On the impact of GitHub Actions on CI services in GitHub

Mehdi Golzadeh
Software Engineering Lab
University of Mons, Belgium
mehdi.golzadeh@umons.ac.be

Constantly committing small changes to a project's codebase enables distributed developers to contribute to the same codebase and keep their copy of that code as up-to-date as possible. The CI tools ensure rapid feedback among developers and improve code quality by automating builds and tests of the codebase. Continuous integration is therefore a crucial part of collaborative software development.

There are a considerable number of CI tools and services available to software engineers, such as Cruise Control, Jenkins, CircleCI, and Travis to name a few. Developers can sign up for these services and have them automatically perform builds, tests and code quality checks whenever they commit code or receive external contributions in the associated version control repository. CI services can also be integrated by social coding platforms such as GitHub and GitLab. In 2015, GitLab introduced its integrated CI/CD service called *GitLabCI*. Bitbucket introduced *Bitbucket pipelines* in October 2019 and GitHub introduced GitHub Actions (abbreviated to GHA) in August 2019.

Given the popularity of GitHub, which hosts millions of repositories and is by far the largest social coding platform, we believe the introduction of GHA has had an impact on the use of CI services on GitHub. In particular, we believe that the introduction of GHA has raised awareness of the need for CI, lowered the entry barrier for CI use, and may even have caused projects to move away from other CI services in favour of GHA.

The goal of this presentation is to show if and how GitHub Actions has changed the CI landscape in GitHub. To do so, we performed a quantitative analysis of CI usage in around 200K GitHub repositories hosting reusable Node.JS packages available via the npm package registry. We identified CI usage by looking for the presence of some CI-specific configuration files (e.g., `.travis.yml` for Travis). We found that the use of a CI is widespread as almost half of the repositories in our dataset use a CI. However, not all CIs are equally used by repositories. We found that the 7 most popular CIs (out of an initial list of 20 CI services) already capture 99% of all CI usage. This list is composed of Travis, GHA, CircleCI, AppVeyor, Azure, GitLab CI and Jenkins.

Prior to the introduction of GHA, Travis was clearly the dominant CI, used by a majority of repositories. Since their introduction in 2014, CircleCI and AppVeyor have attracted repositories well and received a good share of repositories, respectively used by 12.4% and 3.9% of the repositories. GitLab CI, Azure, and Jenkins attracted a smaller share of GitHub repositories (about 1.1% for each). By the end of the observation period in May 2021, Travis and GHA are by far the most popular CIs and together cover more than 90% of all repositories with a CI. This may be surprising for GHA since it was only made available in 2019. In just 18 months, GHA has even surpassed Travis in popularity, being used by more than 51.7% of the repositories with a CI. For comparison, Travis is used by "only" 42.5% of the repositories. It is interesting since from the only study on GHA [1] we are aware of, only 1% of repositories are using GHA (926 projects).

The rapid growth of GHA coincides with changes in the adoption and discontinuation rates of other CIs. We found that the adoption rate of Travis, CircleCI, AppVeyor and Azure decreased. At the same time, the discontinuation rate for Travis, Azure and GitLab CI increased. To determine whether these changes in trend are due to the introduction of GHA, we relied on the statistical technique of the (linear) Regression Discontinuity Design. We confirmed that Travis, Azure and CircleCI were impacted by the introduction of GHA in 2019, resulting in a decrease in the growth rate of their use for Travis and CircleCI, and even a decrease in the use of Azure.

Not only did GHA manage to attract many more new repositories than its competitors, but we found it also managed to make many repositories migrating away from them and adopting GHA. We found that migrations towards GHA account for a quarter of the repositories using GHA. Moreover, GHA is the CI that attracted most migrations, regardless of the "source CI" of the migration. Our results highlight how CIs were impacted by the introduction of GHA on GitHub.

To find out why GHA is popular, a deeper analysis is needed, but we believe that reasons such as being fully integrated with GitHub and easily accessible for all repositories can serve a good reason. However, if these insights can be generalized beyond npm packages, they provide deep and relevant insight to open-source practitioners and CI service providers. Finally, We will also present future avenues of work to complete our empirical analysis with more refined qualitative analyses.

This research is supported by the joint FNRS / FWO Excellence of Science project SECO-ASSIST and FNRS PDR T.0017.18.

REFERENCES

[1] Timothy Kinsman, Mairieli Wessel, Marco A. Gerosa, and Christoph Treude. How do software developers use GitHub actions to automate their workflows? In *International Conference on Mining Software Repositories (MSR)*, 2021.