A CONCEPTUAL REPLICATION OF CONTINUOUS INTEGRATION PAIN POINTS
A Conceptual Replication of Continuous Integration Pain Points

David Widder
Michael Hilton
Christian Kästner
Bogdan Vasilescu
WHAT WILL I TALK ABOUT?
WHAT WILL I TALK ABOUT?

1. WHAT IS CONTINUOUS INTEGRATION? WHY DO PEOPLE USE IT?
WHAT WILL I TALK ABOUT?

1. WHAT IS CONTINUOUS INTEGRATION? WHY DO PEOPLE USE IT?

2. WHY DO A REPLICATION?
WHAT WILL I TALK ABOUT?

1. WHAT IS CONTINUOUS INTEGRATION? WHY DO PEOPLE USE IT?

2. WHY DO A REPLICATION?

3. OUR MIXED METHOD
WHAT WILL I TALK ABOUT?

1. WHAT IS CONTINUOUS INTEGRATION? WHY DO PEOPLE USE IT?

2. WHY DO A REPPLICATION?

3. OUR MIXED METHOD

4. HIGHLIGHTS OF FINDINGS: WHAT REPLICATES + WHAT DOESN’T?
WHAT IS CONTINUOUS INTEGRATION?
WHAT IS CONTINUOUS INTEGRATION?
WHAT IS CONTINUOUS INTEGRATION?
What is continuous integration?
WHAT IS CONTINUOUS INTEGRATION?
WHAT IS CONTINUOUS INTEGRATION?
WHAT IS CONTINUOUS INTEGRATION?
WHAT IS CONTINUOUS INTEGRATION?
CI IS A BEST PRACTICE
CI IS A BEST PRACTICE

6 REASONS TO USE CI

WHY WE NEED CI

WHAT IS CI AND WHY USE IT?

WHAT IS CI AND WHY DO YOU NEED IT?

7 REASONS YOU SHOULD USE CI

WHY CI IS IMPORTANT
LOTS OF RESEARCH SHOWING CI BENEFITS
Lots of research showing CI benefits

- Helps catch bugs faster
- Release twice as often
- Fewer pull requests rejected
- Higher pull request throughput
- Encourages people to write tests
Also, lots of research showing CI pain points
Also, lots of research showing CI pain points:

- Hard to configure
- Too many options
- Unsupported features
- Insecure
- Long builds
WE BELIEVE IT IS TIME TO REVIEW AND REPLICATE CI PAIN POINTS
WHY REVIEW AND REPLICATE CI PAIN POINTS?
WHY REVIEW AND REPLICATE CI PAIN POINTS?

- EVALUATE & SYNTHESIZE PAST RESEARCH
WHY REVIEW AND REPLICATE CI PAIN POINTS?

- EVALUATE & SYNTHESIZE PAST RESEARCH
- PROVIDE REPLICATED GUIDANCE TO PRACTITIONERS
WHY REVIEW AND REPlicate CI PAIN POINTS?

- EVALUATE & SYNTHESIZE PAST RESEARCH
- PROVIDE REPLICATED GUIDANCE TO PRACTITIONERS
- FOCUS FUTURE RESEARCH ON AREAS OF UNCERTAINTY
WHAT’S A CONCEPTUAL REPPLICATION?

CURRENT CI USERS

SWITCHERS & ABANDONERS
What's a conceptual replication?

Current CI Users

Switchers & Abandoners

?
WHY STUDY LEAVERS INSTEAD OF CURRENT USERS?
WHY STUDY LEAVERS INSTEAD OF CURRENT USERS?
CURRENT CI USERS
INDUSTRY & OSS

TRAUSICI LEAVERS
OSS

LIT REVIEW

PAIN POINTS

SURVEY & INTERVIEWS

TRIANGULATION

LOGISTIC REGRESSIONS

PAIN POINTS REPLICATED
(OR NOT)
CURRENT CI USERS INDUSTRY & OSS

LIT REVIEW

SURVEY & INTERVIEWS

TRIANGULATION

PAIN POINTS

LOGISTIC REGRESSIONS

PAIN POINTS REPLICAED (OR NOT)
CURRENT CI USERS
INDUSTRY & OSS

TRAVISCI LEAVERS
OSS

LIT REVIEW

SURVEY & INTERVIEWS

PAIN POINTS

LOGISTIC REGRESSIONS

PAIN POINTS
REPLICATED
(OR NOT)

TRIANGULATION
CURRENT CI USERS
INDUSTRY & OSS

TRAVIS CI LEAVERS
OSS

LIT REVIEW

SURVEY &
INTERVIEWS

TRIANGULATION

PAIN POINTS

LOGISTIC
REGRESSIONS

PAIN POINTS
REPLICATED
(OR NOT)
CURRENT CI USERS INDUSTRY & OSS

TRAVISCI LEAVERS OSS

LIT REVIEW

SURVEY & INTERVIEWS

TRIANGULATION

PAIN POINTS

LOGISTIC REGRESSIONS

PAIN POINTS REPLICATED (OR NOT)
LIT REVIEW
37 PAPERS
35 PAIN POINTS
132 Survey Responses

“Why did your project stop using Travis CI, and what has the CI situation been like since then?”
Science in progress! afternoon spent card sorting ~140 survey responses w @michaelhilton 🌟: @b_vasilescu #mixedmethods
132 survey responses + 12 interviews
132 survey responses + 12 interviews
WHY LOGISTIC REGRESSION?

PEOPLE WHO LEAVE CI VS THOSE WHO DON’T
ABANDONMENT VS SWITCHING

[Image of CI tool with a success symbol]

Commits on Aug 7, 2019

[maven-release-plugin] prepare for next development iteration
kohsuke committed 4 days ago

All checks have failed
1 failing check

× continuous-integration/jenkins/branch — This c... Details
ABANDONMENT VS SWITCHING

![CI](CI.png)

Commits on Aug 7, 2019

[maven-release-plugin] prepare for next development iteration
- kohsuke committed 4 days ago

All checks have failed
- continuous-integration/jenkins/branch — This c...
ABANDONMENT VS SWITCHING

2 LOGISTIC REGRESSIONS, 6,239 REPOS
ABANDONMENT VS SWITCHING

2 LOGISTIC REGRESSIONS, 6,239 REPOS
TRIANGULATED RESULTS
C#}

UNSUPPORTED TECHNOLOGIES
UNSUPPORTED TECHNOLOGIES
“I switched to Wercker, a container based CI pipeline, which means it can execute any scripts with much more flexibility.” (P89)
“I switched to Wercker, a container based CI pipeline, which means it can execute any scripts with much more flexibility.” (P89)

NEEDING DOCKER:
9.5X INCREASE IN SWITCHING,
5.25X INCREASE IN ABANDONING
“I switched to Wercker, a container based CI pipeline, which means it can execute any scripts with much more flexibility.” (P89)

NEEDING DOCKER:
9.5X INCREASE IN SWITCHING,
5.25X INCREASE IN ABANDONING

LACKING LANGUAGE SUPPORT:
1.5X INCREASE IN SWITCHING AND ABANDONING
UNSUPPORTED TECHNOLOGIES
C# UNSUPPORTED TECHNOLOGIES NEEDING DOCKER AFFECTS NEW TRAVISCI USERS ONLY
NEEDING DOCKER AFFECTS NEW TRAVISCI USERS ONLY

WHEN CAN PEOPLE HACK THROUGH TECHNOLOGY SUPPORT CHALLENGES, AND WHEN ARE THE NOT ABLE TO?
LACK OF TESTS
LACK OF TESTS
LACK OF TESTS

“The goal [of adopting Travis] was to ‘force’ myself to add some real tests (to have green Travis badge again!) but this failed so far :)
” (P111)
“The goal [of adopting Travis] was to ‘force’ myself to add some real tests (to have green Travis badge again!) but this failed so far :)” (P111)

1% INCREASE IN TESTS = 16% LOWER CHANCE OF ABANDONING (NO EFFECT ON SWITCHING)
CI CONSISTENCY
CI CONSISTENCY
“I had some open source projects running in TravisCI and some in CircleCI. I just wanted to consolidate the project to one place and I’m sorry to say that at that time TravisCI lost the battle.” (P57)
CI CONSISTENCY

“I had some open source projects running in TravisCI and some in CircleCI. I just wanted to consolidate the project to one place and I’m sorry to say that at that time TravisCI lost the battle.” (P57)

EXPOSURE TO LEAVERS:
~1.5X INCREASE IN CHANCES OF SWITCHING & ABANDONING
CI CONSISTENCY
PEOPLE’S PAST EXPERIENCE IS VERY IMPORTANT IN DETERMINING BEHAVIOR ON FUTURE PROJECTS!
PEOPLE’S PAST EXPERIENCE IS VERY IMPORTANT IN DETERMINING BEHAVIOR ON FUTURE PROJECTS!

BUT NOT OBSERVED IN PAST LITERATURE, SO NEEDS REPLICATION.
SINGLE METHOD OR CONFLICTING RESULTS
LONG BUILD TIMES
LONG BUILD TIMES
“We stopped using Travis CI because it was too slow for us.” (P125)
“We stopped using Travis CI because it was too slow for us.” (P125)

Longer build times associated with a decreased chance of switching and abandoning.
“We stopped using Travis CI because it was too slow for us.” (P125)

LONGER BUILD TIMES ASSOCIATED WITH A DECREASED CHANCE OF SWITCHING AND ABANDONING.

PRESENCE OF VERY LONG BUILDS ASSOCIATED WITH INCREASED RISK OF ABANDONING.
LONG BUILD TIMES
LONG BUILD TIMES

SO, WHEN DO LONG BUILDS STOP PROVIDING VALUE, AND START BECOMING ANNOYING?
A Conceptual Replication of Continuous Integration Pain Points in the Context of Travis CI

David Gray Widder
widder@cmu.edu
Carnegie Mellon

Michael Hilton
hilton@cmu.edu
Carnegie Mellon

Christian Kästner
kaestner@cmu.edu
Carnegie Mellon

Bogdan Vasilescu
vasilescu@cmu.edu
Carnegie Mellon

ABSTRACT
Continuous integration (CI) is an established software quality assurance practice, and the focus of much prior research with a diverse range of methods and populations. In this paper, we first conduct a literature review of 37 papers on CI pain points. We then conduct a conceptual replication study on 37 papers using a triangulation design consisting of a survey with 152 responses, 12 interviews, and two logistic regressions predicting Travis CI abandonment and switching on a dataset of 6,239 GitHub projects. We report and discuss which (if any) of the papers we were able to replicate, those for which we found conflicting evidence, those for which we did not find evidence, and the implications of these findings.

CCS CONCEPTS
• Software and its engineering → Software maintenance tools.

KEYWORDS
Continuous integration, open source software, replication

ACM Reference Format:

1 INTRODUCTION
Continuous integration (CI) has enjoyed tremendous popularity as a quality assurance mechanism during software development, by automating the execution of builds, tests, and other tasks. CI adoption is primarily driven by practices including DevOps, but research has shown that CI practices have a positive effect on software quality and productivity [20, 61, 75].

Despite the widespread adoption of CI, it has long been established by contingency theory [27, 60] that a single “universal best practice” is unlikely, whatever the actual practice. Moreover, for CI specifically, the literature abounds with studies (we counted 37 papers; Section 1) that each touch on some CI pain points. For example, research has shown that it can take significant effort to set up and customize CI infrastructure [27, 34], and the cost of CI systems requires effort to overcome and can cause unwanted interruptions [35], especially without developer buy-in and in the presence of frequent false positives from flaky tests and platform instabilities [27]. Rollbacks or fixes to issues with a specific CI tool can turn developers away from CI in a practice, even when more customizable tool solutions exist [36].

Our main results (Sec. 3) confirm our literature review, the fact that developers find troubleshooting build failures difficult, desire consistency in CI tools across their projects, that it is difficult to use CI with complex tool setups including Docker or to use CI with unsupported languages, find long build times annoying, and find CI less useful without enough tests.

In summary, we conclude (1) a literature review of general CI pain points; (2) an analysis of 152 survey responses about reasons for abandoning or switching Travis CI; (3) regression models on a dataset of 6,239 GitHub Travis CI projects; testing observations from literature; and (4) a discussion of results and implications.

2 STUDY DESIGN
What are the major pain points that turn people away from CI?
To answer this research question, we conduct a conceptual replication [26, 61, 69], i.e., we attempt to corroborate observations from past research using a different experimental design, on a different population. The importance of replication studies in software engineering is increasingly recognized. Our conceptual replication, as opposed to an exact replication, represents a more robust design.
LIT REVIEW
SURVEYS, INTERVIEWS

PAIN POINTS
2 LOGISTIC REGRESSIONS

TRIANGULATION

DAVIDTHEWID
We were unable to cleanly replicate some pain points, they need further study to validate their existence.
We were unable to cleanly replicate some pain points, they need further study to validate their existence. We were able to cleanly replicate some pain points, now we can focus on solutions.
we were unable to cleanly replicate some pain points, they need further study to validate their existence

we were able to cleanly replicate some pain points, now we can focus on solutions

future work should focus more than just TravisCI, and our commit status context method can help
We were unable to cleanly replicate some pain points, they need further study to validate their existence.

We were able to cleanly replicate some pain points, now we can focus on solutions.

Future work should focus more than just TravisCI, and our commit status context method can help.

CI consistency: Social ties impact a project's tooling choices.