An Exploratory Study of Project Activity Changepoints in Open Source Software Evolution

James Walden
Department of Computer Science
Northern Kentucky University
Highland Heights, KY USA
waldenj@nku.edu

Noah Burgin
Department of EE and Computer Science
University of Tennessee
Knoxville, TN USA
noah22@vols.utk.edu

Kuljit Kaur
Department of Computer Science
Guru Nanak Dev University
Amritsar, India
kuljitchahal.cse@gndu.ac.in

***Abstract*—We applied changepoint analysis methods to a dataset of open source projects in order to determine the smoothness of open source evolution. We used a nonparametric changepoint detection algorithm on a dataset of 8,919 projects with at least four years of project activity selected from the World of Code, searching for changepoints in the number of commits and the number of unique contributing authors per month. We show that 99% of projects have changepoints in both time series, with a typical range between one and six changepoints. Increases and decreases in project activity occur with about equal frequency, while the size of changes varies tremendously.**

***Index Terms*—software evolution, changepoints**

I. Introduction

We performed an exploratory study of changepoints in open source project activity during the MSR 2021 hackathon. We analyzed project activity time series obtained from the World of Code [1], an archive cross-referencing over 120 million git repositories from multiple forges. We selected 8,919 projects from the World of Code that had sufficient history to compute monthly time series of commit activity, author activity, and the number of files changed.

Changepoints are data points in a time series, where the statistical properties of the data points before and after the changepoint differ significantly. While Lehman's laws of software evolution [2] point towards general tendencies of software evolution, such as increasing number of features and complexity, these laws do not describe whether a project gradually follows those trends or whether a project moves slowly at one point then rapidly at another point in its evolution.

Changepoint analysis has previously been used to identify instances when software performance changed [3], [4]. Two prior case studies of open source projects visually identified changepoints in development activity of a single project [5], [6] without using a changepoint detection algorithm.

We focused our study on two research questions:

1) How common are changepoints in open source project activity?
2) What are the sizes and directions of changes at changepoints?

II. Data

We selected projects from the World of Code [1], a research archive containing billions of commits from free, libre, and open source software (FLOSS) git repositories. In order to have a sufficiently long time series for changepoint analysis, we chose projects that had at a lifespan of at least four years, with at least 50 authors and 5000 commits. We found 8,919 projects that met our criteria.

We identified projects that met our criteria using the MongoDB `WoC.proj_metadata.R`. During the course of the multiweek virtual hackathon, World of Code transitioned from version R to S. We adapted our data collection scripts and procedures to use the new version S, in order to gain access to the new `rootfork` field it provided. Forges like GitHub contain many forks of popular projects, making it difficult to identify the repository that is used by the project team for development. Prior to version S, the only measure of centrality in a cluster of projects was algorithmically determined within WoC. The `rootfork` field was retrieved from GitHub, and therefore is the true root project.

We collected three monthly time series for each project: number of commits, number of unique authors, and number of files changed per commit. Time series were computed using the `getValues` commands that access data in pre-computed maps and tables within WoC. To get all commits for a certain project, we used the `p2c` map. These commits were then piped to `c2ta` to retrieve the timestamp and author of each commit. From here, a python script grouped each commit by month and counted the number of commits per month and how many unique authors made those commits in that month.

Noah: Can you provide performance data for both project selection and computing time series?

Since World of Code does not provide a map from commits to time and files changed, we used WoC's python interface, `oscar.py` to compute time series for the number of files changed. Commit objects in the python interface provided the needed timestamps and files changed per commit data. Commits were grouped by month and were used to compute the number of files changed per month.

Noah: Can you explain the zero size and zero row time series files I found? What about the post-2020 dates?

III. Changepoint analysis

As our time series data was not normally distributed, we used a nonparametric algorithm to detect changepoints. In particular, we used the implementation of the nonparametric PELT algorithm found in version 1.0.2 of the R `changepoint.np` package [7]. We used the algorithms default parameters, with the exception of specifying the minimum segment length be three months, as we wanted to find changes in activity that were at least slightly durable instead of looking for anomalous months.

We did not perform hyperparameter optimization, as we did not have a dataset of project activity time series with labeled changepoints to evaluate algorithm performance. An evaluation of multiple changepoint algorithms on a variety of time series [8] from different fields of study found little improvement from hyperparameter optimization of this algorithm in any case.

We found that more than 99% of projects have changepoints in project activity. Only 55 projects had no changepoints in their author time series, with most projects having between one and five changepoints. There are outliers, with six projects having 10 changepoints and one project having 14 changepoints. No project had between 11 and 13 changepoints. We can see the distribution of projects by number of changepoints in Figure 1.

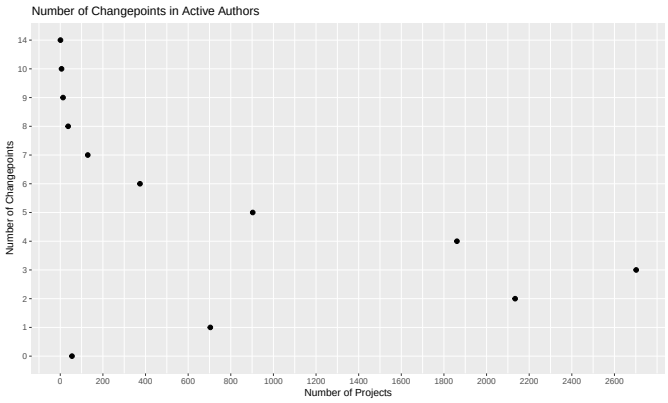

Fig. 1. Number of Changepoints in Author Time Series

Only 32 projects had no changepoints in their commit time series, with most projects having between one and six changepoints. Outliers include 27 projects with ten or more changepoints, with a single project having 16 changepoints.

We found a total of 31,416 changepoints in project commit time series, of which 15,342 (48.8%) were increases in commit activity and 16,047 (51.1%) were reductions in activity. We computed the magnitude of a changepoint as the difference in means in the number of monthly commits before and after the changepoint. The size of most changes were relatively small, with the interquartile range (IQR) ranging between -75 to 87 commits per month, but there was a substantial tail in both directions as can be seen in Figure 4.

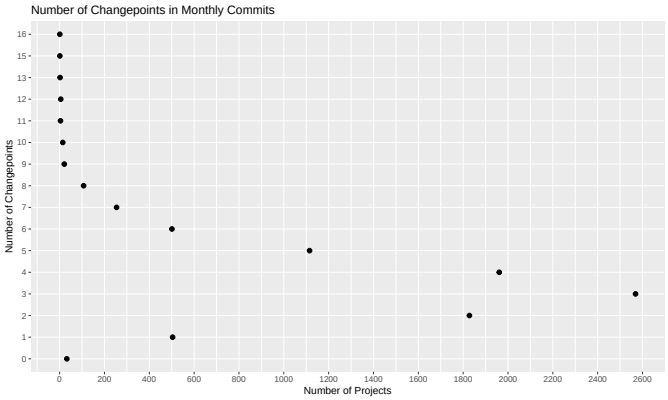

Fig. 2. Number of Changepoints in Commit Time Series

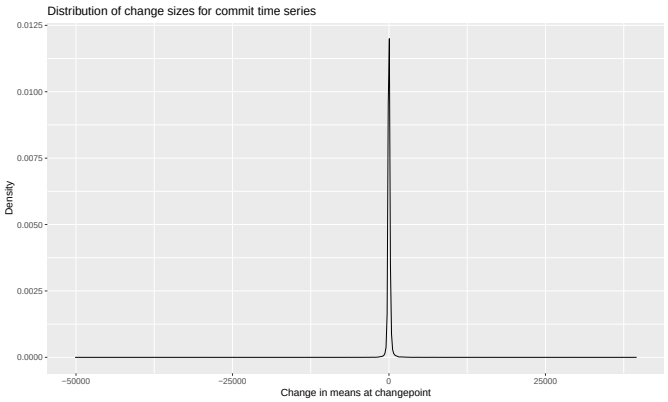

Fig. 3. Size of Changes in Commit Time Series

Our results for the signs and magnitude of author time series changes were similar. We found 28,671 changepoints in author time series, of which 12,114 (42.2%) were reductions in activity and 16,557 (57.7%) were increases in activity. Changes in the number of contributing authors per month were relatively small, with IQR ranging between -3.4 to 5.8 authors per month, but there was a substantial tail in both directions as can be seen in Figure **??**.

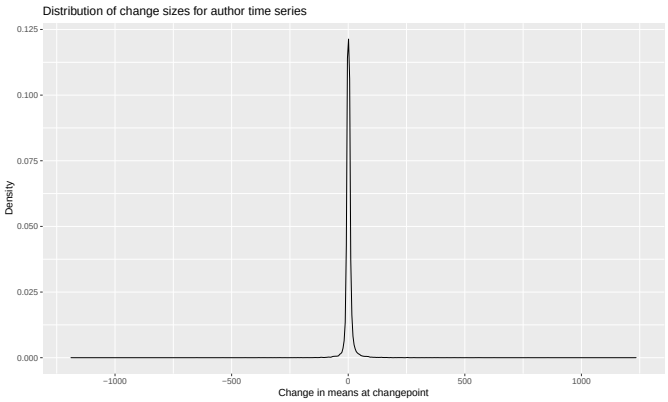

Fig. 4. Size of Changes in Author Time Series

IV. Conclusion

We found that open source evolution is rarely smooth and typically includes changepoints, points where the size and/or direction of evolution changes significantly.

The data and code used in this project can be found in the project's git repository at https://github.com/woc-hack/inflection-points.

In the future, we would like to examine our data in more detail to identify if there are common patterns of changepoints in open source evolution and to see what makes the outliers in number and magnitude of changepoints so different from the majority of projects. We would also like to gather WoC `P2c` commit time series data, which provides commits for all projects in a cluster. This would allow us to understand how the community's contributions to a project evolve rather than inspecting a single project at a time.

References

[1] Y. Ma, C. Bogart, S. Amreen, R. Zaretzki, and A. Mockus, "World of code: an infrastructure for mining the universe of open source vcs data," in *2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR)*. IEEE, 2019, pp. 143–154.

[2] M. M. Lehman, "Laws of software evolution revisited," in *European Workshop on Software Process Technology*. Springer, 1996, pp. 108–124.

[3] J. Cito, D. Suljoti, P. Leitner, and S. Dustdar, "Identifying root causes of web performance degradation using changepoint analysis," in *International Conference on Web Engineering*. Springer, 2014, pp. 181–199.

[4] S. Mühlbauer, S. Apel, and N. Siegmund, "Identifying software performance changes across variants and versions," in *2020 35th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 2020, pp. 611–622.

[5] J. M. Gonzalez-Barahona, G. Robles, I. Herraiz, and F. Ortega, "Studying the laws of software evolution in a long-lived floss project," *Journal of Software: Evolution and Process*, vol. 26, no. 7, pp. 589–612, 2014.

[6] J. Walden, "The impact of a major security event on an open source project: The case of openssl," in *2020 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR)*. IEEE, 2020.

[7] R. Killick and I. Eckley, "changepoint: An r package for changepoint analysis," *Journal of statistical software*, vol. 58, no. 3, pp. 1–19, 2014.

[8] G. J. van den Burg and C. K. Williams, "An evaluation of change point detection algorithms," *arXiv preprint arXiv:2003.06222*, 2020.