

# Continuous Test Generation: Enhancing Continuous Integration with Automated Test Generation

**José Campos**

jose.campos@sheffield.ac.uk  
University of Sheffield  
United Kingdom

**Andrea Arcuri**

arcuri82@gmail.com  
Certus Software V&V Center  
Simula Research Laboratory, Norway

**Gordon Fraser**

gordon.fraser@sheffield.ac.uk  
University of Sheffield  
United Kingdom

**Rui Abreu**

rui@computer.org  
University of Porto  
Portugal

September 17th, 2014

29th IEEE/ACM International Conference on Automated Software Engineering (ASE)  
Västerås, Sweden

# Continuous **Test Generation** (**CTG**)

# EVOSUITE

<http://www.evosite.org/>



<http://www.pexforfun.com/>

Package Explorer

commons-math3

src/main/java

- org.apache.commons.math3
- org.apache.commons.math3.analysis
- org.apache.commons.math3.analysis.differenti
- org.apache.commons.math3.analysis.function
- org.apache.commons.math3.analysis.integratio
- org.apache.commons.math3.analysis.integratio
- org.apache.commons.math3.analysis.interpolat
- org.apache.commons.math3.analysis.polynomial
- org.apache.commons.math3.analysis.solvers
- org.apache.commons.math3.complex
- org.apache.commons.math3.dfp
- org.apache.commons.math3.distribution
- org.apache.commons.math3.distribution.fittin
- org.apache.commons.math3.exception
- org.apache.commons.math3.exception.util
- org.apache.commons.math3.filter
- org.apache.commons.math3.fitting
- org.apache.commons.math3.fitting.leastsquare
- org.apache.commons.math3.fraction
- org.apache.commons.math3.genetics
- org.apache.commons.math3.geometry
- org.apache.commons.math3.geometry.enclosing
- org.apache.commons.math3.geometry.euclidean

commons-math3



## Package Explorer

commons-math3

src/main/java

- org.apache.commons.math3
  - org.apache.commons.math3.analysis
    - org.apache.commons.math3.analysis.differenti
    - org.apache.commons.math3.analysis.function
    - org.apache.commons.math3.analysis.integrati
    - org.apache.commons.math3.analysis.integrati
    - org.apache.commons.math3.analysis.interpolat
    - org.apache.commons.math3.analysis.polynomial
    - org.apache.commons.math3.analysis.solvers
  - org.apache.commons.math3.complex
    - ComplexField.java
    - ComplexFormat.java
    - ComplexUtils.java
    - package-info.java
    - Quaternion.java
    - RootsOfUnity.java
  - org.apache.commons.math3.dfp
  - org.apache.commons.math3.distribution
  - org.apache.commons.math3.distribution.fittin
  - org.apache.commons.math3.exception
  - org.apache.commons.math3.exception.util
  - org.apache.commons.math3.filter

## ComplexField.java

```
28  * This class is a singleton.
29  * </p>
30  * @see Complex
31  * @version $Id: ComplexField.java 1416643 2012-12-03 19:37:14Z
32  * @since 2.0
33  */
34  public class ComplexField implements Field<Complex>, Serializabl
35
36  /** Serializable version identifier. */
37  private static final long serialVersionUID = -61303626887007
38
39  /** Private constructor for the singleton.
40   */
41  private ComplexField() {
42  }
43
44  /** Get the unique instance.
45   * @return the unique instance
46   */
47  public static ComplexField getInstance() {
48      return LazyHolder.INSTANCE;
49  }
50
51  /** {@inheritDoc} */
52  public Complex getOne() {
53      return Complex.ONE;
54  }
55
56  /** {@inheritDoc} */
57  public Complex getZero() {
58      return Complex.ZERO;
59  }
60
61  /** {@inheritDoc} */
62  public Class<? extends FieldElement<Complex>> getRuntimeClass
63      return Complex.class;
```

Java - commons-math3

File Edit Source Refactor

Package Explorer

- commons-math3
  - src/main/java
    - org.apache.commons
    - org.apache.commons
    - org.apache.commons
    - org.apache.commons
    - org.apache.commons
    - org.apache.commons
    - org.apache.commons
    - org.apache.commons
    - org.apache.commons
    - org.apache.commons
      - Complex.java
      - ComplexField.java**
      - ComplexFormat.java
      - ComplexUtils.java
      - package-info.java
      - Quaternion.java
      - RootsOfUnity.java
    - org.apache.commons
    - org.apache.commons
    - org.apache.commons
    - org.apache.commons
    - org.apache.commons
    - org.apache.commons

Open With

Open Type Hierarchy F4

Show In Shift+Alt+W

Copy Ctrl+C

Copy Qualified Name

Paste Ctrl+V

Delete Delete

Remove from Context Shift+Ctrl+Alt+Down

Build Path

Source Shift+Alt+S

Refactor Shift+Alt+T

Import...

Export...

References

Declarations

Refresh F5

Assign Working Sets...

Debug As

Run As

Validate

Team

Compare With

Replace With

Restore from Local History...

EvoSuite

Properties Alt+Enter

Generate JUnit Test Suite

ComplexField.java - Eclipse

```
...eton.  
...Field.java 1416643 2012-12-03 19:37:14Z  
...d implements Field<Complex>, Serializabl  
...sion identifier. */  
... long serialVersionUID = -61303626887007  
...tor for the singleton.  
...()  
...instance.  
...e instance  
...exField getInstance() {  
...r.INSTANCE;  
...e() {  
...NE;  
...ro() {  
...ZERO;  
...ds FieldElement<Complex>> getRuntimeClass
```

org.apache.commons.math3.complex.ComplexField.java - commons-math3/src/main/java

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

commons-math3

src/main/java

org.apache.commons.math3

org.apache.commons.math3.analysis

org.apache.commons.math3.complex

org.apache.commons.math3.dfp

org.apache.commons.math3.distribution

org.apache.commons.math3.distribution.fitting

org.apache.commons.math3.exception

org.apache.commons.math3.exception.util

org.apache.commons.math3.filter

org.apache.commons.math3.optimization

org.apache.commons.math3.optimization.nonlinear

org.apache.commons.math3.optimization.nonlinear.gradient

org.apache.commons.math3.optimization.nonlinear.nelder-mead

org.apache.commons.math3.optimization.unconstrained

org.apache.commons.math3.optimization.unconstrained.nelder-mead

org.apache.commons.math3.optimization.unconstrained.nelder-mead.fmin

org.apache.commons.math3.optimization.unconstrained.nelder-mead.fmin.fmin

org.apache.commons.math3.optimization.unconstrained.nelder-mead.fmin.fmin.fmin

org.apache.commons.math3.optimization.unconstrained.nelder-mead.fmin.fmin.fmin.fmin

org.apache.commons.math3.optimization.unconstrained.nelder-mead.fmin.fmin.fmin.fmin.fmin

org.apache.commons.math3.optimization.unconstrained.nelder-mead.fmin.fmin.fmin.fmin.fmin.fmin

org.apache.commons.math3.optimization.unconstrained.nelder-mead.fmin.fmin.fmin.fmin.fmin.fmin.fmin

org.apache.commons.math3.optimization.unconstrained.nelder-mead.fmin.fmin.fmin.fmin.fmin.fmin.fmin.fmin

org.apache.commons.math3.optimization.unconstrained.nelder-mead.fmin.fmin.fmin.fmin.fmin.fmin.fmin.fmin.fmin

org.apache.commons.math3.optimization.unconstrained.nelder-mead.fmin.fmin.fmin.fmin.fmin.fmin.fmin.fmin.fmin.fmin

org.apache.commons.math3.complex.ComplexField.java - commons-math3/src/main/java

ComplexField.java

```
28 * This class is a singleton.  
29 * </p>  
30 * @see Complex  
31 * @version $Id: ComplexField.java 1416643 2012-12-03 19:37:14Z  
32 * @since 2.0  
33 */
```

EvoSuite Test Generation: org.apache.commons.math3.complex.ComplexField

EvoSuite test suite generation

Generating test cases

☐ Always run in background

Cancel

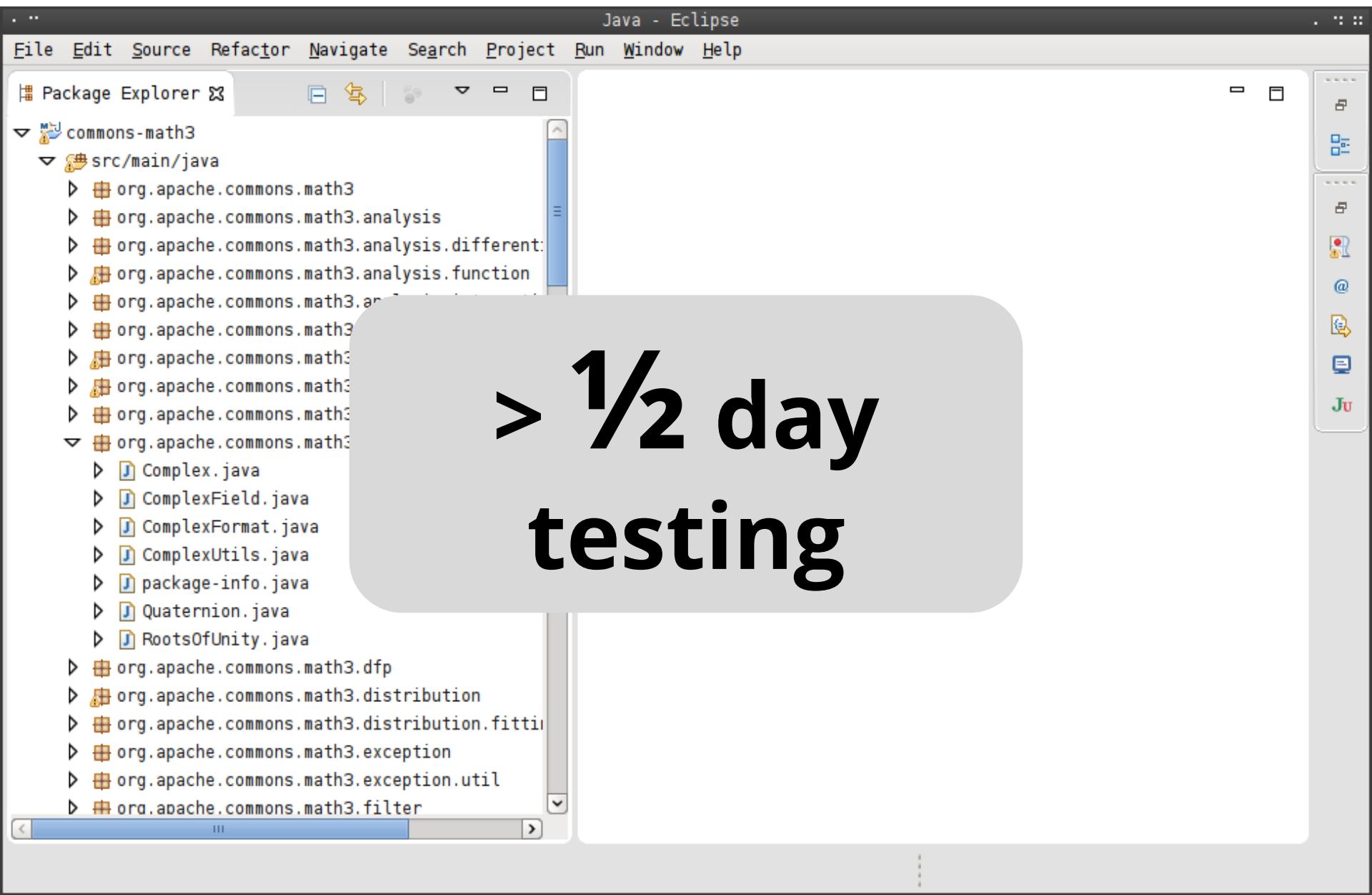
Details &gt;&gt;

Run in Background

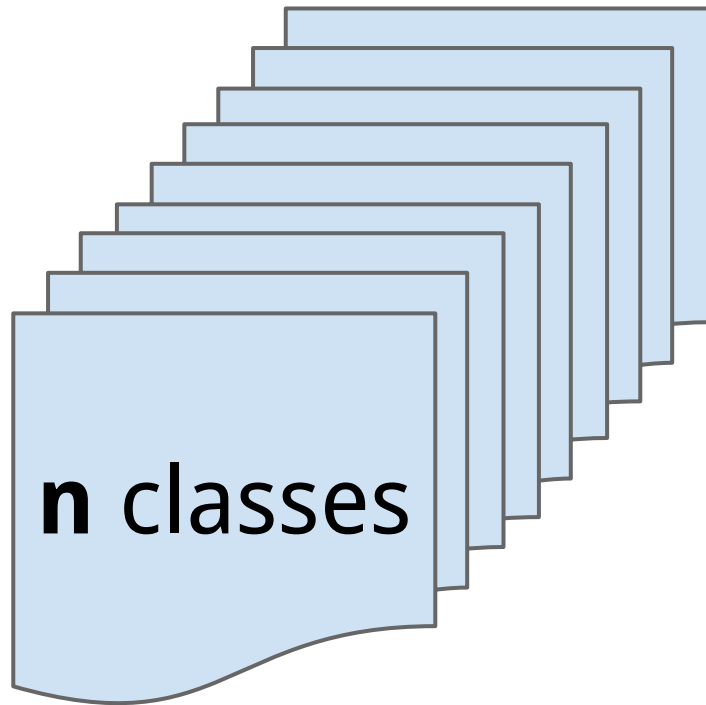
```
51 /** {@inheritDoc} */  
52 public Complex getOne() {  
53     return Complex.ONE;  
54 }  
55  
56 /** {@inheritDoc} */  
57 public Complex getZero() {  
58     return Complex.ZERO;  
59 }  
60  
61 /** {@inheritDoc} */  
62 public Class<? extends FieldElement<Complex>> getRuntimeClass()  
63     return Complex.class;
```

EvoSuite Test Genera...ield: (13%)





# Testing Whole Projects

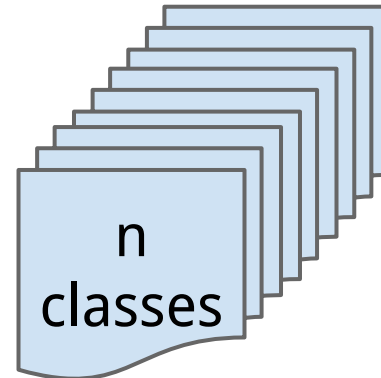


limit time budget

# Testing Whole Projects - *state-of-art*



$time_{per\ class} =$



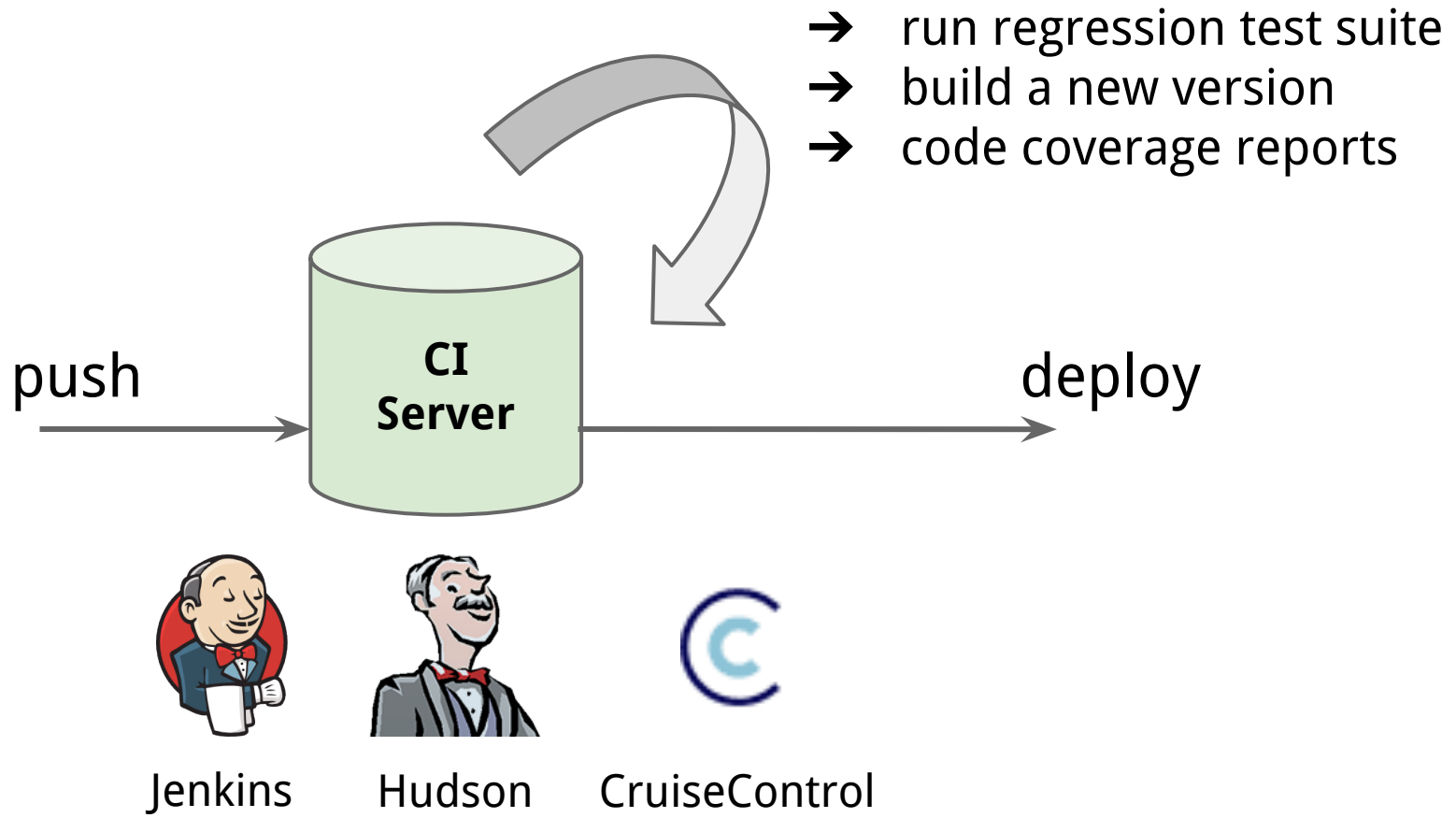
n  
classes

# Testing Whole Projects - *state-of-art*

	time
Class A	5 min
Class B	5 min
Class Z	5 min
total time	<b>15 minutes</b>

# Continuous Test Generation (CTG)

# Continuous Integration



# **Continuous Test Generation (CTG)**

# Continuous Test Generation (CTG)



# Continuous Test Generation (CTG)

→ Time Budget Allocation based on complexity

# Continuous Test Generation (CTG)

- Time Budget Allocation based on complexity
- Seeding Objects

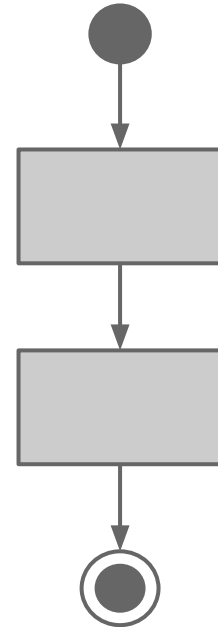
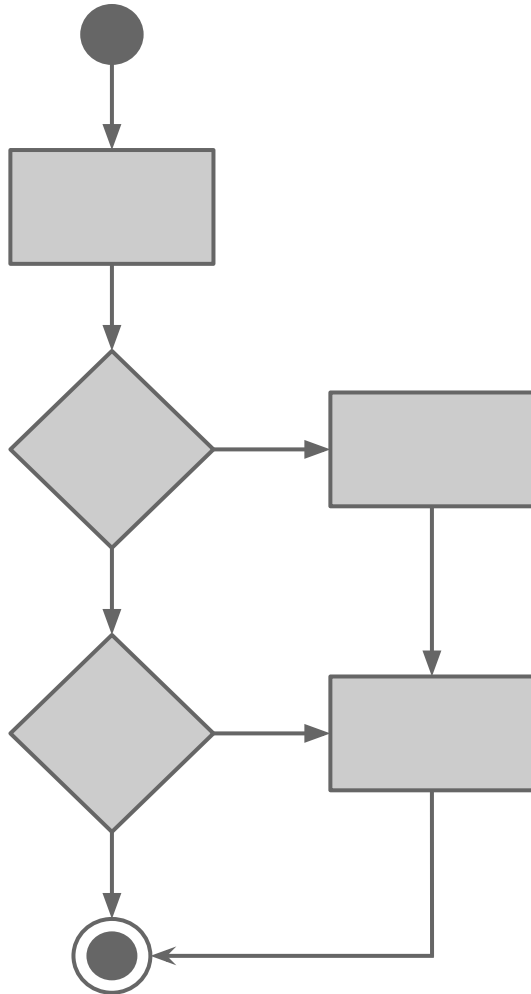
# Continuous Test Generation (CTG)

- Time Budget Allocation based on complexity
- Seeding Objects
- Continuous Test Generation based on History

# Continuous Test Generation (CTG)

- Time Budget Allocation based on complexity
- Seeding Objects
- Continuous Test Generation based on History

# Time Budget Allocation



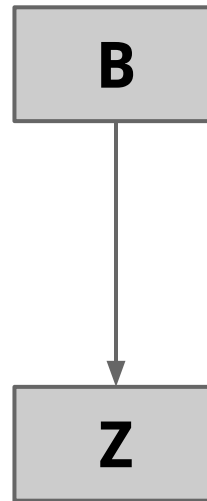
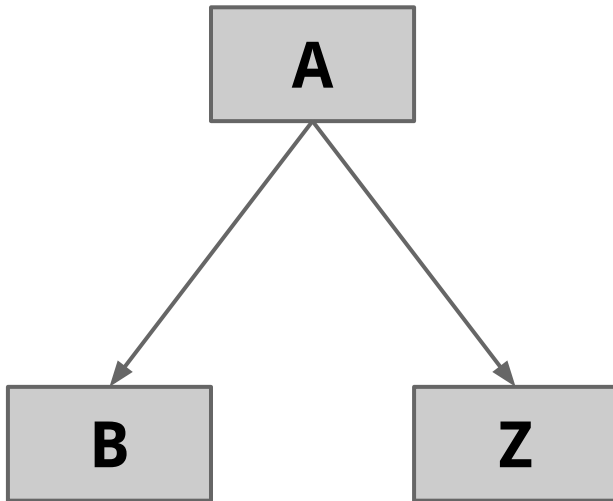
# Time Budget Allocation

	# Branches	time
Class A	10	9.33 min
Class B	1	1 min
Class Z	5	4.67 min
total time		<b>15 minutes</b>

# Continuous Test Generation (CTG)

- Time Budget Allocation based on complexity
- Seeding Objects
- Continuous Test Generation based on History

# Seeding Objects





# Seeding Objects

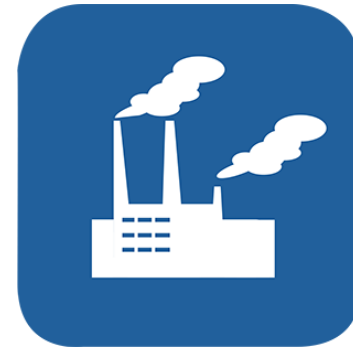
	Order	time
Class A	3rd	5 min
Class B	2nd	5 min
Class Z	1st	5 min
total time		<b>15 minutes</b>

# Evaluation I - Subjects

**SOURCE**  
**forge**  
**100**

**10 random  
projects\***

**total of 279  
classes**



**5 industrial  
projects**

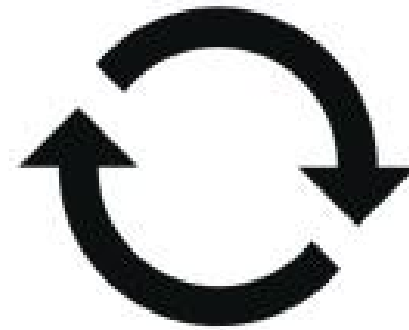
**total of 1307  
classes**

\* available at <http://www.evosuite.org/subjects/>

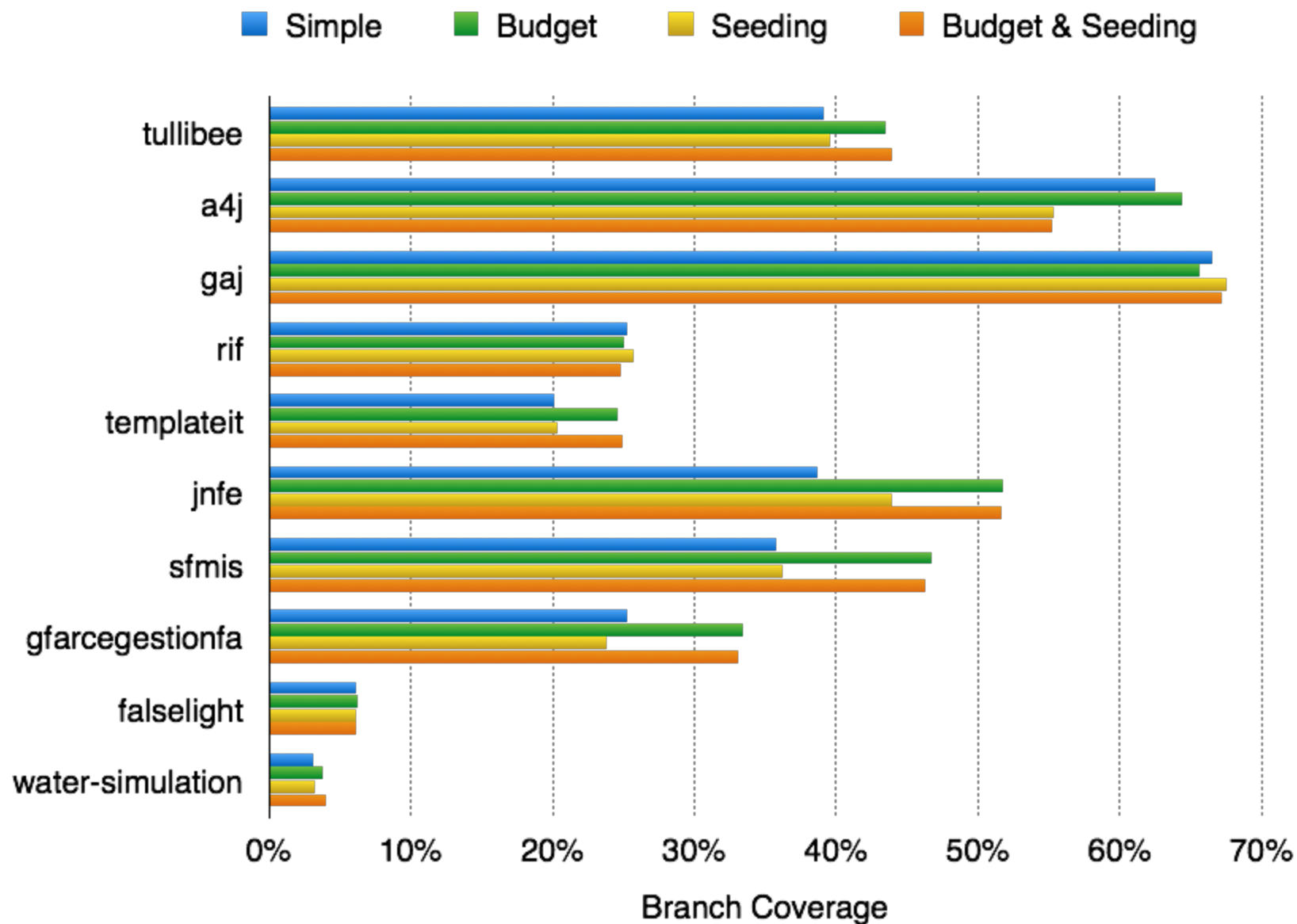
# Evaluation I - Setup I

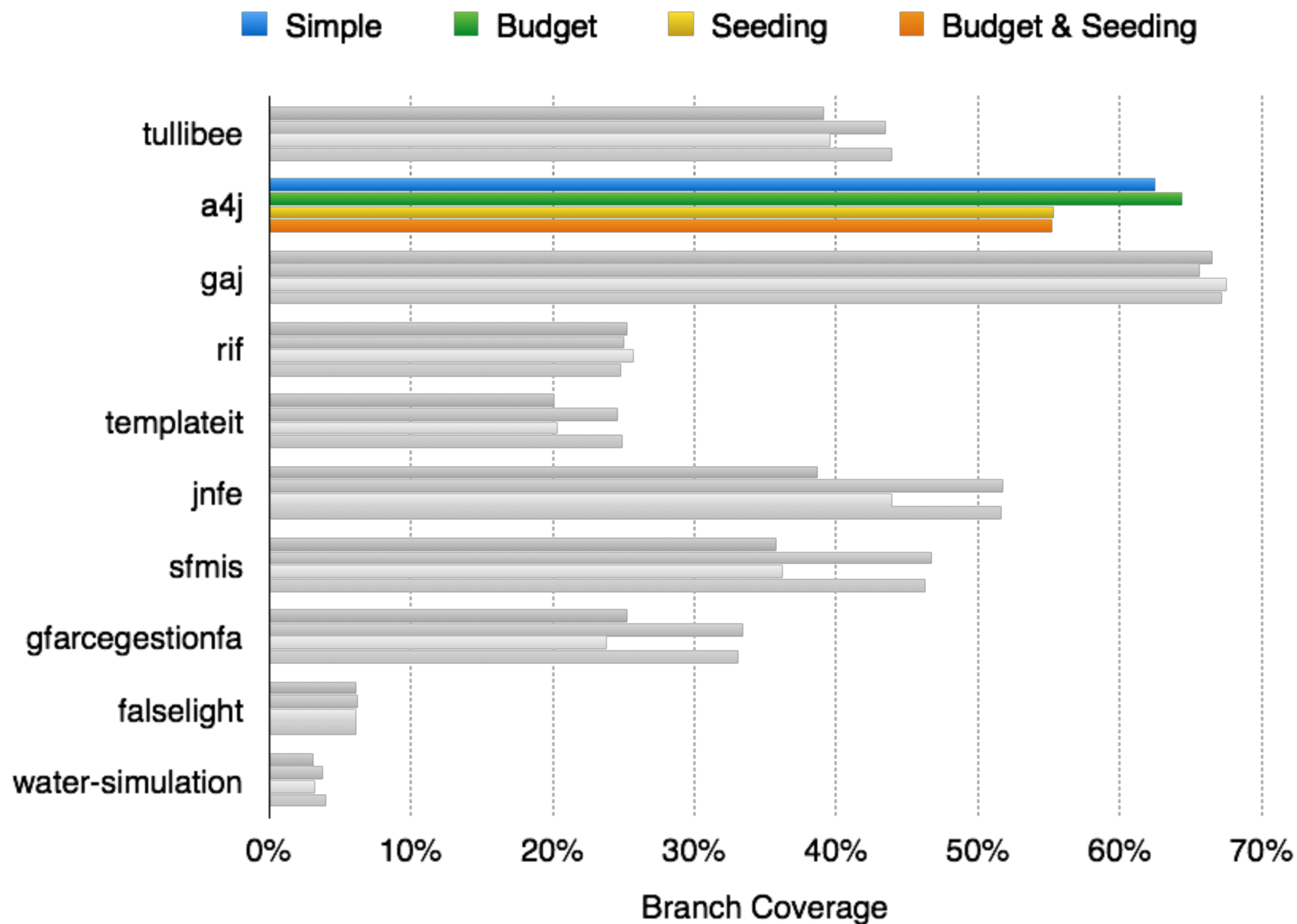


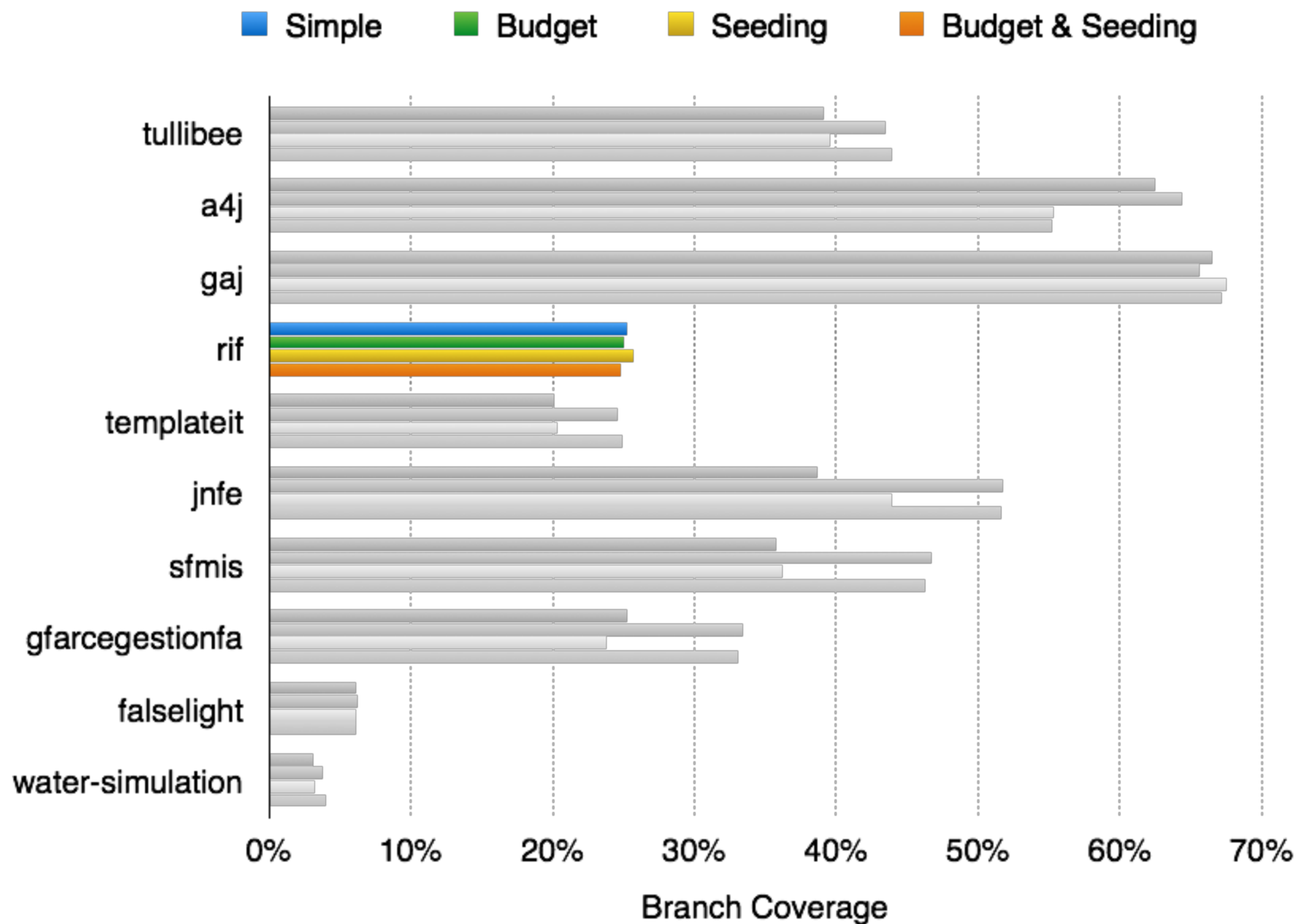
per class



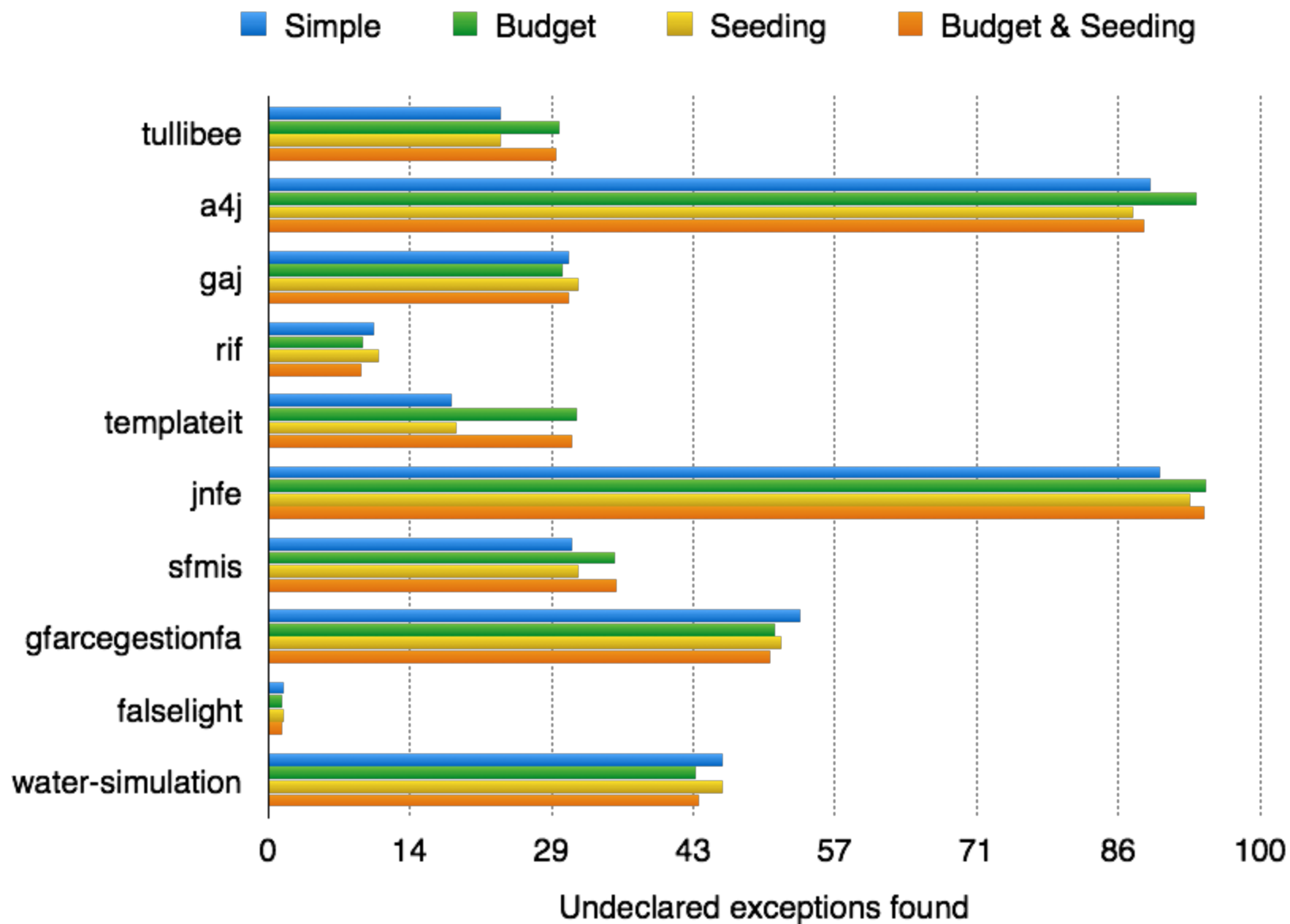
SF100 - 50 times  
Industrial - 1 time

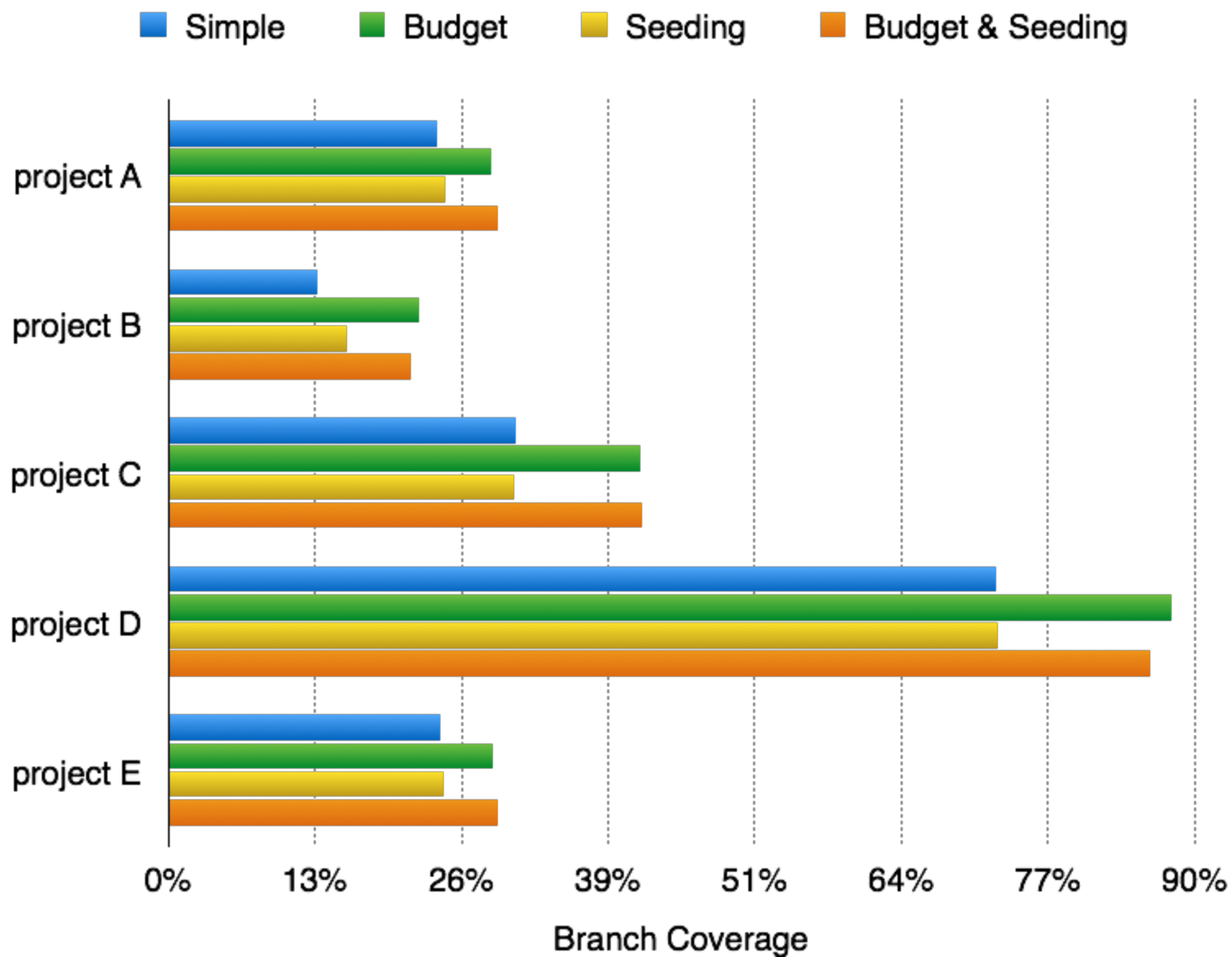






SF100 - Branch Coverage



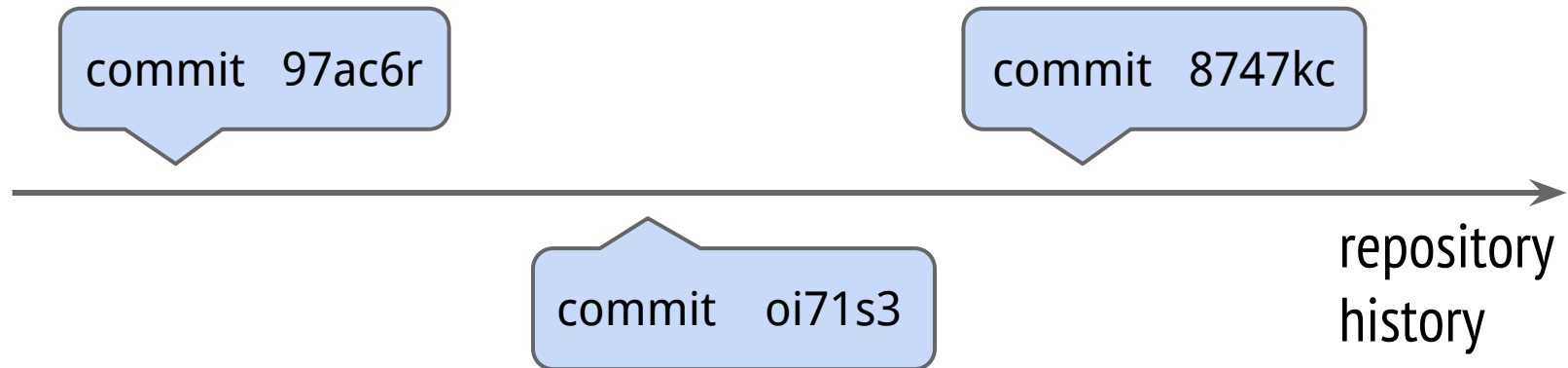




# Continuous Test Generation (CTG)

- Time Budget Allocation based on complexity
- Seeding Objects
- Continuous Test Generation based on History

# History Strategy



- information about history changes
- class coverage of each test generation
- uses previous test suite generated

# History Strategy

commit 97ac6r

N class A

N class B

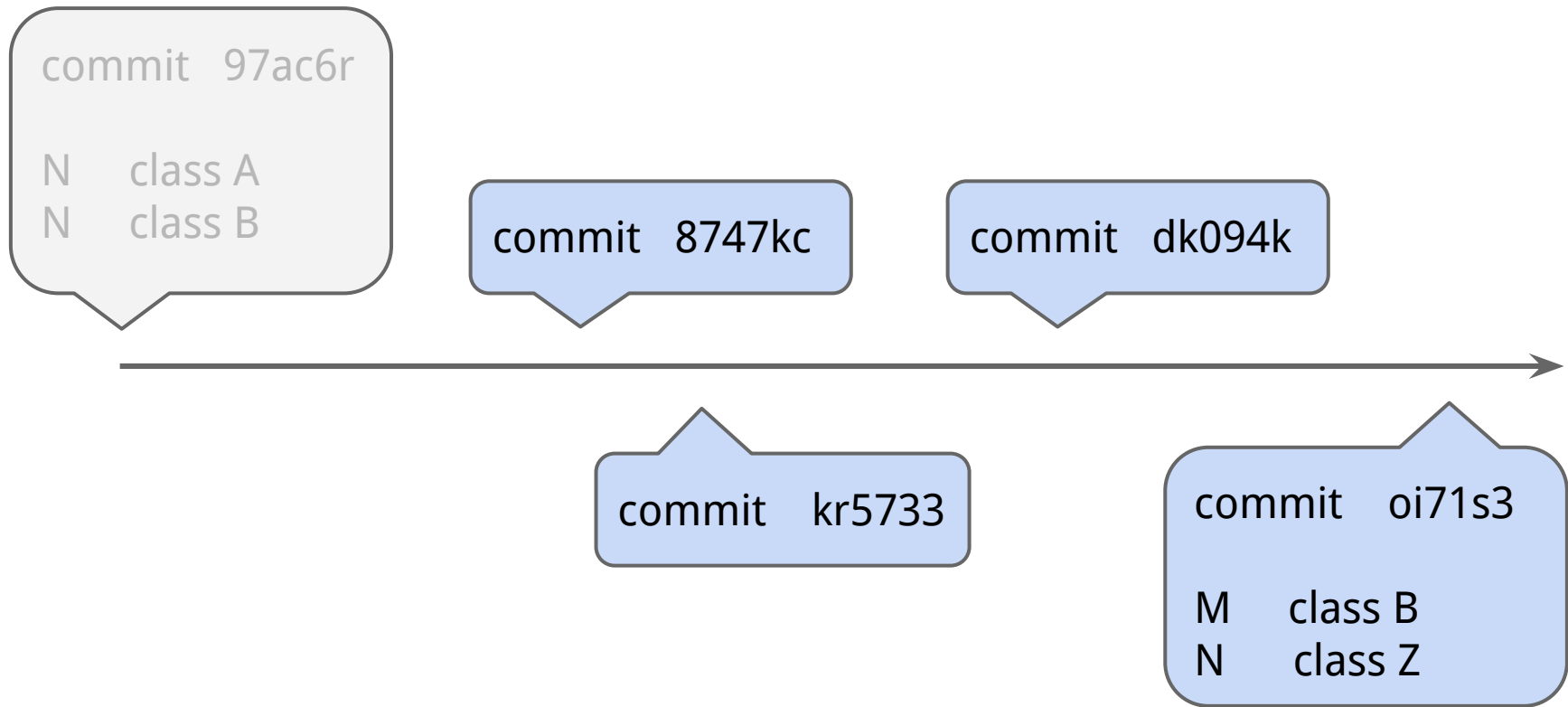


repository  
history

# History Strategy

	# Branches	status	time
Class A	10	new	13 min
Class B	1	new	2 min
total time			<b>15 minutes</b>

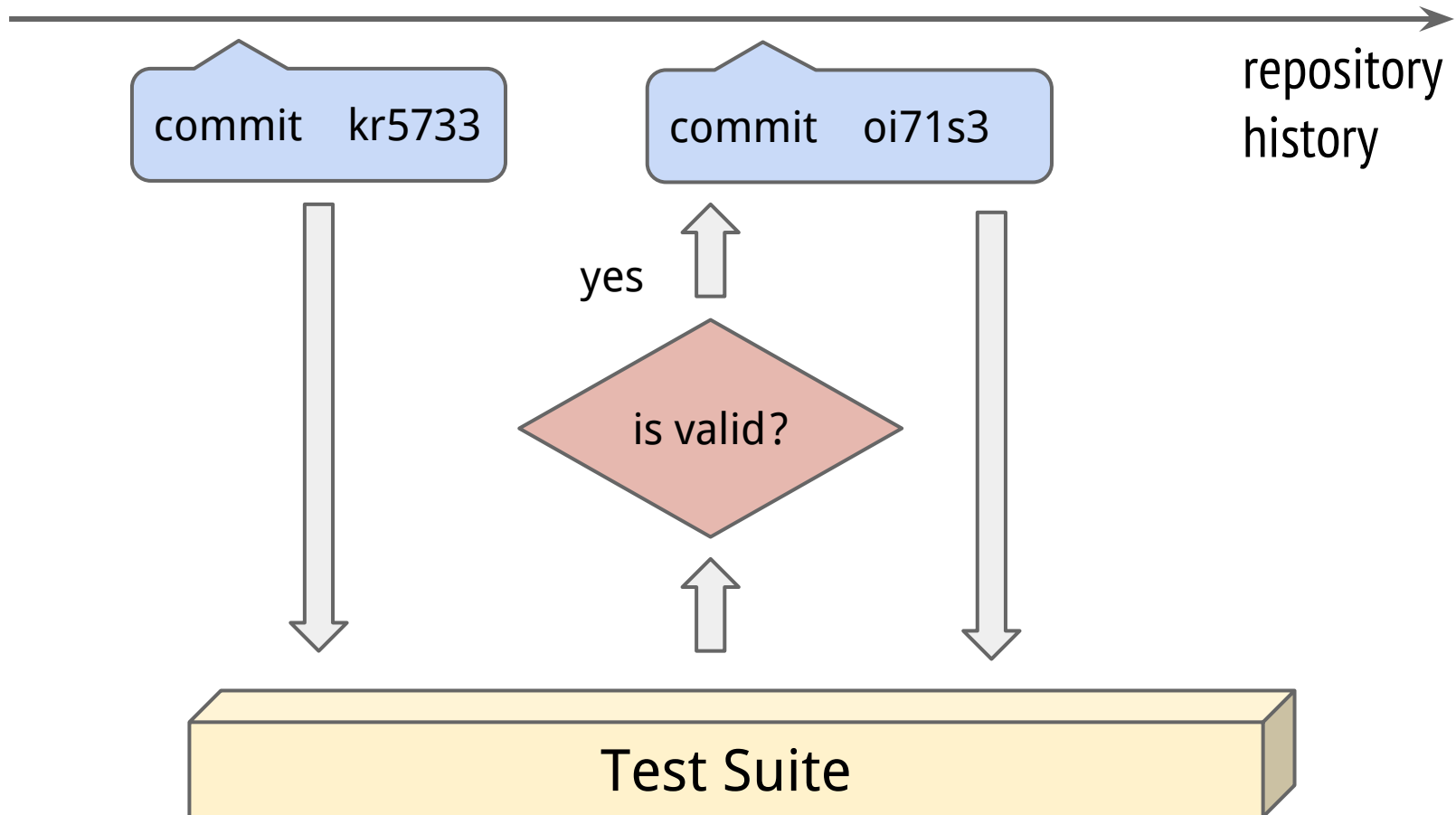
# History Strategy



# History Strategy

	# Branches	status	time
Class A	10	-	0 min
Class B	2	modified	5 min
Class Z	4	new	10 min
total time			<b>15 minutes</b>

# History Strategy - *Reusing Previous Suites*



# Evaluation II - Subjects



**GitHub**

**8 popular  
projects\***

**total of 475  
classes**

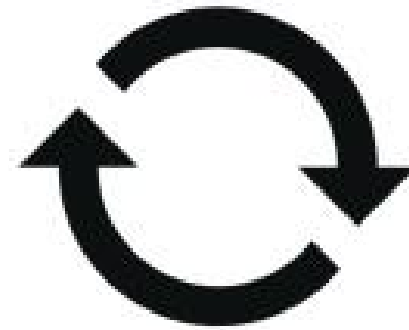
\* available at <http://www.evosuite.org/subjects/>



# Evaluation II - Setup I

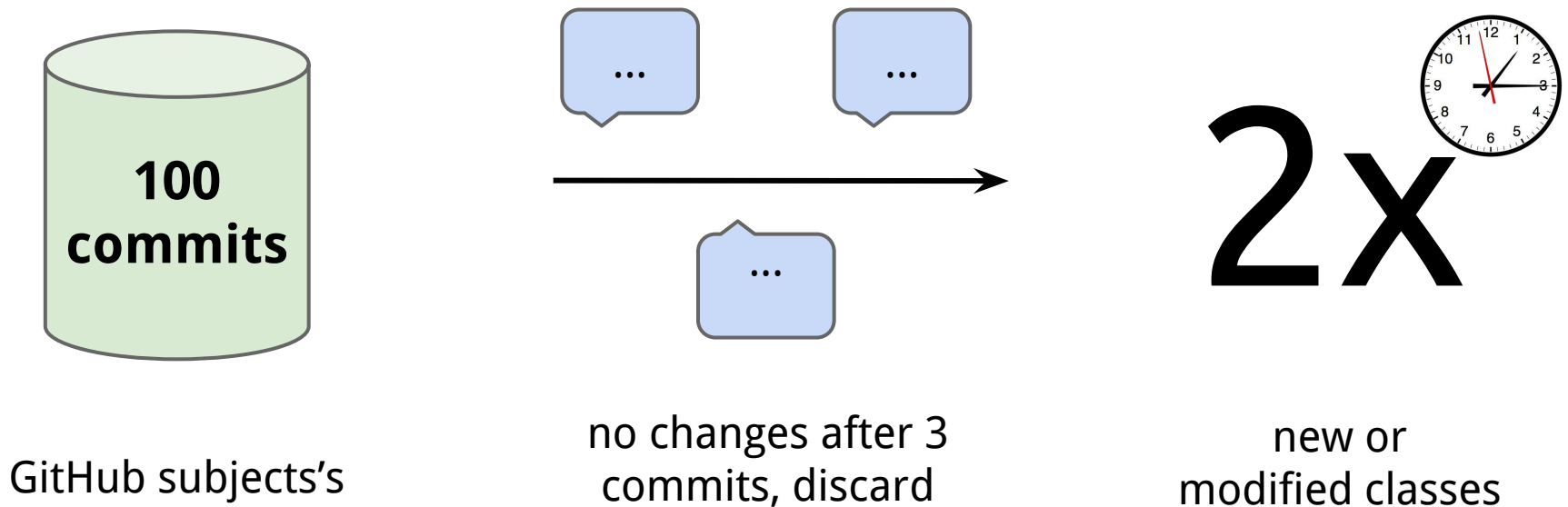


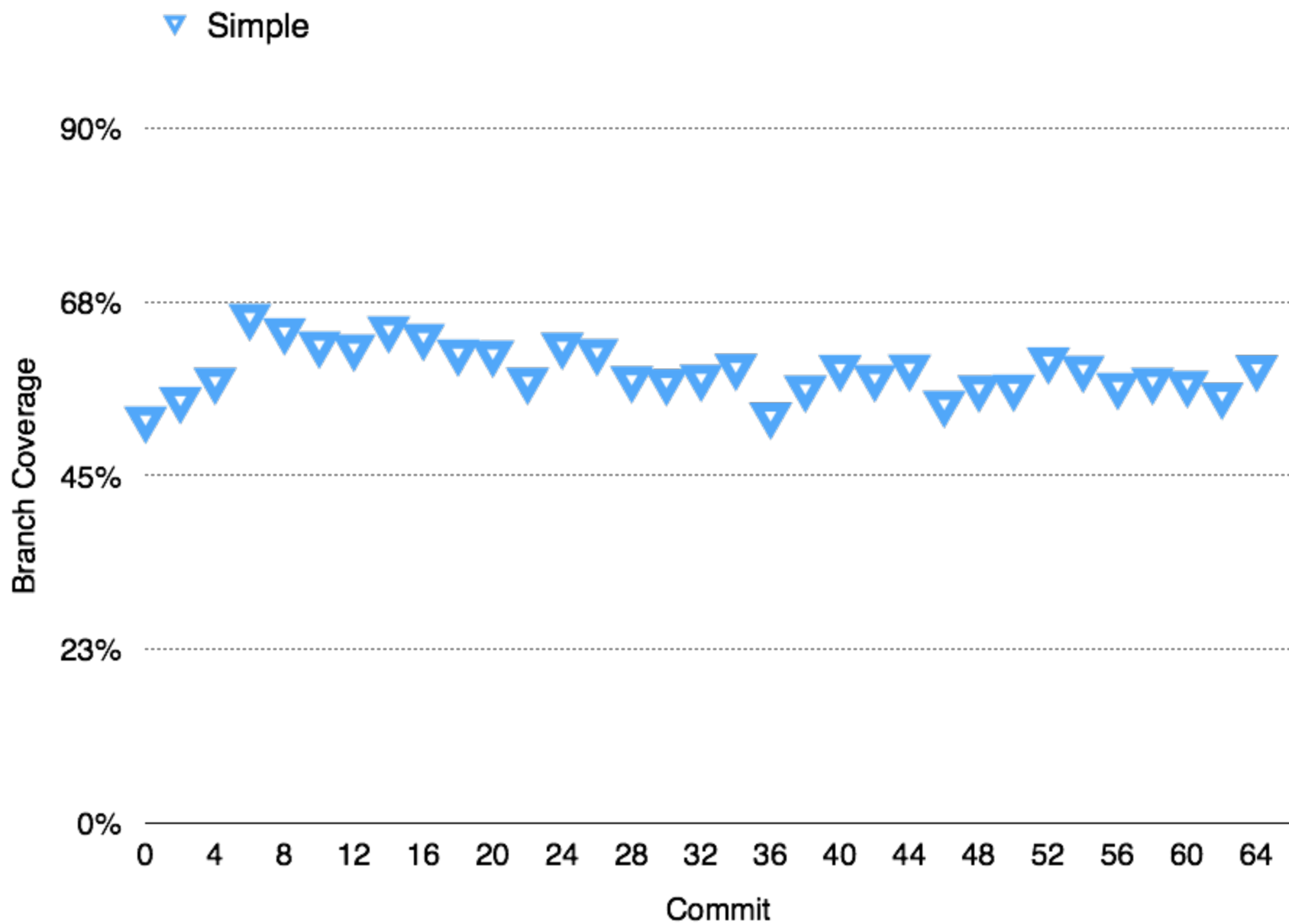
per class

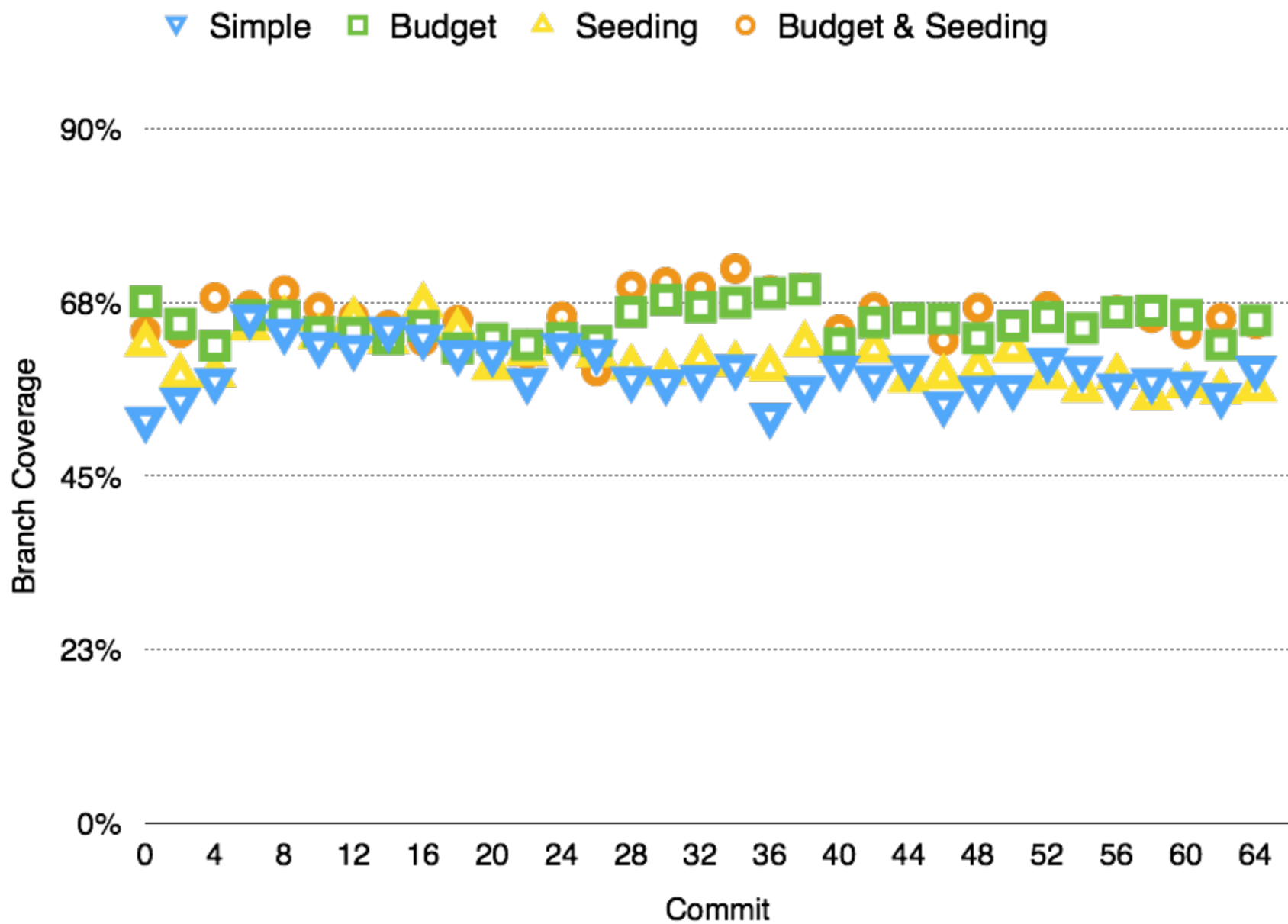


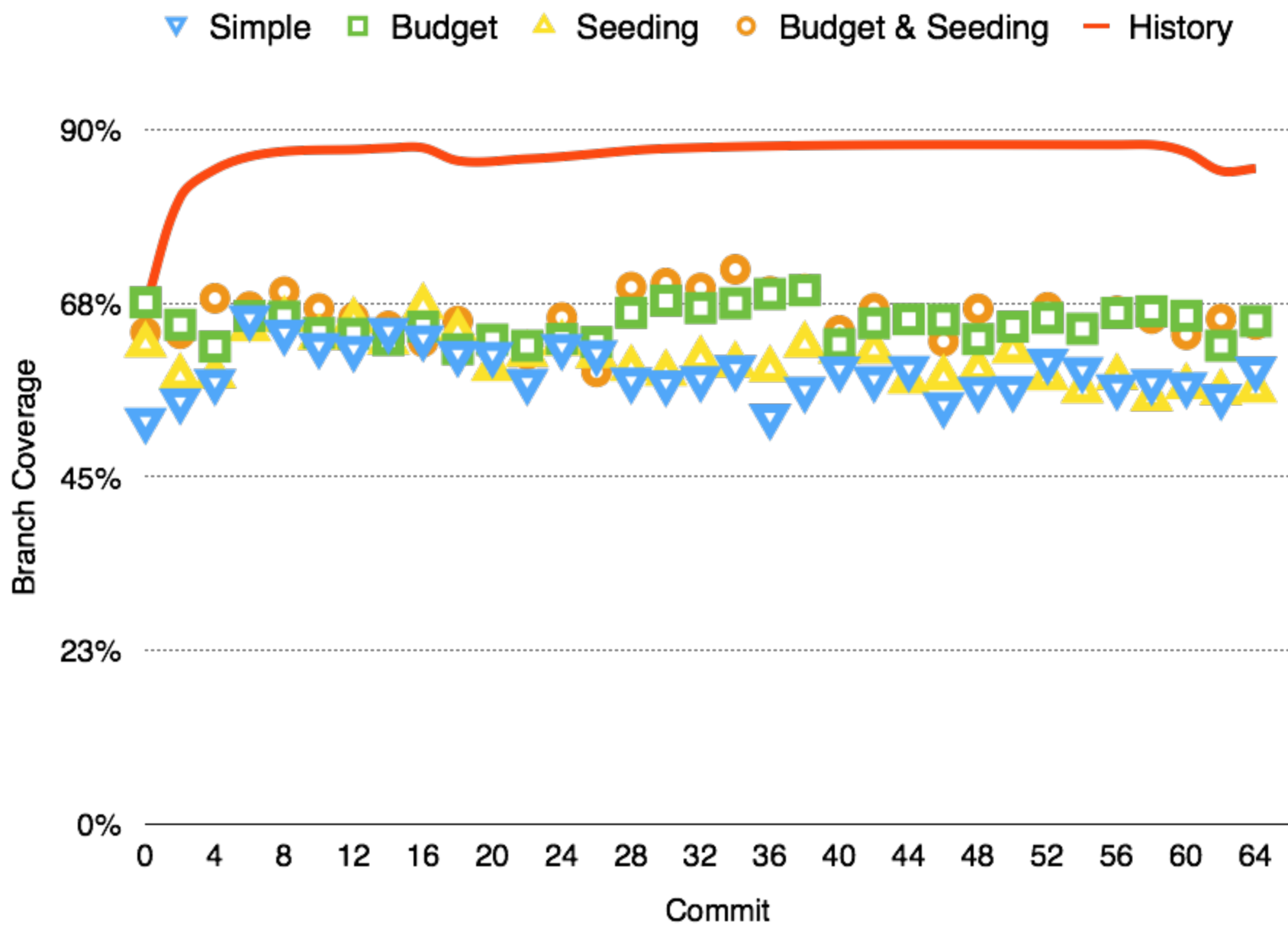
GitHub - 5 times

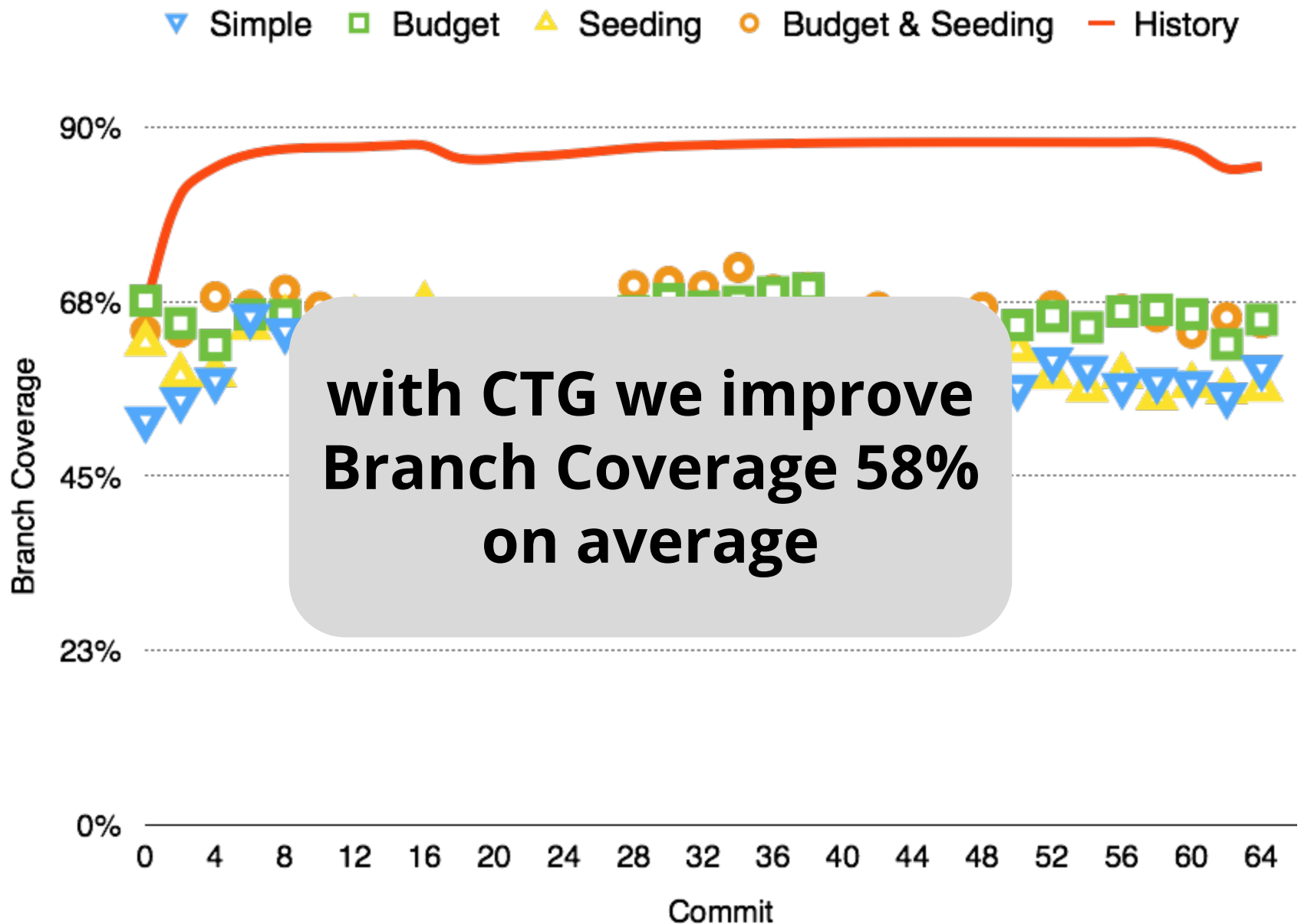
# Evaluation II - Setup II

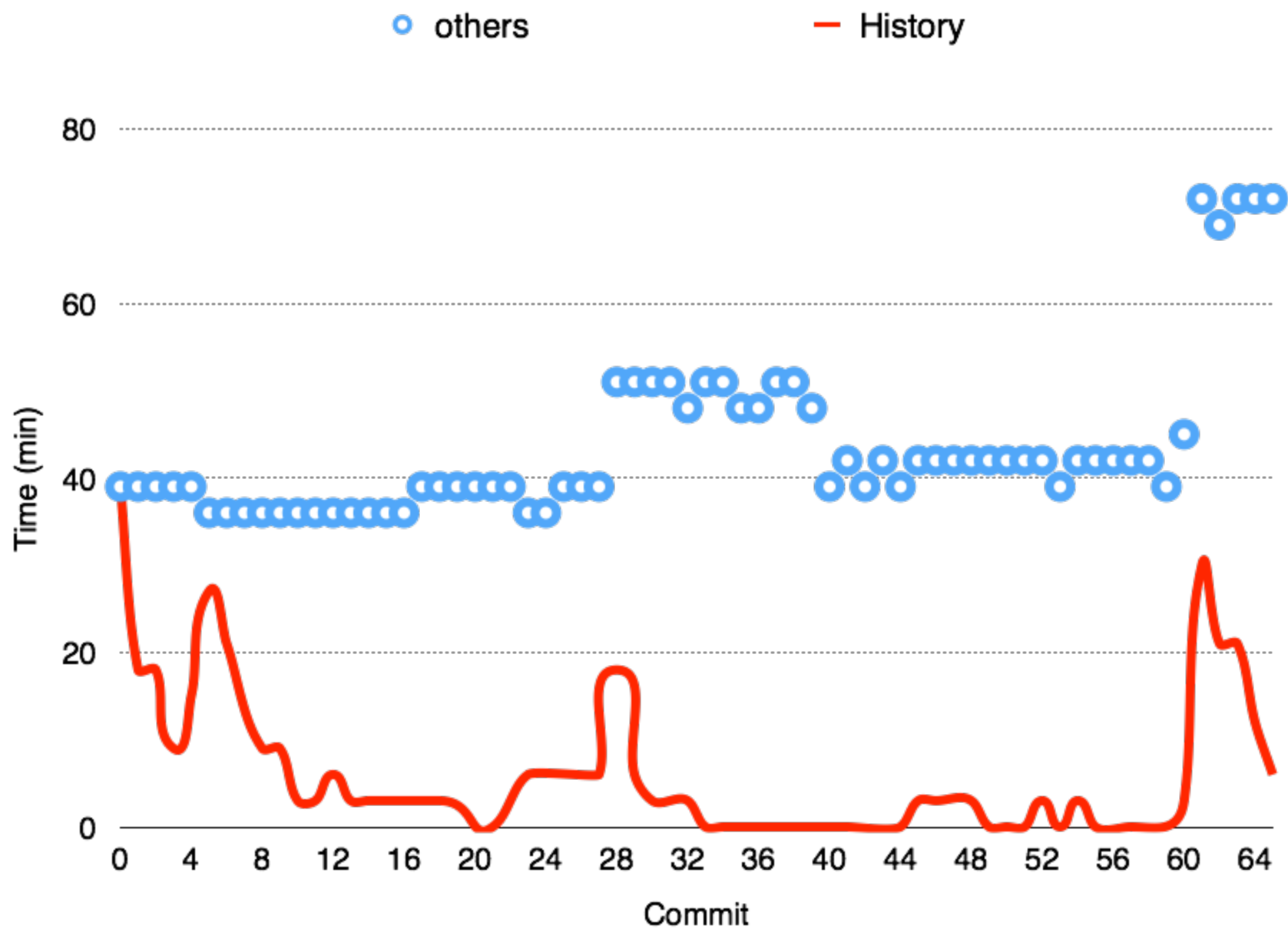


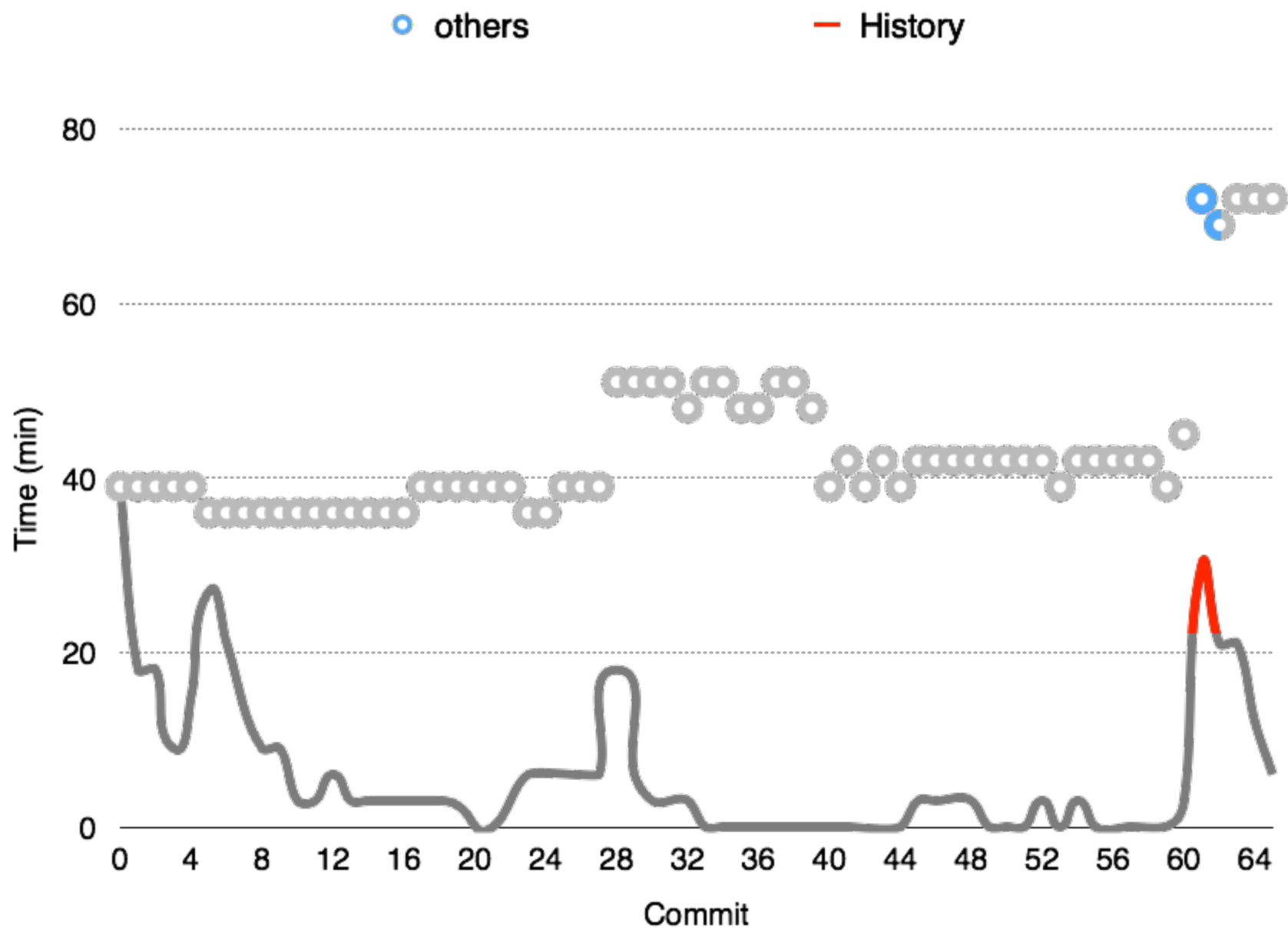




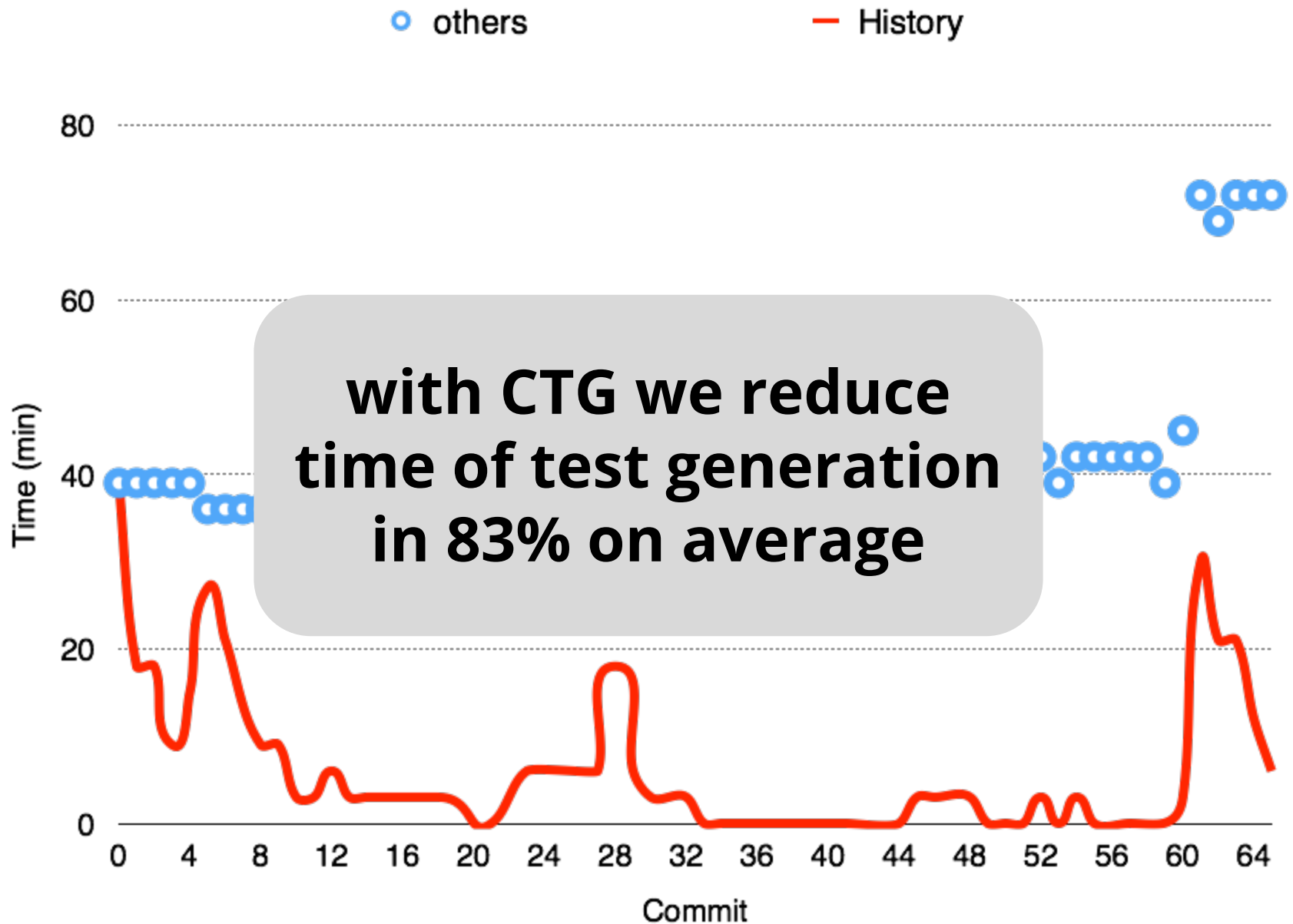


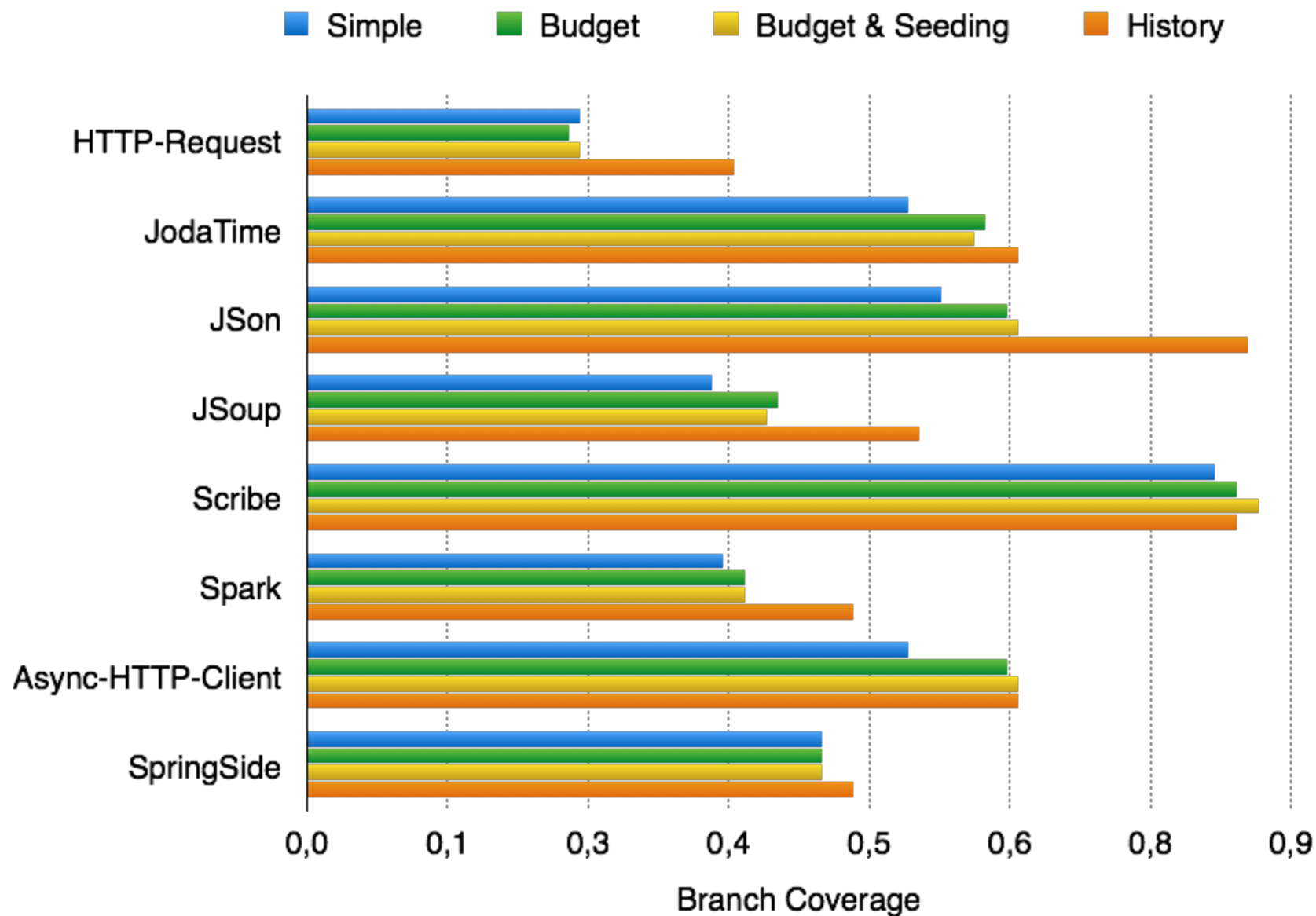


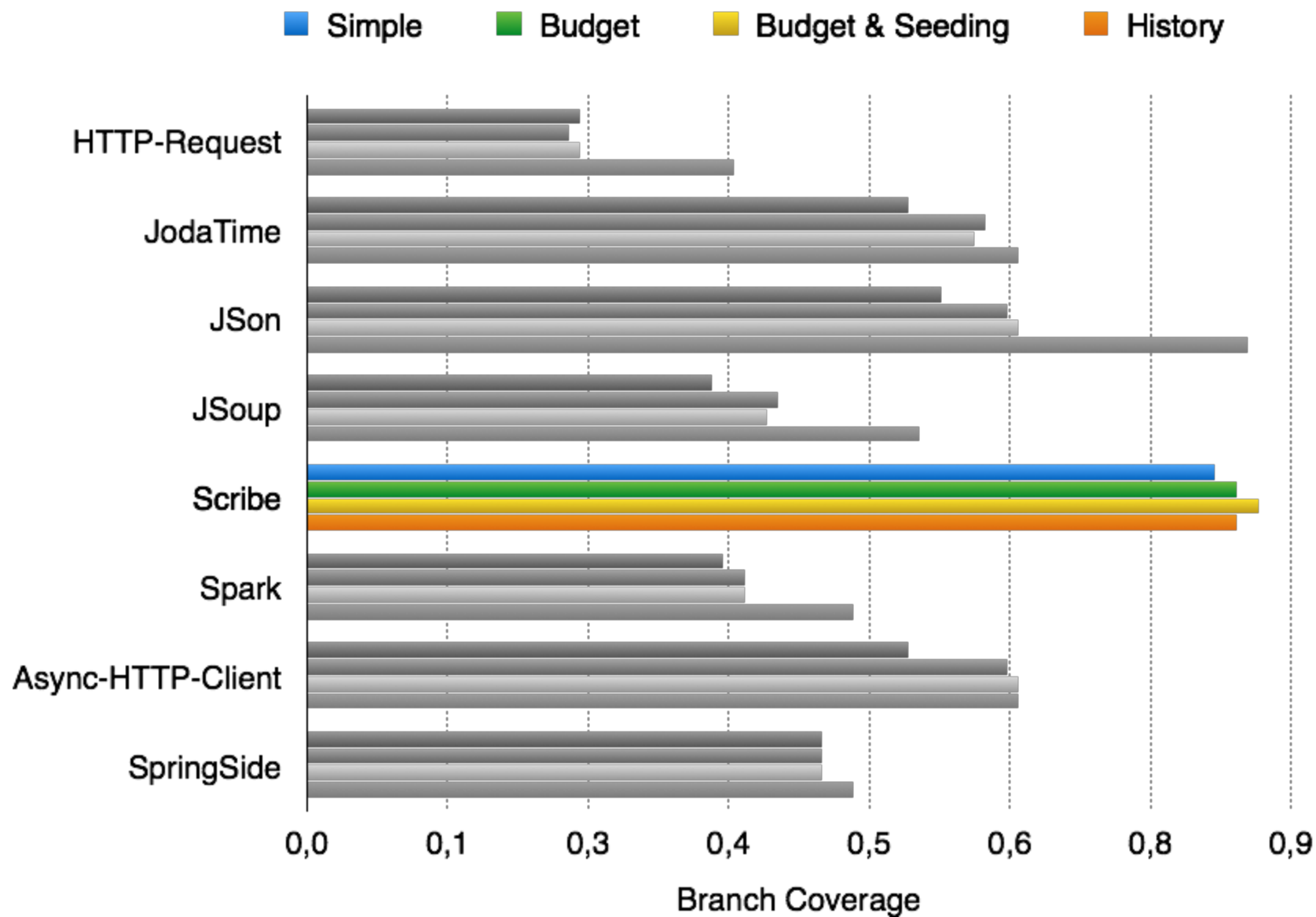








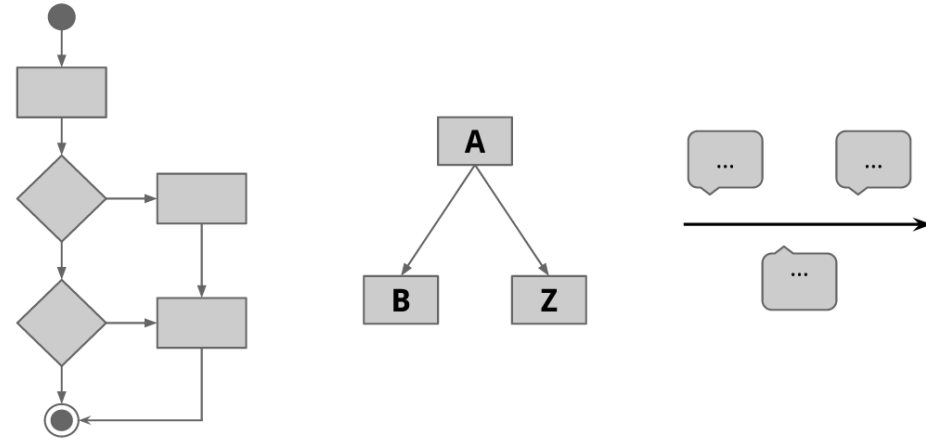
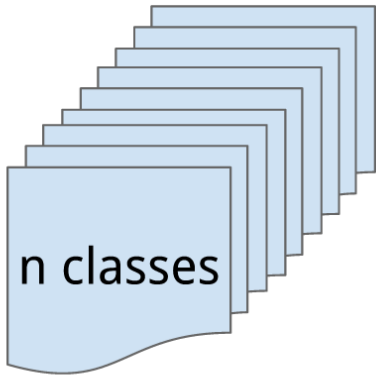




GitHub - Average coverage over time

# Future Work

- intelligent Seeding
- improve History strategy using fault prediction models
- other coverage criteria (statement, mutation testing, entropy)



<http://www.evosuite.org/>

**SOURCE  
forge  
100**

10 random  
projects\*

total of 279  
classes



5 industrial  
projects

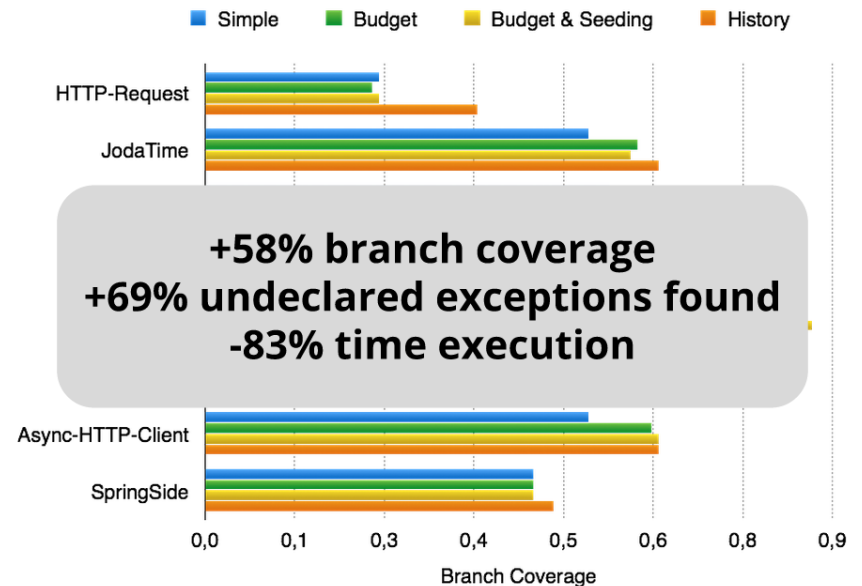
total of 1307  
classes



**GitHub**

8 popular  
projects\*

total of 475  
classes



**Annex**

# CTG vs Test Suite Augmentation

1. CTG answers the question of how to implement test suite augmentation (e.g., how to allocate the computational budget to individual classes);
2. While CTG can benefit from information about changes, it can also be applied without any software changes;
3. CTG is not tied to an individual coverage criterion; for example, one could apply CTG such that once coverage of one criterion is saturated, test generation can target a different, more rigorous criterion;
4. The implementation as part of continuous integration makes it possible to automatically notify developers of any faults found by automated oracles such as assertions or code contracts.