *Research design: Qualitative, quantitative, and mixed methods approaches.* Sage publications.

[4] Laura Dabbish, Colleen Stuart, Jason Tsay, and Jim Herbsleb. 2012. Social coding in GitHub: transparency and collaboration in an open software repository. In *Proc. Conf. Computer Supported Cooperative Work (CSCW)*. ACM, 1277–1286.

[5] Karl Fogel. 2005. *Producing open source software: How to run a successful free software project.* O'Reilly Media, Inc.

[6] Georgios Gousios, Martin Pinzger, and Arie van Deursen. 2014. An exploratory study of the pull-based software development model. In *Proc. Int'l Conf. Software Engineering (ICSE)*. ACM, 345–355.

[7] Linus Nyman. 2014. Hackers on forking. In *Proc. Int'l Symposium on Open Collaboration (OpenSym)*. ACM, 6.

[8] Donna Spencer. 2009. *Card sorting: Designing usable categories.* Rosenfeld Media.

[9] Shurui Zhou, Ștefan Stănciulescu, Olaf Leßenich, Yingfei Xiong, Andrzej Wąsowski, and Christian Kästner. 2018. Identifying Features in Forks. In *Proc. Int'l Conf. Software Engineering (ICSE)*. ACM Press, 105–116.

[10] Shurui Zhou, Bogdan Vasilescu, and Christian Kästner. 2019. What the Fork: A Study of Inefficient and Efficient Forking Practices in Social Coding. In *Proc. Europ. Software Engineering Conf./Foundations of Software Engineering (ESEC/FSE)*. ACM Press, New York, NY, 350–361.

[11] Shurui Zhou, Bogdan Vasilescu, and Christian Kästner. 2020. How Has Forking Changed in the Last 20 Years? A Study of Hard Forks on GitHub. In *Proceedings of the 42nd International Conference on Software Engineering (ICSE)*. ACM Press, New York, NY.