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

Yangyang Zhao  Alexander Serebrenik  Yuming Zhou  Vladimir Filkov  Bogdan Vasilescu
Nanjing U  TU Eindhoven  Nanjing U  DECAL at UC Davis  STRUDEL at CMU
@aserebrenik  @vfilkov  @b_vasilescu
Happy Halloween!
Interventions are common in software engineering

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

Interventions are common in software engineering

- SVN —> git
- push —> pull request
- ? —> continuous integration
- ...

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

(change in slope)
Evaluating the effects of an intervention: before vs. after

change in slope

t-test no difference

before 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

slope before

change in level

slope after
Interrupted time series

Multiple regression w/ controls for confounds
slope before

change in level

slope after

time:

1  2  3 … … …  100  101  102 … … …  200
time:  

1  2  3 … … …  100  101  102 … … … 200  

time after intervention:  

0  0  0 … … …  0  1  2 … … … 100
time: 1 2 3 … … … 100 101 102 … … … 200

intervention: F F F … … … T T T … … … T

time after intervention: 0 0 0 … … … 0 1 2 … … … 100

change in level

slope before

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

- $\beta \sim 1$
- $\gamma \sim -50$
- $\beta + \delta \sim -0.5$

**Dependent variable:**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$y$</td>
<td></td>
</tr>
<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>
<tr>
<td>Observations</td>
<td>200</td>
</tr>
<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

-375 days
-45 days
-15 days
+15 days
+45 days
+375 days

-12

...
Adoption of Travis CI

Travis CI adoption (first `.travis.yml` commit)

Unstable period excluded

Starting sample:
165,549 GitHub projects using Travis
Adoption of Travis CI

Travis CI adoption
(first .travis.yml commit)

Starting sample:
165,549 GitHub projects using Travis

24 active periods
x
7 programming languages
More frequent commits

Smaller code changes

RQs

Impact on automated testing?

More issues and pull requests closed

Quick pull requests resolution
## 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 (α)</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 (β)</td>
<td>-0,007</td>
<td>-0,012*</td>
</tr>
<tr>
<td>interventionTrue (γ)</td>
<td>0,071</td>
<td>0,220**</td>
</tr>
<tr>
<td>time_after_intervention (δ)</td>
<td>-0,009</td>
<td>-0,022**</td>
</tr>
<tr>
<td></td>
<td>Churn in non-merge commits</td>
<td>Churn in merge commits</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>----------------------------</td>
<td>------------------------</td>
</tr>
<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>0,122</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 in non-merge commits is not affected by time or Travis CI.
Churn in non-merge commits is **not affected** by time or Travis CI.

Discontinuity in merge commits: preparation for transition, clean-up.
## 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>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
- 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
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

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

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

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

Increasing trend only before adopting Travis CI

Number

Latency

Mean PR latency
Impact on automated testing?

- **↓ missing files/dep**
- **↑ comp/exec errors**
- **↑ failed tests**

**RQs**

- More frequent commits
- Smaller code changes
- Increasing trend slowed down
- # increases pre-Travis, flattened out by Travis
- Duration not affected
- More issues and pull requests closed
- Quick pull requests resolution

Both before and after Travis

Affected only for merge commits
Interrupted time series

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

Yangyang Zhao
Alexander Serebrenik
Yuming Zhou
Vladimir Filkov
Bogdan Vasilescu