International Journal of Open Source Software and Processes
Volume 13 • Issue 1

An Empirical Analysis of Inferences From Commit, Fork, and Branch Rates of Top GitHub Projects

Ekbal Rashid, RTC Institute of Technology, India*
Mohan Prakash, Jharkhand Rai University, India

ABSTRACT

GitHub repositories have been used to understand some important parameters related to quantitative growth of software projects. The rate at which fork, commits, and branches have been done in different repositories which are supposed to be the topmost ones when it comes to the number of contributors is the main point of study in this paper. The desirable goal is to find some sort of similarity in the trends of all these software projects. Similar trends tend to lead to interesting inferences regarding the quantitative growth and even the quality of these software projects. Is the rate of forking, branching, committing anything to do with the success of software projects? The authors have taken up this question in the current study.

KEYWORDS

Activity, Commit, Effort, Fork, GitHub, Project, Repositories, Software

INTRODUCTION

In an earlier work the authors had tried to analyze data of top ten GitHub projects of 2019 published in 'The State of the Octoverse' (https://octoverse.github.com/2019/). The data was collected from GitHub itself taking help from the methods elucidated in the documentation of GitHub (https://docs.github.com/en/github) and then some parameters were defined. The terms that were defined were related to the concepts of 'effort' and 'activity' of software projects. An attempt was made to understand how different kinds of efforts namely – commit-effort, fork-effort, and branch-effort could play significant roles in understanding the health of the software, or one can even say, the quality of the software. Similarly, some activity parameters were defined which could also help understand the quantitative growth of software projects in general. The activities that were defined in that particular research work were issue activity and pull request activity. It was felt during that research that these ratios were not adequate to understand the quantitative evolution of the software project. Take for example we have two software projects A and B. Let us assume that these projects have the same values for effort and activity but are working for different spans of time. A suppose is working for a period

DOI: 10.4018/IJOSSP.300751 *Corresponding Author

of 30 months and B for a period of 60 months. Let us assume that both have 1000 forks and 1000 contributors. That makes the fork-effort of both the projects equal to one. But the fact that B has been in business for twice the amount of time as compared to A may make one wonder whether that can play a significant role in understanding the situation of the project. An intuitive understanding is that this may indeed make us say that the time for which the project is in operation has indeed a lot to do with the understanding of the quantitative growth of the project. Whether the time factor has anything to do with the quality of the software is also a matter of deep inspection. Can we say that software projects that are developed faster can be better qualitatively just because they have been developed faster? Doesn't seem to be right. Because at first instance one may think what has the speed got to do with the quality? So, the present paper will be attempting to examine commit, fork and branch rates/ First these quantities have been formally defined and then the data collected has been presented. This is followed by the analysis and inferencing part and then the post research analysis. There is no doubt about the fact that researchers are looking towards GitHub repositories with a view to gathering data, especially the data of open-source software projects and then drawing interesting conclusions from it. They are also trying to find out why software projects succeed and why such projects fail. Invariably, such attempts will in a way lead to development of newer engineering techniques in the study of software projects.

A lot of work is also being done to understand the quality aspect of such software projects. GitHub is the storehouse for many such projects, mostly collaborative and open source in nature. The authors believe that quantitative growth has a deep and dialectical link with quality of software. We may understand quality in terms of quantity also and that changes in quantity however imperceptible they might be, bring about change in quality. Similarly, the change in quality is also a kind of quantitative change if looked at from the proper viewpoint. The direction of quantitative change may adversely or positively affect the quality of the software.

We can mine the GitHub data to see which projects are active for how long and also for how many commits, forks and branches have been completed in this span of time. This has been done in this paper. The data has been tabulated. Now, the reason why only the ten top projects have been considered is that the authors feel that understanding these projects may be a key to finding out whether these projects follow any kind of pattern in this regard. If there is a pattern then it may be linked with the quality of the software projects also. Of course, it goes beyond doubt that there may be the need of studying the data related to other software projects. The projects that have been listed here are the top projects of 2019 based upon the number of contributors. However, there are other ways of looking at software also. We may study the top ones from other years besides 2019. Or we study the data of software that have the highest rating. We may study software that have the biggest size, so on and so forth. All these may be the part of future study. At present, we are looking towards contributors and for the year 2019.

There is no doubt whatsoever that open-source collaborative software are ruling the world. Even proprietary ones are losing the race and are moving towards open-source. That brought about the interest of Microsoft in GitHub and eventually made the former acquire the latter. Right back in 2018, the acquisition had been complete as per a post published in the official blog of Microsoft. (https://blogs.microsoft.com/blog/2018/10/26/microsoft-completes-github-acquisition/). The post says, "…GitHub will retain its developer-first ethos, operate independently, and remain an open platform. Together, the two companies will work together to empower developers to achieve more at every stage of the development lifecycle, accelerate enterprise use of GitHub, and bring Microsoft's developer tools and services to new audiences…"(https://blogs.microsoft.com/blog/2018/10/26/microsoft-completes-github-acquisition/). This clearly elucidates GitHub's importance and also underlines the reason as to why the authors have decided to chose this platform to collect data from.