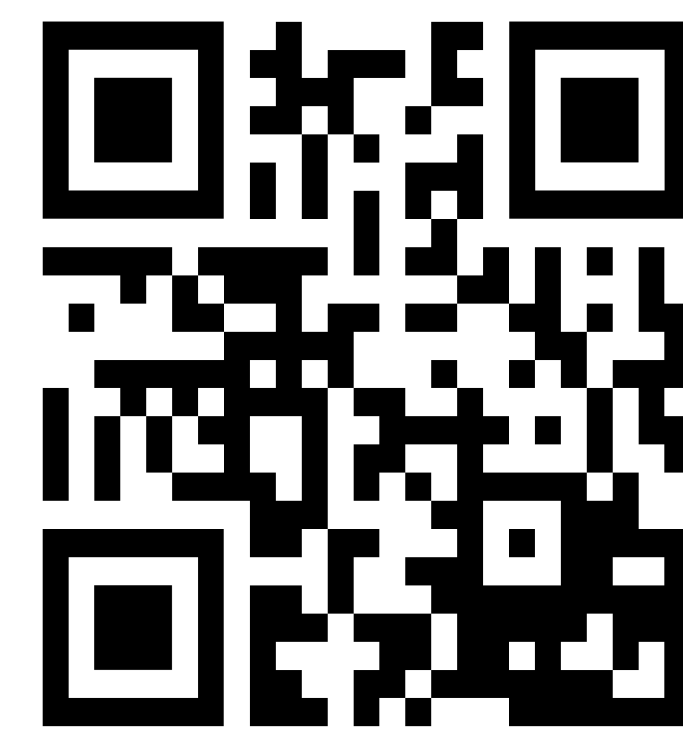


Inadequate Testing, Time Pressure, and (Over) Confidence: A Tale of Continuous Integration Users

Gustavo Pinto, Marcel Rebouças, Fernando Castor
gpinto@ufpa.br, {mscr,castor}@cin.ufpe.br



CI is a core agile practice.

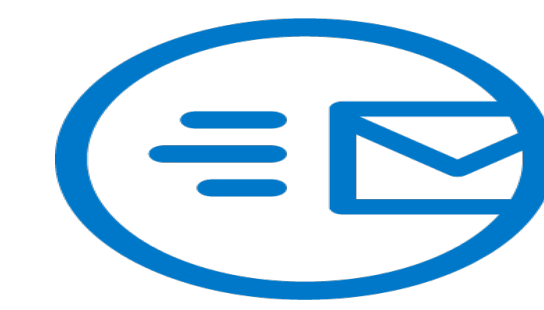
In a project that uses CI, errors can be detected quickly, their causes can be located with less effort, and the overall development process can be sped up.



A survey with CI users

We selected the top 50 most popular OSS projects written in the most used programming languages, hosted on Github, with Travis enabled.

666 projects: 737K builds (182K failed)



1,100 CI users
158 responses (14%)

Little is known about CI practice

RQ.

What are the **perceptions** from CI users in terms of **reasons** for build breakage, and the **benefits** and **problems** of CI usage?

Technical reasons for build breakage

Inadequate testing (33 occurrences)

“Badly written that fail with minor bugfixes”

“Not enough tests”

Dependency management (8 occurrences)

“people sometimes don’t update dependencies, so the CI server detects errors that do not happen locally”

Lack of domain knowledge (18 occurrences)

“unfamiliarity with the architecture of the code and overall module interactions”

Social reasons for build breakage

(self imposed) Time pressure (36 occurrences)

“eagerness to help”

Lack of testing culture (17 occurrences)

“people not running the tests and build on their machines before pushing the changes”

Overconfidence (11 occurrences)

*“the change is alright” “this is a trivial change”
“this is only a small fix, it should not break anything”*

Lack of communication (6 occurrences)

“not knowing who to ask for help”

Benefits

Catch problems earlier (32 occ.)

“Being aware of when/where breakage occurs greatly accelerates solution”

cross-platform testing (16 occ.)

“compiling for multiple different targets (x86,arm,window,linux etc)”

Confidence (10 occurrences)

“some sort of confidence that introduced changes don’t break the current behavior”

Problems

Configuring the build env. (32 occ.)

“some sort of confidence that introduced changes don’t break the current behavior”

False sense of confidence (25 occ.)

A passing build can encourage a reviewer to merge code without a thorough review.

Monetary costs (7 occurrences)

“Services like travis are not free if you want to use them at bigger scale”