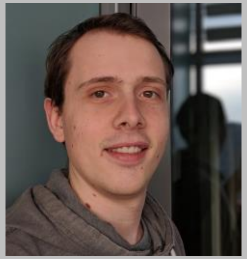
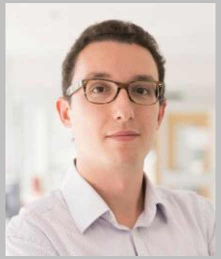
The Silent Helper: The Impact of Continuous Integration on Code Reviews



Nathan Cassee

Eindhoven University of Technology, Netherlands

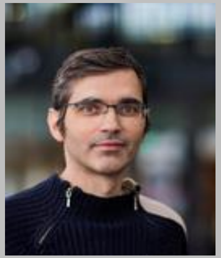
@NathanCassee



Bogdan Vasilescu

Carnegie Mellon University, USA

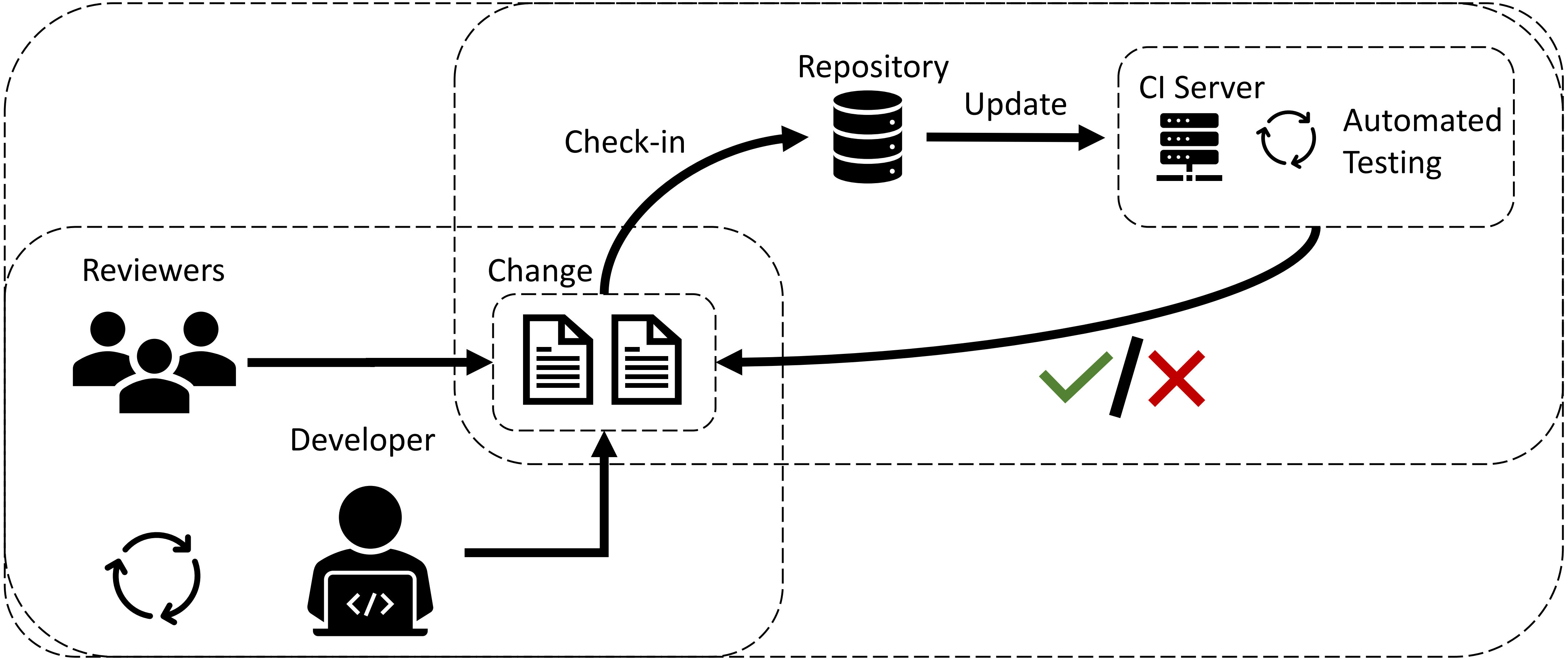
@b_vasilescu



Alexander Serebrenik

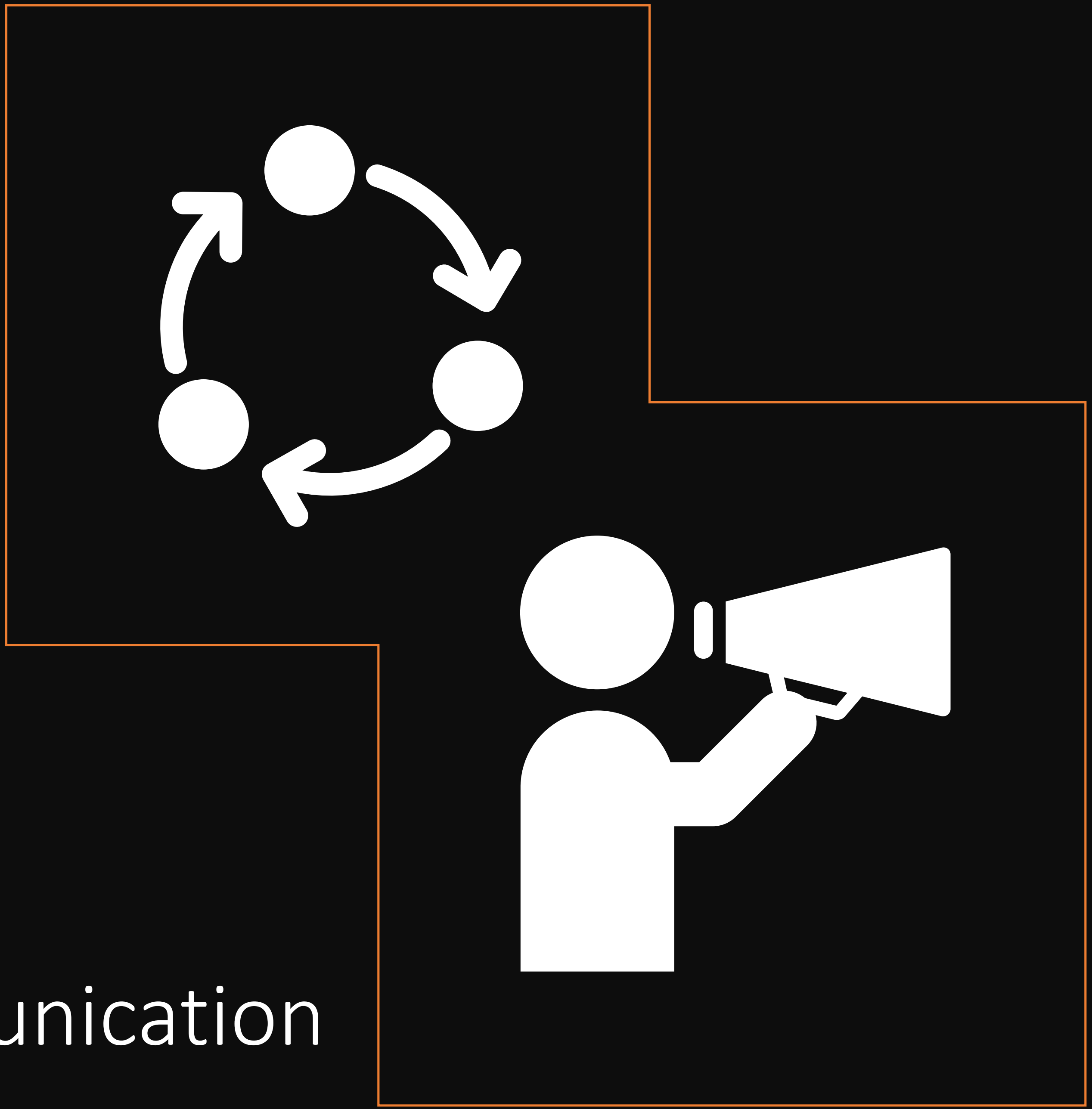
Eindhoven University of Technology, Netherlands

@aserebrenik



Continuous Integration and Code Reviews

Research questions



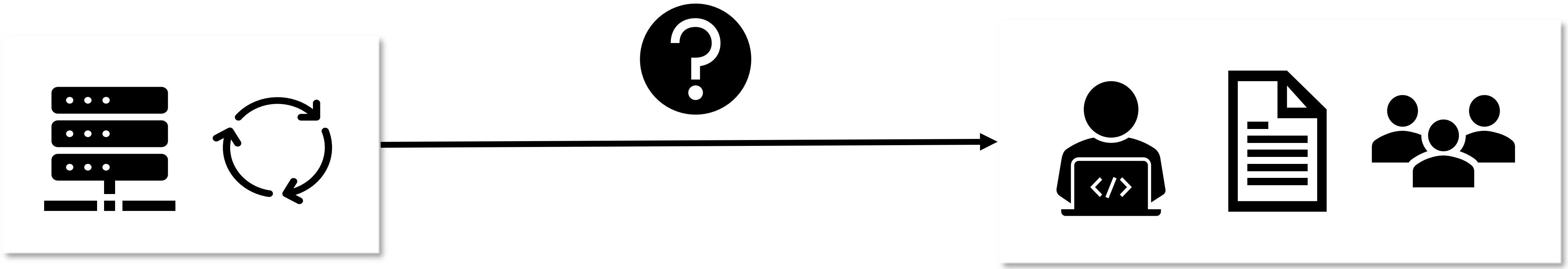
Code Changes

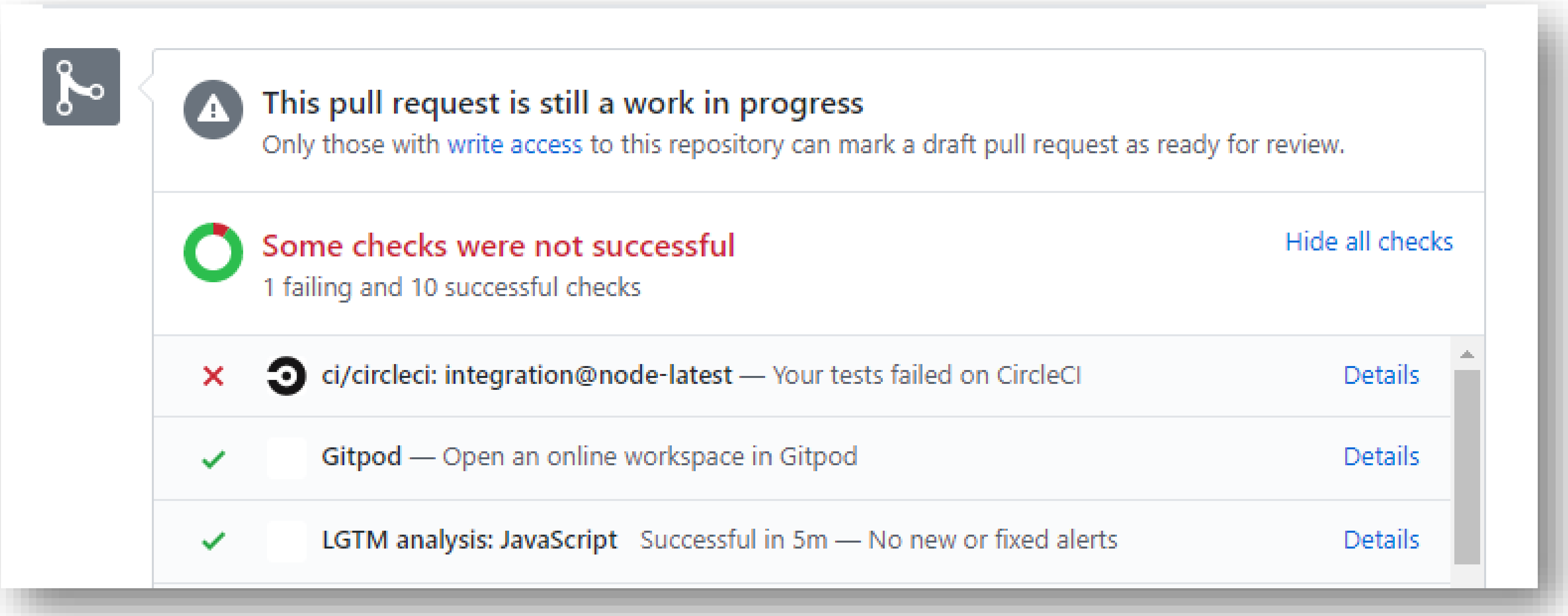
Communication

Take away Message

Continuous Integration fulfills the role of a **Silent Helper** during Code Reviews, leading to a **33% drop** in code review **comments**, without a similar drop in code modifications a year after the adoption of Continuous Integration.

How did we get here?





This pull request is still a work in progress

Only those with write access to this repository can mark a draft pull request as ready for review.

Some checks were not successful

1 failing and 10 successful checks

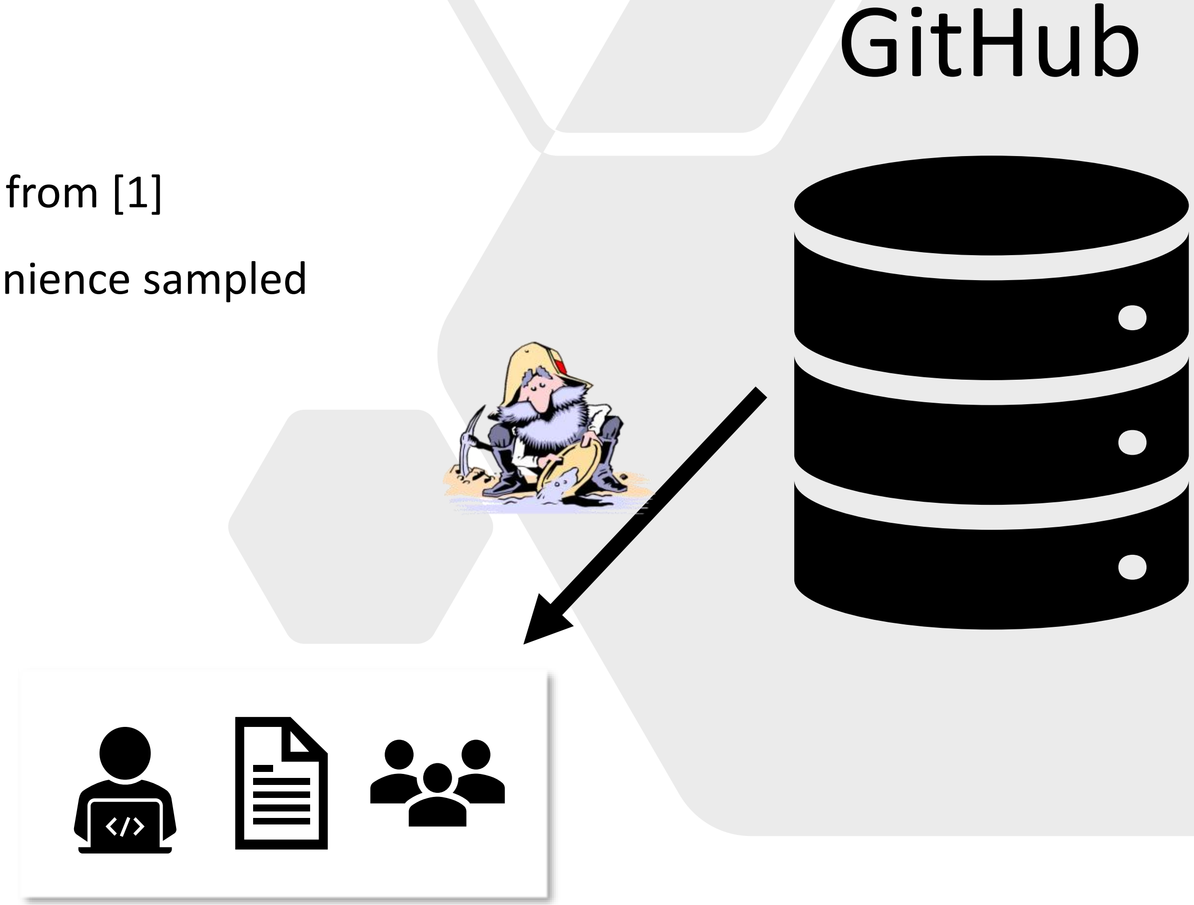
✗ ci/circleci: integration@node-latest — Your tests failed on CircleCI

✓ Gitpod — Open an online workspace in Gitpod

✓ LGTM analysis: JavaScript Successful in 5m — No new or fixed alerts

Data collection

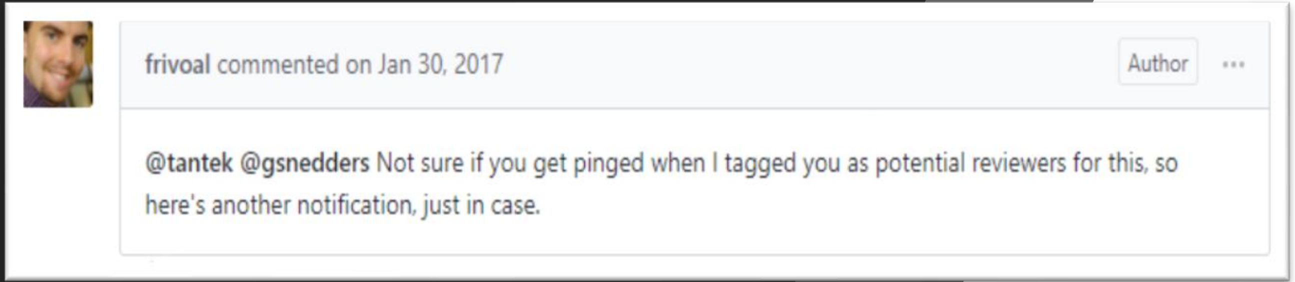
- 685 Projects
  - 50% taken from [1]
  - 50% convenience sampled
- 1.2 million Code Reviews
- 5 million Commits



[1]: Y. Zhao, A. Serebrenik, Y. Zhou, V. Filkov, and B. Vasilescu, “The impact of continuous integration on other software development practices: A large-scale empirical study,” in ASE. IEEE, 2017

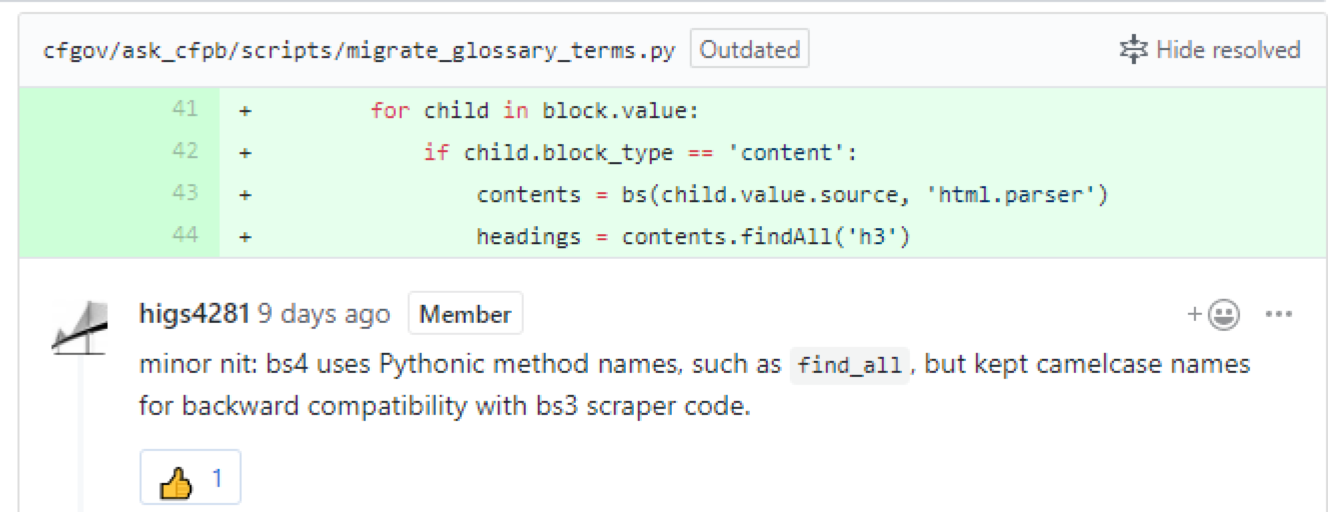
Operationalizations:

Communication



frivoal commented on Jan 30, 2017

@tantek @gsnedders Not sure if you get pinged when I tagged you as potential reviewers for this, so here's another notification, just in case.



cfgov/ask_cfpb/scripts/migrate_glossary_terms.py

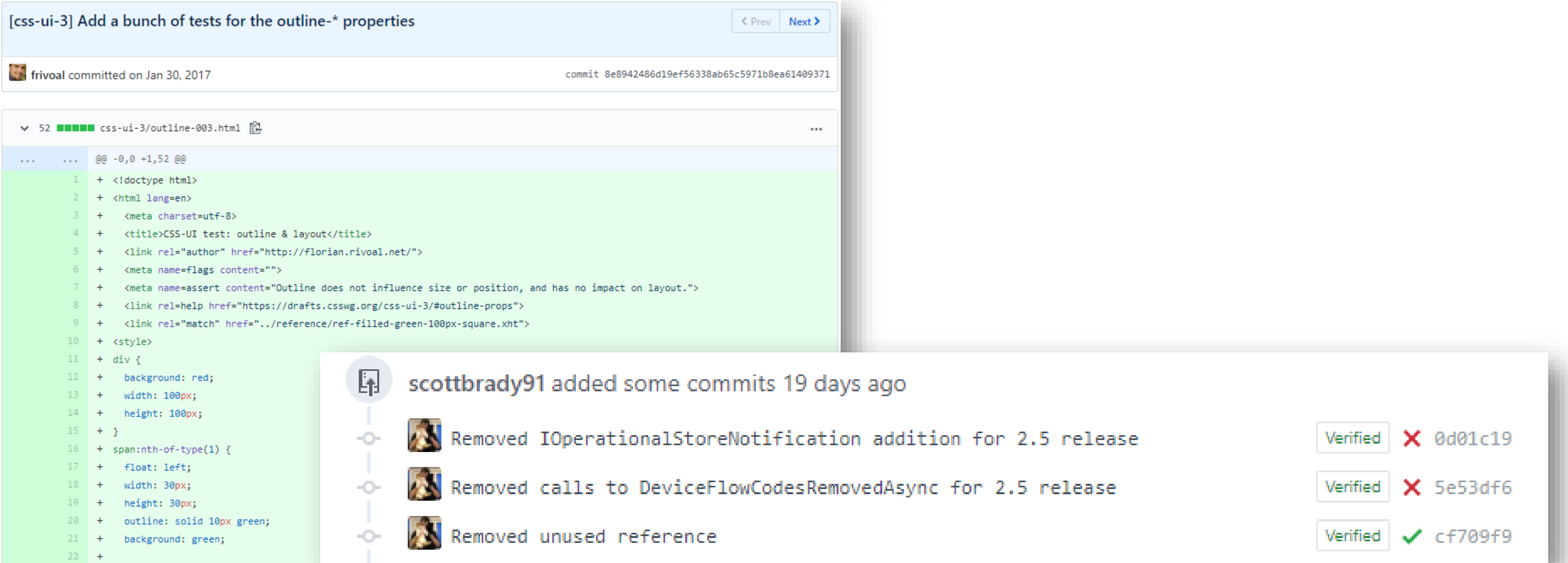
```
+           for child in block.value:
+               if child.block_type == 'content':
+                   contents = bs(child.value.source, 'html.parser')
+                   headings = contents.findAll('h3')
```

higs4281 9 days ago

minor nit: bs4 uses Pythonic method names, such as `find_all`, but kept camelcase names for backward compatibility with bs3 scraper code.

Operationalizations:

Commits



[css-ui-3] Add a bunch of tests for the outline-* properties

< Prev | Next >

frivoal committed on Jan 30, 2017

commit 8e8942486d19ef56338ab65c5971b8ea61409371

52 css-ui-3/outline-003.html

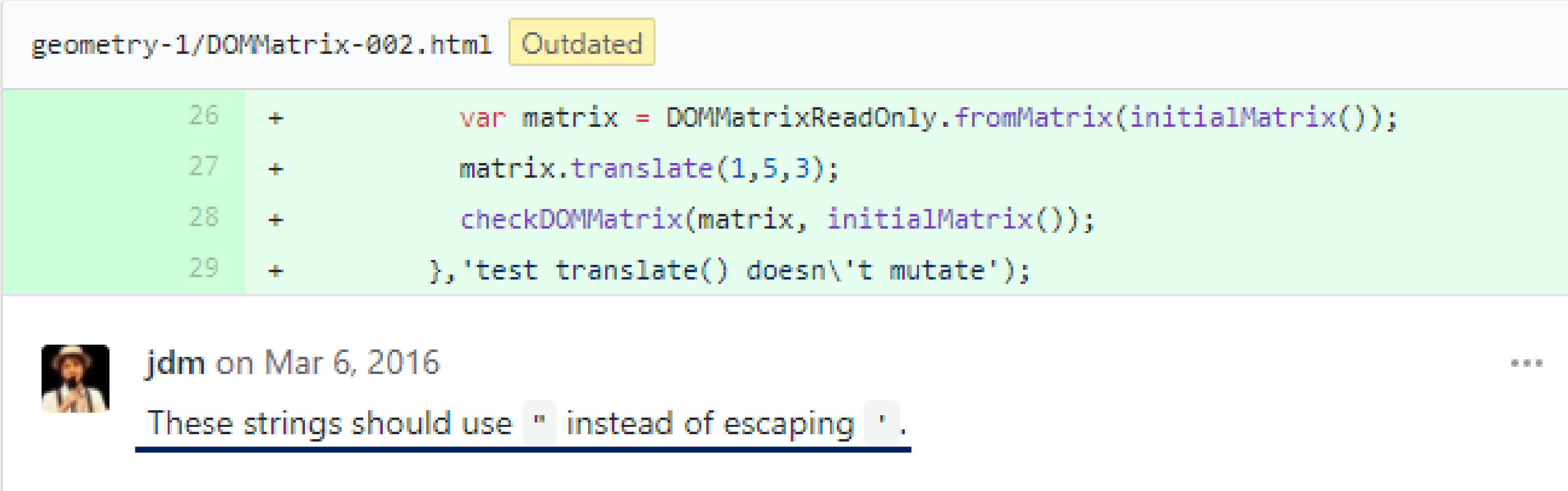
```
@@ -0,0 +1,52 @@
+ <!doctype html>
+ <html lang=en>
+   <meta charset=utf-8>
+   <title>CSS-UI test: outline & layout</title>
+   <link rel="author" href="http://florian.rivoal.net/">
+   <meta name=flags content="">
+   <meta name=assert content="Outline does not influence size or position, and has no impact on layout.">
+   <link rel=help href="https://drafts.csswg.org/css-ui-3/#outline-props">
+   <link rel="match" href="../reference/ref-filled-green-100px-square.xht">
+ <style>
+ div {
+   background: red;
+   width: 100px;
+   height: 100px;
+ }
+ span:nth-of-type(1) {
+   float: left;
+   width: 30px;
+   height: 30px;
+   outline: solid 10px green;
+   background: green;
+
```

scottbrady91 added some commits 19 days ago

- Removed IOperationalStoreNotification addition for 2.5 release — Verified ✗ 0d01c19
- Removed calls to DeviceFlowCodesRemovedAsync for 2.5 release — Verified ✗ 5e53df6
- Removed unused reference — Verified ✓ cf709f9

Operationalizations:

Effective comments

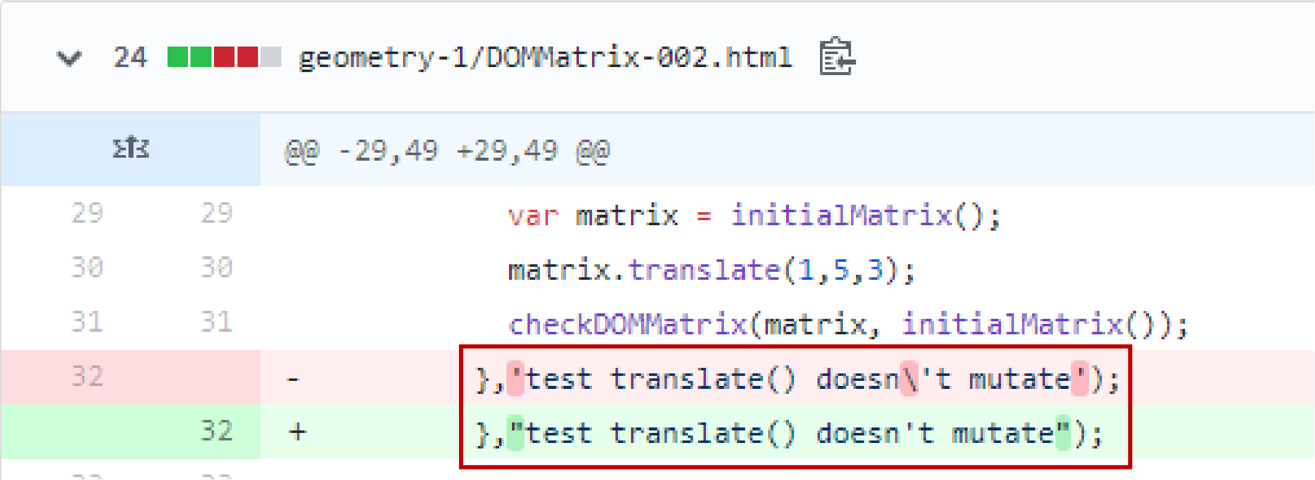


geometry-1/DOMMatrix-002.html Outdated

```
+        var matrix = DOMMatrixReadOnly.fromMatrix(initialMatrix());
+        matrix.translate(1,5,3);
+        checkDOMMatrix(matrix, initialMatrix());
+      },'test translate() doesn\'t mutate');
```

jdm on Mar 6, 2016

These strings should use " instead of escaping '.



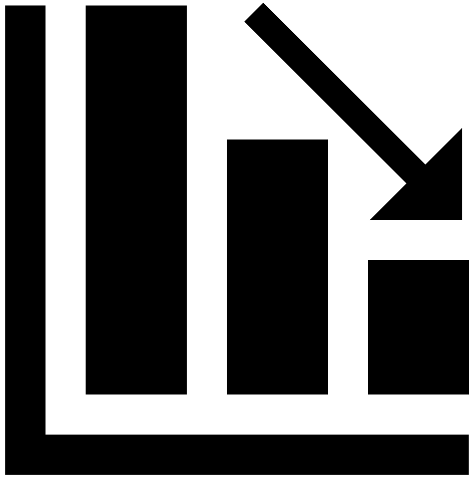
24 🟩🟩🟥🟥 geometry-1/DOMMatrix-002.html

```
@@ -29,49 +29,49 @@
        var matrix = initialMatrix();
        matrix.translate(1,5,3);
        checkDOMMatrix(matrix, initialMatrix());
-      },'test translate() doesn\'t mutate');
+      },"test translate() doesn't mutate");
```

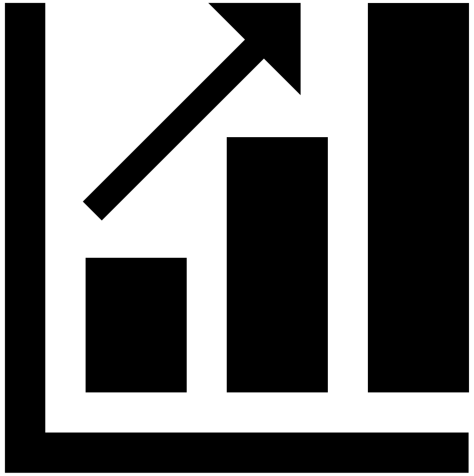
A. Bosu, M. Greiler, and C. Bird, “Characteristics of useful code reviews: An empirical study at Microsoft,” MSR, 2015

Data Analysis:

Regression Discontinuity Design

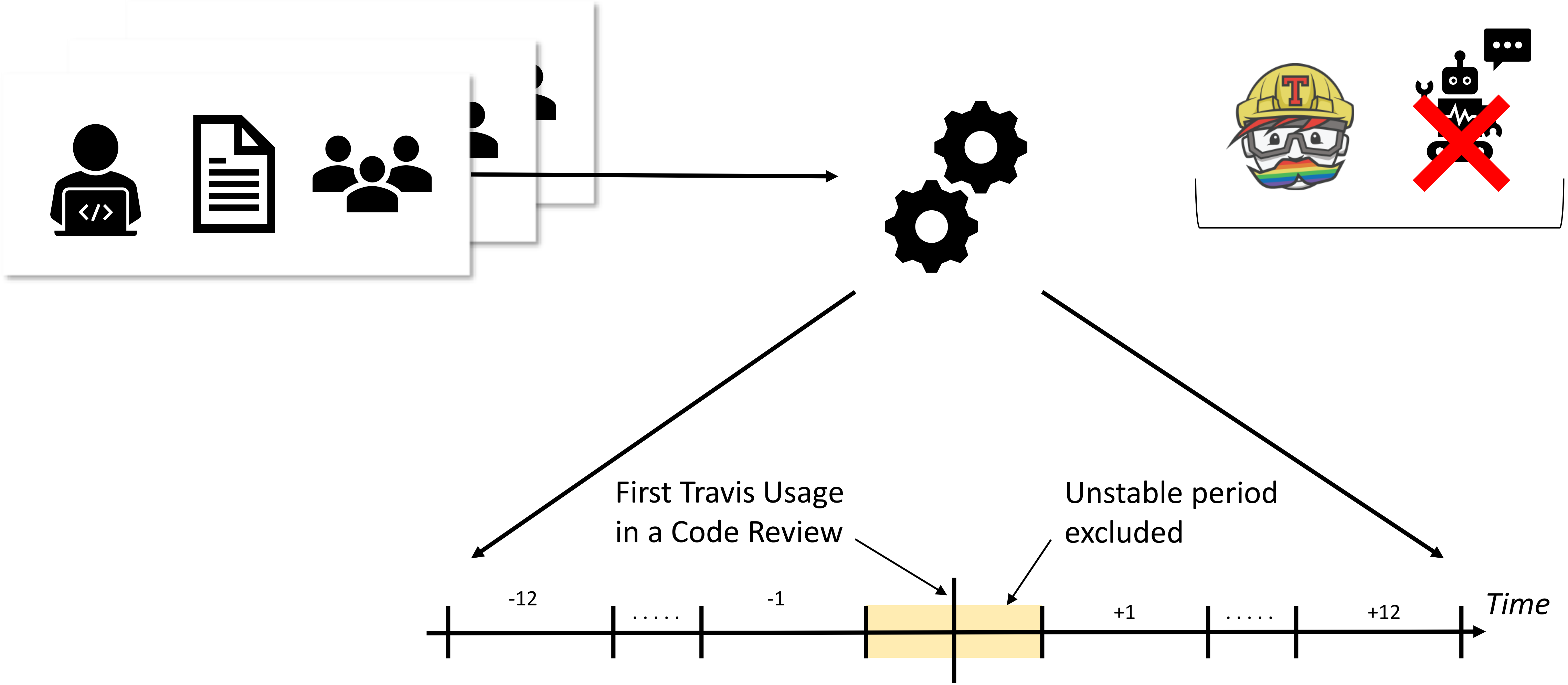


Before intervention



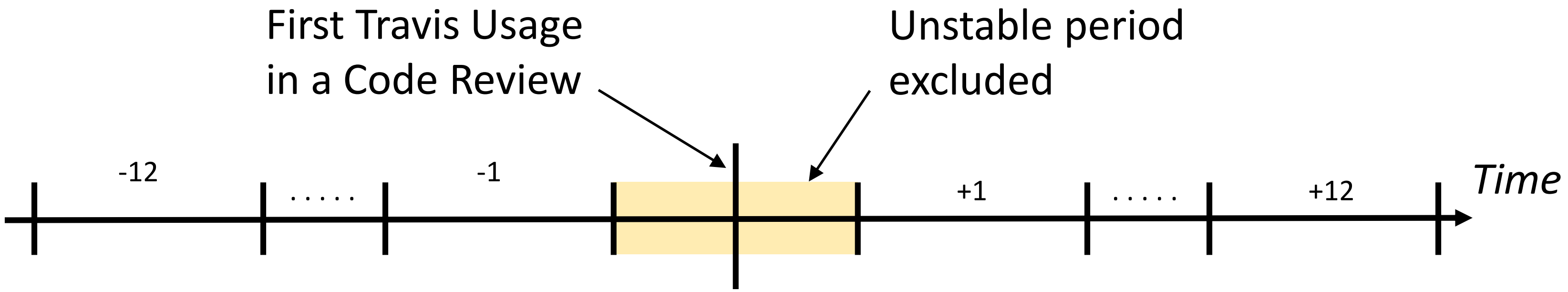
After intervention

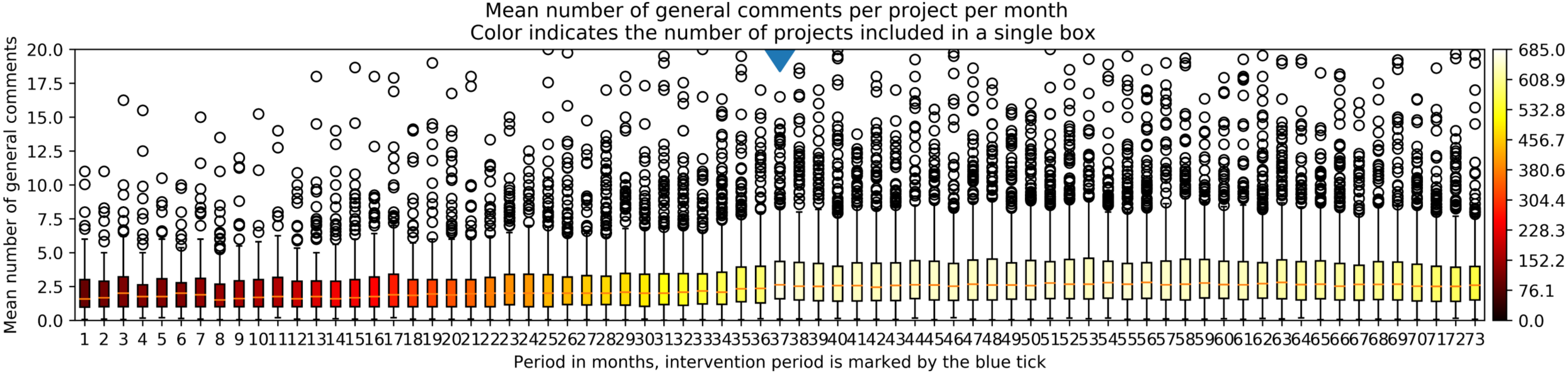
From 1.2 million code review to 685 time series



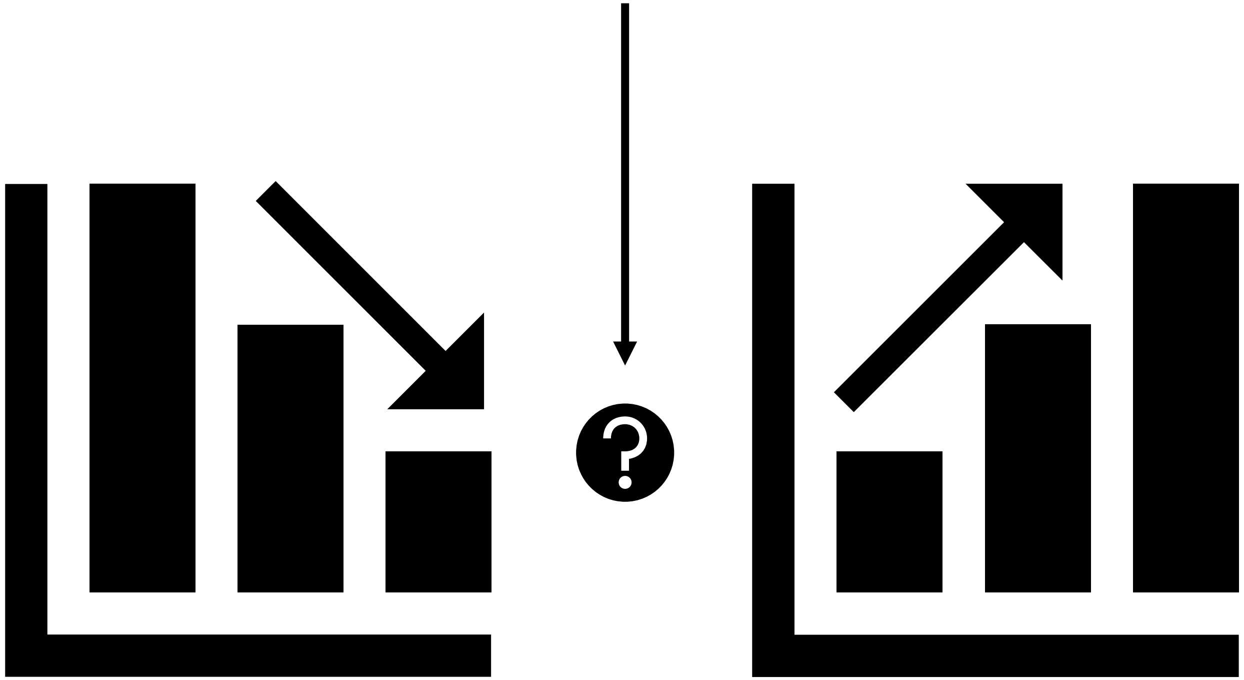
First Travis Usage in a Code Review

Unstable period excluded


First Travis Usage
in a Code Review
Unstable period
excluded
-12
.....
-1
+1
.....
+12
Time


Mean number of general comments per project per month
Color indicates the number of projects included in a single box
Mean number of general comments
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
Period in months, intervention period is marked by the blue tick
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73
685.0
608.9
532.8
456.7
380.6
304.4
228.3
152.2
76.1
0.0

Regression Discontinuity Design



- Response variables: *Review Comments, General Comments, Effective Comments, Commits After Create.*

- Accepted Controls: *Lines added, Reviewers involved*

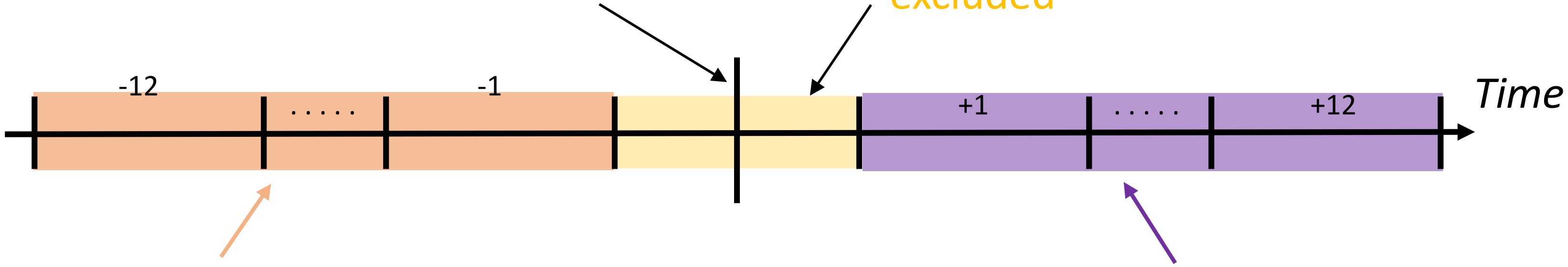
- Influence of Continuous Integration on response variables (**Time series**)

- **Account for per project differences**

Interpreting RDD models

Intervention
(Continuous Integration)

Unstable period
excluded

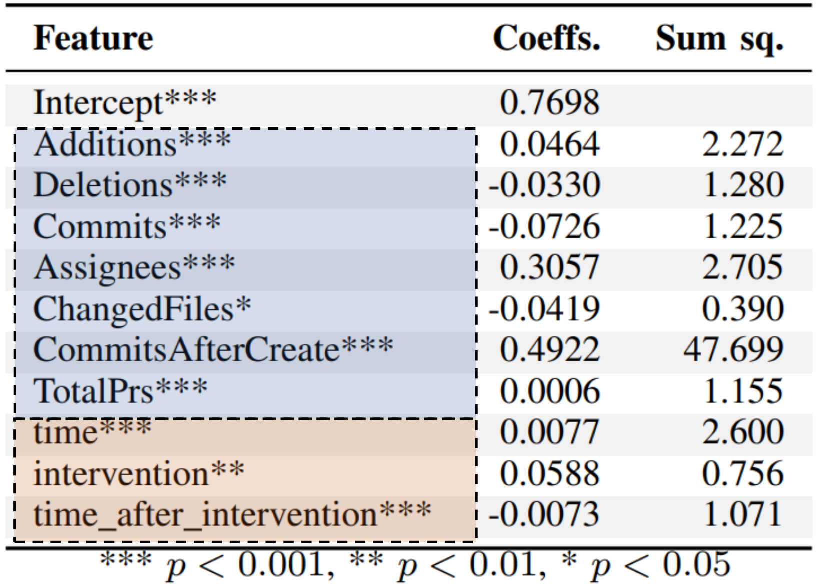


Trend before intervention

Trend after intervention

Trend: 📈 \ 📉

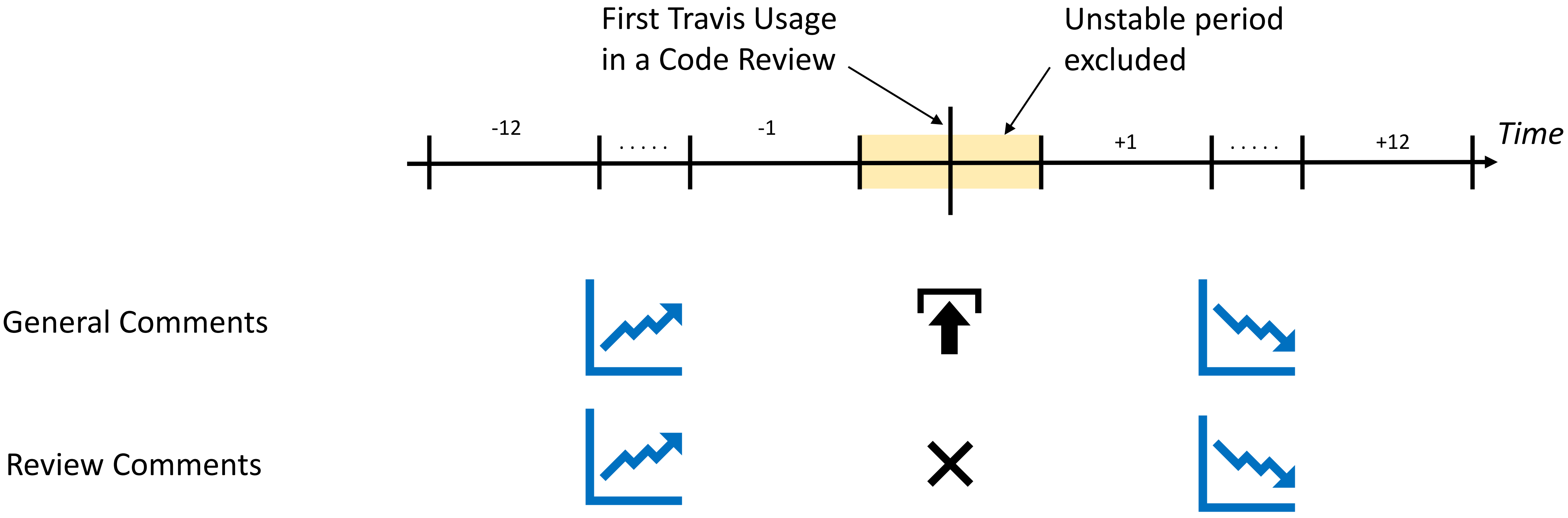
Intervention: ⬆ \ ⬇



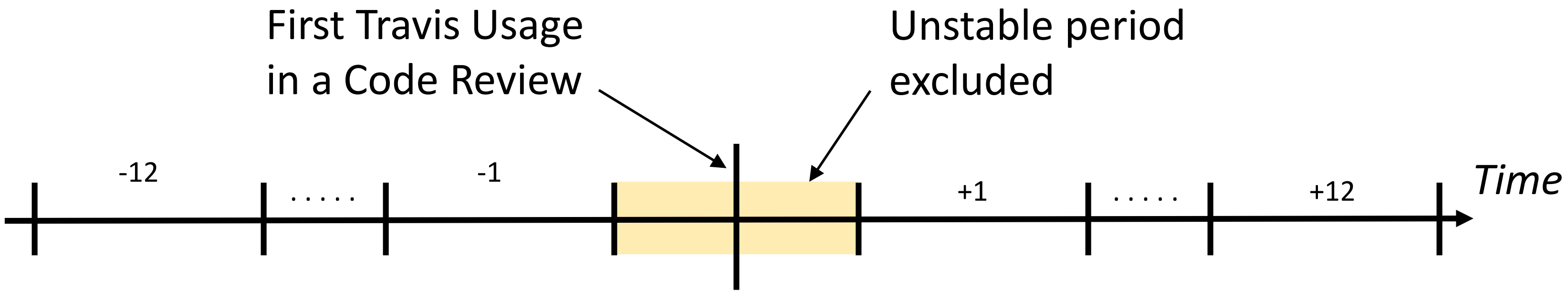
Feature	Coeffs.	Sum sq.
Intercept***	0.7698	
Additions***	0.0464	2.272
Deletions***	-0.0330	1.280
Commits***	-0.0726	1.225
Assignees***	0.3057	2.705
ChangedFiles*	-0.0419	0.390
CommitsAfterCreate***	0.4922	47.699
TotalPrs***	0.0006	1.155
time***	0.0077	2.600
intervention**	0.0588	0.756
time_after_intervention***	-0.0073	1.071

*** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$

When reviewing, developers communicate less

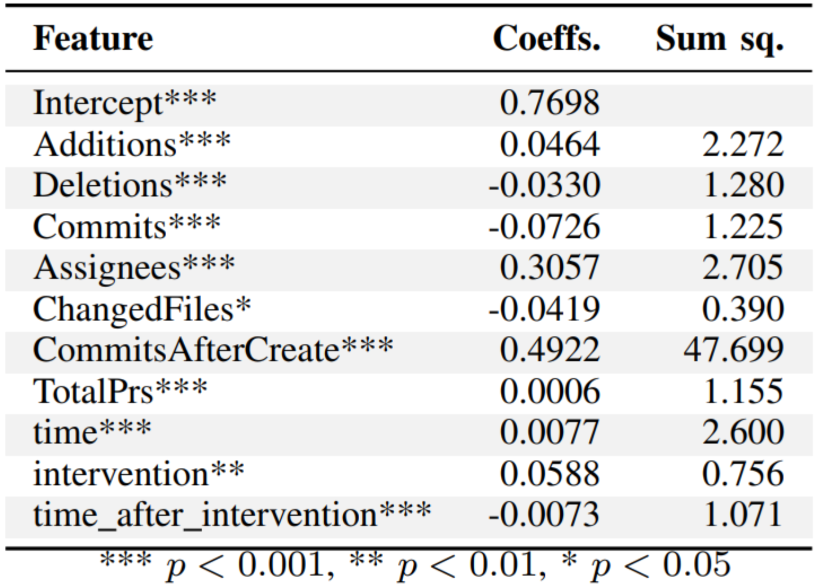


Commits are still being made after CI adoption



Effective Comments	↗	✗	↘
Commits	↗	✗	✗

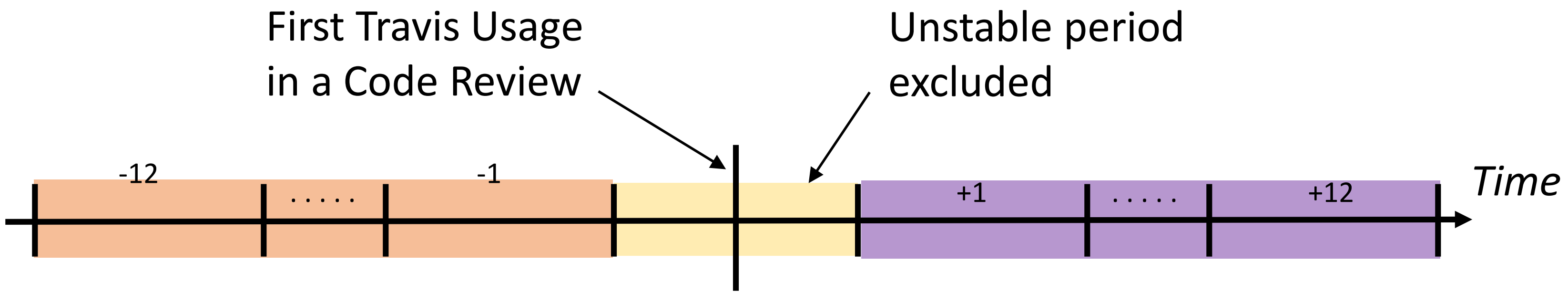
Approximating the Effect



Feature	Coeffs.	Sum sq.
Intercept***	0.7698	
Additions***	0.0464	2.272
Deletions***	-0.0330	1.280
Commits***	-0.0726	1.225
Assignees***	0.3057	2.705
ChangedFiles*	-0.0419	0.390
CommitsAfterCreate***	0.4922	47.699
TotalPrs***	0.0006	1.155
time***	0.0077	2.600
intervention**	0.0588	0.756
time_after_intervention***	-0.0073	1.071

*** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$

- One less general comment per code review
- On a mean of three comments per code reviews (33% drop)



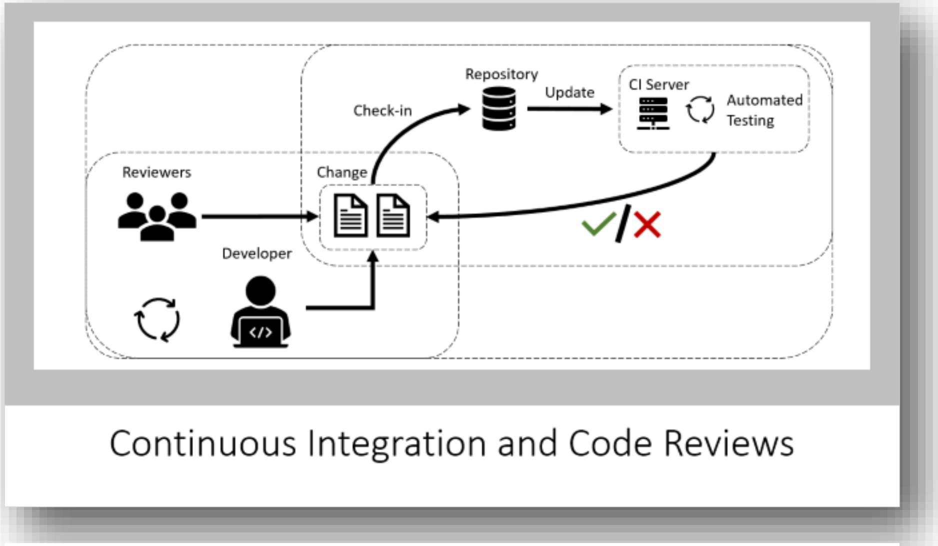
First Travis Usage in a Code Review

Unstable period excluded

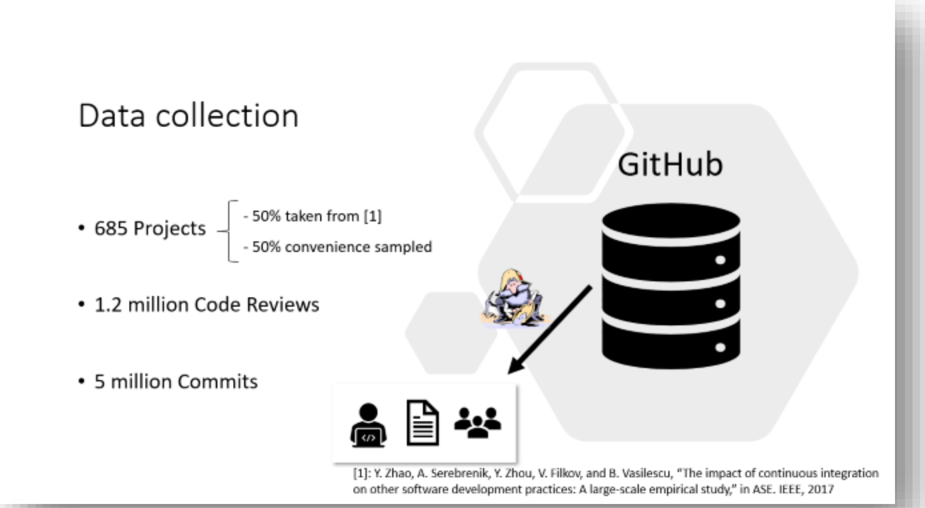
Take away Message

Continuous Integration fulfills the role of a **Silent Helper** during Code Reviews, leading to a **33% drop** in code review **comments**, without a similar drop in code modifications a year after the adoption of Continuous Integration.

In Conclusion



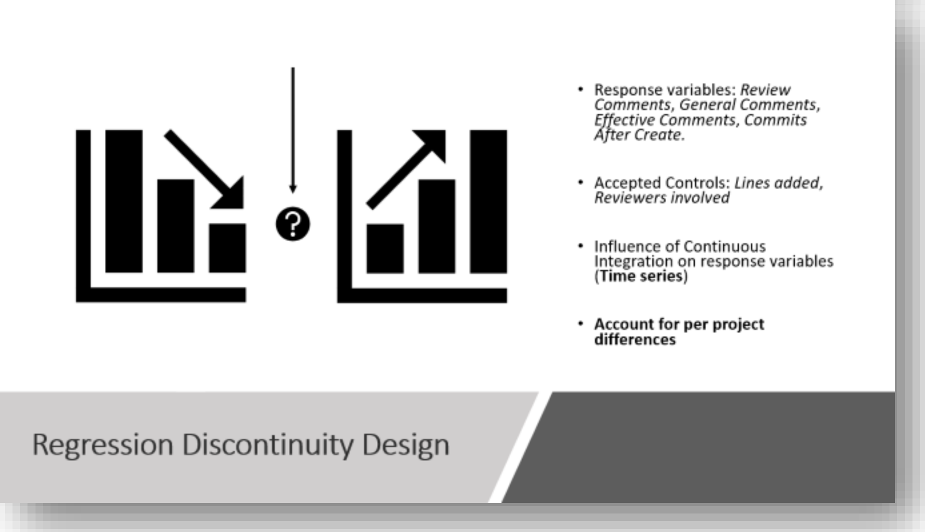
Continuous Integration and Code Reviews



Data collection

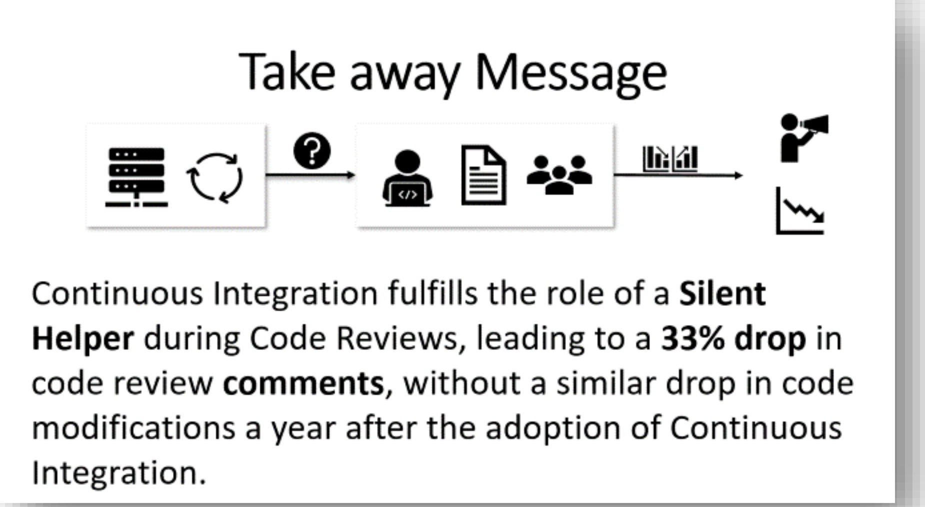
- 685 Projects
  - 50% taken from [1]
  - 50% convenience sampled
- 1.2 million Code Reviews
- 5 million Commits

[1]: Y. Zhao, A. Serebrenik, Y. Zhou, V. Filkov, and B. Vasilescu, "The impact of continuous integration on other software development practices: A large-scale empirical study," in ASE. IEEE, 2017



- Response variables: *Review Comments, General Comments, Effective Comments, Commits After Create.*
- Accepted Controls: *Lines added, Reviewers involved*
- Influence of Continuous Integration on response variables (**Time series**)
- **Account for per project differences**

Regression Discontinuity Design



Take away Message

Continuous Integration fulfills the role of a **Silent Helper** during Code Reviews, leading to a **33% drop** in code review **comments**, without a similar drop in code modifications a year after the adoption of Continuous Integration.