

# Modern Code Reviews in Open-Source Projects: Which Problems Do They Fix?



Delft University of Technology

Moritz Beller

Delft University of Technology



Andy Zaidman, Georgios Gousios, Alberto Bacchelli,  
Elmar Jürgens, Radjino Bohlanath, Shane McIntosh

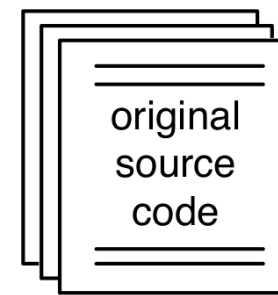
# **Modern Code Review Workflow**

**Old**

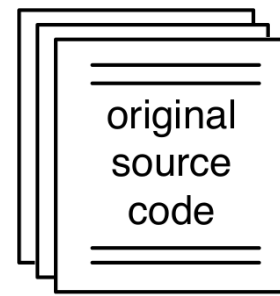


**Modern Code Review Workflow**

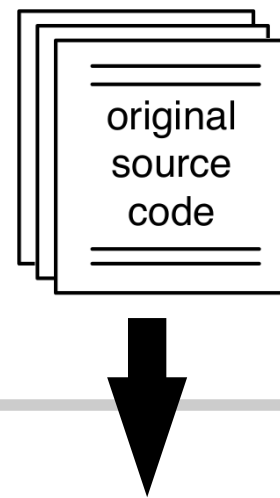
# **Modern Code Review Workflow**



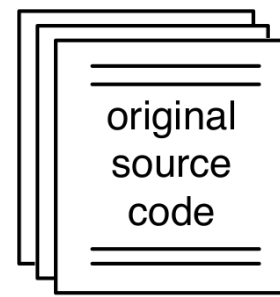
# Modern Code Review Workflow



# Modern Code Review Workflow

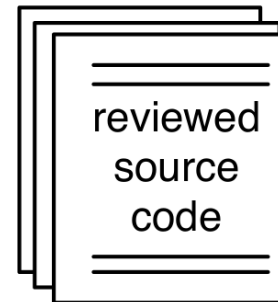
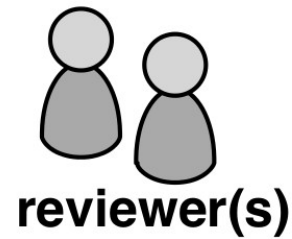
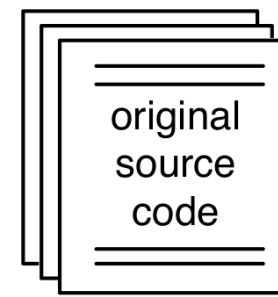


# Modern Code Review Workflow



# Modern Code Review Workflow

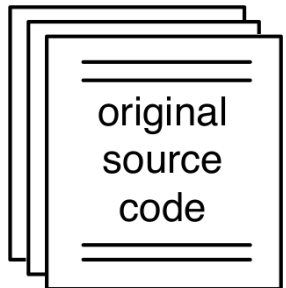




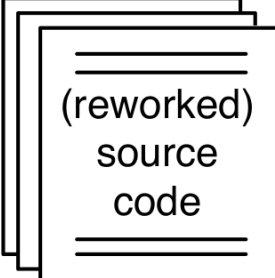
# Modern Code Review Workflow



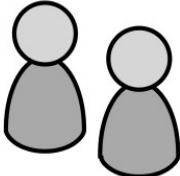
**author**



original  
source  
code



(reworked)  
source  
code

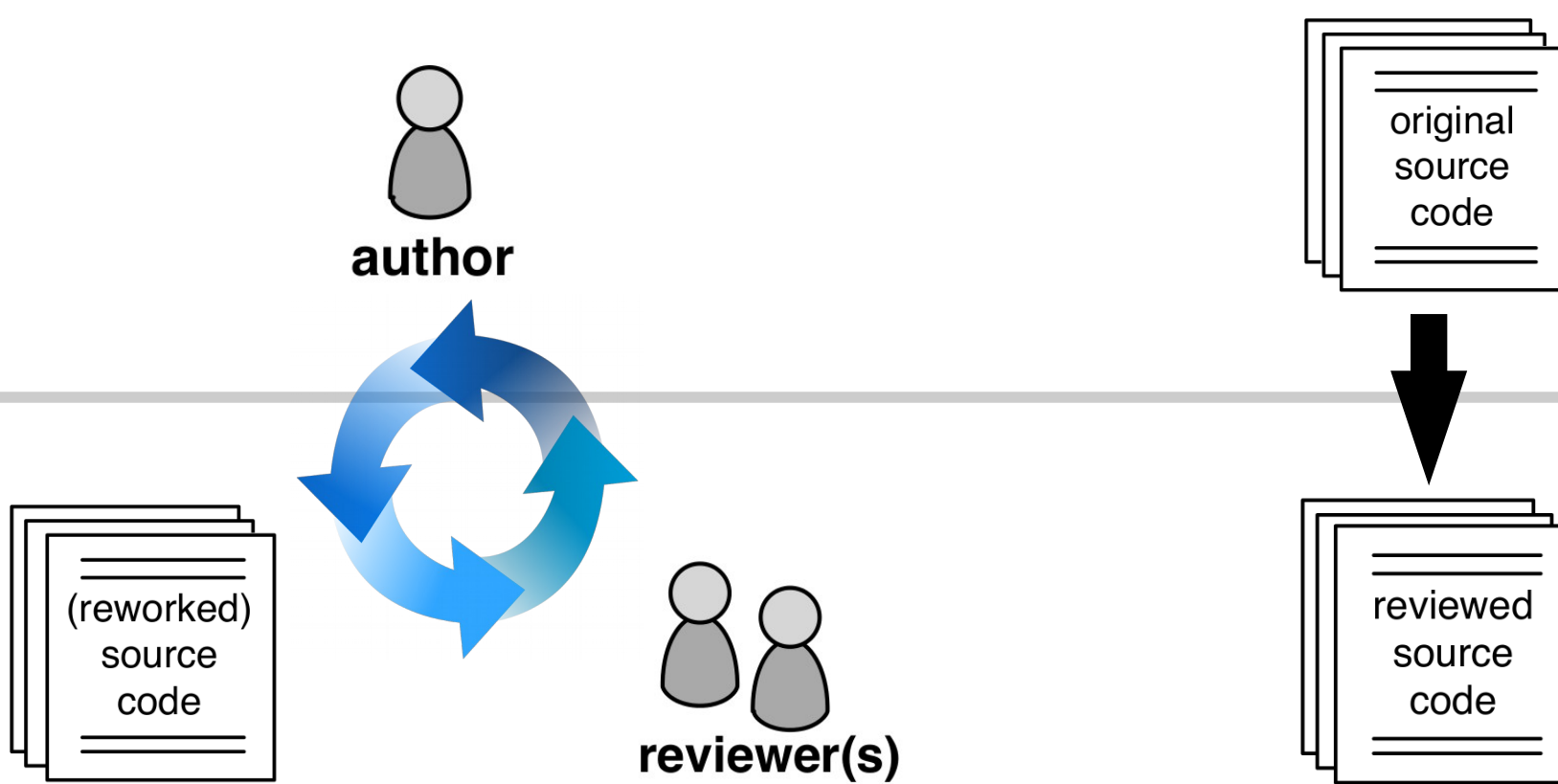


**reviewer(s)**

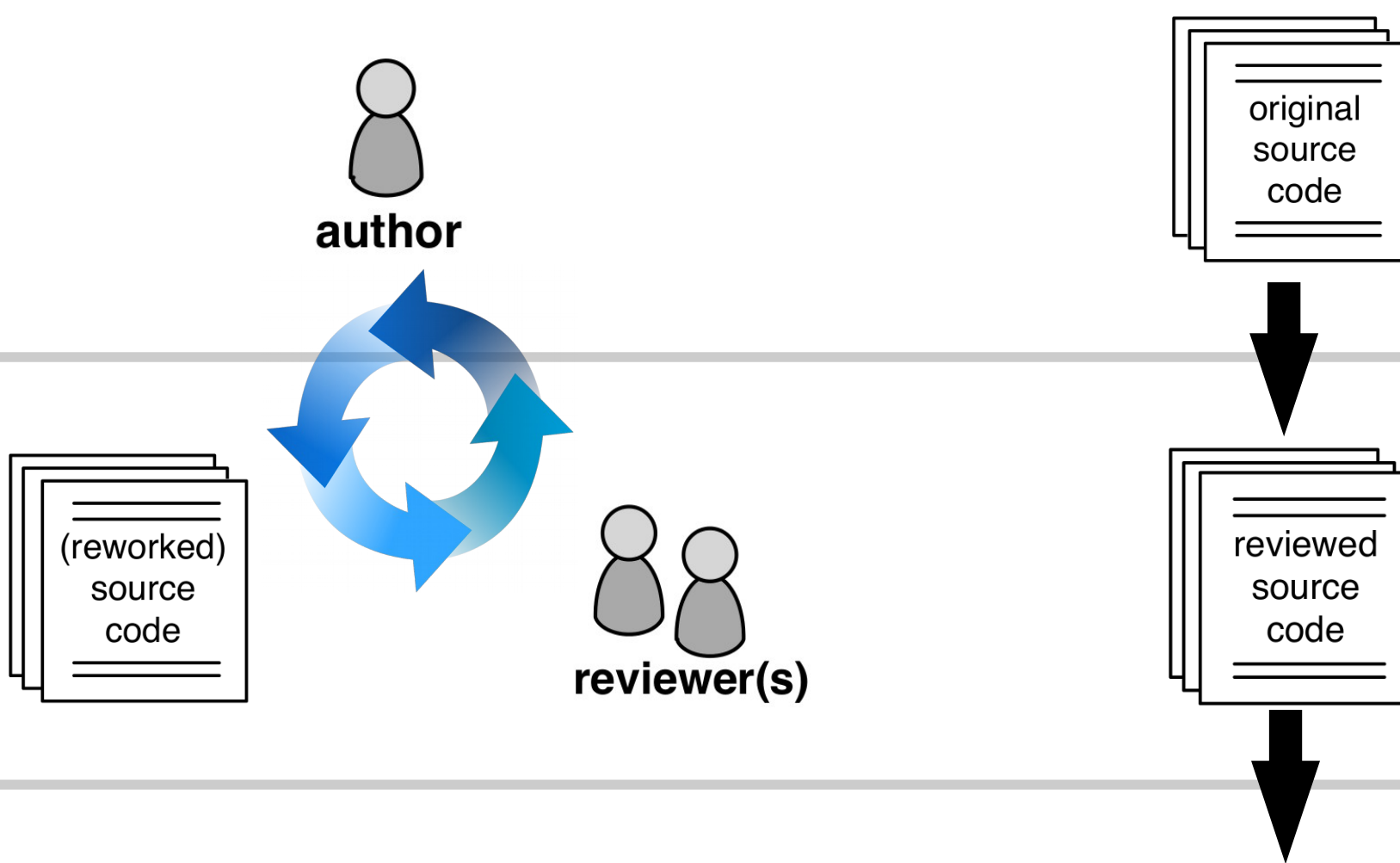


reviewed  
source  
code

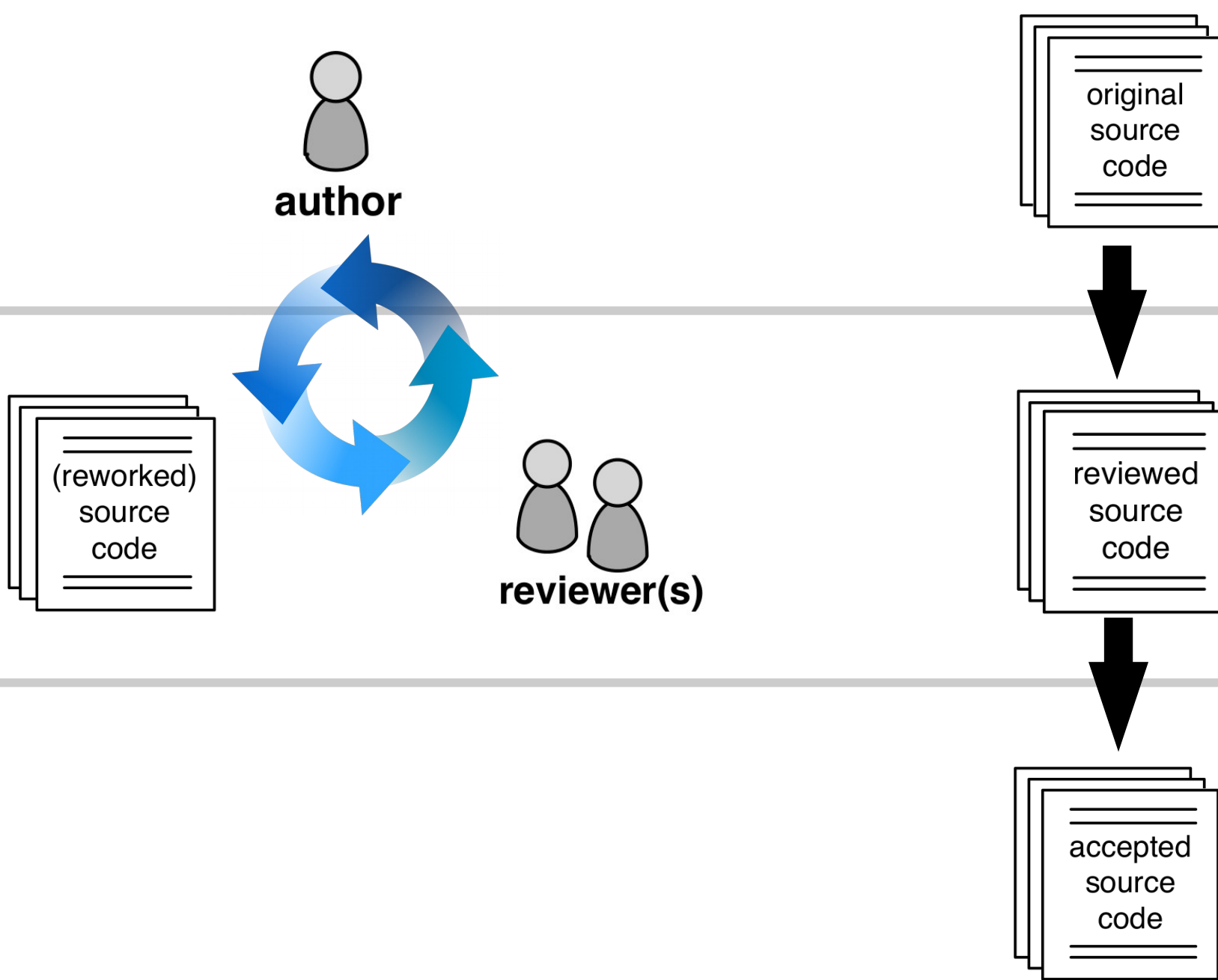
# Modern Code Review Workflow



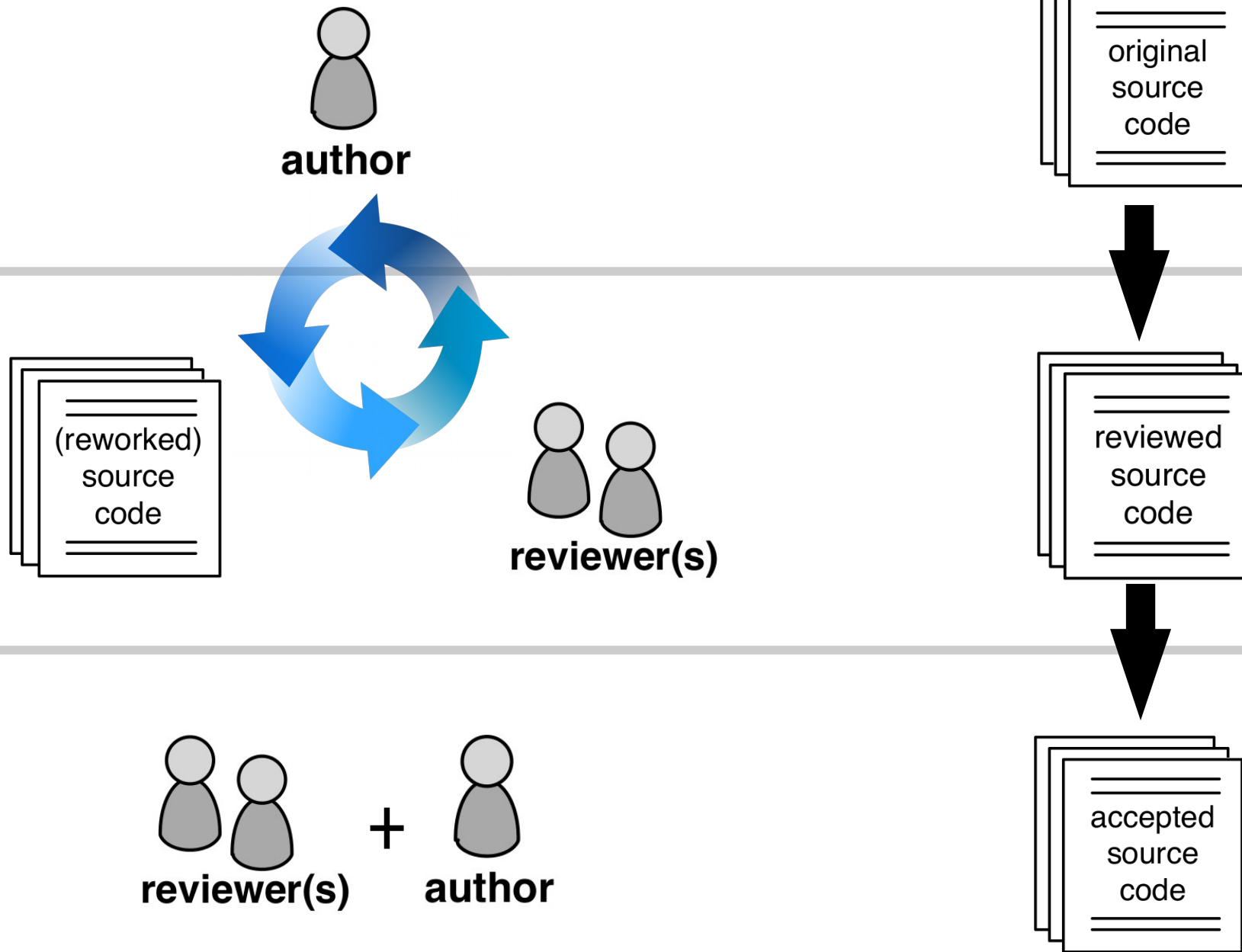
# Modern Code Review Workflow



# Modern Code Review Workflow



# Modern Code Review Workflow



# Modern Code Review Workflow





```
/** The ui thread for WatchDog registration. */  
public Runnable watchDog = new StartupUIThread();
```





```
/** The ui thread for WatchDog registration. */  
public Runnable watchDog = new StartupUIThread();
```



```
/** The ui thread for WatchDog registration. */  
// TODO (GG) Provide a more descriptive name  
public Runnable watchDog = new StartupUIThread();
```



The diagram illustrates the evolution of a code snippet. It features two horizontal lines with large downward-pointing arrows between them. Above the first arrow is a stack of document icons labeled 'original', and below the second arrow is a stack labeled 'reviewed source code'. The code itself is shown in two stages: the top stage is a single line of code, and the bottom stage adds a comment before the same line of code.

```
/** The ui thread for WatchDog registration. */  
public Runnable watchDog = new StartupUIThread();
```

```
/** The ui thread for WatchDog registration. */  
// TODO (GG) Provide a more descriptive name  
public Runnable watchDog = new StartupUIThread();
```

The diagram illustrates the evolution of a code snippet through three stages, represented by a vertical stack of three document icons. The top icon is labeled 'original', the middle 'reviewed', and the bottom 'accepted'. A large grey '1' is on the left. Arrows point from the original to the reviewed stage, and from the reviewed to the accepted stage.

```
/** The ui thread for WatchDog registration. */  
public Runnable watchDog = new StartupUIThread();
```

```
/** The ui thread for WatchDog registration. */  
// TODO (GG) Provide a more descriptive name  
public Runnable watchDog = new StartupUIThread();
```

```
/** The ui thread for WatchDog registration. */  
public Runnable watchDogUiThread = new  
StartupUIThread();
```

The diagram illustrates the process of code review and acceptance. It consists of three stages of code snippets, each represented by a stack of papers. The first stage is labeled 'original' and shows a code snippet. A large black arrow points down to the second stage, labeled 'reviewed', which shows the same code snippet with a green annotation. Another large black arrow points down to the third stage, labeled 'accepted', which shows the code snippet with the variable name underlined in green. The background features a large, light gray 'L' shape on the left side.

```
/** The ui thread for WatchDog registration. */  
public Runnable watchDog = new StartupUIThread();
```

```
/** The ui thread for WatchDog registration. */  
// TODO (GG) Provide a more descriptive name  
public Runnable watchDog = new StartupUIThread();
```

```
/** The ui thread for WatchDog registration. */  
public Runnable watchDogUiThread = new  
StartupUIThread();
```



```
/** The ui thread for WatchDog registration. */  
public Runnable watchDog = new StartupUIThread();
```

The diagram illustrates a sequence of three code snippets, each representing a different state of a variable declaration. The snippets are connected by downward-pointing arrows, indicating a progression or a series of changes. The first snippet shows a variable named 'watchDog' of type 'Runnable' initialized with a 'StartupUIThread' object. The second snippet is identical to the first but includes a comment '// TODO (GG) Provide a more descriptive name'. The third snippet shows the variable name changed to 'watchDogUiThread' and underlined in green, indicating a review-triggered change. The background of the diagram features a large, light gray 'L' shape on the left and a stack of three document icons on the right, each labeled 'original', 'reviewed', and 'accepted' respectively, further emphasizing the process of code review and change.

```
/** The ui thread for WatchDog registration. */  
// TODO (GG) Provide a more descriptive name  
public Runnable watchDog = new StartupUIThread();
```

```
/** The ui thread for WatchDog registration. */  
public Runnable watchDogUiThread = new  
StartupUIThread();
```

# Review-triggered Change



```
/** The ui thread for WatchDog registration. */  
public Runnable watchDog = new StartupUIThread();
```



```
/** The ui thread for WatchDog registration. */  
// TODO (GG) Provide a more descriptive name  
public Runnable watchDog = new StartupUIThread();
```

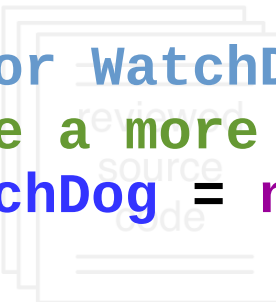




```
/** The ui thread for WatchDog registration. */  
public Runnable watchDog = new StartupUIThread();
```



---



```
/** The ui thread for WatchDog registration. */  
// TODO (GG) Provide a more descriptive name  
public Runnable watchDog = new StartupUIThread();
```


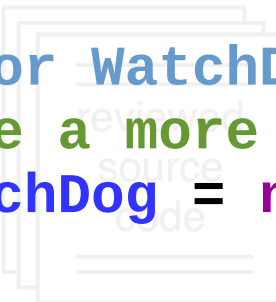




```
/** The ui thread for WatchDog registration. */  
public Runnable watchDog = new StartupUIThread();
```



```
/** The ui thread for WatchDog registration. */  
// TODO (GG) Provide a more descriptive name  
public Runnable watchDog = new StartupUIThread();
```



```
/** The ui thread for WatchDog registration. */  
private Runnable watchDogUiThread = new  
StartupUIThread();
```





```
/** The ui thread for WatchDog registration. */  
public Runnable watchDog = new StartupUIThread();
```



```
/** The ui thread for WatchDog registration. */  
// TODO (GG) Provide a more descriptive name  
public Runnable watchDog = new StartupUIThread();
```



```
/** The ui thread for WatchDog registration. */  
private Runnable watchDogUiThread = new  
StartupUIThread();
```



The diagram illustrates a sequence of code changes. It features three code snippets arranged vertically, connected by large black downward-pointing arrows. The first snippet is labeled 'original' and shows a public Runnable named 'watchDog'. The second snippet is labeled 'reviewed source code' and adds a TODO comment. The third snippet is labeled 'accepted code' and changes the variable name to 'watchDogUiThread' and makes it private. A large blue number '2' is on the left side of the image.

```
/** The ui thread for WatchDog registration. */  
public Runnable watchDog = new StartupUIThread();
```

```
/** The ui thread for WatchDog registration. */  
// TODO (GG) Provide a more descriptive name  
public Runnable watchDog = new StartupUIThread();
```

```
/** The ui thread for WatchDog registration. */  
private Runnable watchDogUiThread = new  
StartupUIThread();
```

# Self-motivated + Review-triggered Change



```
/** The ui thread for WatchDog registration. */  
public Runnable watchDog = new StartupUIThread();
```



```
/** The ui thread for WatchDog registration. */  
// TODO (GG) Provide a more descriptive name  
public Runnable watchDog = new StartupUIThread();
```






```
/** The ui thread for WatchDog registration. */  
public Runnable watchDog = new StartupUIThread();
```



```
/** The ui thread for WatchDog registration. */  
// TODO (GG) Provide a more descriptive name  
public Runnable watchDog = new StartupUIThread();
```


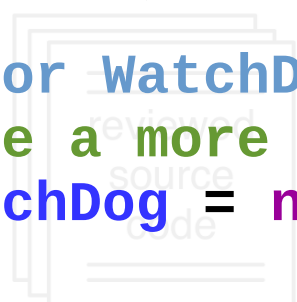




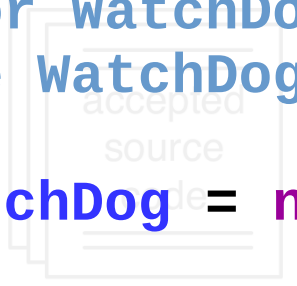
```
/** The ui thread for WatchDog registration. */  
public Runnable watchDog = new StartupUIThread();
```



```
/** The ui thread for WatchDog registration. */  
// TODO (GG) Provide a more descriptive name  
public Runnable watchDog = new StartupUIThread();
```



```
/**  
 * The ui thread for WatchDog registration. From  
 * this thread, the WatchDog program is started.  
 */  
public Runnable watchDog = new StartupUIThread();
```


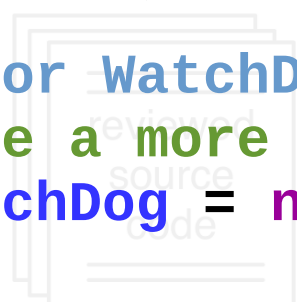




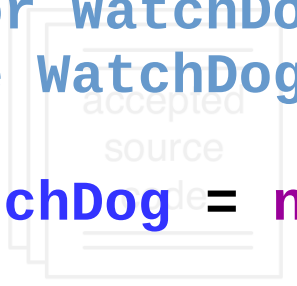
```
/** The ui thread for WatchDog registration. */  
public Runnable watchDog = new StartupUIThread();
```



```
/** The ui thread for WatchDog registration. */  
// TODO (GG) Provide a more descriptive name  
public Runnable watchDog = new StartupUIThread();
```



```
/**  
 * The ui thread for WatchDog registration. From  
 * this thread, the WatchDog program is started.  
 */  
public Runnable watchDog = new StartupUIThread();
```



3

```
/** The ui thread for WatchDog registration. */  
public Runnable watchDog = new StartupUIThread();
```



original

```
/** The ui thread for WatchDog registration. */  
// TODO (GG) Provide a more descriptive name  
public Runnable watchDog = new StartupUIThread();
```

reviewed  
source code

```
/**  
 * The ui thread for WatchDog registration. From  
 * this thread, the WatchDog program is started.  
 */  
public Runnable watchDog = new StartupUIThread();
```

accepted  
source code

# Self-motivated Change



```
/** The ui thread for WatchDog registration. */  
public Runnable watchDog = new StartupUIThread();
```



```
/** The ui thread for WatchDog registration. */  
// TODO (GG) Provide a more descriptive name  
public Runnable watchDog = new StartupUIThread();
```






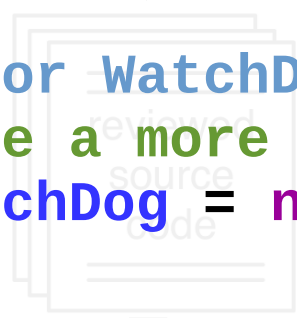


```
/** The ui thread for WatchDog registration. */  
public Runnable watchDog = new StartupUIThread();
```



---

```
/** The ui thread for WatchDog registration. */  
// TODO (GG) Provide a more descriptive name  
public Runnable watchDog = new StartupUIThread();
```






```
/** The ui thread for WatchDog registration. */  
public Runnable watchDog = new StartupUIThread();
```



```
/** The ui thread for WatchDog registration. */  
// TODO (GG) Provide a more descriptive name  
public Runnable watchDog = new StartupUIThread();
```



```
/** The ui thread for WatchDog registration. */  
public Runnable watchDog = new StartupUIThread();
```


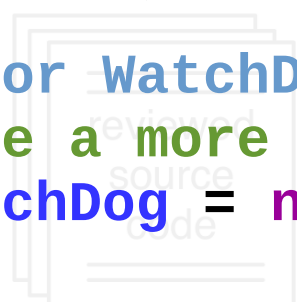




```
/** The ui thread for WatchDog registration. */  
public Runnable watchDog = new StartupUIThread();
```



```
/** The ui thread for WatchDog registration. */  
// TODO (GG) Provide a more descriptive name  
public Runnable watchDog = new StartupUIThread();
```



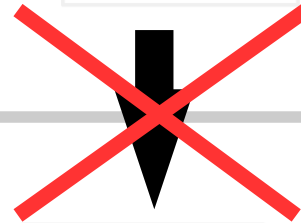
```
/** The ui thread for WatchDog registration. */  
public Runnable watchDog = new StartupUIThread();
```



**No Change**



```
/** The ui thread for WatchDog registration. */  
public Runnable watchDog = new StartupUIThread();
```



```
/** The ui thread for WatchDog registration. */  
// TODO (GG) Provide a more descriptive name  
public Runnable watchDog = new StartupUIThread();
```



```
/** The ui thread for WatchDog registration. */  
public Runnable watchDog = new StartupUIThread();
```

**No Change**



```
/** The ui thread for WatchDog registration. */  
public Runnable watchDog = new StartupUIThread();
```

```
/** The ui thread for WatchDog registration. */  
// TODO (GG) Provide a more descriptive name  
public Runnable watchDog = new StartupUIThread();
```

```
/** The ui thread for WatchDog registration. */  
public Runnable watchDog = new StartupUIThread();
```

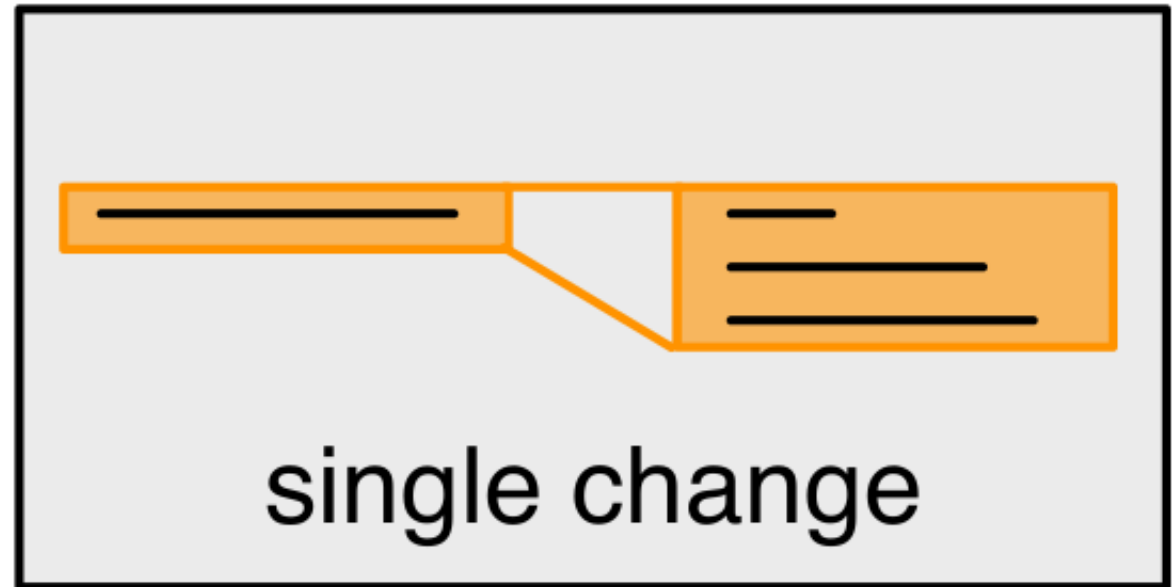
**No Change**

**Does it matter?**

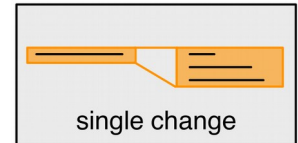
>1,400 changes  
>245 tasks

**GROMACS** FAST.  
FLEXIBLE.  
FREE.

**conQAT**  
CQSE



# Motivation for Change



1

```
/** The ui thread for WatchDog registration. */  
public Runnable watchDog = new StartupUIThread();
```

2

```
/** The ui thread for WatchDog registration. */  
// TODO (GG) Provide a more descriptive name  
public Runnable watchDog = new StartupUIThread();
```

3

```
/** The ui thread for WatchDog registration. */  
public Runnable watchDogUiThread = new  
StartupUIThread();
```

Review-triggered Change

```
/** The ui thread for WatchDog registration. */  
public Runnable watchDog = new StartupUIThread();
```

```
/** The ui thread for WatchDog registration. */  
// TODO (GG) Provide a more descriptive name  
public Runnable watchDog = new StartupUIThread();
```

```
/** The ui thread for WatchDog registration. */  
private Runnable watchDogUiThread = new  
StartupUIThread();
```

Self-motivated + Review-triggered Change

```
/** The ui thread for WatchDog registration. */  
public Runnable watchDog = new StartupUIThread();
```

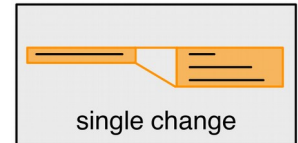
```
/** The ui thread for WatchDog registration. */  
// TODO (GG) Provide a more descriptive name  
public Runnable watchDog = new StartupUIThread();
```

```
/**  
 * The ui thread for WatchDog registration. From  
 * this thread, the whole WatchDog program is run.  
 */  
private Runnable watchDogUiThread = new  
StartupUIThread();
```

Self-motivated + Review-triggered Change



# Motivation for Change



1

```
/** The ui thread for WatchDog registration. */  
public Runnable watchDog = new StartupUIThread();
```

2

```
/** The ui thread for WatchDog registration. */  
// TODO (GG) Provide a more descriptive name  
public Runnable watchDog = new StartupUIThread();
```

3

```
/** The ui thread for WatchDog registration. */  
public Runnable watchDogUiThread = new  
StartupUIThread();
```

Review-triggered Change

```
/** The ui thread for WatchDog registration. */  
public Runnable watchDog = new StartupUIThread();
```

```
/** The ui thread for WatchDog registration. */  
// TODO (GG) Provide a more descriptive name  
public Runnable watchDog = new StartupUIThread();
```

```
/** The ui thread for WatchDog registration. */  
private Runnable watchDogUiThread = new  
StartupUIThread();
```

Self-motivated + Review-triggered Change

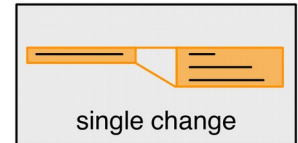
```
/** The ui thread for WatchDog registration. */  
public Runnable watchDog = new StartupUIThread();
```

```
/** The ui thread for WatchDog registration. */  
// TODO (GG) Provide a more descriptive name  
public Runnable watchDog = new StartupUIThread();
```

```
/**  
 * The ui thread for WatchDog registration. From  
 * this thread, the whole WatchDog program is run.  
 */  
private Runnable watchDogUiThread = new  
StartupUIThread();
```

Self-motivated + Review-triggered Change

# Motivation for Change



1

```
/** The ui thread for WatchDog registration. */  
public Runnable watchDog = new StartupUIThread();
```

2

```
/** The ui thread for WatchDog registration. */  
// TODO (GG) Provide a more descriptive name  
public Runnable watchDog = new StartupUIThread();
```

78-90%

3

```
/** The ui thread for WatchDog registration. */  
public Runnable watchDogUiThread = new  
StartupUIThread();
```

Review-triggered Change

```
/** The ui thread for WatchDog registration. */  
public Runnable watchDog = new StartupUIThread();
```

```
/** The ui thread for WatchDog registration. */  
// TODO (GG) Provide a more descriptive name  
public Runnable watchDog = new StartupUIThread();
```

```
/** The ui thread for WatchDog registration. */  
private Runnable watchDogUiThread = new  
StartupUIThread();
```

Self-motivated + Review-triggered Change

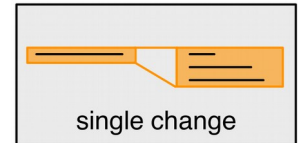
```
/** The ui thread for WatchDog registration. */  
public Runnable watchDog = new StartupUIThread();
```

```
/** The ui thread for WatchDog registration. */  
// TODO (GG) Provide a more descriptive name  
public Runnable watchDog = new StartupUIThread();
```

```
/**  
 * The ui thread for WatchDog registration. From  
 * this thread, the whole WatchDog program is run.  
 */  
private Runnable watchDogUiThread = new  
StartupUIThread();
```

Self-motivated + Review-triggered Change

# Motivation for Change



1

```
/** The ui thread for WatchDog registration. */  
public Runnable watchDog = new StartupUIThread();
```

2

```
/** The ui thread for WatchDog registration. */  
// TODO (GG) Provide a more descriptive name  
public Runnable watchDog = new StartupUIThread();
```

78-90%

3

```
/** The ui thread for WatchDog registration. */  
public Runnable watchDogUiThread = new  
StartupUIThread();
```

Review-triggered Change

```
/** The ui thread for WatchDog registration. */  
public Runnable watchDog = new StartupUIThread();
```

```
/** The ui thread for WatchDog registration. */  
// TODO (GG) Provide a more descriptive name  
public Runnable watchDog = new StartupUIThread();
```

```
/** The ui thread for WatchDog registration. */  
private Runnable watchDogUiThread = new  
StartupUIThread();
```

Self-motivated + Review-triggered Change

22-10%

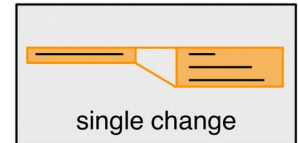
```
/** The ui thread for WatchDog registration. */  
public Runnable watchDog = new StartupUIThread();
```

```
/** The ui thread for WatchDog registration. */  
// TODO (GG) Provide a more descriptive name  
public Runnable watchDog = new StartupUIThread();
```

```
/**  
 * The ui thread for WatchDog registration. From  
 * this thread, the whole WatchDog program is run.  
 */  
private Runnable watchDogUiThread = new  
StartupUIThread();
```

Self-motivated + Review-triggered Change

# Motivation for Change



1

```
/** The ui thread for WatchDog registration. */  
public Runnable watchDog = new StartupUIThread();
```

2

```
/** The ui thread for WatchDog registration. */  
// TODO (GG) Provide a more descriptive name  
public Runnable watchDog = new StartupUIThread();
```

78-90%

3

```
/** The ui thread for WatchDog registration. */  
public Runnable watchDogUiThread = new  
StartupUIThread();
```

Review-triggered Change

```
/** The ui thread for WatchDog registration. */  
public Runnable watchDog = new StartupUIThread();
```

```
/** The ui thread for WatchDog registration. */  
// TODO (GG) Provide a more descriptive name  
public Runnable watchDog = new StartupUIThread();
```

```
/** The ui thread for WatchDog registration. */  
private Runnable watchDogUiThread = new  
StartupUIThread();
```

Self-motivated + Review-triggered Change

22-10%

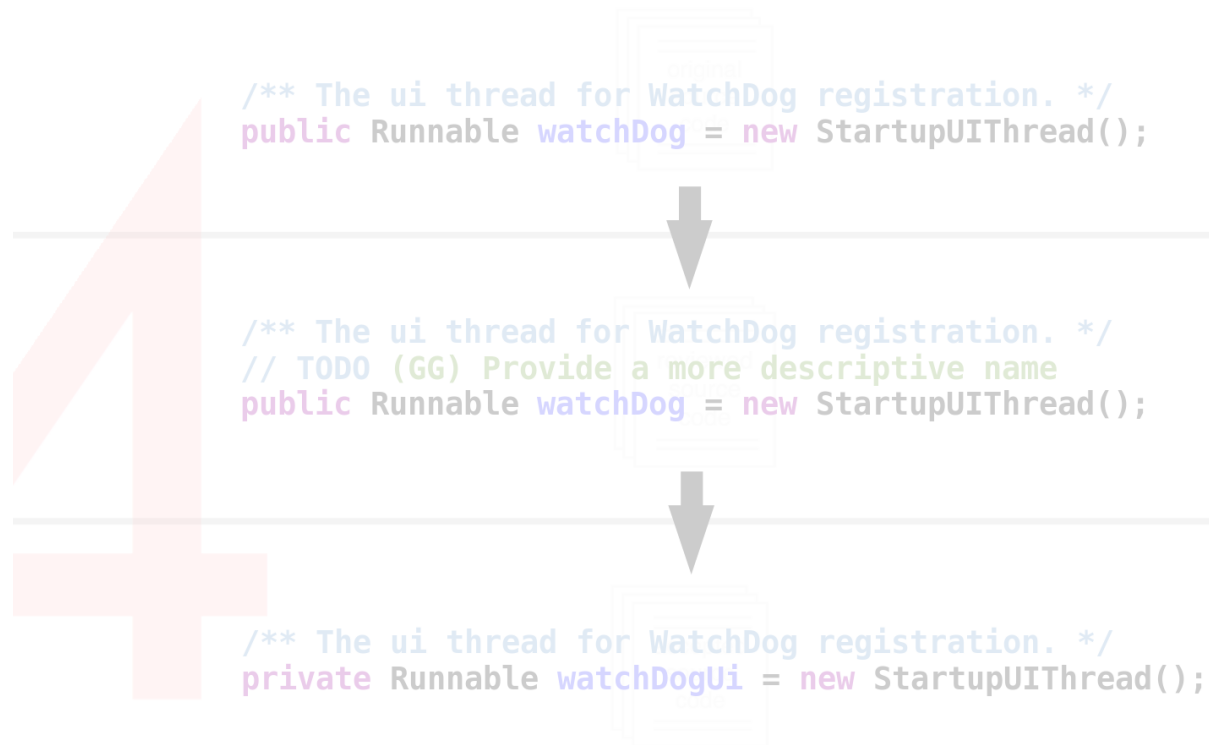
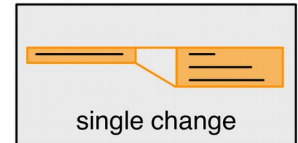
```
/** The ui thread for WatchDog registration. */  
public Runnable watchDog = new StartupUIThread();
```

```
/** The ui thread for WatchDog registration. */  
// TODO (GG) Provide a more descriptive name  
public Runnable watchDog = new StartupUIThread();
```

```
/**  
 * The ui thread for WatchDog registration. From  
 * this thread, the whole WatchDog program is run.  
 */  
private Runnable watchDogUiThread = new  
StartupUIThread();
```

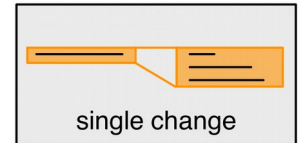
Self-motivated + Review-triggered Change

# Motivation for No Change



No Change

# Motivation for No Change



original

```
/** The ui thread for WatchDog registration. */  
public Runnable watchDog = new StartupUIThread();
```



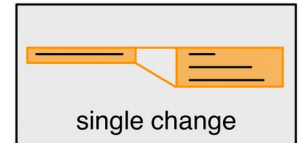
```
/** The ui thread for WatchDog registration. */  
// TODO (GG) Provide a more descriptive name  
public Runnable watchDog = new StartupUIThread();
```



```
/** The ui thread for WatchDog registration. */  
private Runnable watchDogUi = new StartupUIThread();
```

**No Change**

# Motivation for No Change



original

```
/** The ui thread for WatchDog registration. */  
public Runnable watchDog = new StartupUIThread();
```



```
/** The ui thread for WatchDog registration. */  
// TODO (GG) Provide a more descriptive name  
public Runnable watchDog = new StartupUIThread();
```

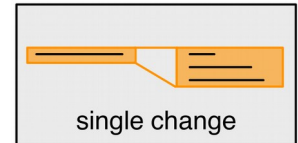
7-35%



```
/** The ui thread for WatchDog registration. */  
private Runnable watchDogUi = new StartupUIThread();
```

No Change

# Motivation for No Change



original

```
/** The ui thread for WatchDog registration. */  
public Runnable watchDog = new StartupUIThread();
```



```
/** The ui thread for WatchDog registration. */  
// TODO (GG) Provide a more descriptive name  
public Runnable watchDog = new StartupUIThread();
```

7-35%

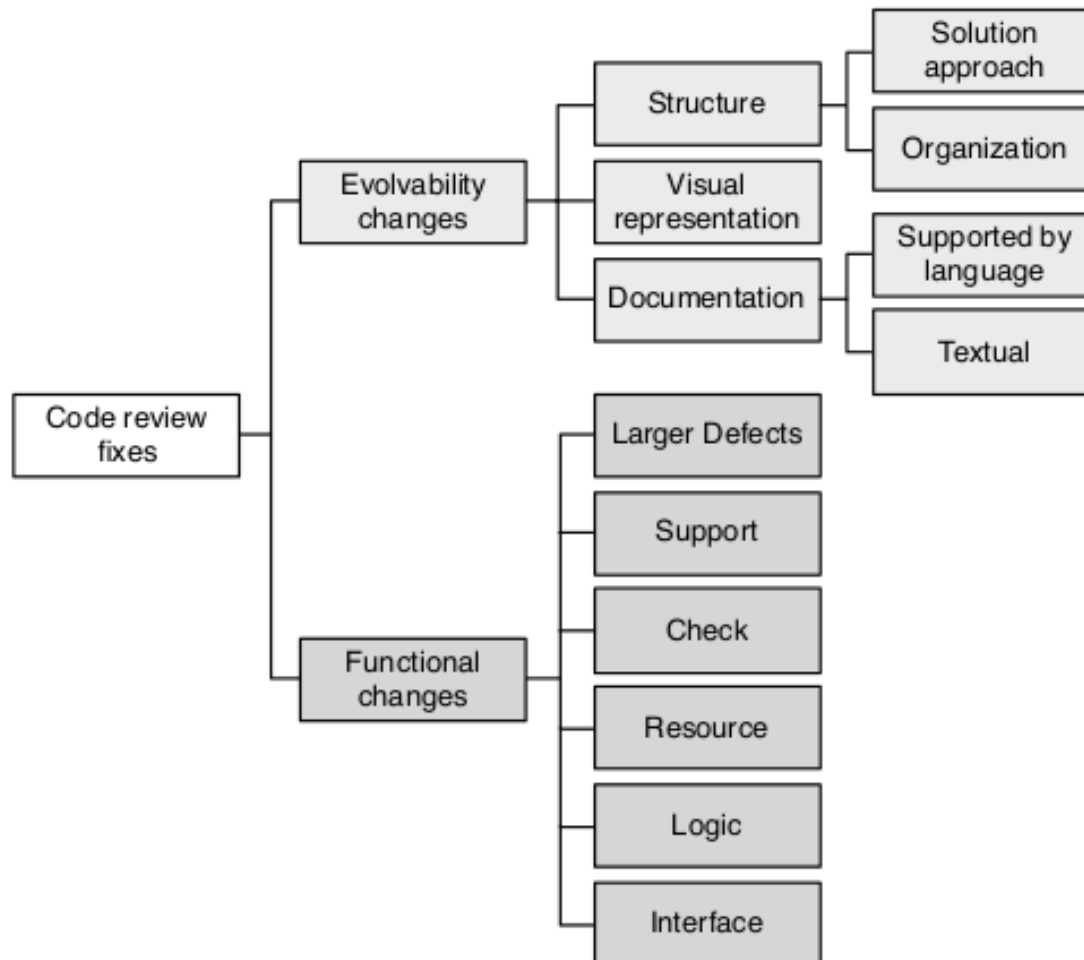


```
/** The ui thread for WatchDog registration. */  
private Runnable watchDogUi = new StartupUIThread();
```

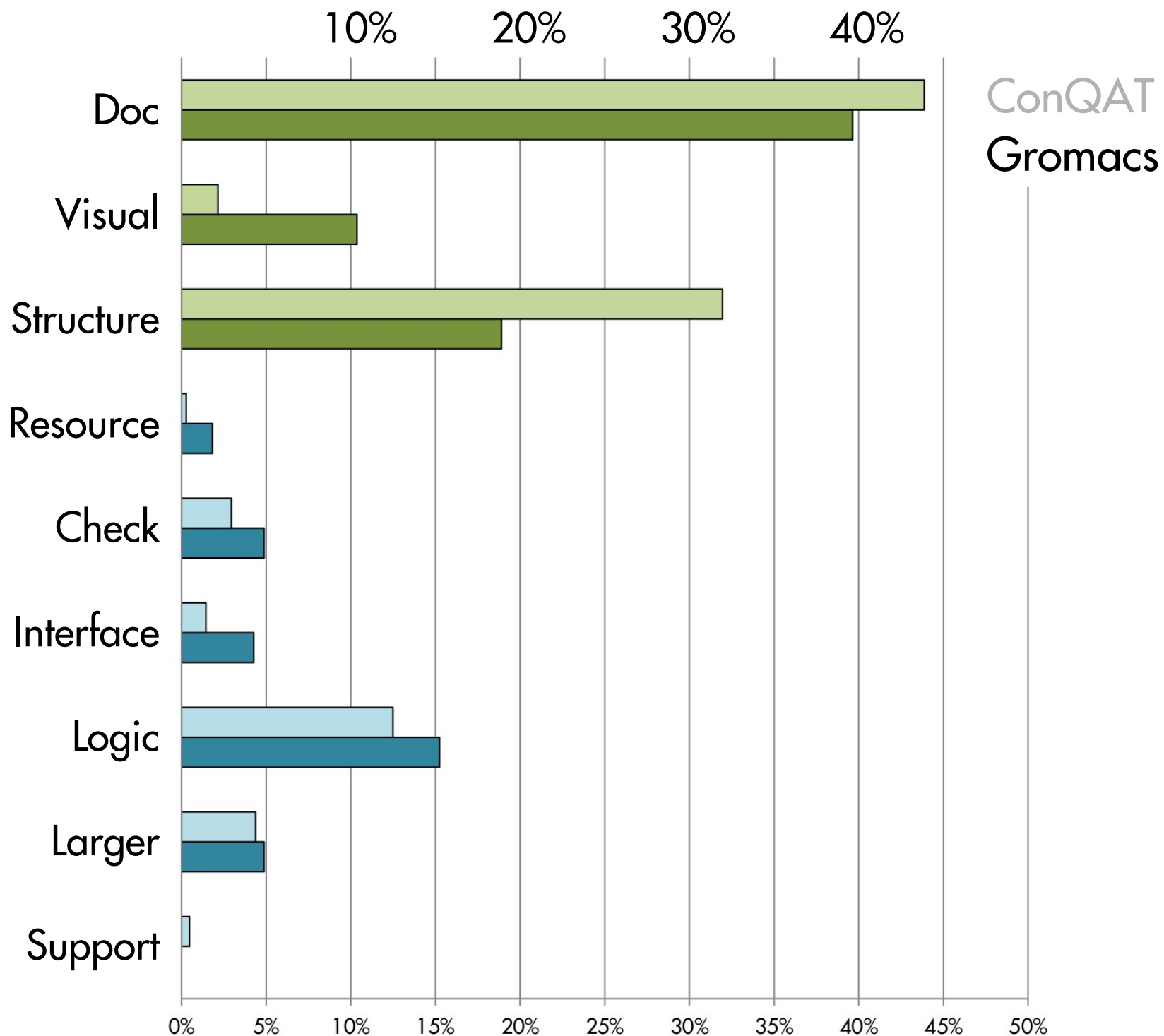
No Change



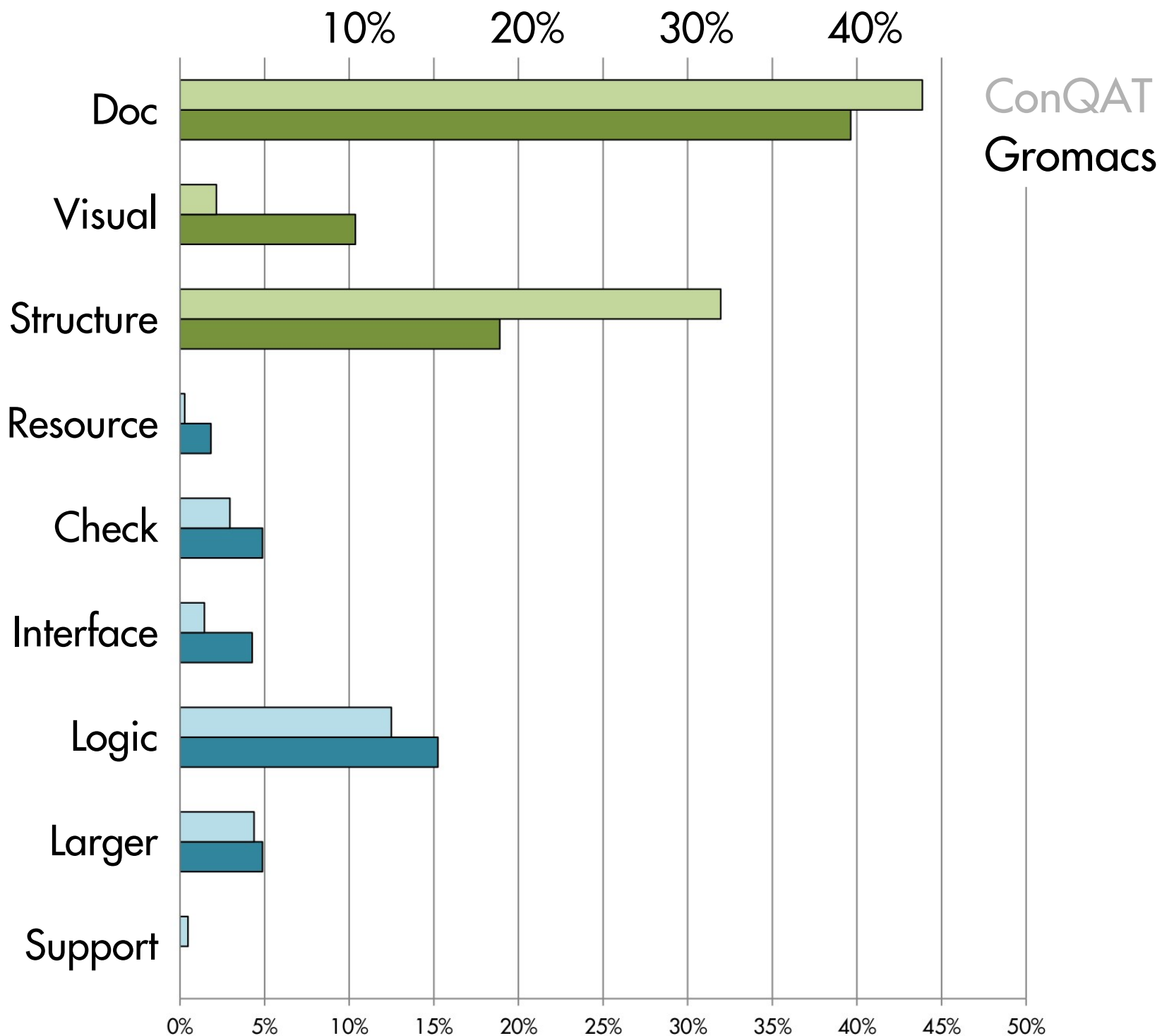
# Type of Change



manual change classification



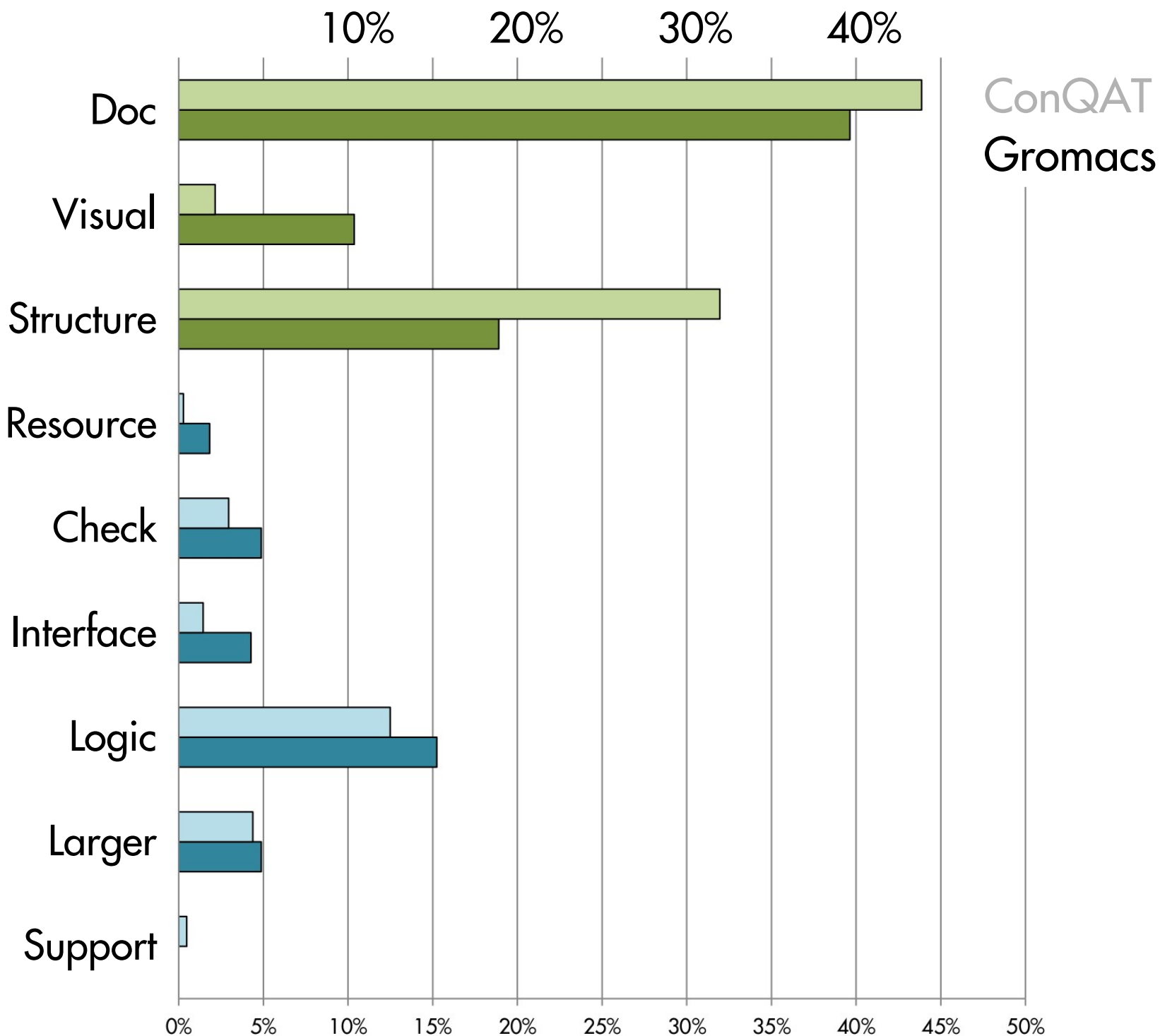
75



75

•

•

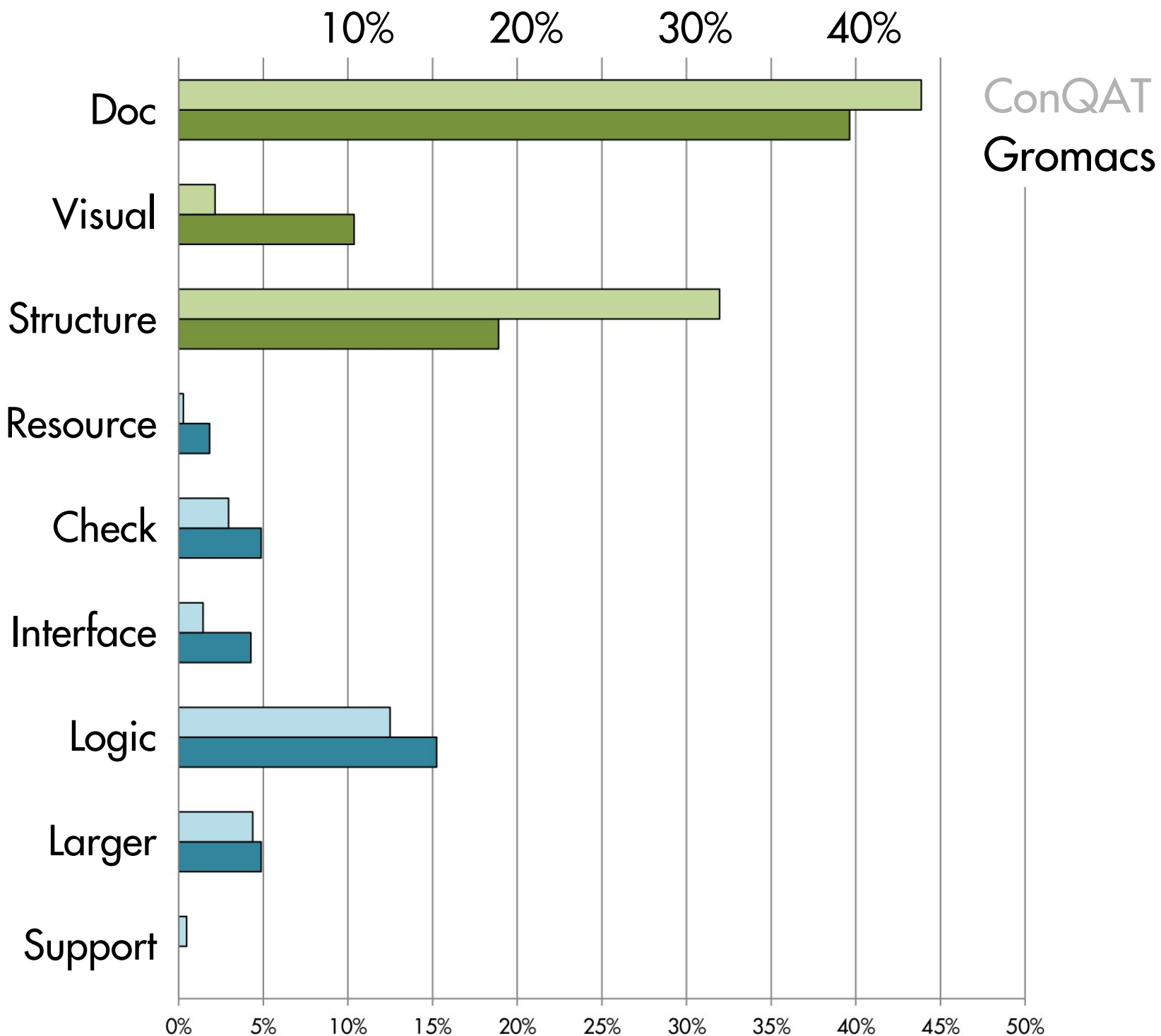


75

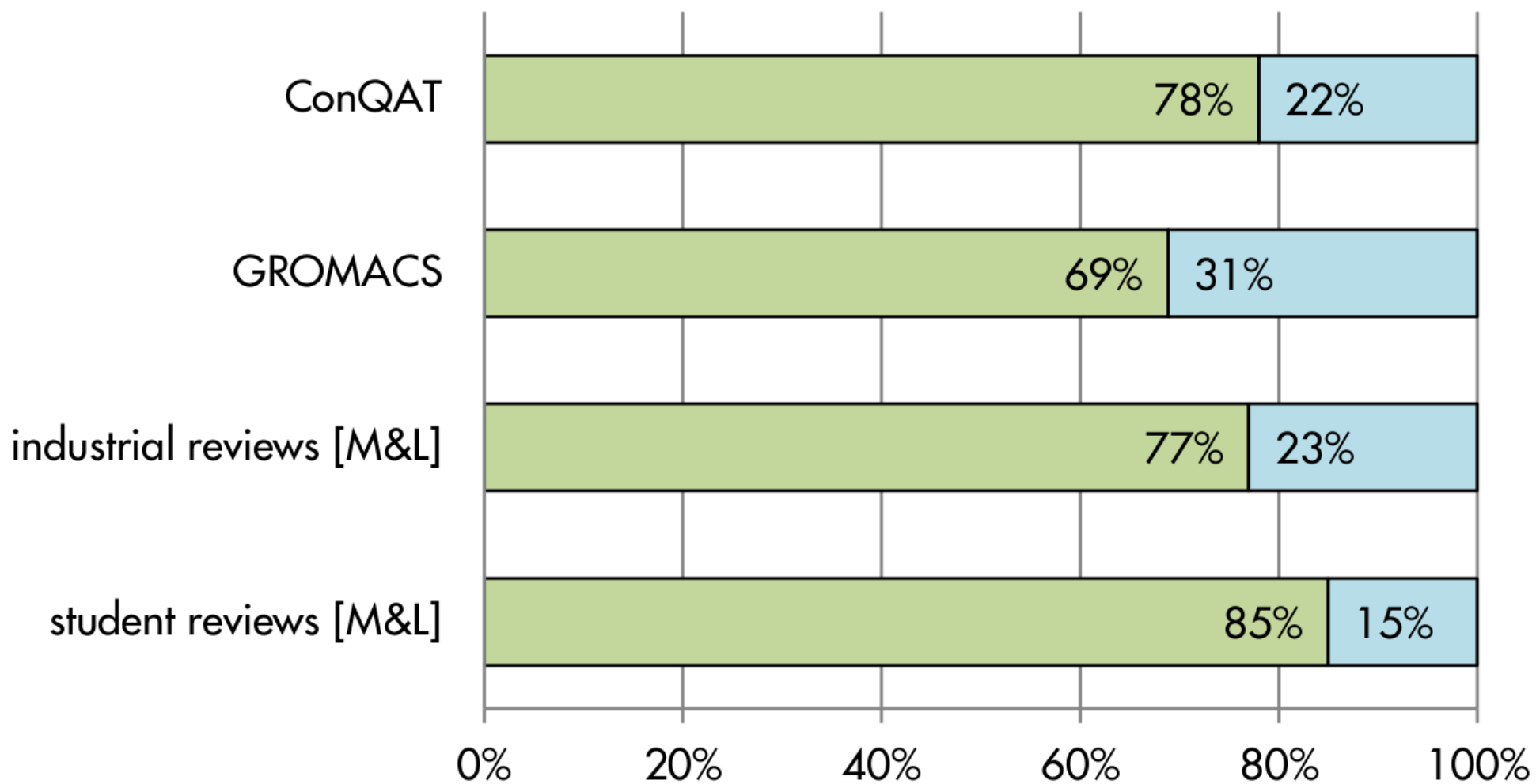
•

•

25

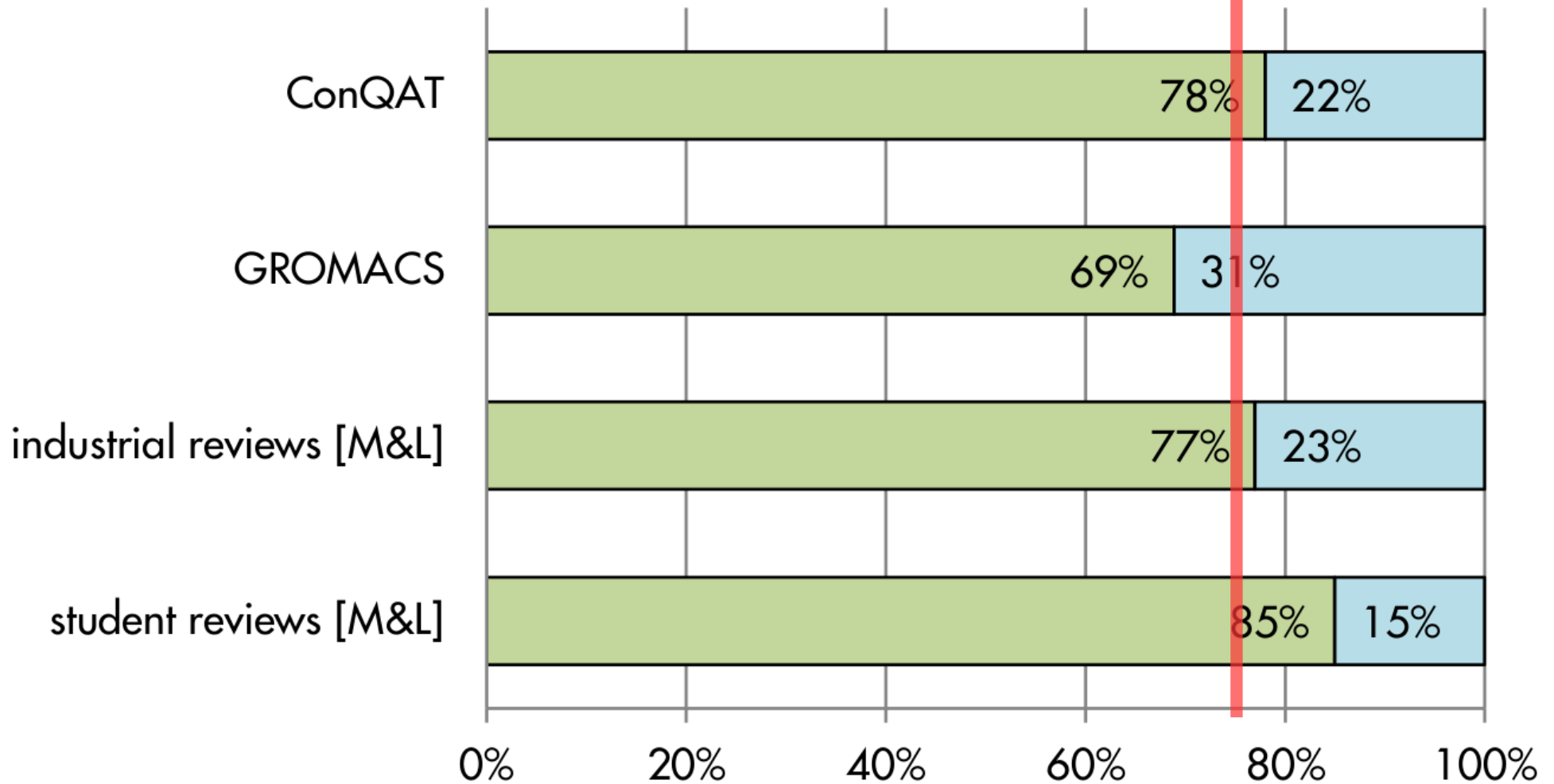


■ evolvability changes    ■ functional changes



75 : 25

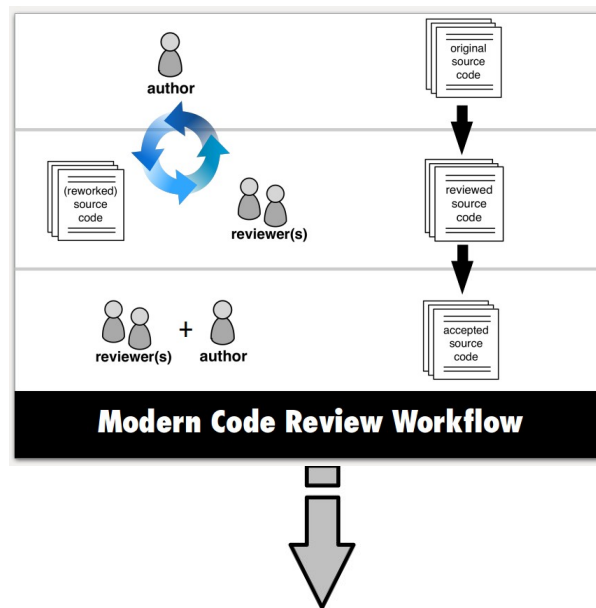
■ evolvability changes   ■ functional changes



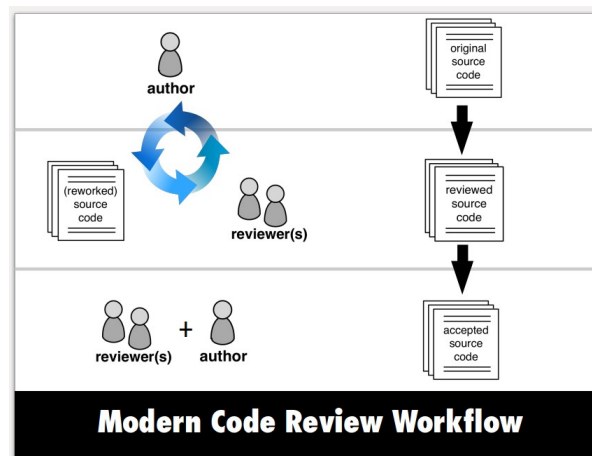
# What Influences Code Review?



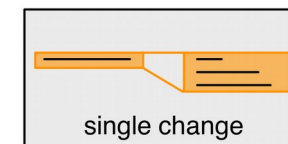
# What Influences Code Review?



# What Influences Code Review?



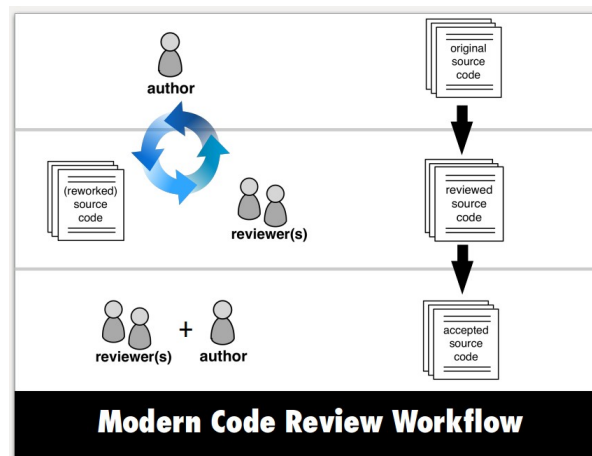
number of changes  
in code under review



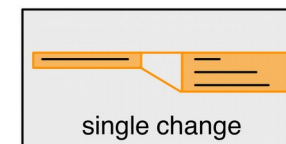
# What Influences Code Review?

conQAT

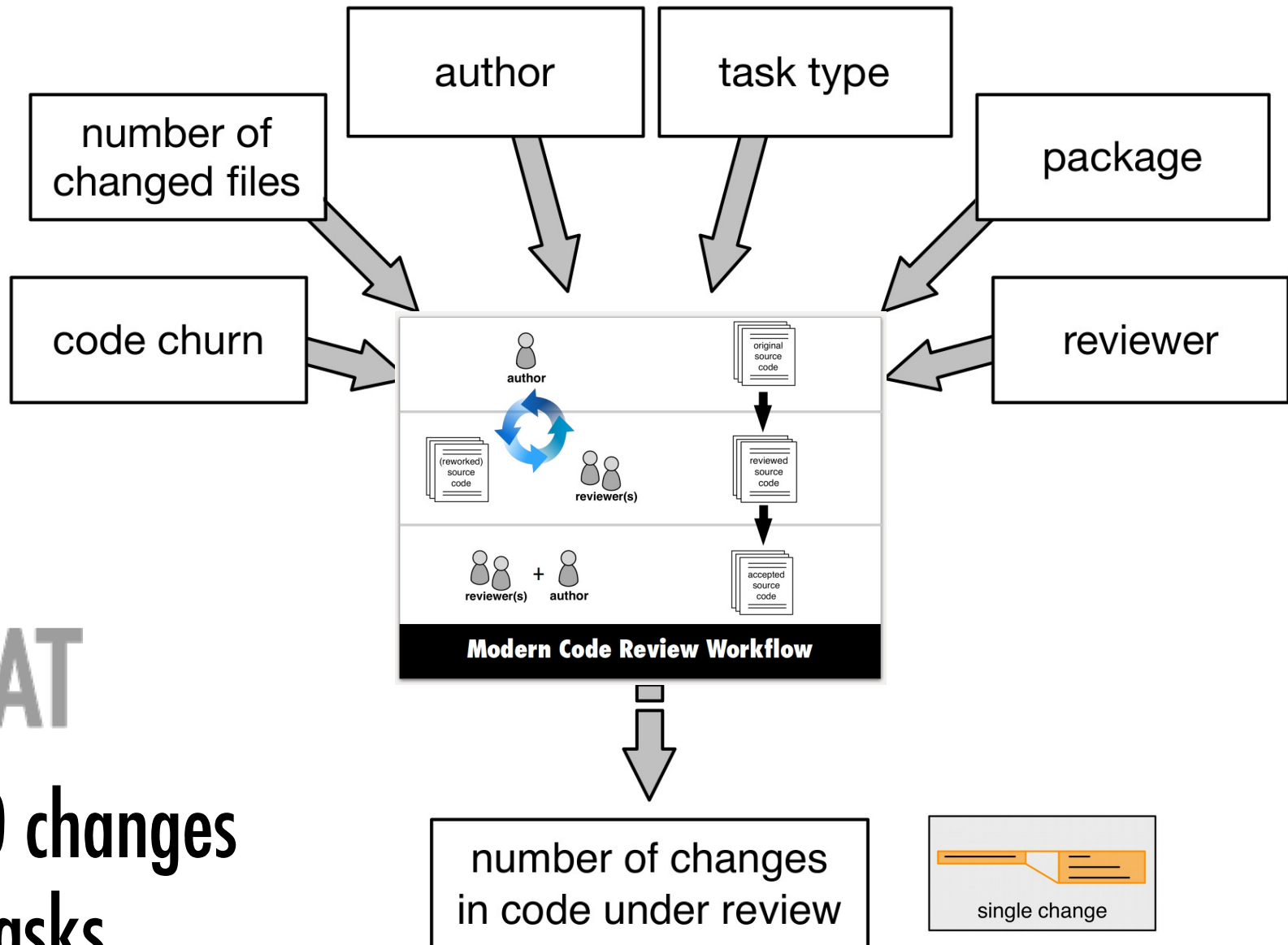
> 2,800 changes  
> 970 tasks



number of changes  
in code under review



# What Influences Code Review?

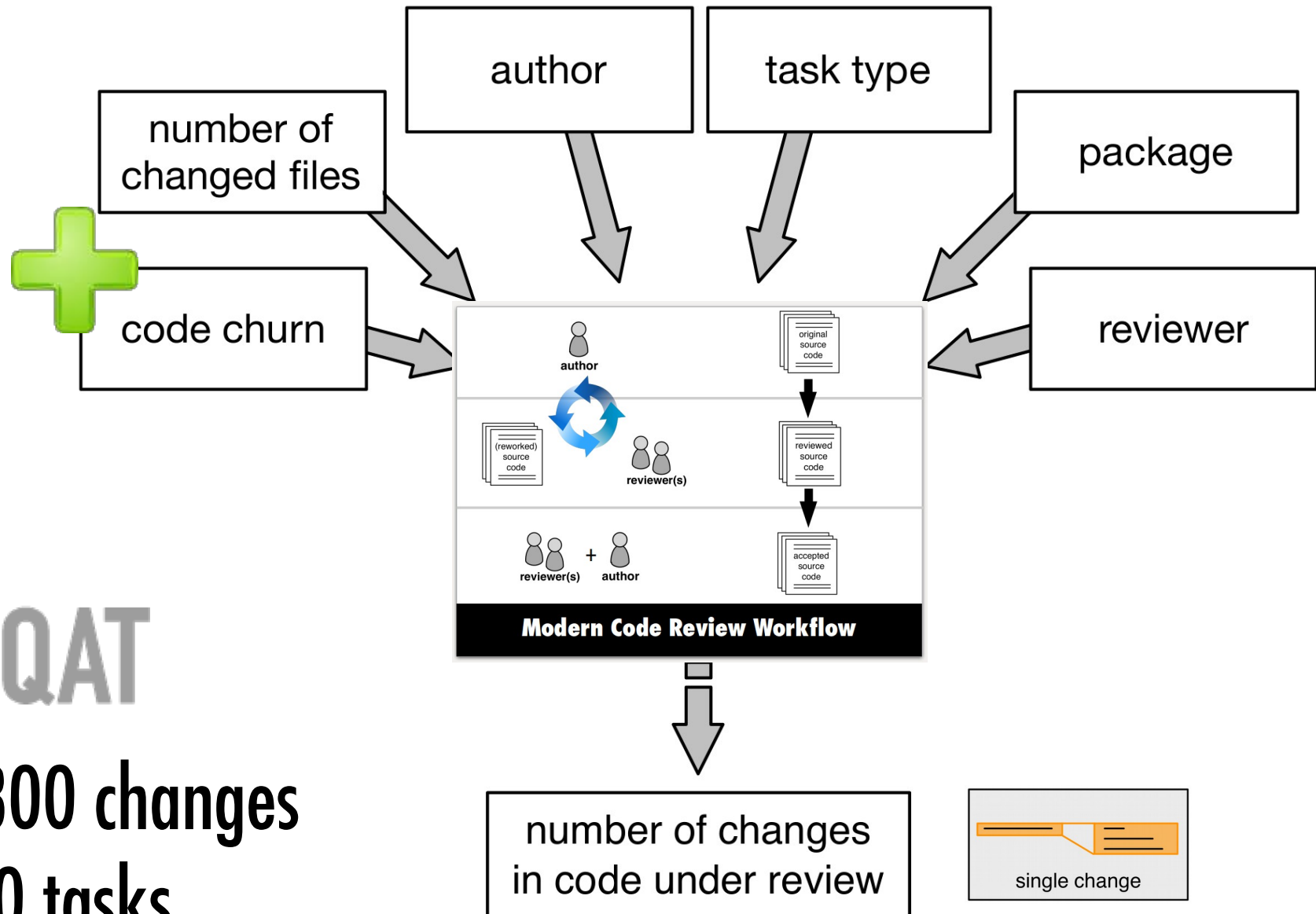


conQAT

> 2,800 changes

> 970 tasks

# What Influences Code Review?

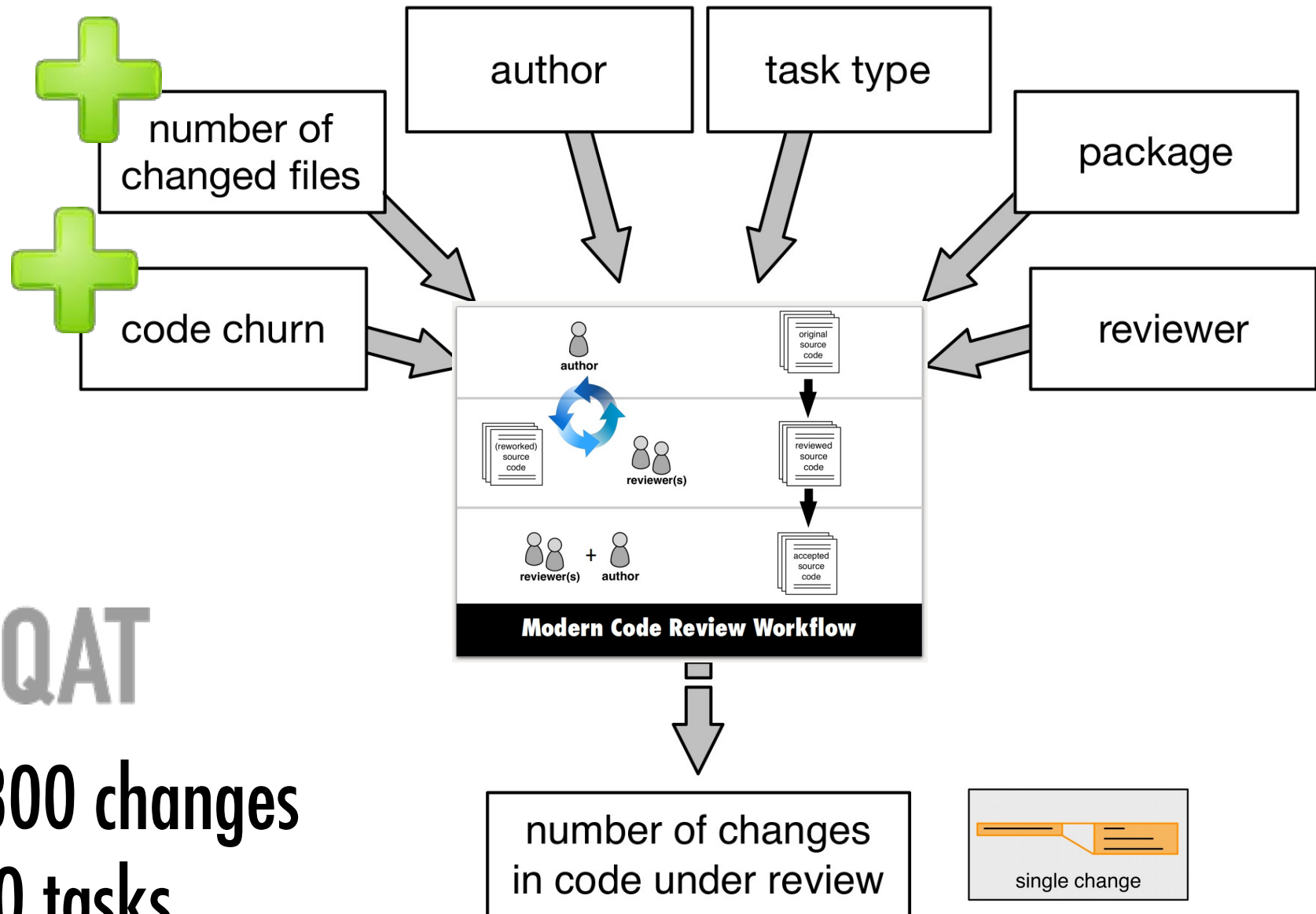


conQAT

> 2,800 changes

> 970 tasks

# What Influences Code Review?

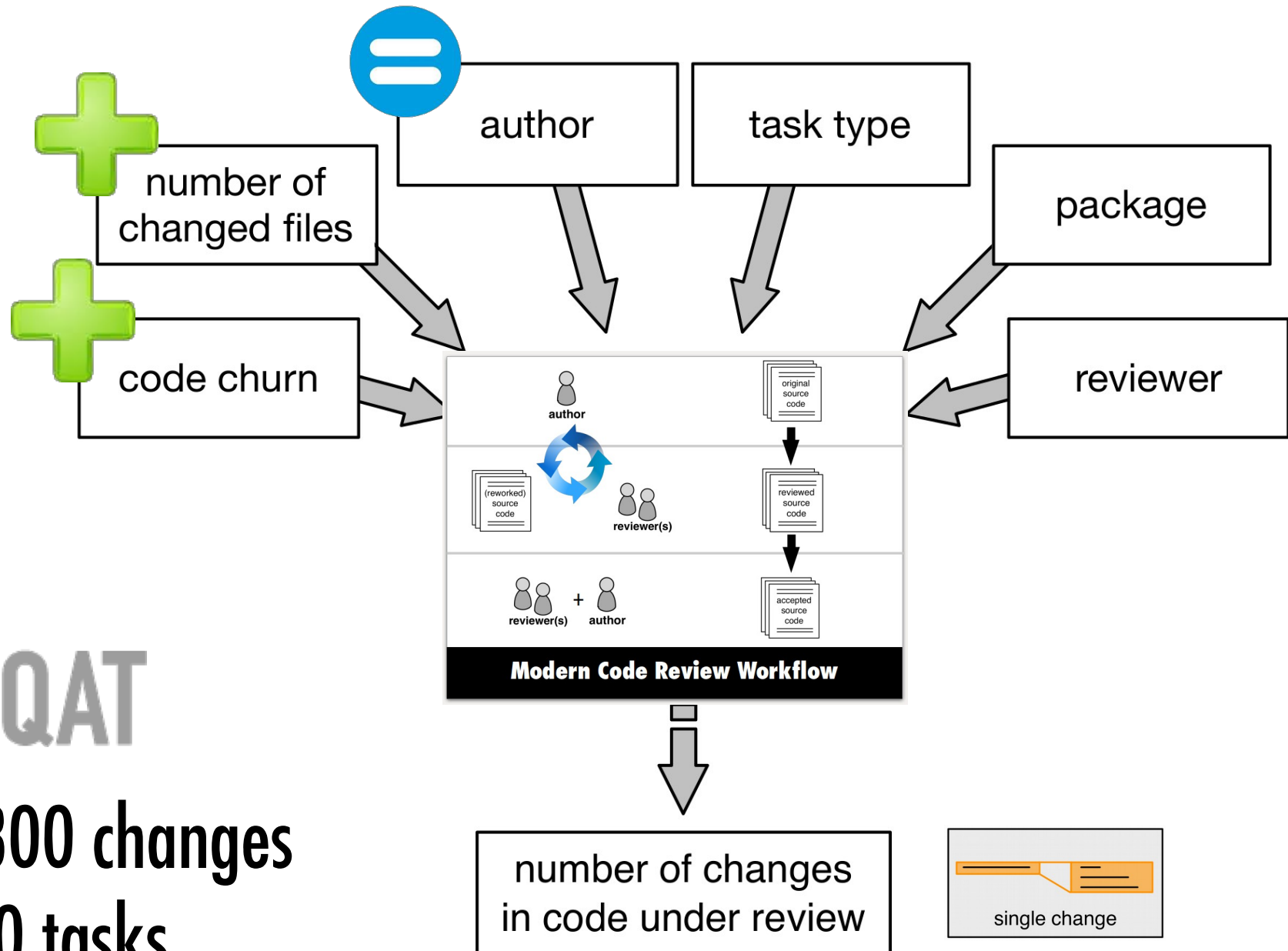


conQAT

> 2,800 changes

> 970 tasks

# What Influences Code Review?

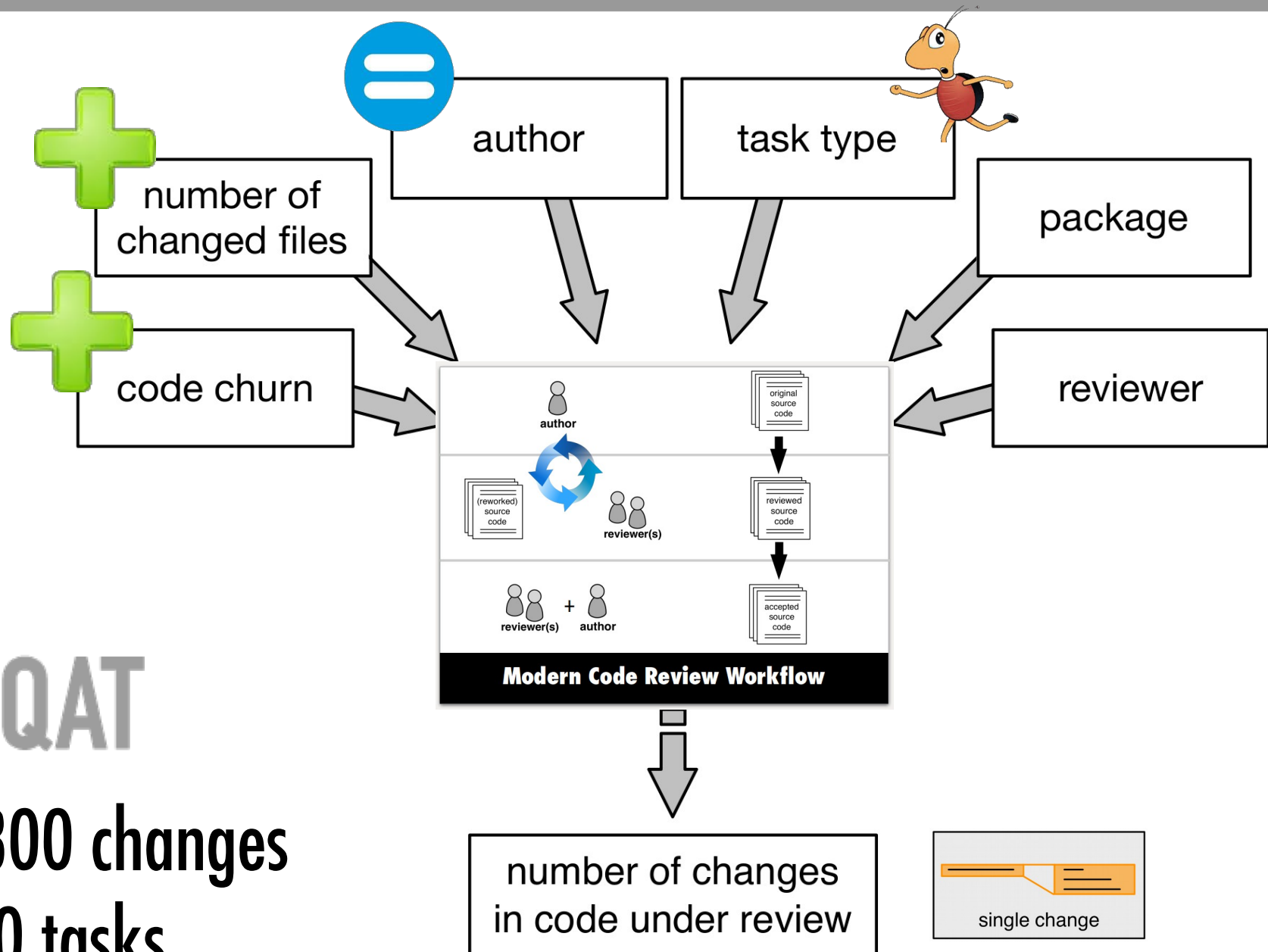


conQAT

> 2,800 changes

> 970 tasks

# What Influences Code Review?

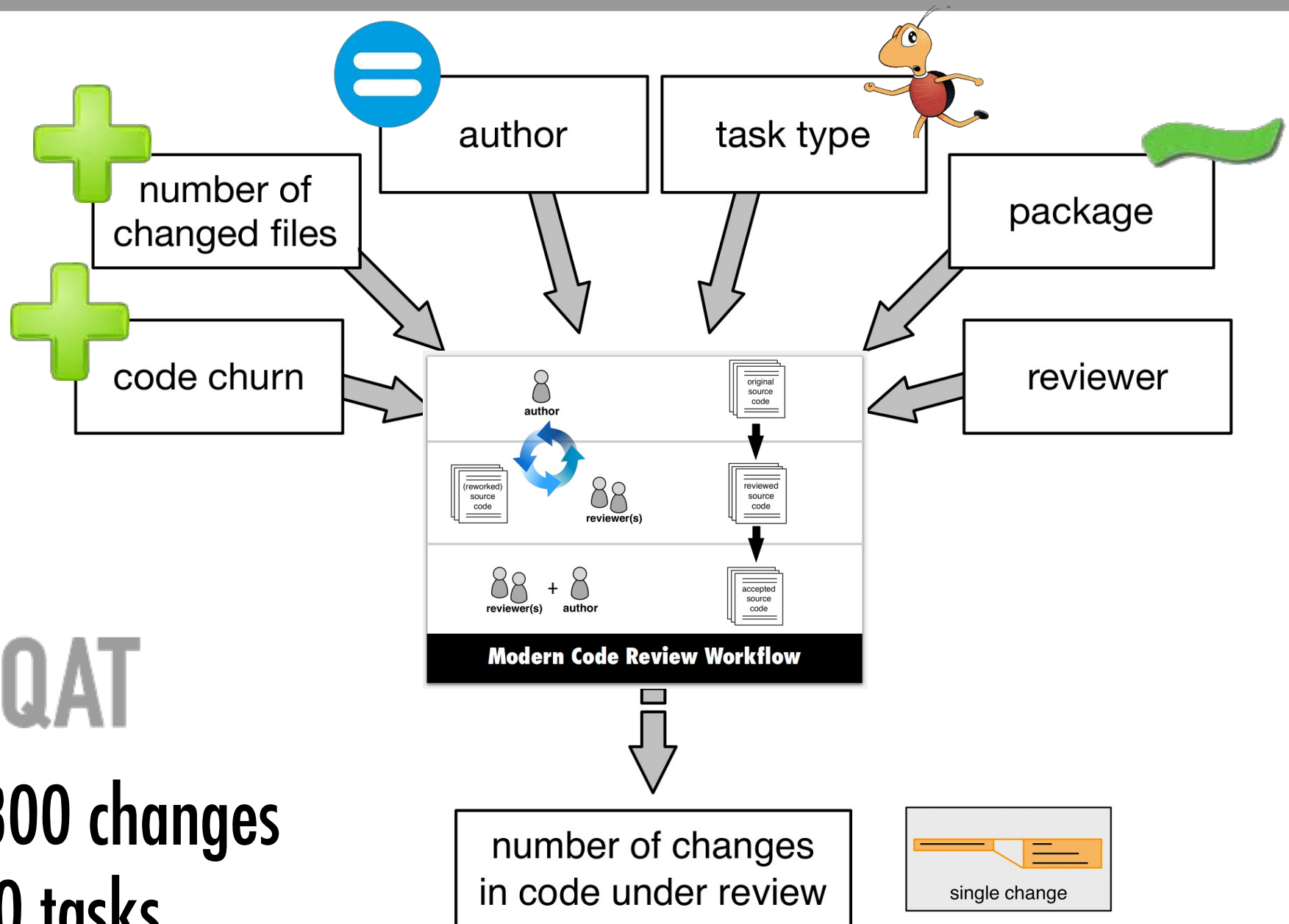


conQAT

> 2,800 changes  
> 970 tasks



# What Influences Code Review?

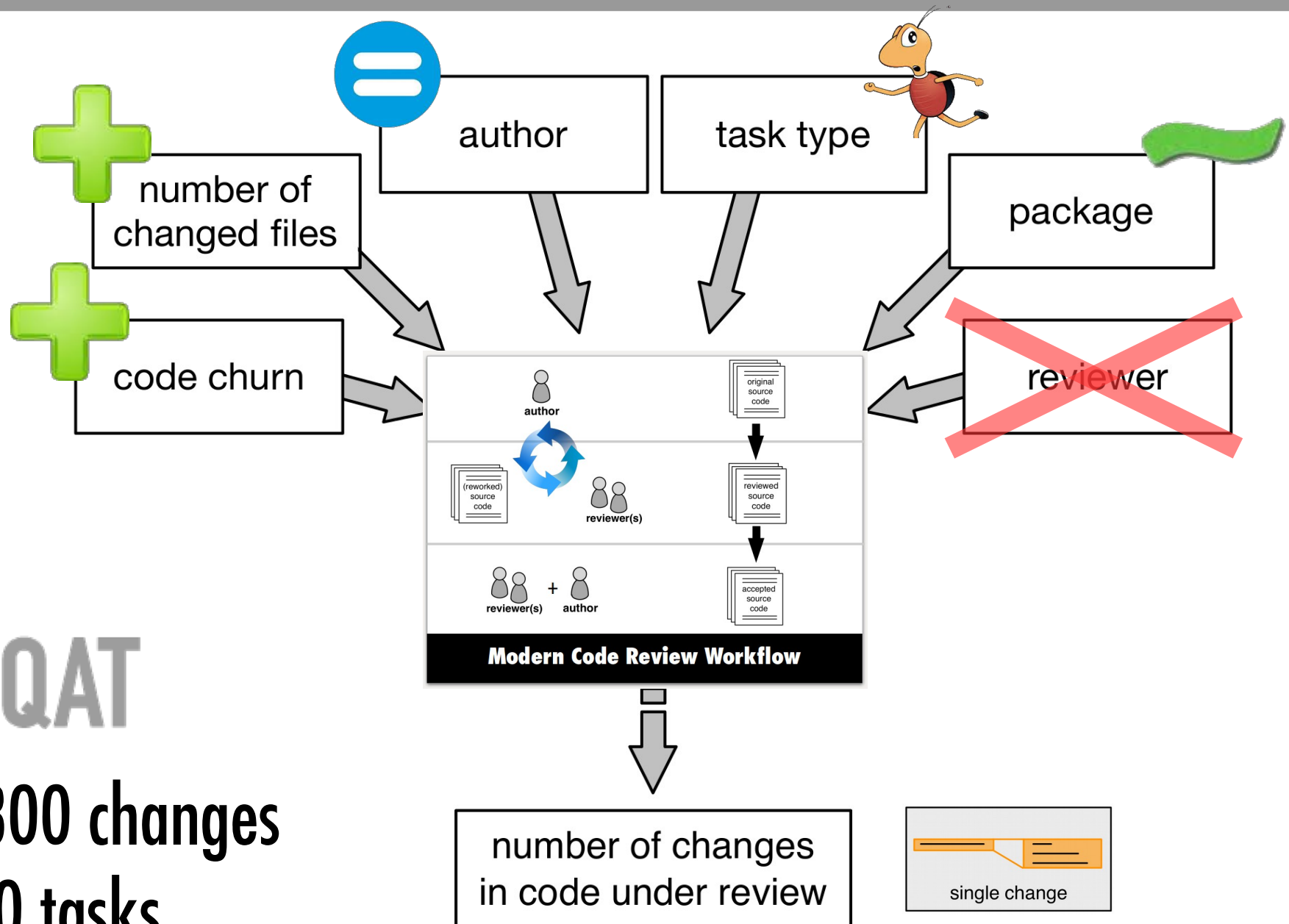


conQAT

> 2,800 changes

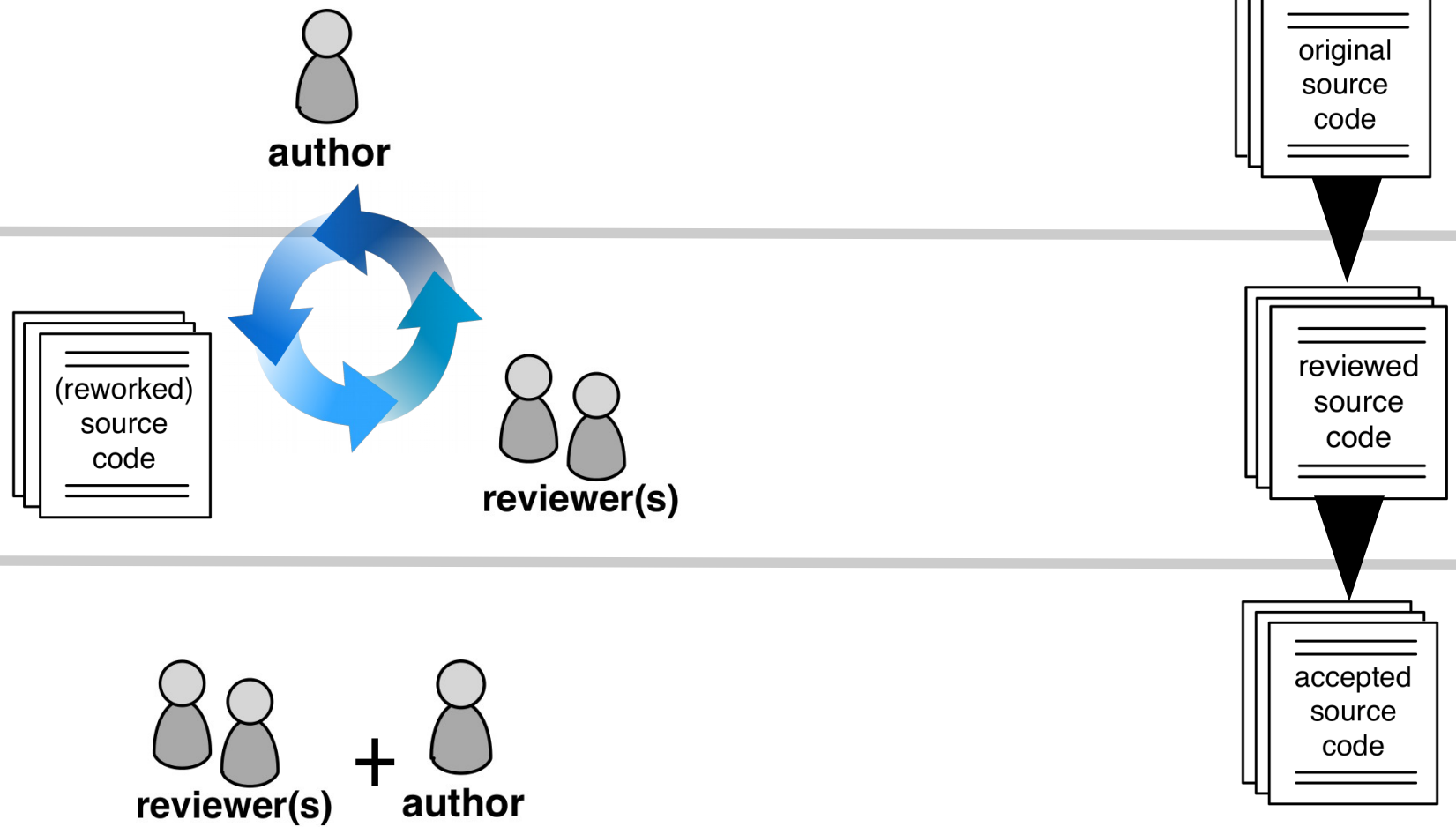
> 970 tasks

# What Influences Code Review?

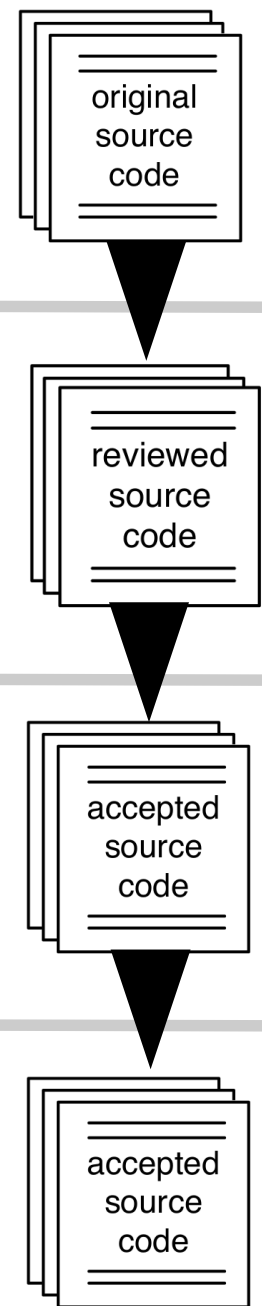
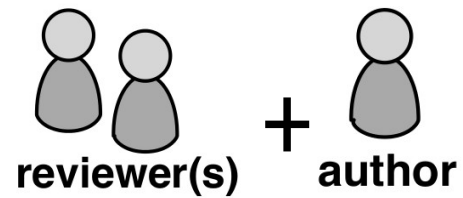
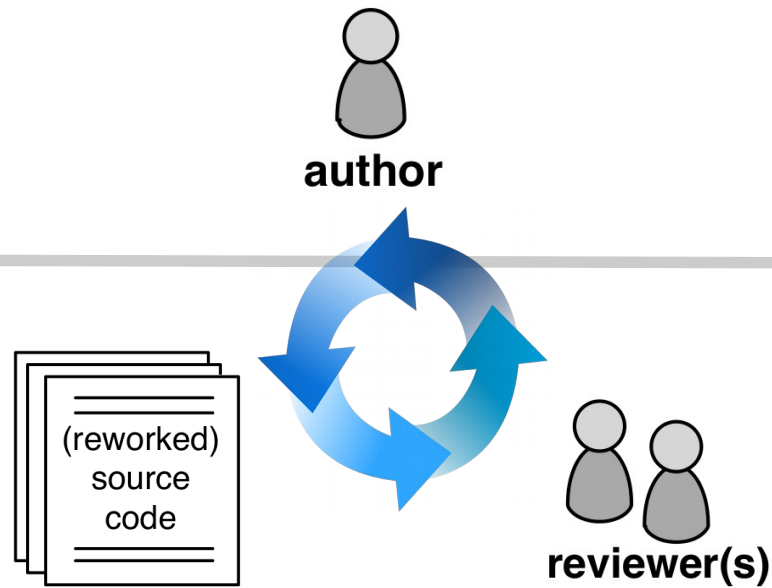


conQAT

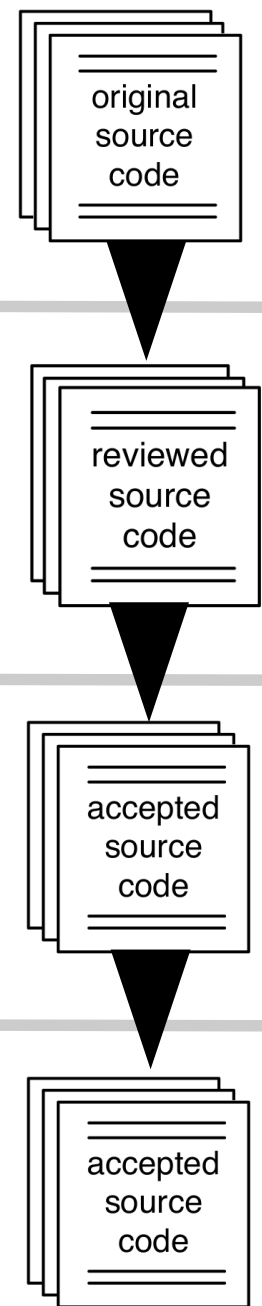
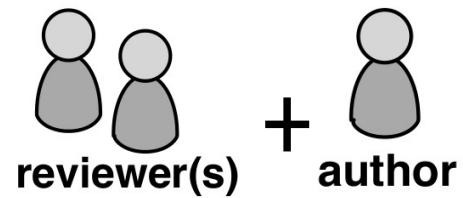
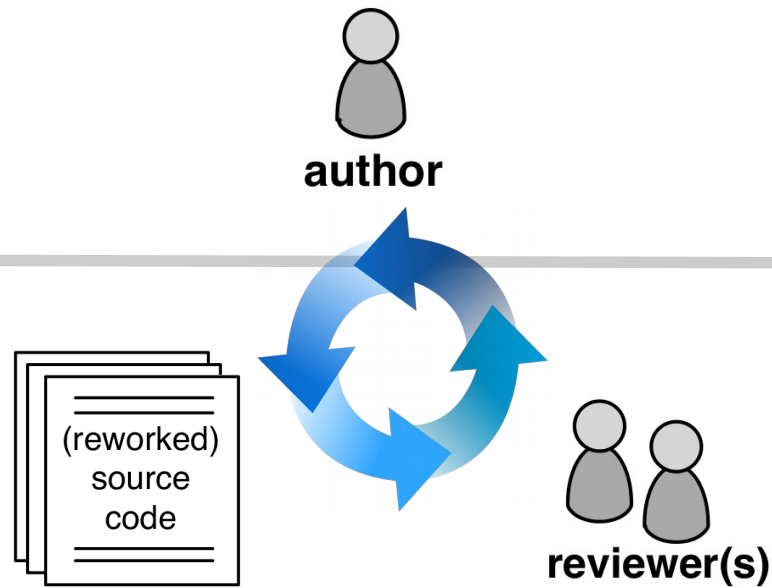
> 2,800 changes  
> 970 tasks



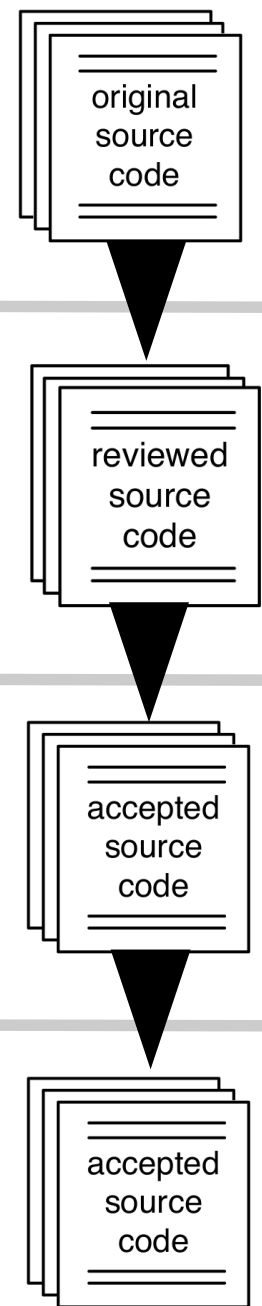
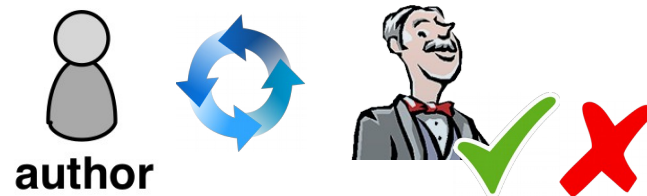
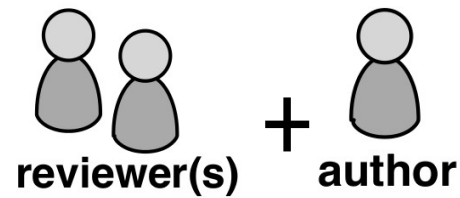
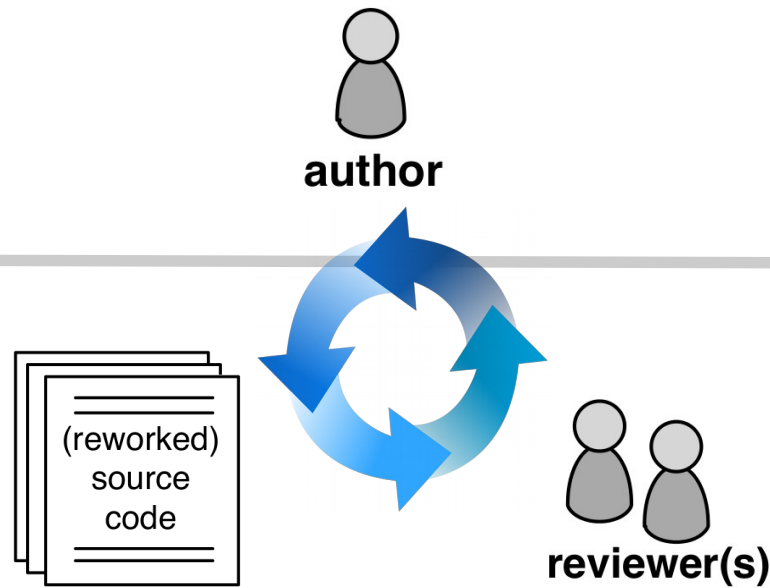
# **"Continuous" Integration Then (2014)**



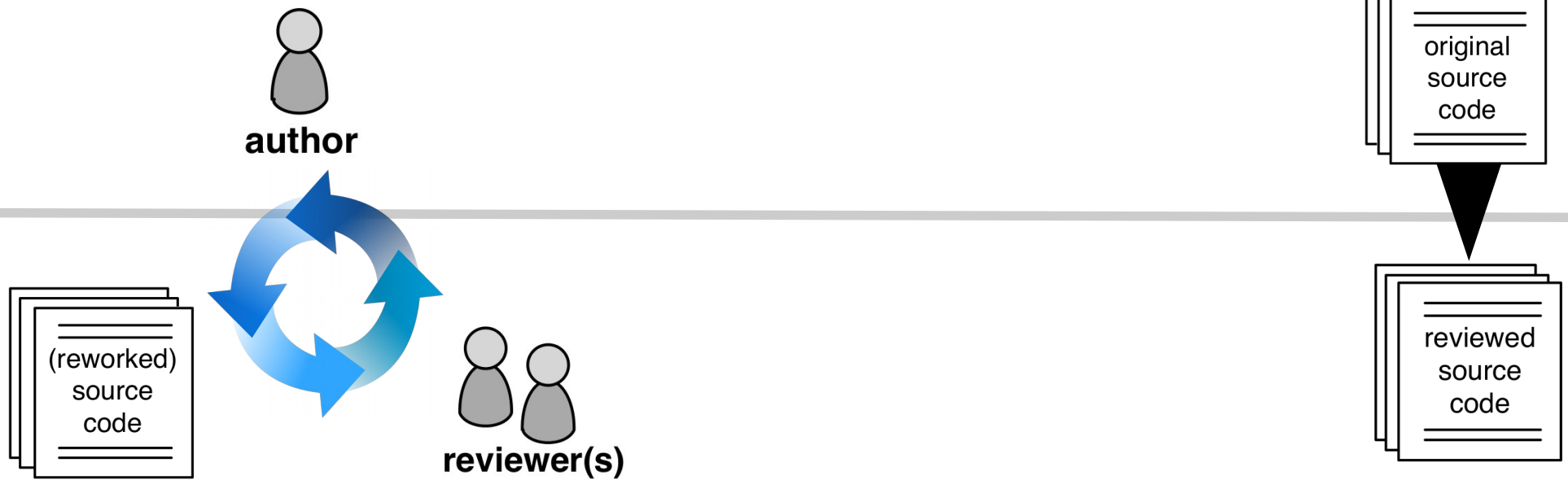
# **"Continuous" Integration Then (2014)**



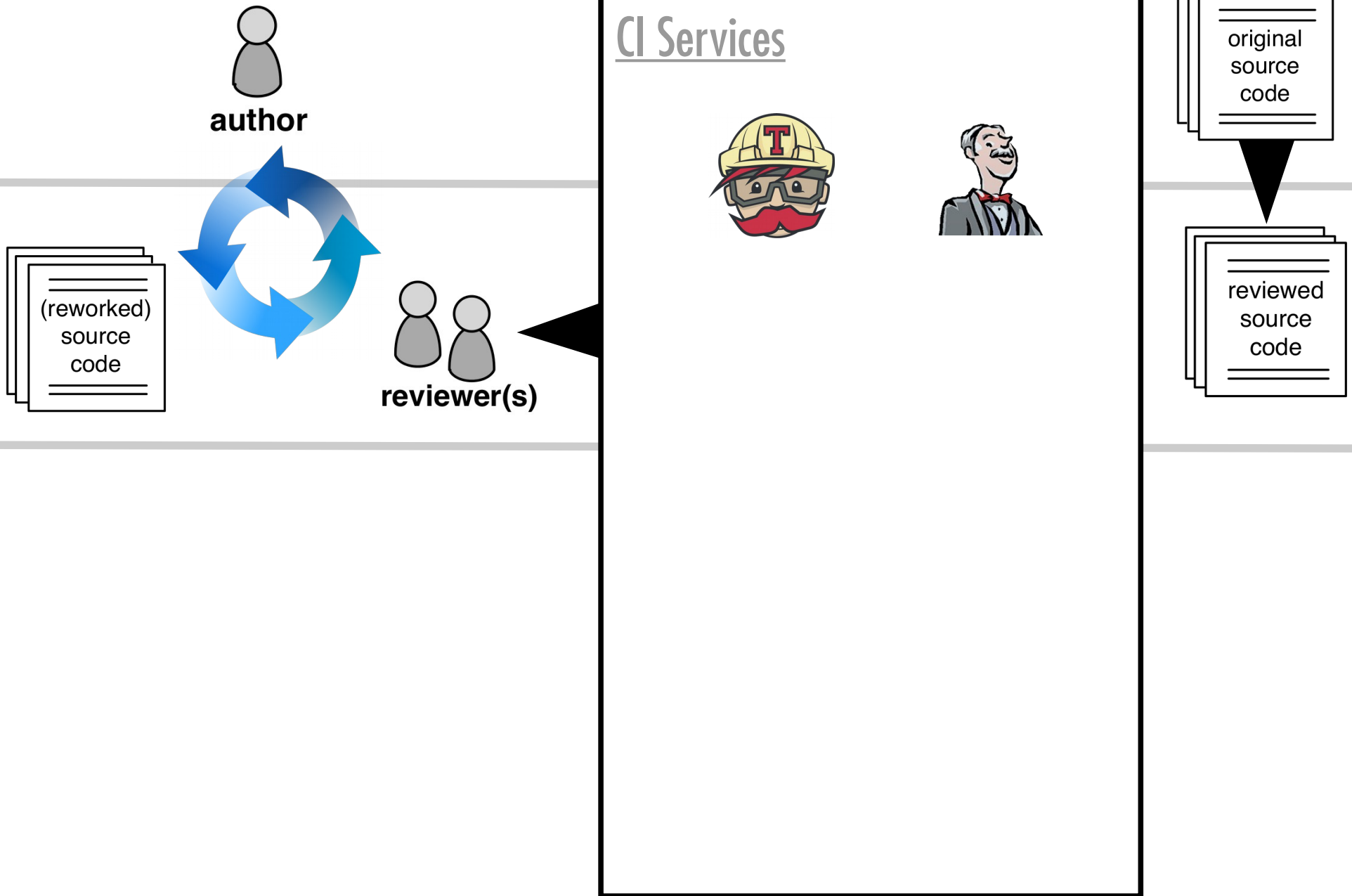
# "Continuous" Integration Then (2014)



# "Continuous" Integration Then (2014)

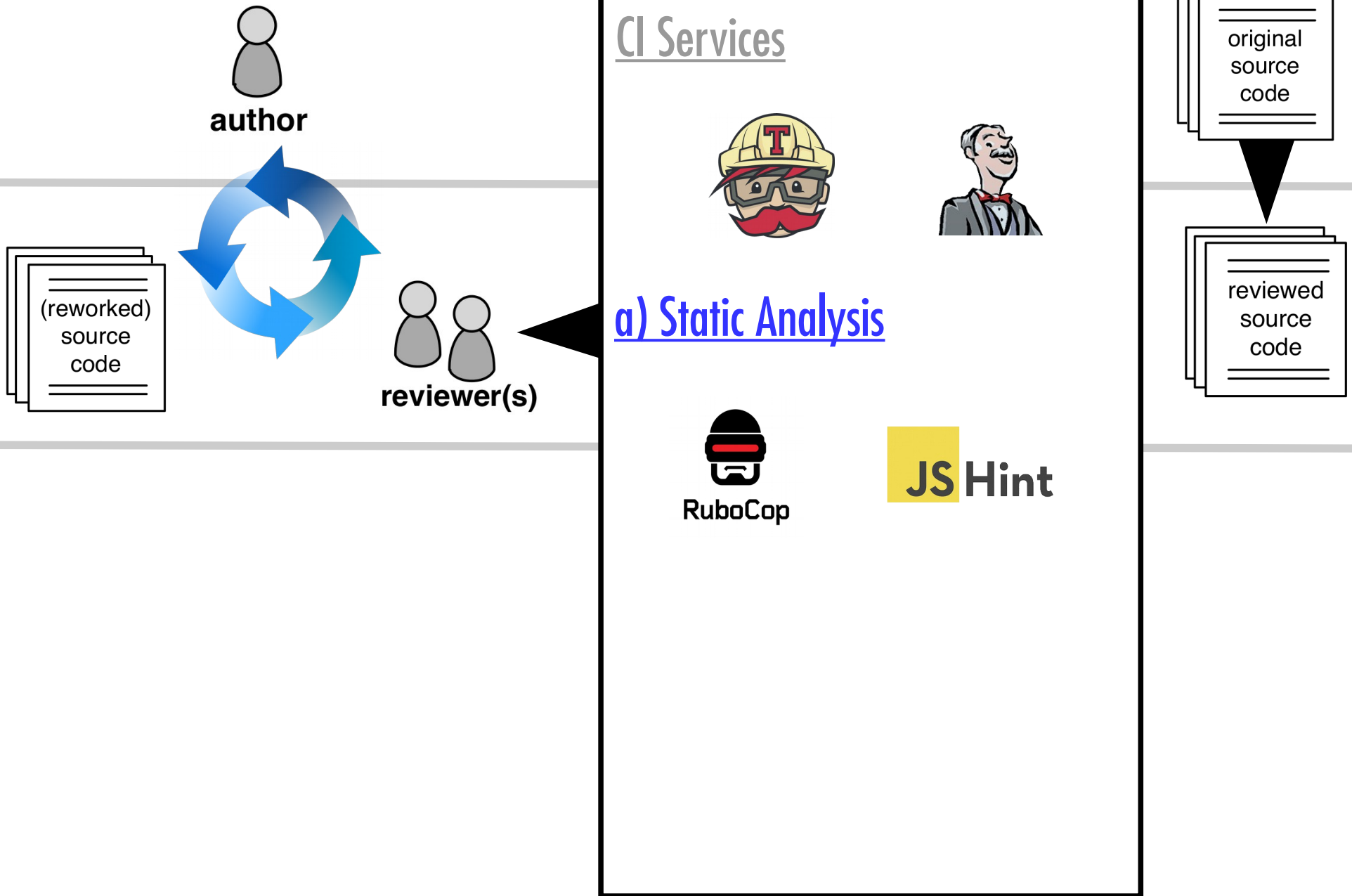


# Continuous Integration Now (2017)

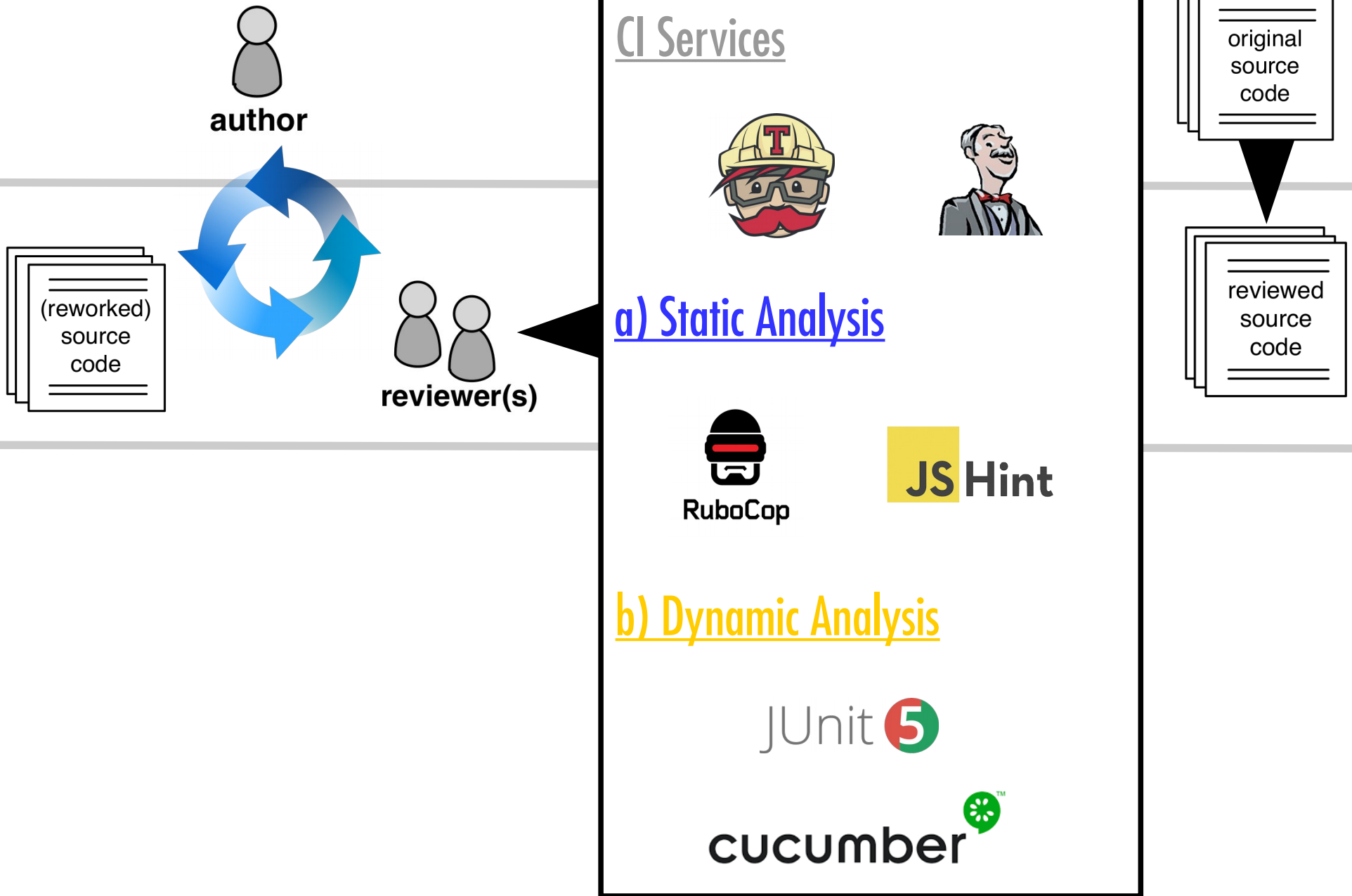


# Continuous Integration Now (2017)

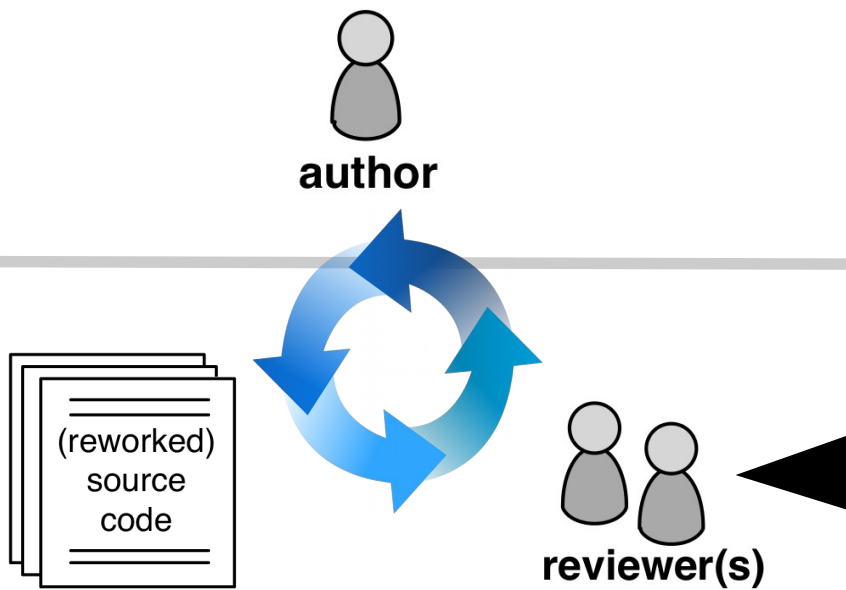




# Continuous Integration Now (2017)



# Continuous Integration Now (2017)



## CI Services



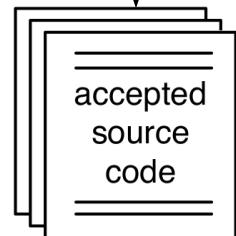
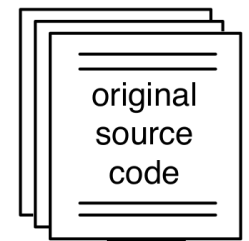
### a) Static Analysis



RuboCop

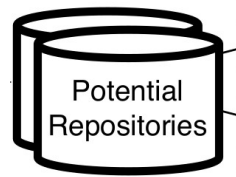


### b) Dynamic Analysis

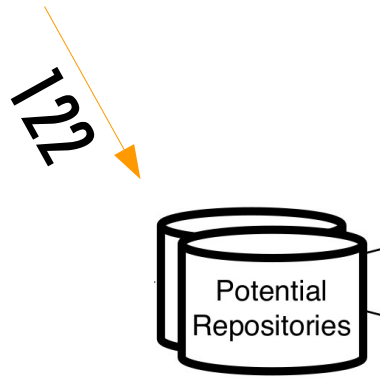


# Continuous Integration Now (2017)

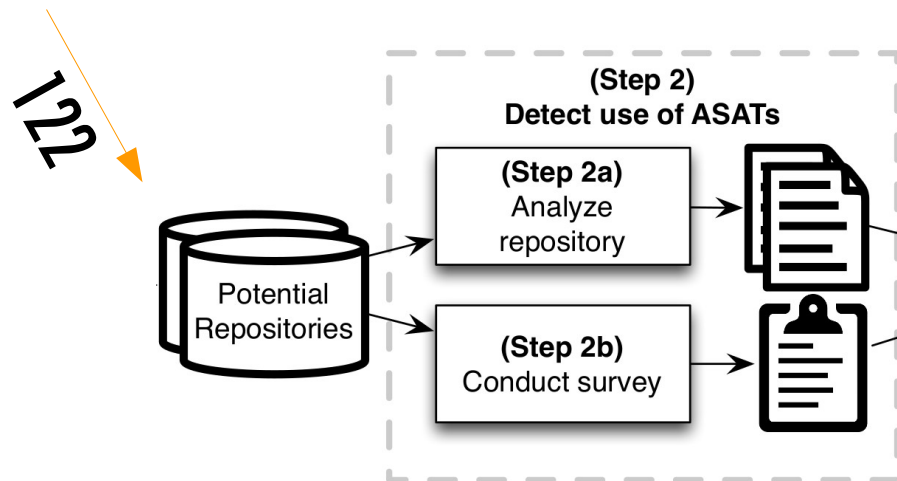
# a) How Prevalent Is Static Analysis?



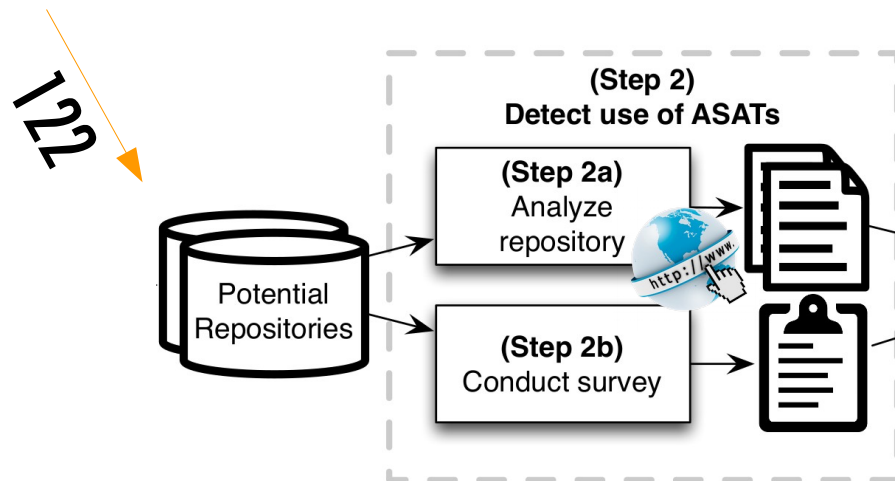
# a) How Prevalent Is Static Analysis?



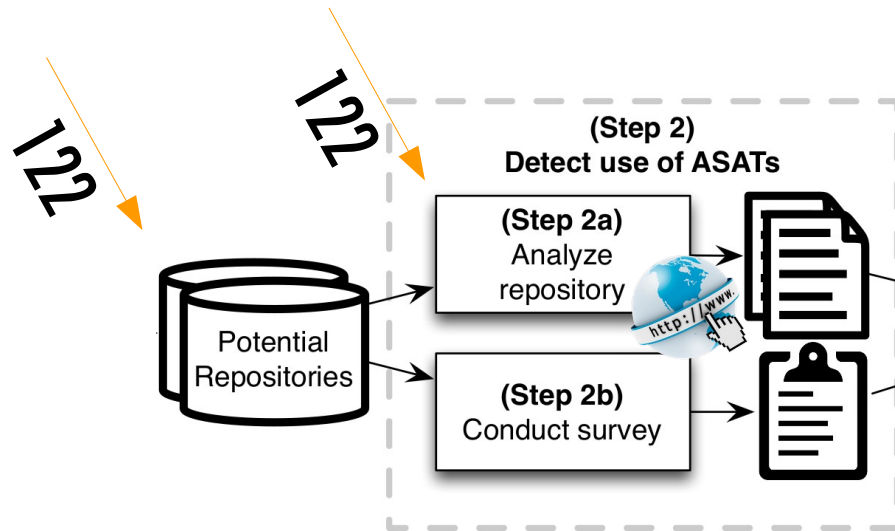
# a) How Prevalent Is Static Analysis?



# a) How Prevalent Is Static Analysis?

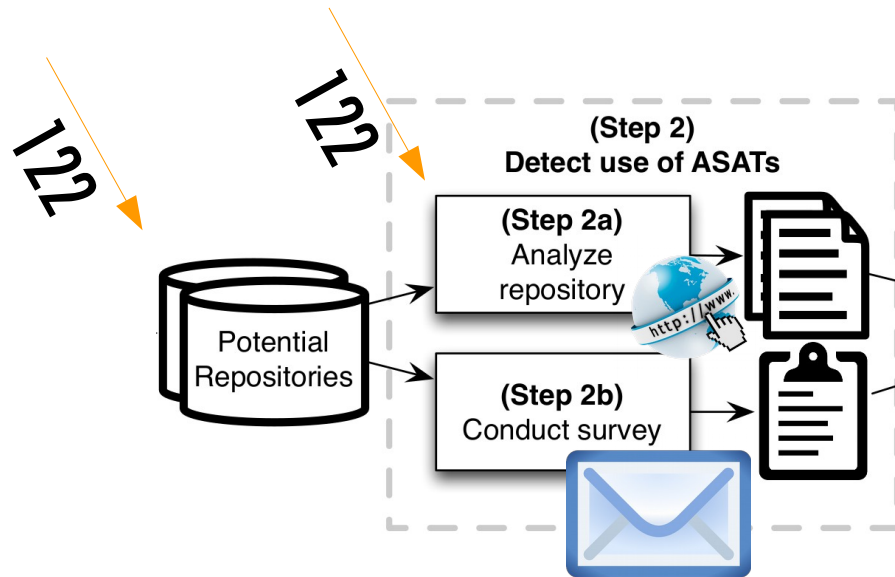


# a) How Prevalent Is Static Analysis?

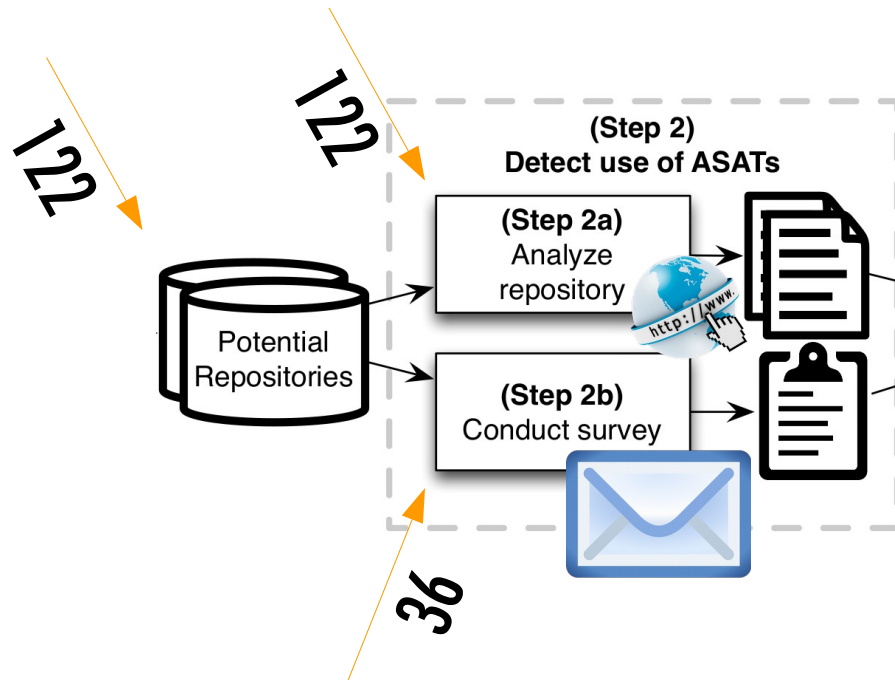




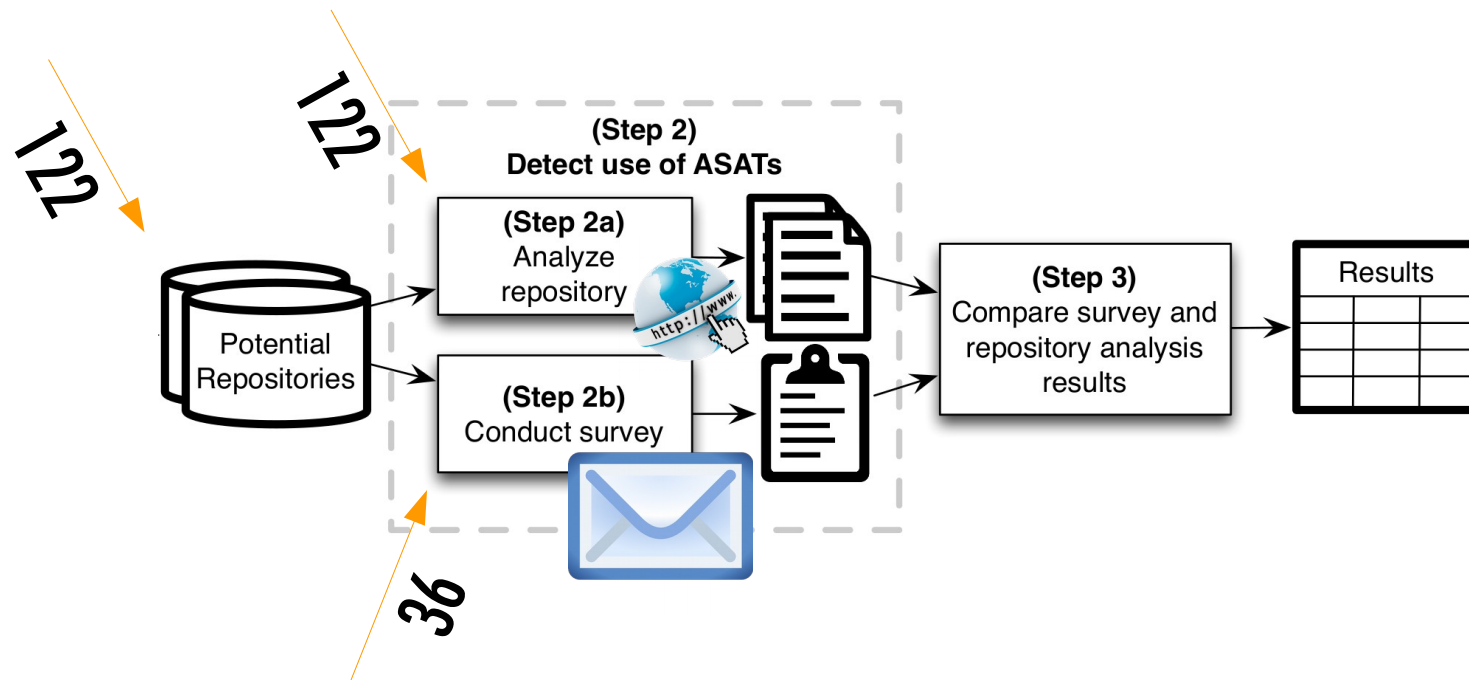
# a) How Prevalent Is Static Analysis?



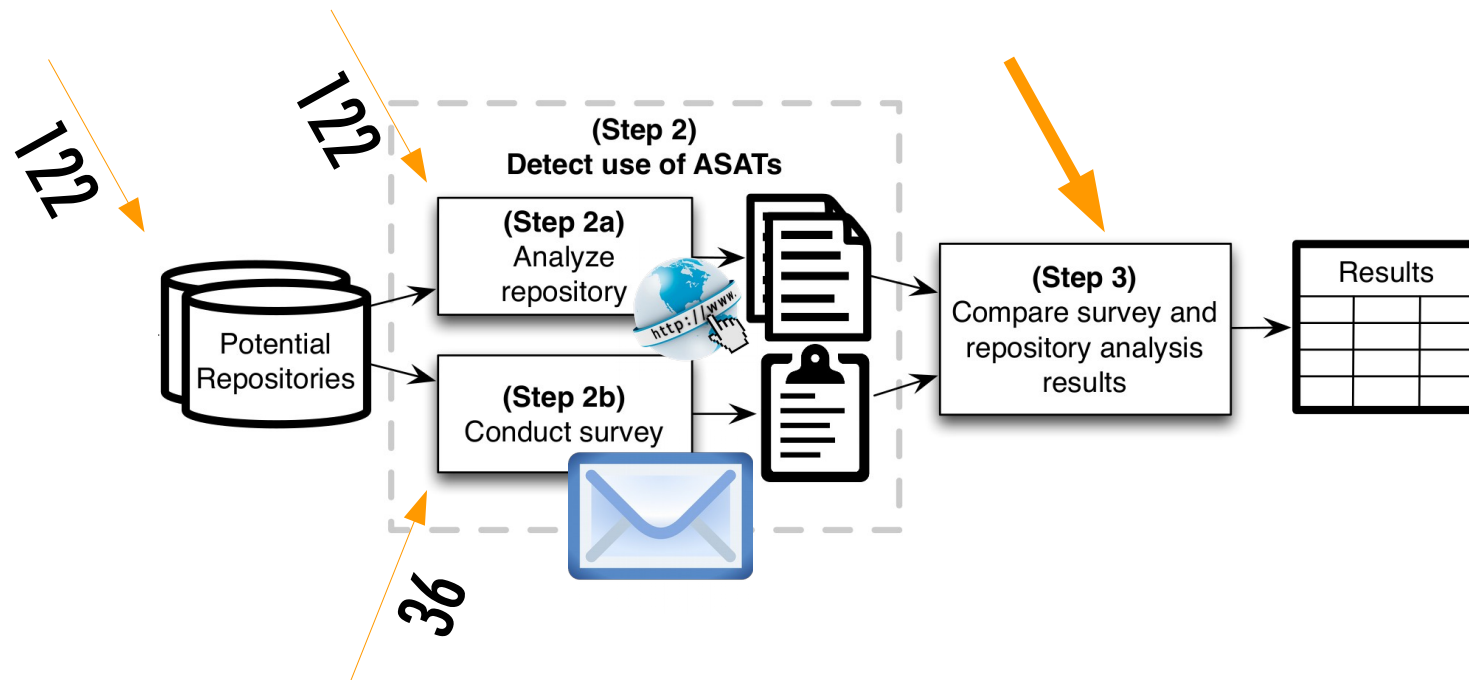
# a) How Prevalent Is Static Analysis?



# a) How Prevalent Is Static Analysis?





# a) How Prevalent Is Static Analysis?





# **a) How Prevalent Is Static Analysis?**



# a) How Prevalent Is Static Analysis?

<i>Source</i>	<i>Amount of Projects</i>	<i>Using ASATs</i>	<i>Using &gt;1 ASAT</i>	<i>Enforcing ASAT</i>
				
				

# a) How Prevalent Is Static Analysis?



<i>Source</i>	<i>Amount of Projects</i>	<i>Using ASATs</i>	<i>Using &gt;1 ASAT</i>	<i>Enforcing ASAT</i>
	122			
	36			

# a) How Prevalent Is Static Analysis?



<i>Source</i>	<i>Amount of Projects</i>	<i>Using ASATs</i>	<i>Using &gt;1 ASAT</i>	<i>Enforcing ASAT</i>
	122	59%		
	36	77%		





# a) How Prevalent Is Static Analysis?

<i>Source</i>	<i>Amount of Projects</i>	<i>Using ASATs</i>	<i>Using &gt;1 ASAT</i>	<i>Enforcing ASAT</i>
	122	59%	23%	
	36	77%	36%	

# a) How Prevalent Is Static Analysis?

<i>Source</i>	<i>Amount of Projects</i>	<i>Using ASATs</i>	<i>Using &gt;1 ASAT</i>	<i>Enforcing ASAT</i>
	122	59%	23%	-
	36	77%	36%	36%

# a) How Prevalent Is Static Analysis?

<i>Source</i>	<i>Amount of Projects</i>	<i>Using ASATs</i>	<i>Using &gt;1 ASAT</i>	<i>Enforcing ASAT</i>
	122	59%	23%	-
	36	77%	36%	36%



Moritz Beller, Radjino Bholanath, Shane McIntosh, Andy Zaidman:

***Analyzing the State of Static Analysis: A Large-Scale Evaluation in Open Source Software***, SANER, Osaka (Japan), 2016

## **b) How Prevalent Is Dynamic Analysis?**

## **b) How Prevalent Is Dynamic Analysis?**

***CI is the best practice in which developers not only integrate their work into a shared mainline frequently, but also verify the quality continuously through testing.***

## **b) How Prevalent Is Dynamic Analysis?**

***CI is the best practice in which developers not only integrate their work into a shared mainline frequently, but also verify the quality continuously through testing.***

It is known.



<TL;DR> . . . </TL;DR>



<TL;DR> . . . </TL;DR>

---

T E S T S

---

Running nl.tudelft.watchdog.ClientVersionCheckerTest

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time  
elapsed: 0.04 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

<TL;DR>...</TL;DR>

---

## T E S T S

---

Running nl.tudelft.watchdog.ClientVersionCheckerTest

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time  
elapsed: 0.04 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

<TL;DR>...</TL;DR>

<TL;DR>...</TL;DR>

---


## T E S T S

---

Running nl.tudelft.watchdog.ClientVersionCheckerTest

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time  
elapsed: 0.04 sec

Results :



Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

<TL;DR>...</TL;DR>

<TL;DR>...</TL;DR>

---

## T E S T S

---

Running nl.tudelft.watchdog.ClientVersionCheckerTest

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time  
elapsed: 0.04 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

<TL;DR>...</TL;DR>

<TL;DR>...</TL;DR>

---

## T E S T S

---

Running nl.tudelft.watchdog.ClientVersionCheckerTest

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time  
elapsed: 0.04 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

<TL;DR>...</TL;DR>

<TL;DR> . . . </TL;DR>

-----  
T E S T S  
-----

Running nl.tudelft.watchdog.ClientVersionCheckerTest  
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time  
elapsed: 0.04 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

<TL;DR> . . . </TL;DR>

- Parsing Log files for Java: Maven, Gradle, Ant

<TL;DR> . . . </TL;DR>

-----  
T E S T S  
-----

Running nl.tudelft.watchdog.ClientVersionCheckerTest  
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time  
elapsed: 0.04 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

<TL;DR> . . . </TL;DR>

- Parsing Log files for Java: Maven, Gradle, Ant



<TL;DR> . . . </TL;DR>

-----  
T E S T S  
-----

Running nl.tudelft.watchdog.ClientVersionCheckerTest  
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time  
elapsed: 0.04 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

<TL;DR> . . . </TL;DR>

- Parsing Log files for Java: Maven, Gradle, Ant





<TL;DR> . . . </TL;DR>

-----  
T E S T S  
-----

Running nl.tudelft.watchdog.ClientVersionCheckerTest  
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time  
elapsed: 0.04 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

<TL;DR> . . . </TL;DR>

- Parsing Log files for Java: Maven, Gradle, Ant



- Parsing Log files for Ruby: bundler and rake

<TL;DR> . . . </TL;DR>

-----  
T E S T S  
-----

Running nl.tudelft.watchdog.ClientVersionCheckerTest  
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time  
elapsed: 0.04 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

<TL;DR> . . . </TL;DR>

- Parsing Log files for Java: Maven, Gradle, Ant



- Parsing Log files for Ruby: bundler and rake



## **b) How Prevalent Is Dynamic Analysis?**

## **b) How Prevalent Is Dynamic Analysis?**

- **69% of Java and 87.5% of the Ruby projects:  
test runs**

## **b) How Prevalent Is Dynamic Analysis?**

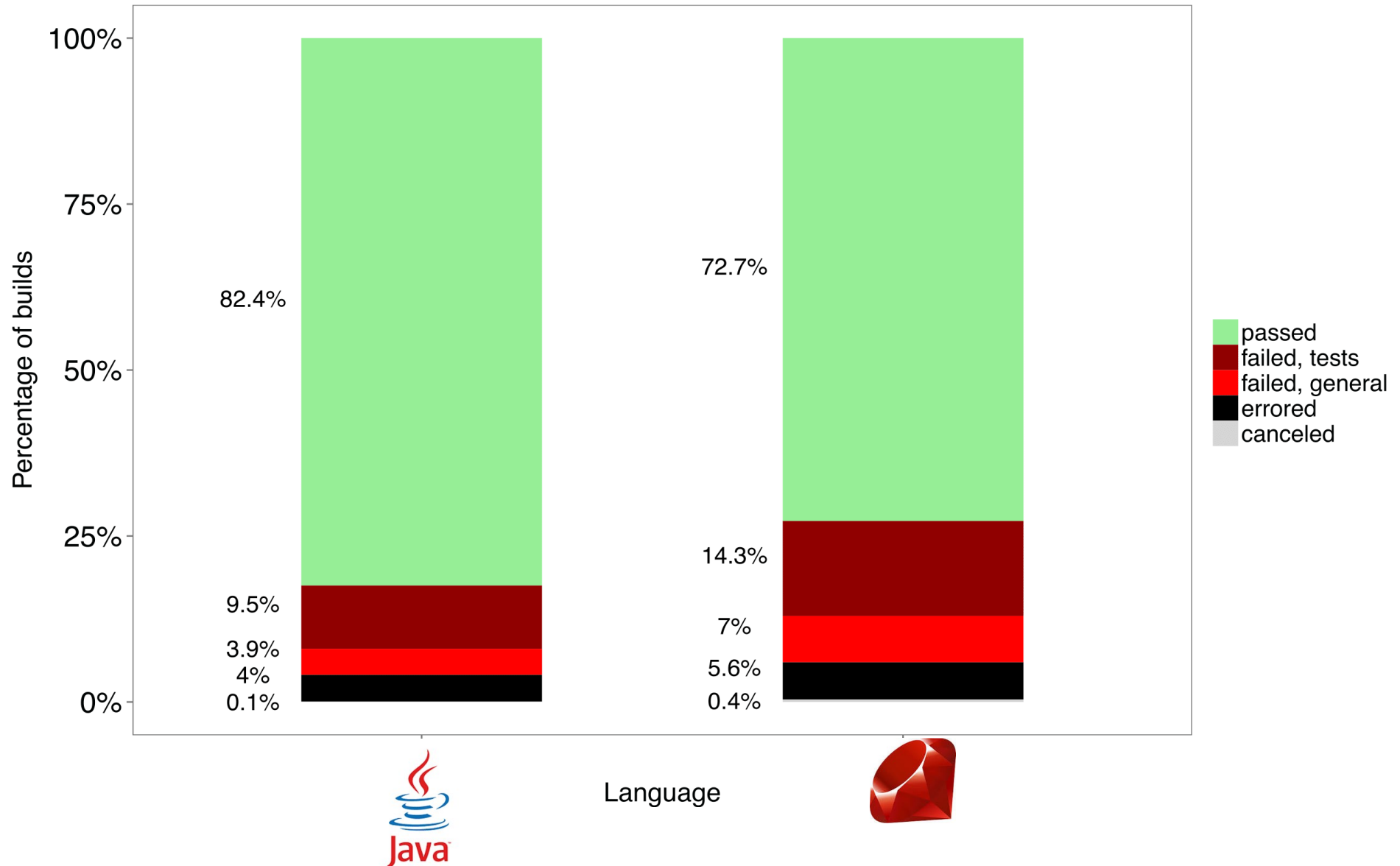
- **69% of Java and 87.5% of the Ruby projects:  
test runs**
- **Overall, 81% of the projects:  
test runs**

## **b) How Prevalent Is Dynamic Analysis?**

- **69% of Java and 87.5% of the Ruby projects:  
test runs**
- **Overall, 81% of the projects:  
test runs**
- **96% of builds:  
test runs**

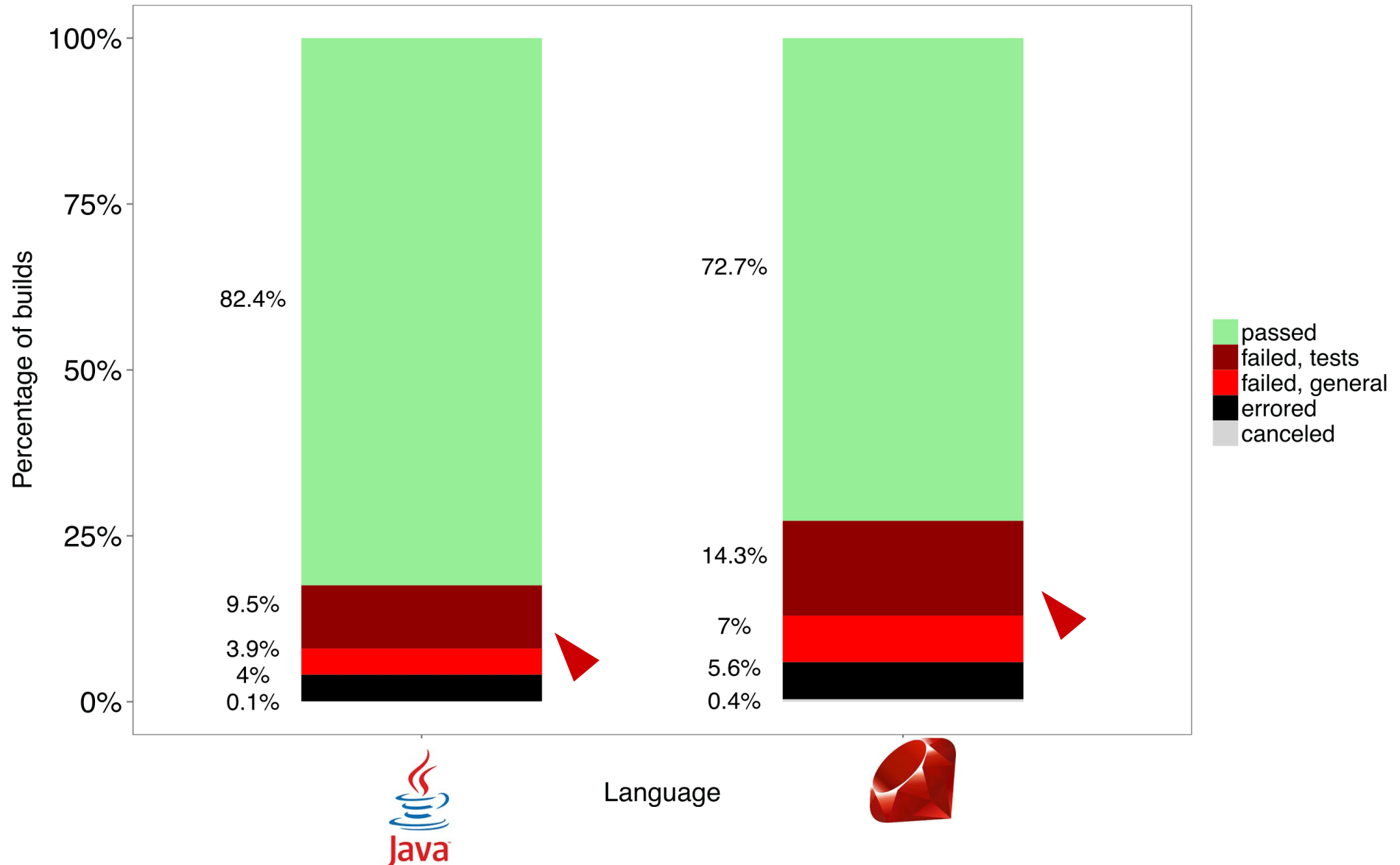
## **b) How Prevalent Is Dynamic Analysis?**

## b) How Prevalent Is Dynamic Analysis?





## b) How Prevalent Is Dynamic Analysis?



## **b) Testing Is Central To CI**

***Testing is the single most important reason for broken builds, more prevalent than compile errors, missing dependencies, build cancellations and provisioning problems together.***

## **b) Testing Is Central To CI**

***Testing is the single most important reason for broken builds, more prevalent than compile errors, missing dependencies, build cancellations and provisioning problems together.***



Moritz Beller, Georgios Gousios, Andy Zaidman:

***Oops, My Tests Broke the Build: An Explorative Analysis of Travis CI with GitHub, MSR, Buenos Aires (Argentina), 2017***

# What Is TravisTorrent?

[HOME](#)[ACCESS TRAVIS](#)

## TravisTorrent

Free and Open Travis Analytics for Everyone

TravisTorrent, a [GHTorrent](#) partner project, provides free and easy-to-use [Travis CI](#) build analyses to the masses through its open database.

TravisTorrent is the [MSR'2017 Mining Challenge](#). By extracting cool facts from the data set, you can win awesome prizes, courtesy of Travis CI (deadline 20 Feb. 2017)!

TravisTorrent provides access to a database of hundreds of thousands of analyzed travis builds in **under 10 seconds**.

[Access TravisTorrent now](#)

# What Is TravisTorrent?

[HOME](#)[ACCESS TRAVIS](#)

## TravisTorrent

Free and Open Travis Analytics for Everyone

TravisTorrent, a [GHTorrent](#) partner project, provides free and easy-to-use [Travis CI](#) build analyses to the masses through its open database.

TravisTorrent is the [MSR'2017 Mining Challenge](#). By extracting cool facts from the data set, you can win awesome prizes, courtesy of Travis CI (deadline 20 Feb. 2017)!

TravisTorrent provides access to a database of hundreds of thousands of analyzed travis builds in **under 10 seconds**.

[Access TravisTorrent now](#)



**MSR Mining  
Challenge 2017**

**Google**

**Public Data Set**

# What Is TravisTorrent?

[HOME](#)[ACCESS TRAVIS](#)

## TravisTorrent

Free and Open Travis Analytics for Everyone

TravisTorrent, a [GHTorrent](#) partner project, provides free and easy-to-use [Travis CI](#) build analyses to the masses through its open database.

Free\* SQL-like cloud query interface on Google BQ

TravisTorrent provides access to a database of hundreds of thousands of analyzed Travis builds in **under 10 seconds**.

[Access TravisTorrent now](#)



**MSR Mining  
Challenge 2017**

**Google**

**Public Data Set**

# What Is TravisTorrent?



HOME ACCESS TRAVI

# TravisTorrent.testroots.org

Free and Open Travis Analytics for Everyone

TravisTorrent, a [GHTorrent](#) partner project, provides free and easy-to-use [Travis CI](#) build analyses to the masses through its open database.

Free\* SQL-like cloud query interface on Google BQ

TravisTorrent provides access to a database of hundreds of thousands of analyzed travis builds in **under 10 seconds**.

Access TravisTorrent now



MSR Mining  
Challenge 2017

Google

Public Data Set

# **Research on Continuous Code Review Enabled By TravisTorrent**



# Research on Continuous Code Review

## Enabled By TravisTorrent



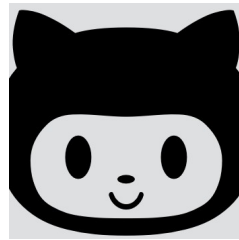
- git\_branch
- git\_all\_built\_commits
- ...

# Research on Continuous Code Review

## Enabled By TravisTorrent



- git\_branch
- git\_all\_built\_commits
- ...

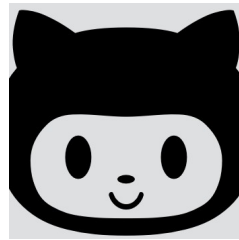


- gh\_project\_name
- gh\_is\_pr
- gh\_team\_size
- gh\_diff\_tests\_added
- gh\_first\_commit\_...

# Research on Continuous Code Review Enabled By TravisTorrent



- git\_branch
- git\_all\_built\_commits
- ...



- gh\_project\_name
- gh\_is\_pr
- gh\_team\_size
- gh\_diff\_tests\_added
- gh\_first\_commit\_...



- tr\_prev\_build
- tr\_status
- tr\_log\_tests\_ran
- tr\_log\_tests\_failed
- ...

# Research on Continuous Code Review Enabled By TravisTorrent

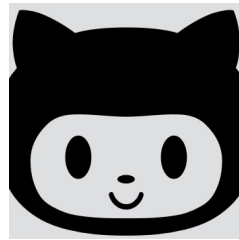


## TravisTorrent

Free and Open Travis Analytics for Everyone



- git\_branch
- git\_all\_built\_commits
- ...



62 columns

- gh\_project\_name
- gh\_is\_pr
- gh\_team\_size
- gh\_diff\_tests\_added
- gh\_first\_commit\_...



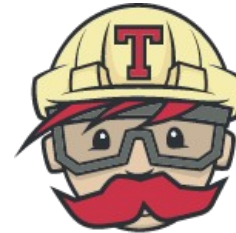
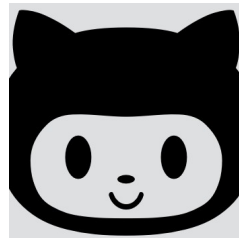
- tr\_prev\_build
- tr\_status
- tr\_log\_tests\_ran
- tr\_log\_tests\_failed
- ...

# Research on Continuous Code Review Enabled By TravisTorrent



## TravisTorrent

Free and Open Travis Analytics for Everyone



- git\_branch
- git\_all\_built\_commits
- ...

62 columns

- gh\_project\_name
- gh\_is\_pr
- gh\_team\_size
- gh\_diff\_tests\_added
- gh\_first\_commit\_...

- tr\_prev\_build
- tr\_status
- tr\_log\_tests\_ran
- tr\_log\_tests\_failed
- ...

# Research on Continuous Code Review Enabled By TravisTorrent



## TravisTorrent

Free and Open Travis Analytics for Everyone



- git\_branch
- git\_all\_built\_commits
- ...

62 columns

- gh\_project\_name
- gh\_is\_pr
- gh\_team\_size
- gh\_diff\_tests\_added
- gh\_first\_commit\_...

- tr\_prev\_build
- tr\_status
- tr\_log\_tests\_ran
- tr\_log\_tests\_failed
- ...

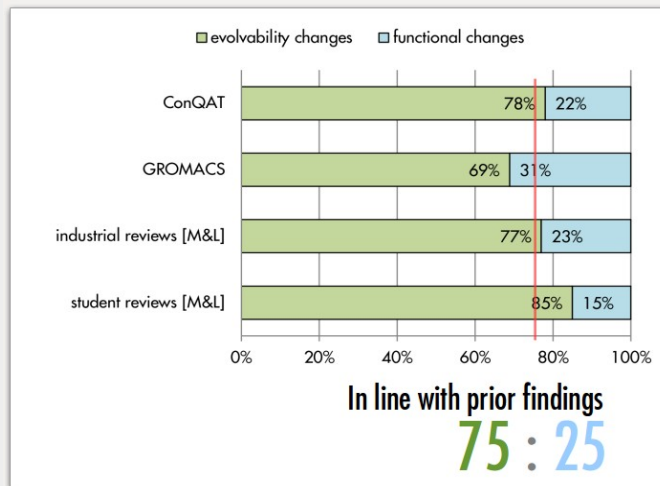
# **Modern Code Reviews in Open-Source Projects: Which Problems Do They Fix?**



**@Inventitech**

Moritz Beller, TU Delft

# Modern Code Reviews in Open-Source Projects: Which Problems Do They Fix?

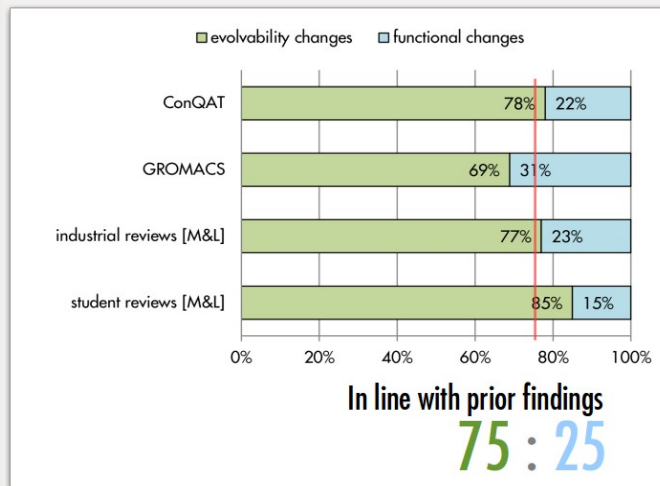


@Inventitech

Moritz Beller, TU Delft



# Modern Code Reviews in Open-Source Projects: Which Problems Do They Fix?



## RQ1: How Prevalent Are ASATs?

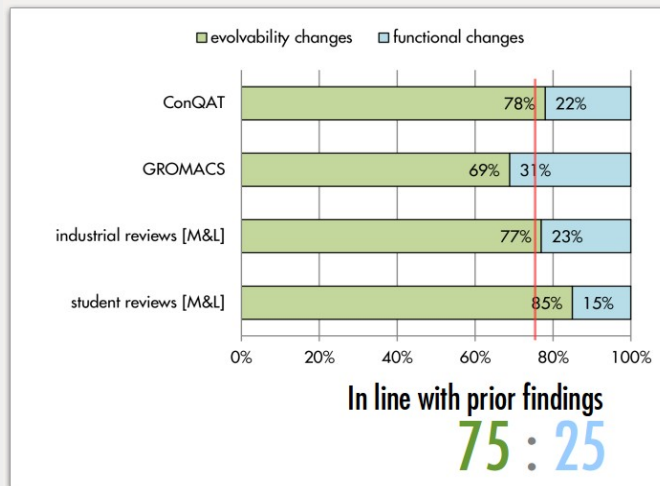
Source	Amount of Projects	% of Projects Using ASATs
Using ASATs	28/36	77%
Using >1 ASAT	13/36	36%
Enforcing ASAT	13/36	36%



@Inventitech

Moritz Beller, TU Delft

# Modern Code Reviews in Open-Source Projects: Which Problems Do They Fix?



## RQ1: How Prevalent Are ASATs?

Source	Amount of Projects	% of Projects Using ASATs
Using ASATs	28/36	77%
Using >1 ASAT	13/36	36%
Enforcing ASAT	13/36	36%

### What Is TravisTorrent?

HOME ACCESS TRAVIS

## TravisTorrent.testroots.org

Free and Open Travis Analytics for Everyone

TravisTorrent, a GitHub partner project, provides free and easy-to-use Travis CI build analyses to the masses through its open database.

Free\* SQL-like cloud query interface on Google BQ

TravisTorrent provides access to a database of hundreds of thousands of analyzed Travis builds in under 10 seconds.

Access TravisTorrent now

MSR Mining Challenge 2017

Google Public Data Set



@Inventitech

Moritz Beller, TU Delft