

Project Proposal: DevOps for Github

Siyang Zhang, sz2741, Tsai-Chen Hsieh, th2990

Project description and motivation

We plan to work on a Google Chrome extension that can enhance the experience of GitHub, with a focus on DevOps evaluation metrics. The extension would allow users to view DevOps-related metrics on the repository page, possibly presented via visual charts. DevOps evaluation is a relatively new field, with no standard established at the time of writing. However, it is crucial for teams that want to improve DevOps further. The vanilla GitHub does not have many metrics that focus on the evaluation of DevOps, and we want to change that. Software engineering includes more than technical details but also many methodologies and standards. In this project, we want to build some helpful metrics which allow developers to evaluate the workflow and improve better. ~~We are aware of some existing tools that might provide similar functionality to our proposal, but these tools are usually enterprise solutions, which are not suitable for a non-working environment, such as class projects.~~ The main feature that differentiates us from other tools is the ease of use. While other existing tools require software installation on the computer or additional account creation (such as Jenkins and Grafana), the chrome extension requires minimal installation and only needs a GitHub account. For functionality, we focus on intuition rather than customization. We want to let the developers be able to start using them without much knowledge of DevOps. In cases such as class projects or open source projects, the metrics not only help experienced developers become more efficient, but newcomers like students can also use the extension to learn more about DevOps. Current Computer Science education does not teach much about DevOps, so the project can help students understand DevOps from the evaluation perspective. In short, we view our project as a gentle introduction to DevOps rather than a sophisticated tool used in production. With the popularity of GitHub, our focus on the platform can benefit the most users.

Currently, there are a few key metrics presented in the midterm paper “Challenges of DevOps Implementation in the Current Age” that we can start with:

Deployment Frequency - calculation can be either done by number of deployments / the number of releases or number of deployments / time units.

Defect Density - calculation can be done by: number of issues / size of software, or number of issues / number of deployments

From the above examples, we can see that there are different ways to calculate the same metric. We will provide as much raw data as possible, which the user can customize to get the metrics that fit their goal. These metrics can be used for DevOps evaluation, such as the time it takes to commit to production, project risk, and productivity. There will also be default data for every metric so that the user can start using it immediately. The metrics mentioned above are documented by one or more research papers, and we would try to make sure that each metric is backed up with a paper.

~~In addition, we are currently facing a challenge: finding or building a testing tool that can adapt to multiple languages and generate a suitable bug report for our extension is hard. Without a reliable approach to run testing for committed codes, we cannot record any data we need and finish our evaluation. We will continue to search for a possible solution as we have enough time. Otherwise, we will limit our project scope to a specific language in which we can find a tool and obtain the required data.~~

This project is related to Tsai-Chen Hsieh's midterm paper "Challenges of DevOps Implementation in the Current Age", specifically the section on DevOps evaluation.

Implementation

We would use the development tools provided by Google Chrome to develop the extension, which we will mainly write in Javascript. The data would be provided by GitHub Rest API, where Octokit.js can be used in Javascript. There are also many Javascript libraries that we can use for visualization, such as D3.js.

The project will be divided into two parts: metrics that focus on GitHub information and metrics that require language-specific test cases. Deployment frequency is an example of the former, where we can get the data from GitHub Actions. For the latter, defect density is an example because we need to acquire detailed test case information, which is language specific. For now, we will focus on Flutter since we are relatively more familiar with project developments in Flutter. We will run corresponding Flutter tests using the official packages and generate bug reports so that we can evaluate them with our metrics.