Wait For It: Determinants of Pull Request Evaluation Latency on GitHub

Y. Yu*†, H. Wang*, V. Filkov†, P. Devanbu†, B. Vasilescu†

*College of Computer, National University of Defense Technology. China
†Department of Computer Science, University of California, Davis. USA

1) Motivation

The pull-based model is wildly used in distributed software teams. It offers a much lower barrier to entry for potential contributors, since anyone can submit pull requests to any repository.

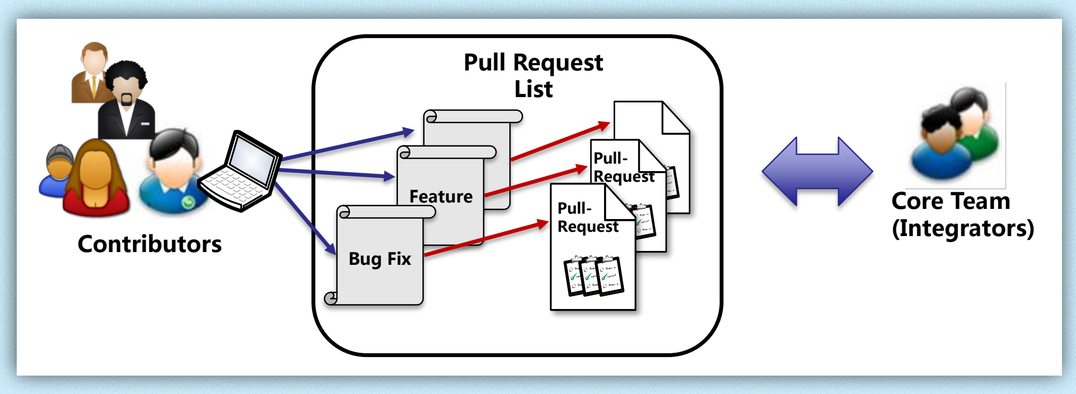

Figure 1. Usage of Pull Requests

The members of a project's core team (*integrators*) are responsible for evaluating the proposed changes and integrating them into the main development line. They often struggle to keep up with the volume of incoming pull requests. Automated testing, or continuous integration (CI), is one technology that helps them manage the high review load.

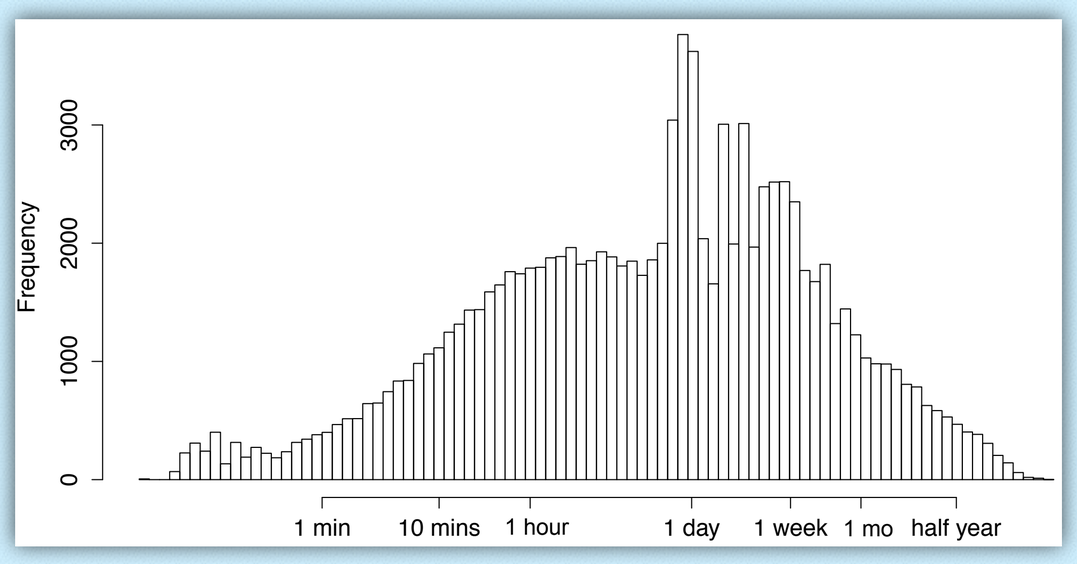

Figure 2. Distribution of Pull Request Evaluation Time

2) Our study

We report on a quantitative study that tries to resolve which factors affect pull request evaluation latency in GitHub. We model the evaluation time of merged pull requests (PRs) tested by Travis-CI.

Attributes	Ruby	Python	JavaScript	Java/C++	**Total**
#Integrators	220	177	190	103	690
#PRs received	28,409	28,903	26,983	18,989	103,284
#PRs merged	20,755	24,039	17,920	13,456	76,170
#PRs merged&CI-tested	11,562	11,955	11,821	5,510	40,848

Table 1. Summary of Our Dataset

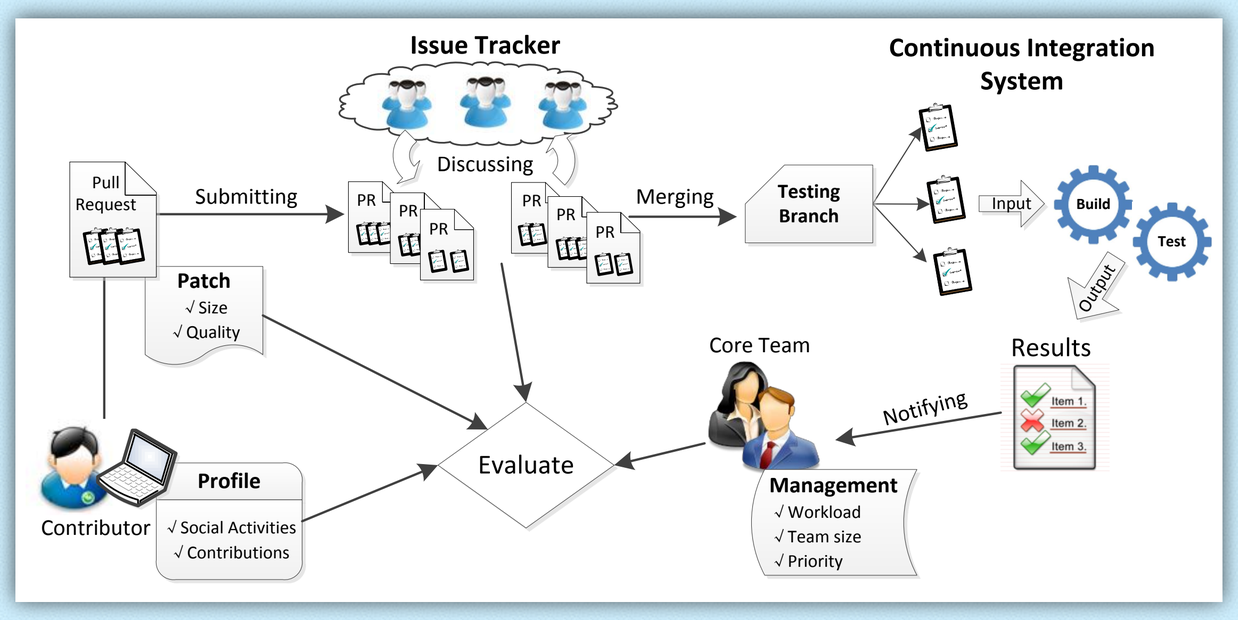

Figure 3. Evaluation Model of Pull Requests

3) Results

The evaluation latency is a complex issue, requiring many independent variables to explain adequately.

We build three different models and confirm the following three hypotheses:

H1. Previously-identified social and technical factors influence pull request latency in expected ways.

H2. Process-related factors have a significant impact on pull request latency.

H3. Continuous integration is a dominant factor of pull request latency.

Features	Model 1 Coeffs(Errors)	Model 1 Sum Sq.	Model 2 Coeffs(Errors)	Model 2 Sum Sq.	Model 3 Coeffs(Errors)	Model 3 Sum Sq.
(Intercept)	0.072 (0.009)***		0.045 (0.009)***		0.155 (0.008)***	
scale(log(proj_age))	0.022 (0.004)***	276.96***	−0.014 (0.004)**	276.96***	−0.028 (0.004)***	276.96***
scale(log(team_size))	−0.055 (0.004)***	7.92***	−0.108 (0.004)***	7.92***	−0.108 (0.004)***	7.92***
scale(log(n_additions + 0.5))	0.064 (0.005)***	3354.64***	0.065 (0.005)***	3354.64***	0.035 (0.004)***	3354.64***
scale(log(n_deletions + 0.5))	−0.016 (0.005)**	54.92***	0.001 (0.005)	54.92***	−0.000 (0.004)	54.92***
scale(log(n_commits + 0.5))	0.147 (0.005)***	3789.65***	0.130 (0.005)***	3789.65***	0.028 (0.004)***	3789.65***
scale(log(hotness + 0.5))	0.016 (0.004)***	74.31***	0.001 (0.004)	74.31***	0.016 (0.003)***	74.31***
pr_includes_testsTRUE	0.108 (0.010)***	194.91***	0.076 (0.009)***	194.91***	0.009 (0.008)	194.91***
scale(log(n_comments + 0.5))	0.409 (0.005)***	5482.16***	0.189 (0.005)***	5482.16***	0.037 (0.005)***	5482.16***
scale(submitter_success_rate)	−0.037 (0.005)***	432.86***	−0.023 (0.004)***	432.86***	−0.016 (0.004)***	432.86***
scale(strength_social_connection)	−0.072 (0.005)***	494.32***	−0.037 (0.005)***	494.32***	−0.052 (0.004)***	494.32***
scale(log(n_followers + 0.5))	−0.090 (0.004)***	358.74***	−0.108 (0.004)***	358.74***	−0.064 (0.004)***	358.74***
submitter_is_integratorTRUE	−0.129 (0.011)***	56.10***	−0.095 (0.010)***	56.10***	−0.078 (0.009)***	56.10***
scale(log(proj_age)):scale(log(team_size))	−0.074 (0.004)***	234.23***	−0.012 (0.004)**	107.07***	−0.016 (0.004)***	70.27***
scale(log(description_complexity))			0.115 (0.004)***	960.00***	0.087 (0.004)***	960.00***
scale(log(availability + 0.5))			0.037 (0.004)***	124.96***	0.033 (0.003)***	124.96***
scale(log(n_open_pr + 0.5))			0.166 (0.005)***	908.33***	0.151 (0.004)***	908.33***
Friday_effectTRUE			0.068 (0.010)***	34.68***	0.062 (0.009)***	34.68***
issue_tagTRUE			0.096 (0.009)***	56.72***	0.081 (0.008)***	56.72***
mention_tagTRUE			−0.060 (0.013)***	14.11***	−0.020 (0.012)	14.11***
scale(log(first_rsp + 0.5))			0.274 (0.005)***	1892.64***	0.243 (0.004)***	1892.64***
scale(log(team_size)):scale(log(workload+0.5))			−0.071 (0.004)***	163.80***	−0.041 (0.004)***	60.39***
scale(log(total_ci_time))					0.481 (0.005)***	3855.79***
ci_errorTRUE					−0.401 (0.009)***	977.41***
ci_failTRUE					−0.016 (0.009)	0.27
scale(log(first_rsp+0.5)):scale(log(total_ci_time))					−0.102 (0.003)***	434.71***
Adjusted R-squared	0.362		0.461		0.587	

$^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$

Table 2. Pull Request Latency Models and Results

yuyue@nudt.edu.cn • vasilescu@ucdavis.edu