A ALL OBSERVATIONS

For completeness and to assist future work, we list all observations from literature here.

A.1 Tested Observations

O1.a People find it difficult to troubleshoot a CI build failure.
O2.e Projects which change infrequently are less likely to need to use CI.
O2.f Projects with a rigorous testing culture are less likely to use CI.
O2.n Some company supported projects are reluctant to use a public CI service
O3.a Projects which attempt to use CI often run into configuration challenges, especially when: converting an old build environment to use CI; when testing third party components; or when attempting to self-host their CI.
O3.b CI configuration files can be confusing, leading some to prefer a GUI.
O3.c Projects which use many other developer tools find it difficult to connect these to CI.
O3.d GitHub projects which attempt to use a CI service which does not support the project's programming language run into challenges.
O4.a People do not like long CI builds, which prohibit quick feedback.
O4.b Projects which use CI are likely to see an increase in pull request latency, because of CI's associated long build times.
O4.c Projects with long CI build times are more likely to have an error prone CI configuration.
O4.d In data intensive domains, CI becomes expensive at best or impractial at worst due to intense hardware needs.
O5.a Projects with few or no tests see less benefit from CI.
O5.d Projects with flaky tests make CI less useful, often causing people to trust CI's build results less.

A.2 Non-Operationalized Observations

O1.b It is difficult to understand the overall state of a large project by looking at CI results, especially when the project is split into sub projects.
O1.c People report a need for better notifications fromCI, especially to provide relevant information to different stakeholders.
O1.d Some suggest that CI can contribute to performance creep, because small frequent builds make performance changes harder to notice.
O2.b CI is seen as requiring a certain "culture", which may be at odds with existing institutional culture.
O2.c Without buy in from leaders, CI adopting is unlikely to be successful and its use is less likely to be helpful.
O2.i CI is sometimes seen as an unreachable standard, especially in a evolving organization.
O2.j Some developers fear damage to their reputation when breaking the CI build, making them less willing to expose early versions of their work.
O2.k Newcomers to a project often don't understand what CI is, or the way it is used on the specific project they are joining.
O3.e CI users dislike when tools force a specific workflow.
O5.b Projects which are primarily GUI code, or other cases in which automatic test cases are hard to write, see less benefit from CI.
O5.c People don't like writing tests, find it difficult or time consuming, or not a part of their team's norms, and thus CI is less useful due to lack of tests.

A.3 Out of Scope Observations

O2.a It is hard to change habits in large companies, and since CI requires changing many such habits, adopting CI is hard.
O2.d CI can have a negative effect on company culture, including by creating too many meetings.
O2.g Failure to agree about the goals for adopting CI between a client and a company is a barrier to effective CI use.
O2.h Uneven CI knowledge prevents teams from using CI effectively.
O2.l Because CI is yet another skill that prospective hires must have, CI use is seen as making it harder to hire developers with requisite skills.
O2.m Projects in health care or similar regulation-heavy spaces are less likely to use CI.
O5.e Difficulties recreating the production environment also diminish the level of trust in CI build results.

B INTERVIEW PROTOCOL

For replicability, we list the questions used in our semi-structured interviews here.

Basic Questions

(1) How would you describe your role in the project?
(2) Why did your project start using Continuous Integration?
(3) Why did your project start using TravisCI specifically?
(4) Why did your project stop using TravisCI? Were there specific problems?
(5) Did you adopt a different CI system? If so, which system, and what factors caused you to choose it specifically? If not, what has it been like without CI?
(6) (if yes to above: continue, else: skip to final question)
(7) On a scale of 1 to 5 how happy are you with new system?
(8) (if less than 4) What problems are there with new system? Did these surprise you?
(9) Are you considering switching CI again?
(10) (if 4 more or more) What have been the biggest advantages? Did these surprise you?

Community Questions

(1) Tell me about the decision to change CI tools: How many people were involved in the discussion to start using TravisCI, to stop using it, and to adopt the new system?
(2) What did these discussions look like? Was there a lot of discussion? Any disagreements?
(3) Was there a learning curve for the new CI system for you, or for other contributors who were not involved in the switch?
(4) Is there agreement from other contributors that the new CI system is good or bad?
(5) Anything else you think I should know?