

# Understanding and Improving Continuous Integration

Michael Hilton  
Oregon State University, USA  
hiltonm@eecs.oregonstate.edu

## ABSTRACT

Continuous Integration (CI) has been widely adopted in the software development industry. However, the usage of CI in practice has been ignored for far too long by the research community. We propose to fill this blind spot by doing in-depth research into CI usage in practice. We will answer *how* questions by using quantitative methods, such as investigating open source data that is publicly available. We will answer *why* questions using qualitative methods, such as semi-structured interviews and large scale surveys. In the course of our research, we plan on identifying barriers that developers face when using CI. We will develop techniques to overcome those barriers via automation.

This work is advised by Professor Danny Dig.

## CCS Concepts

•Software and its engineering → *Agile software development*; Software testing and debugging;

## Keywords

Continuous Integration; Software Engineering

## 1. RESEARCH PROBLEM

Continuous Integration (CI) is emerging as one of the biggest success stories in automated software engineering. CI systems automate the compilation, building, and testing of software. For example, such automation has been reported [2] to help Flickr deploy to production more than 10 times per day. Others claim [4] that by adopting CI and a more agile planning process, a product group at HP reduced development costs by 78%.

These success stories have led to CI growing in interest and popularity. Travis CI [1], a popular CI service, reports that over 300,000 projects are using Travis. The State of Agile industry survey [7], with 3,880 participants, found 50% of respondents use CI. Another large survey, The State of

DevOps report [8] finds one of the indicators of “high performing IT organizations” is CI.

Despite CI rising as a big success story in automated software engineering, it has received almost no attention from the research community. For example, how widely is CI used in practice, and what are some costs and benefits associated with CI? Without answering such questions, developers, tool builders, and researchers make decisions based on hunches instead of data.

Not only will this work allow us to build tools which will have immediate benefit to the many developers who are using CI, our results will open new avenues of research into the CI practice. This will benefit the entire research community.

## 2. RELATED WORK

The most related work in this area is by Vasilescu et al. [9]. This paper examines quality and productivity outcomes related to CI in Github. The authors collect a data set of 246 projects on Github. While this paper points to positive benefits of CI, the authors state that this paper must be considered a preliminary study. The authors claim that the exact mechanisms behind this should be better understood. This deeper understanding is exactly what we hope our work will achieve.

Miller et al. [6] present a case study of a team which adopted CI. The authors describe the first 100 days of a team which transitioned from not using CI to using CI. This paper does not claim to provide broad conclusions about the benefits of CI since they are only observing one team.

Laukkanen et al [5] also performed a case study of the adoption of CI. Again, this study is a single case study, and not generalizable.

## 3. PROPOSED APPROACH

Our proposed approach has three stages. First we will study the current usage of CI in a very large corpus of open source projects. Second, we will understand what are the barriers developers face when using CI. Third we will develop automated techniques to assist developers in overcoming these barriers.

In the first stage we will study *what* are some of the current practices for using CI. For this stage, we will ask questions such as: “How many projects use CI?”, “How does the use of CI differ between more and less successful projects?” and “What are the costs and benefits of CI?” To answer these questions, we will mine GitHub to find which projects are using CI, and learn how those projects use CI. This approach will also help us learn the differences between how

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author.

Copyright is held by the owner/author(s).

FSE’16, November 13–18, 2016, Seattle, WA, USA  
ACM 978-1-4503-4218-6/16/11...\$15.00  
<http://dx.doi.org/10.1145/2950290.2983952>

successful and unsuccessful projects use CI.

In the second stage we will study *why* developers use CI. We will ask questions to the developers via semi-structured interviews as well as surveys. We will ask developers about how and why they use CI, what barriers they experience when using CI, and how to they make sense of the information they get from their CI tool(s).

In the third stage we will develop automated techniques to assist developers in overcoming these barriers. We will develop automated tools to help developers overcome the barriers we previously discovered. For example, we plan to develop tools to help with the initial setup, running, and maintenance of CI systems.

## 4. CONTRIBUTIONS

The first contribution of this work is to advance the knowledge of the research community about CI usage. Without this understanding, it will be impossible for researchers to make contributions that are useful to developers. This increased awareness will help researchers to pay attention to CI because it is not a passing trend. Our study will also help developers by listing situations where CI provides the most value. Moreover, for tool builders we can discover several opportunities where automation can help.

We believe that this work will help software engineering researchers better understand the barriers to adopting and using CI, and that this better understanding will open up new avenues for research. For tool builders, a better understanding of barriers will help them know what to focus on when developing new features. For developers, raising the visibility of issues they struggle with will help researchers and tool builders find solutions for them, which will help developers be even more productive.

We will release all of our tools as open source, so that they can be freely used by the development community. We will also release companion websites for our research work, to make our results more accessible to the development community.

## 5. PRELIMINARY RESULTS

Our initial work consists of a study of the Usage, Costs, and Benefits of Continuous Integration in Open-Source Projects. [3]. In this work, we use three complementary methods to study in-depth the usage of CI in open-source projects. To understand what CI systems developers use, we analyzed 34,544 open-source projects from GitHub. To understand how developers use CI, we analyzed 1,529,291 builds from the most popular CI system. To understand why projects use or do not use CI, we surveyed 442 developers. With this data, we answered 14 questions related to the usage, cost, and benefits of CI.

Our results show there are good reasons for the rise of CI. Compared to projects that do not use CI, projects that use CI: (i) release twice as often, (ii) accept pull requests faster, and (iii) have developers who are less worried about breaking the build. Therefore, it should come at no surprise that 70% of the most popular projects on GitHub heavily use CI.

The trends that we discover point to an expected growth of CI. In the future, CI will have an even greater influence than it has today. We hope that our initial results provide a call to action for the research community to engage with

this important field of automated software engineering.

## 6. EVALUATION PLAN

We will evaluate our work in complementary ways.

The qualitative work is difficult to evaluate, so we will triangulate the results from our semi-structured interviews with surveys as well as GitHub mining to determine if our findings generalize.

When we develop tools, we will evaluate them by asking several research questions:

*Are these tools useful?* One way to answer this question is to send patches generated by our tools to open source projects, and measure the acceptance ratio.

*Are these tools accurate?* For this will measure precision and recall of our generated configuration files. For example, we could compare the config file we generated with existing configs to see if we correctly identified everything necessary to be able to correctly run CI for a given code repository.

*Can the tools help CI run faster?* There has been a lot of work in test suit reduction, selection and prioritization. Are the current mechanisms effective for CI? Does it matter if the CI is local or in the cloud? Are there tests that only make sense to run in the cloud? Are there some tests that don't make sense to run in the cloud?

## 7. ACKNOWLEDGMENTS

The author is advised by Danny Dig, both of whom are members of the Software Evolution Group at Oregon State University.

## 8. REFERENCES

- [1] Travis CI. <https://travis-ci.org/>. Accessed: 2016-04-21.
- [2] J. Allspaw and P. Hammond. 10+ deploys per day: Dev and ops cooperation at Flickr. <https://www.youtube.com/watch?v=LdOe18KhtT4>. Accessed: 2016-04-21.
- [3] M. Hilton, T. Tunnell, D. Marinov, D. Dig, K. Huang, and others. Usage, costs, and benefits of continuous integration in open-source projects.
- [4] J. Humble. Evidence and case studies. <http://continuousdelivery.com/evidence-case-studies/>. Accessed: 2016-04-29.
- [5] E. Laukkanen, M. Paasivaara, and T. Arvonen. Stakeholder perceptions of the adoption of continuous integration: A case study. In *AGILE*, 2015.
- [6] A. Miller. A hundred days of continuous integration. In *AGILE*, 2008.
- [7] V. One. 10th annual state of Agile development survey. <https://versionone.com/pdf/VersionOne-10th-Annual-State-of-Agile-Report.pdf>, 2016.
- [8] Puppet and DevOps Research and Assessments (DORA). 2016 state of DevOps Report. <https://puppet.com/system/files/2016-06/2016%20State%20of%20DevOps%20Report.0.pdf>, 2016.
- [9] B. Vasilescu, Y. Yu, H. Wang, P. Devanbu, and V. Filkov. Quality and productivity outcomes relating to continuous integration in GitHub. In *FSE*, 2015.