value of the number of contributors so that these parameters may actually work as suitable benchmarks. By this we mean that if a project has a certain minimum number of contributors then commit-effort and fork-effort values may help us decide whether a project is quantitatively on the right track, or even if the project is a good one.

2. We have seen that the data of branch-effort is not such that it gives us a value independent of the nature of the software project. It does not give us a value which will simply help us to determine whether the project is up to the mark or not. In this direction, what may be done is that the software projects may be divided into separate categories and then comparison of branch-effort may be performed and comparisons made in each category to see whether there is any specific pattern of interest or not. The authors feel the need of a separate paper to look into this matter. There is every possibility that the same category of software projects will have the same type of branch-effort values.
3. So far, the ratios haven't taken the time factor into consideration. Hence, we are not in a position to differentiate between two projects having similar values of activity and effort but are working for different time intervals. Say, we have two projects A and B having the same value of pull request activity or issue activity. Even though they might have the same values of commit-effort and fork-effort. This may make one conclude that the software projects are of the same quantitative nature, or even of the same quality to a particular extent. But if these two projects have been in operation for different intervals of time. Suppose project A has been around for 30 months and project B has been around for 60 months. Then, should we still treat them as of similar quantitative nature? The authors feel that there should be some parameter that works around the concept of time and this again can be something for future study.

References:

[1] https://octoverse.github.com/

[2] https://docs.github.com/en/github

[3] Padhye, Rohan & Kumarasamy Mani, Senthil Kumar & Sinha, Vibha. (2014) *A study of external community contribution to open-source projects on GitHub*. 10.1145/2597073.2597113. (ResearchGate)

[4] Hata H, Todo T, Onoue S, Matsumoto K (2015) Characteristics of sustainable oss projects: A theoretical and empirical study. In: Proceedings of the 8th international workshop on cooperative and human aspects of software engineering, CHASE '15. IEEE Press, Piscataway, pp 15–21.

[5] Tamburri, D.A., Palomba, F., Serebrenik, A. et al. Discovering community patterns in open-source: a systematic approach and its evaluation. Empire Software Eng 24, 1369–1417 (2019). https://doi.org/10.1007/s10664-018-9659-9 (Springer)

[6] Gamalielsson J, Lundell B (2013) Sustainability of open source software communities beyond a fork: how and why has the libreoffice project evolved? J Syst Softw 3(11):128–145. https://doi.org/10.1016/j.jss.2013.11.1077

[7] Schweik CM (2013) Sustainability in open source software commons: lessons learned from an empirical study of sourceforge projects. Technol Innov Manag Rev 3:13–19. http://timreview.ca/article/645

[8] Tamburri DA, Lago P, van Vliet H (2013a) Organizational social structures for software engineering. ACM Comput Surv,46(1):3,1–3,35. https://doi.org/10.1145/2522968.2522971

[9] https://octoverse.github.com/2019/