The Impact of Continuous Integration on Other Software Development Practices: A Large-Scale Empirical Study

Yangyang Zhao
Nanjing U

Alexander Serebrenik
TU Eindhoven

Yuming Zhou
Nanjing U

Vladimir Filkov
DECAL at UC Davis

Bogdan Vasilescu
STRUDEL at CMU

@aserebrenik
@vfilkov
@b_vasilescu
Interventions are common in software engineering

- SVN $\rightarrow$ git
- push $\rightarrow$ pull request
- ? $\rightarrow$ continuous integration
- ...

How to measure effects using trace data?
Evaluating the effects of an intervention: *before vs. after*
Evaluating the effects of an intervention: *before vs. after*
Evaluating the effects of an intervention: *before vs. after*
Evaluating the effects of an intervention: before vs. after
Today

Methodology to empirically study the effects of an intervention (continuous integration)
Interrupted time series

Multiple regression w/ controls for confounds

change in level

slope before

slope after

Experimental and Quasi-Experimental Designs for Generalized Causal Inference
### Graphical Representation

**Slope Before** and **Slope After**

**Change in Level**

**Time before Intervention:**

- 1
- 2
- 3
- ...
- 100
- 101
- 102
- ...
- 200

**Time after Intervention:**

- 0
- 0
- 0
- ...
- 0
- 1
- 2
- ...
- 100

**Intervention:**

- F
- F
- F
- ...
- T
- T
- T
- ...
- T
\[ y_i = \alpha + \beta \cdot \text{time}_i + \gamma \cdot \text{intervention}_i + \delta \cdot \text{time\_after\_intervention}_i + \varepsilon_i \]
\[ y_i = \beta \cdot \text{time}_i + \gamma \cdot \text{intervention}_i + \delta \cdot \text{time\_after\_intervention}_i + \varepsilon_i \]

- \( \beta \sim 1 \)
- \( \gamma \sim -50 \)
- \( \beta + \delta \sim -0.5 \)

<table>
<thead>
<tr>
<th>Dependent variable:</th>
<th>( y )</th>
</tr>
</thead>
<tbody>
<tr>
<td>time</td>
<td>0.991***</td>
</tr>
<tr>
<td>intervention</td>
<td>-48.678***</td>
</tr>
<tr>
<td>time_after_intervention</td>
<td>-1.500***</td>
</tr>
<tr>
<td>Constant</td>
<td>1.007</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Observations</th>
<th>200</th>
</tr>
</thead>
<tbody>
<tr>
<td>R^2</td>
<td>0.967</td>
</tr>
<tr>
<td>Adjusted R^2</td>
<td>0.967</td>
</tr>
<tr>
<td>Residual Std. Error</td>
<td>4.844 (df = 196)</td>
</tr>
<tr>
<td>F Statistic</td>
<td>1,924.910*** (df = 3; 196)</td>
</tr>
</tbody>
</table>

Note: *p<0.1; **p<0.05; ***p<0.01
Effects of adopting Travis CI
Why CI?

Lots of folklore, e.g., Martin Fowler:

- Everyone Commits To the Mainline Every Day
- Fix Broken Builds Immediately
- Keep the Build Fast
- ...

https://martinfowler.com/articles/originalContinuousIntegration.html
Adoption of Travis CI

Travis CI adoption (first .travis.yml commit) 

Unstable period excluded

Starting sample: 
165,549 GitHub projects using Travis

24 active periods 
 x 
7 programming languages
More frequent commits

RQs

Smaller code changes

Impact on automated testing?

More issues and pull requests closed

Quick pull requests resolution
## Churn

<table>
<thead>
<tr>
<th></th>
<th>Churn in non-merge commits</th>
<th>Churn in merge commits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept ((\alpha))</td>
<td>1.336***</td>
<td>-1.297**</td>
</tr>
<tr>
<td>log(TotalCommits)</td>
<td>0.529**</td>
<td>1.113**</td>
</tr>
<tr>
<td>AgeAtTravis</td>
<td>-0.003*</td>
<td>-0.005**</td>
</tr>
<tr>
<td>log(NumAuthors)</td>
<td>-0.233**</td>
<td>-0.522**</td>
</tr>
<tr>
<td>time ((\beta))</td>
<td>-0.007</td>
<td>-0.012*</td>
</tr>
<tr>
<td>interventionTrue ((\gamma))</td>
<td>0.071</td>
<td>0.220**</td>
</tr>
<tr>
<td>time_after_intervention ((\delta))</td>
<td>-0.009</td>
<td>-0.022**</td>
</tr>
</tbody>
</table>
## Churn

<table>
<thead>
<tr>
<th>Control variables</th>
<th>Non-merge commits</th>
<th>Merge commits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept ($\alpha$)</td>
<td>1,336***</td>
<td>-1,297**</td>
</tr>
<tr>
<td>log(TotalCommits)</td>
<td>0,529**</td>
<td>1,113**</td>
</tr>
<tr>
<td>AgeAtTravis</td>
<td>0,432</td>
<td>-0,005**</td>
</tr>
<tr>
<td>log(NumAuthors)</td>
<td>-0,233**</td>
<td>-0,522**</td>
</tr>
<tr>
<td>time ($\beta$)</td>
<td>-0,007</td>
<td>-0,012*</td>
</tr>
<tr>
<td>interventionTrue ($\gamma$)</td>
<td>0,071</td>
<td>0,220**</td>
</tr>
<tr>
<td>time_after_intervention ($\delta$)</td>
<td>-0,009</td>
<td>-0,022**</td>
</tr>
</tbody>
</table>
Churn in non-merge commits is not affected by time or Travis CI.
Churn

Churn in non-merge commits

<table>
<thead>
<tr>
<th></th>
<th>time ($\beta$)</th>
<th>interventionTrue ($\gamma$)</th>
<th>time_after_intervention ($\delta$)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-0.007</td>
<td>n.s.</td>
<td>-0.009</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>-0.022**</td>
</tr>
</tbody>
</table>

Churn in **non-merge commits** is not affected by time or Travis CI.
### Churn

<table>
<thead>
<tr>
<th>Variable</th>
<th>Churn in non-merge commits</th>
<th>Churn in merge commits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept ($\alpha$)</td>
<td>1,336***</td>
<td>-1,297**</td>
</tr>
<tr>
<td>log(TotalCommits)</td>
<td>0,529**</td>
<td>1,113**</td>
</tr>
<tr>
<td>AgeAtTravis</td>
<td>0,000*</td>
<td>-0,005**</td>
</tr>
<tr>
<td>log(NumAuthors)</td>
<td>-0,233**</td>
<td>-0,522**</td>
</tr>
<tr>
<td>time ($\beta$)</td>
<td>-0,007</td>
<td>-0,012*</td>
</tr>
<tr>
<td>interventionTrue ($\gamma$)</td>
<td>n.s.</td>
<td>0,220**</td>
</tr>
<tr>
<td>time_after_intervention ($\delta$)</td>
<td>-0,009</td>
<td>-0,022**</td>
</tr>
</tbody>
</table>

**Control variables**

Churn in **non-merge commits** is **not affected** by time or Travis CI.

**Discontinuity in merge com.**: preparation for transition, clean-up.
Churn in non-merge commits is not affected by time or Travis CI. Discontinuity in merge commits: preparation for transition, clean-up. Decrease in churn in merge commits is amplified by Travis CI.
Churn in non-merge commits is not affected by time or Travis CI

Discontinuity in merge com.: preparation for transition, clean-up

Decrease in churn in merge commits is amplified by Travis CI
Triangulation: user survey

introduced Travis to their projects
Discontinuity in merge commits: preparation for transition, clean-up

Decrease in churn in merge commits is amplified by Travis CI

R25: “contributors couldn’t be trusted to run test suite on their own”

R38: Travis as “a part of automated package/release effort”

R4: “commits became smaller and more frequent, to check the build; pull requests became easier to check”
Closed PRs

Among others:

- On average, more PRs are being closed per unit time after adopting Travis CI.
Closed PRs

Increasing trend **only before** adopting Travis CI
**RQs**

- **Impact on automated testing?**
  - More frequent commits
  - Smaller code changes
  - Increasing trend slowed down

- **Both before and after Travis**
  - More issues and pull requests closed
  - Quick pull requests resolution

- **Affected only for merge commits**
  - ↓ missing files/dep
  - ↑ comp/exec errors
  - ↑ failed tests

- **# increases pre-Travis, flattened out by Travis**
  - duration not affected

- **Impact on automated testing?**
  - Increasing trend slowed down

- **Quick pull requests resolution**
Interrupted time series

\[ y_i = \alpha + \beta \cdot \text{time}_i + \gamma \cdot \text{intervention}_i + \delta \cdot \text{time}\text{\_after\text{\_intervention}}_i + \epsilon_i \]

Yangyang Zhao
Alexander Serebrenik
Yuming Zhou
Vladimir Filkov
Bogdan Vasilescu