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 used a nonparametric changepoint detection algorithm to measure the smoothness of open source software evolution. Our dataset consisted of 8,919 projects with at least four years of history selected from the World of Code. We found changepoints in project activity time series in 99% of the projects, with most projects having between one and six changepoints. Increases and decreases in project activity occur with roughly equal frequency. While most changes are relatively small, on the order of a few authors or few dozen commits per month, there were long tails of much larger project activity changes.**

***Index Terms*—software evolution, changepoints, world of code**

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 historical data to compute monthly time series of project activity.

Lehman's laws of software evolution [2] describe how time series that describe characteristics of software, such as complexity or functionality, evolve in the long run. However, these laws do not address the question of whether such time series are smooth or punctuated by changepoints. Changepoints are data points in a time series, where the statistical properties of the data points before and after the changepoint differ significantly.

A five stage model of the software lifecycle has been proposed to explain how project activity changes throughout the lifecycle of a project [3]. The model was adapted to account for multiple phases of growth and stabilization found in open source software evolution [4]. Like many studies of software evolution, these papers analyzed software time series visually for a small number of projects.

In contrast, we analyzed thousands of projects using a changepoint detection algorithm [5] to measure the prevalence and size of changepoints in open source software evolution. The two research questions for this exploratory study were:

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

II. Data

In order to have a sufficiently quantity of data for changepoint analysis, we selected open source projects that had at a lifespan of at least four years, with at least 50 commit 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.S` collection. Project selection was completed in a few minutes.

During the course of the multi-week virtual hackathon, World of Code (WoC) data transitioned from version R to version S. We adapted data collection scripts and procedures written for version R to use the new version, 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 identifies the true root project based on data provided by GitHub.

We collected two monthly time series for each project: number of commits and number of authors that made one or more commits during the month. Time series were computed using the `getValues` commands that access data in precomputed maps and tables within WoC. To get all commits for a selected project, we used the `p2c` map. We then used the `c2ta` map to retrieve the timestamp and author of each commit. A python script grouped commits by month, counting the number of commits per month and the number of unique authors who made those commits. Running these processes on World of Code servers took four days. As other hackathon projects were simultaneously using these servers, it may be possible to compute the time series in a shorter amount of time.

III. Changepoint analysis

As our time series data was not normally distributed, we used the nonparametric PELT (Pruned Exact Linear Time) algorithm for changepoint detection. We used the implementation in version 1.0.2 of the R `changepoint.np` package [6]. We used the algorithm's default parameters, with the exception of specifying the minimum segment length to be three months, as we wanted to find changes in activity that were somewhat durable instead of looking for anomalous months.

We found that more than 99% of projects have changepoints in project activity. Only 55 projects had no changepoints in their author time series, and the median number of changepoints was three. Most projects (94%) had between one and six 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.

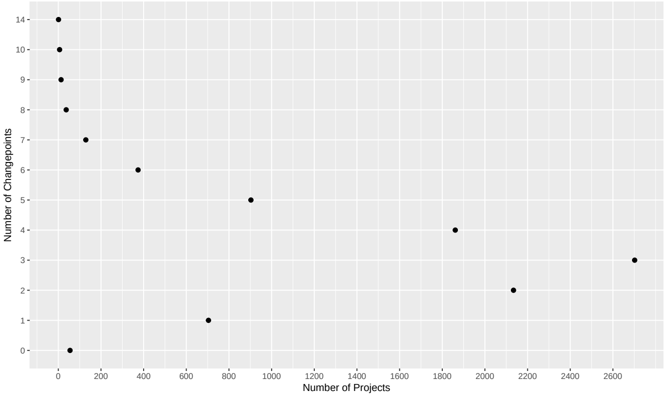

Fig. 1. Number of Changepoints in Author Time Series

The median number of changepoints in the commits per month time series was also three, with 32 projects having no changepoints. Most projects (90%) had between one and six changepoints. Outliers consisted of 27 projects with ten or more changepoints, including a single project with 16 changepoints.

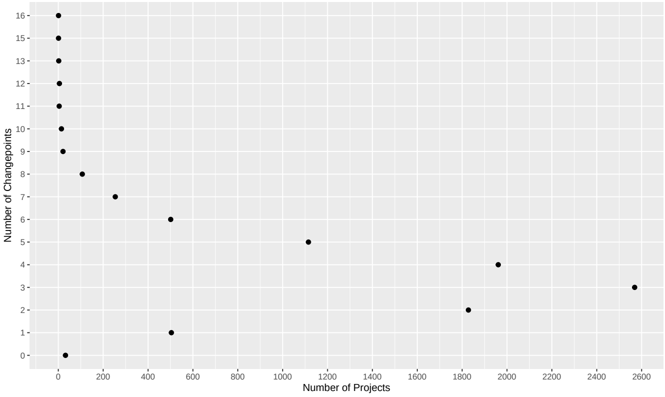

Fig. 2. Number of Changepoints in Commit Time Series

We found a total of 31,416 changepoints in project commit time series, of which 15,342 (49%) were increases in commit activity and 16,047 (51%) 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 3.

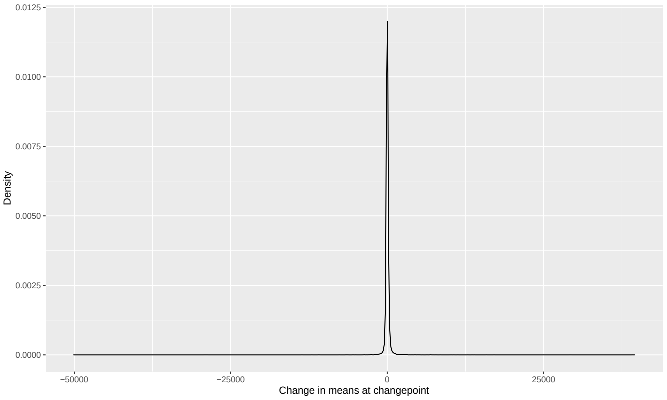

Fig. 3. Size of Changes in Commit Time Series

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%) were reductions in activity and 16,557 (58%) 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. The graph for author time series is identical in appearance to Figure 3 but with smaller values on the x-axis.

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 vast majority of projects have between one and six changepoints in both the number of monthly commits and number of unique authors per month, while some outliers have up to 16 changepoints. Changepoints that decrease or increase project activity occur with roughly equal frequency. While most changepoints are relatively small (a few authors per month, a few dozen commits per month), there is a long tail of much larger changes. 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 plan to study patterns of changepoints and to examine changepoints in software characteristics beyond project activity. Patterns may point towards a common software lifecycle model or common responses to external events, such as security incidents [7]. We also plan to investigate the causes of changepoints, with a focus on the long tail of large changes. We would also like to explore software evolution more broadly. Large open source projects, like the Linux kernel, often use multiple repositories. World of Code provides information about clusters of repositories, which we could use to investigate patterns of software evolution in such projects.

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] V. T. Rajlich and K. H. Bennett, "A staged model for the software life cycle," *Computer*, vol. 33, no. 7, pp. 66–71, 2000.

[4] A. Capiluppi, J. M. González-Barahona, I. Herraiz, and G. Robles, "Adapting the "staged model for software evolution" to free/libre/open source software," in *Ninth international workshop on Principles of software evolution: in conjunction with the 6th ESEC/FSE joint meeting*, 2007, pp. 79–82.

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

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

[7] 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.