Understand the impact of CI in Software Development

Quantitative and Qualitative Insights from 87 GitHub Projects

Ningning Yang

Agenda

- Introduction & Background
- Quantitative Study
- Qualitative Study
- Implications
- The Validity & Limitation of Study
- Conclusion

Introduction

- Brief Overview of CI
  - A software development practice involving the automatic integration of code changes into a shared repository
- Motivation
  - Quicken the delivery of merged PRs in software projects
- Significance
  - The study reveals crucial insights

Background – Study with TRAVISCI

- Study Scope
  - 87 GitHub projects employing TRAVISCI as their chosen CI service.
  - TRAVISCI is known for automating builds and testing processes upon code changes.
- Quantitative Analysis
  - Investigated the impact of TRAVISCI on the delivery time of merged pull requests across these projects.
- Qualitative Insights
  - Complemented the quantitative aspect with a qualitative study, tapping into the perceptions of contributors involved in 73 of these projects.

Quantitative Study

- Q1: Are merged pull requests released more
- quickly using a CI service?
- Q2: Does the increased number of PR submissions after adopting a CI service increase the delivery time of pull requests?
- Q3: What factors impact the delivery time after adopting a CI service?

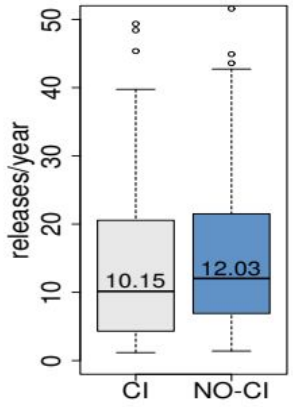


Fig. 3: Releases per year *before* and *after* TRAVISCI.

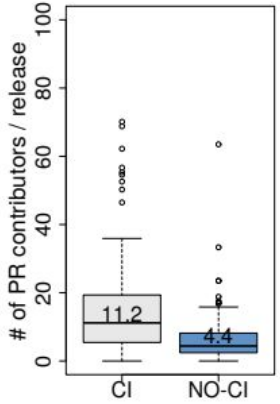
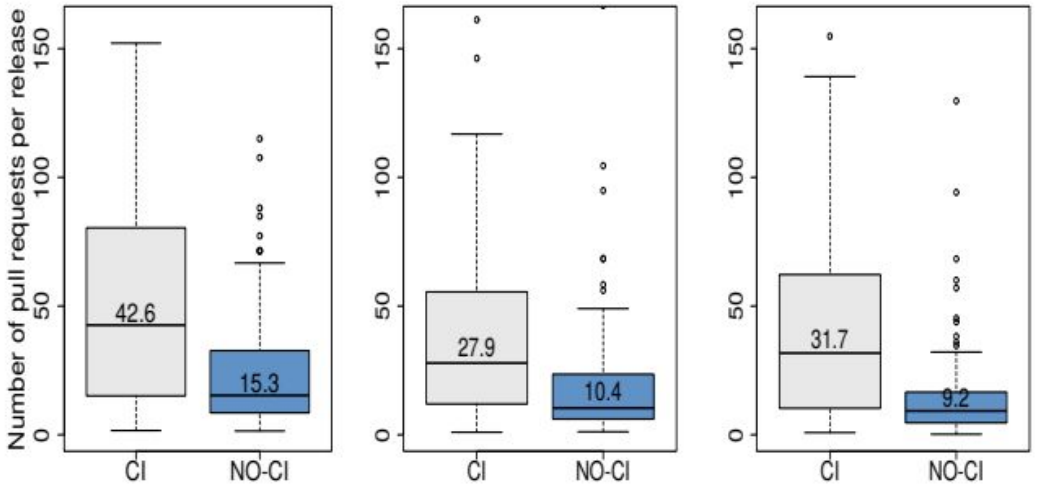


Fig. 4: PR contributors per release.



(a) Submitted PRs (b) Merged PRs (c) Delivered PRs

Fig. 2: PR submission, merge, and delivery rates per release.

Quantitative Study Findings

- Q1: Only 51.3% of projects showed quicker delivery of merged PRs after adopting TRAVISCI.
- Q2: 54% of projects experienced longer PR lifetimes after CI adoption.
- Q3: Increased PR submissions and merge workload were significant factors affecting delivery time.

Qualitative Study

- Q4: What is the perceived influence of CI on the time to deliver merged PRs?
- Q5: What are the perceived causes of delay in the delivery time of merged PRs?
- Q6: What is the perceived influence of CI on the software release process?
- Q7: What is the perceived influence of CI on the code review process?
- Q8: What is the perceived influence of CI on attracting more contributors to open-source projects?

Qualitative Study Findings

- Q4: Contributors have mixed views on CI's impact on delivery time, reflecting the diversity of perceptions.
- Q5: Contributors attribute delivery time delays to increased complexity and PR volume, aligning with quantitative results.
- Q6: CI is seen as facilitating a smoother release process, emphasizing consistent and automated testing.

Qualitative Study Findings

- Q7: CI streamlines code reviews, enhancing focus and efficiency, though complexities arise pre-CI adoption.
- Q8: CI adoption makes projects appealing to new contributors, showcasing commitment to modern practices and code quality.

Implications

- Considered CI Adoption
- Focus Beyond Speed
- Adaptation to Increased Workloads
- Quality over Quantity
- Shorter Release Cycles

The Validity & Limitations of Study

Validity

- Comprehensive Data Analysis
- Mixed-Methods Approach
- Empirical Evidence
- Relevance to Current Practices

Limitations

- Focus on TRAVISCI Only
- Restricted to GitHub Projects
- Potential Biases in Survey Responses
- Open-Source Project Emphasis
- Generalizability Concerns

Conclusions

The study on the impact of CI services like TRAVISCI on the delivery time of merged PRs offers new insights that challenge prevailing assumptions in software development.

While CI brings several advantages, its impact on speeding up PR delivery is not as straightforward as often believed.

Teams must approach CI adoption with a nuanced understanding, focusing on its broader benefits and preparing for potential increases in workload. By doing so, they can effectively harness CI to enhance their software development processes.

Reference

Bernardo, J. H., da Costa, D. A., Kulesza, U., & Treude, C. (2023). The impact of a continuous integration service on the delivery time of merged pull requests. *Empirical Software Engineering*, *28*(4). https://doi.org/10.1007/s10664-023-10327-6