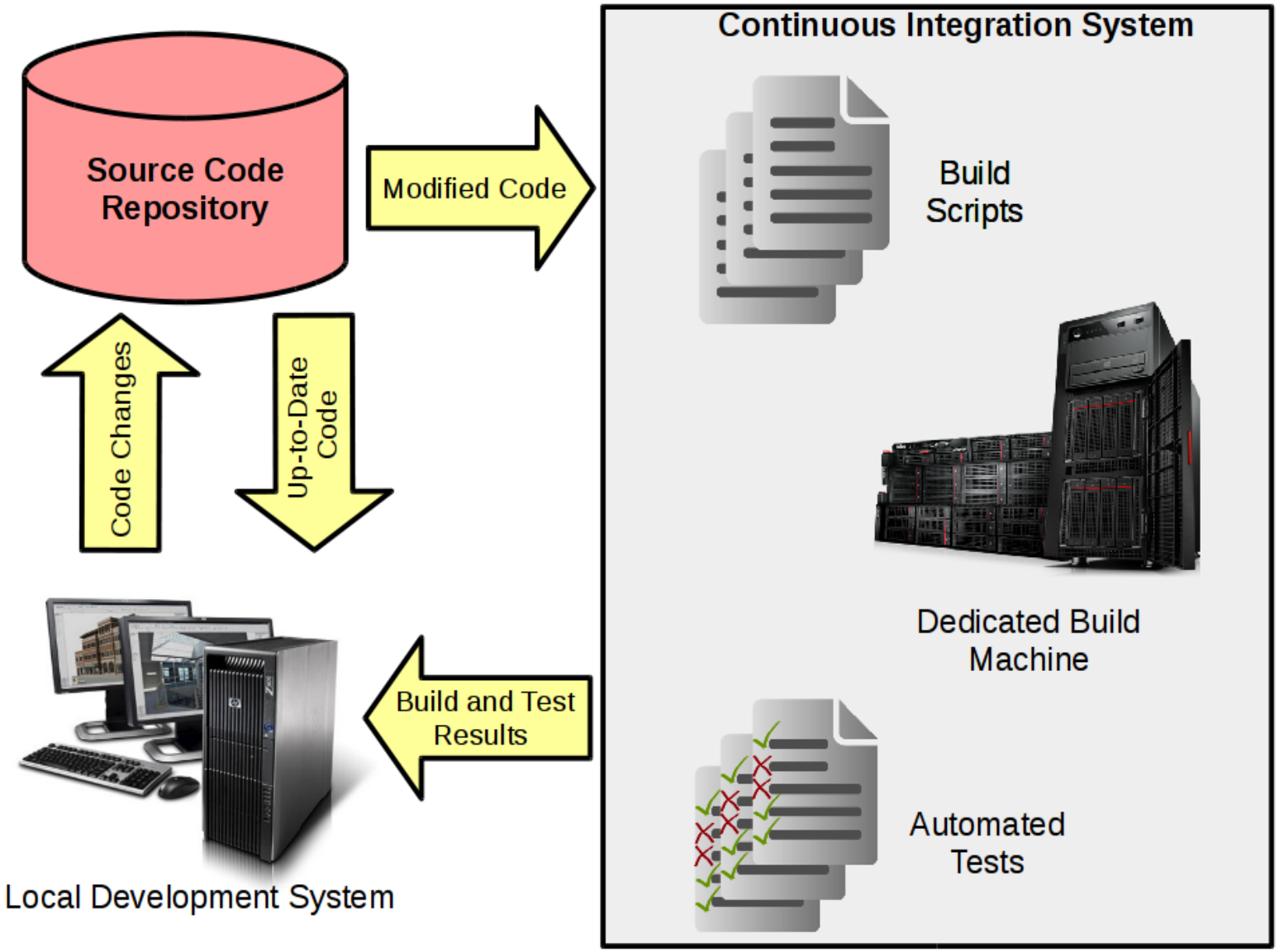
Universiteit Antwerpen

SATToSE 2015

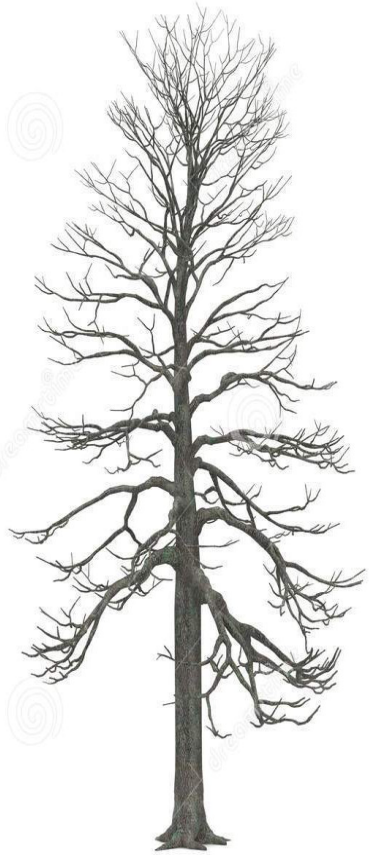
Mutation Testing:
An Industrial Experiment

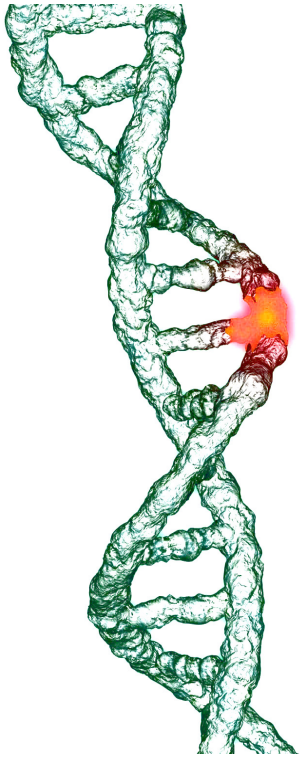
Ali Parsai
Quinten D. Soetens
Serge Demeyer


Continuous Integration System
Source Code Repository
Modified Code
Build Scripts
Code Changes
Up-to-Date Code
Dedicated Build Machine
Build and Test Results
Automated Tests
Local Development System

PROBLEM

Branch Coverage

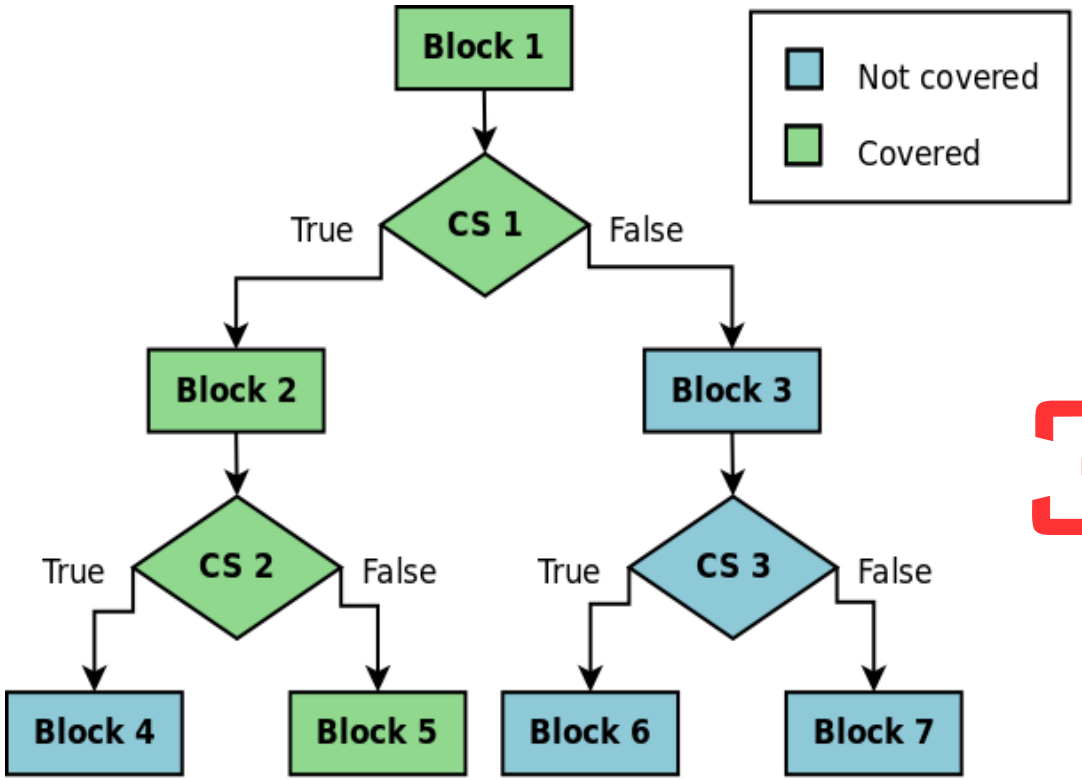


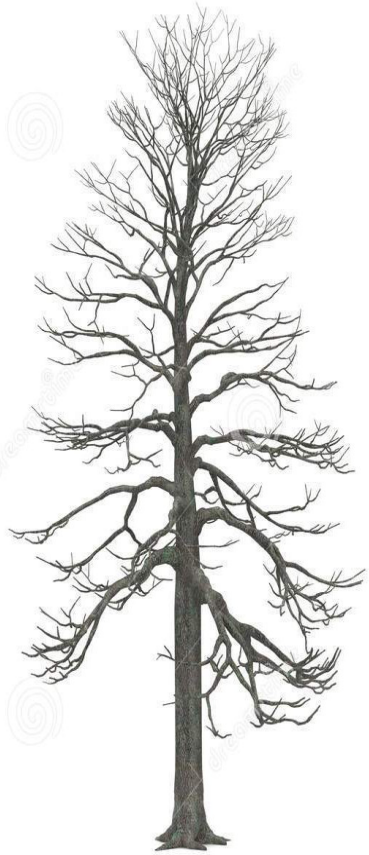


SOLUTION

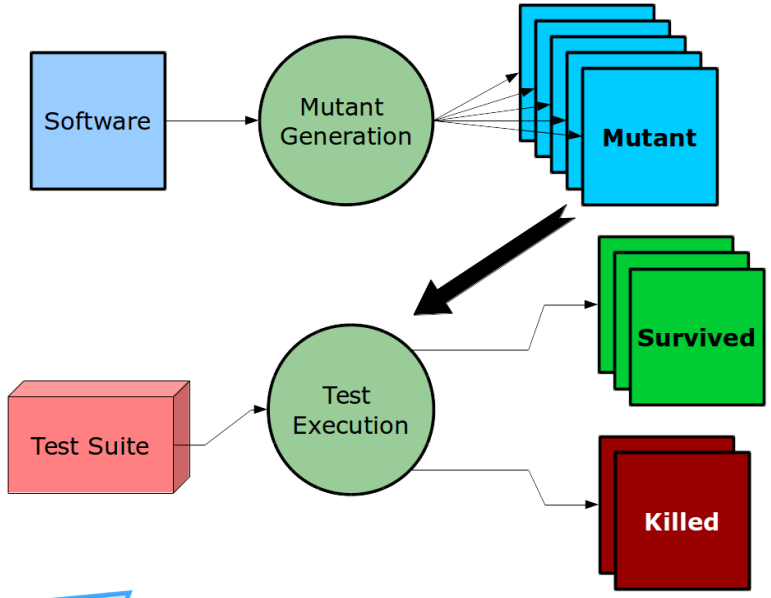
Mutation Testing

Branch Coverage





33%

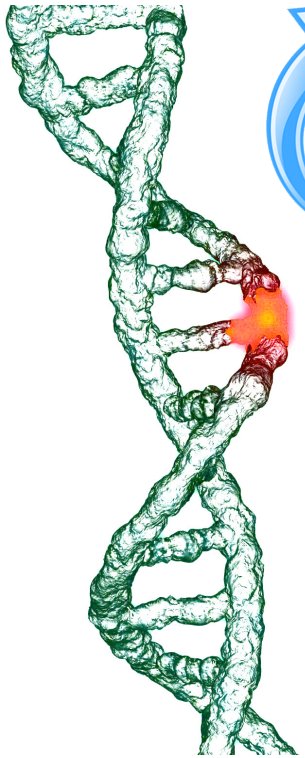


EXAMPLE

```
if ( x > 0 && y <= 0) x++;
```

```
if ( x <= 0 && y <= 0) x++;
if ( x > 0 && y >= 0) x++;
if ( x > 0 || y <= 0) x++;
if ( x > 0 && y <= 0) x--;
```

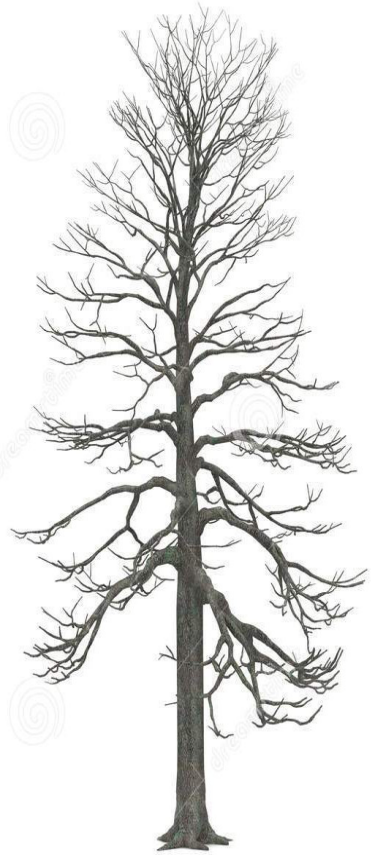
PROCESS



75%

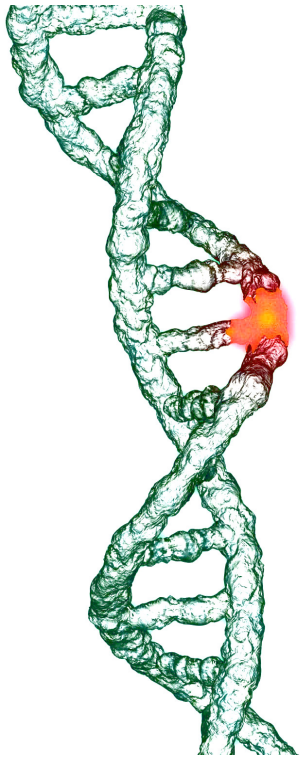
Mutation Testing

Branch Coverage



+ Can be calculated fast
+ Very good tool support – Easily integrable
– Not accurate for finding weaknesses

+ Very accurate
+ Adaptable by customizing the fault model
– Computationally intensive
– Poor tool support – Hard to integrate



Mutation Testing

Works on Java source code
(and potentially other languages)

Works regardless of build system and testing framework

LITTLEDARWIN PROJECT REPORT

Project Summary

Number of Files	Mutation Coverage
26	62.0 295/476

Breakdown by File

Name	Mutation Coverage
../main/java/com/addthis/codec/binary/BufferIn.java	0.0 0/2
../main/java/com/addthis/codec/binary/BufferOut.java	0.0 0/2
../main/java/com/addthis/codec/binary/CodecBin2.java	86.4 89/103
../main/java/com/addthis/codec/config/ConfigNodeCursor.java	81.2 13/16
../main/java/com/addthis/codec/config/ConfigTraversingParser.java	53.3 16/30
../main/java/com/addthis/codec/config/Configs.java	32.6 14/43
../main/java/com/addthis/codec/jackson/CaseIgnoringEnumDeserializer.java	28.6 2/7
../main/java/com/addthis/codec/jackson/CodecBeanDeserializer.java	60.0 6/10
../main/java/com/addthis/codec/jackson/CodecBeanDeserializerModifier.java	50.0 2/4
../main/java/com/addthis/codec/jackson/CodecDeserializers.java	50.0 1/2
../main/java/com/addthis/codec/jackson/CodecIntrospector.java	100.0 14/14
../main/java/com/addthis/codec/jackson/CodecJackson.java	80.0 4/5
../main/java/com/addthis/codec/jackson/CodecTypeDeserializer.java	76.9 20/26
../main/java/com/addthis/codec/jackson/CodecTypeIdResolver.java	42.9 3/7
../main/java/com/addthis/codec/jackson/CodecTypeResolverBuilder.java	100.0 3/3
../main/java/com/addthis/codec/jackson/IndexReportingObjectArrayDeserializer.java	80.0 8/10
../main/java/com/addthis/codec/jackson/Jackson.java	21.7 5/23
../main/java/com/addthis/codec/jackson/KeyReportingMapDeserializer.java	21.9 7/32
../main/java/com/addthis/codec/jackson/MissingPropertyException.java	100.0 1/1
../main/java/com/addthis/codec/jackson/UnderscorePropertyIgnorer.java	100.0 1/1
../main/java/com/addthis/codec/plugins/PluginMap.java	93.1 27/29
../main/java/com/addthis/codec/plugins/PluginRegistry.java	33.3 1/3
../main/java/com/addthis/codec/plugins/Plugins.java	0.0 0/1
../main/java/com/addthis/codec/reflection/CodableClassInfo.java	69.2 9/13
../main/java/com/addthis/codec/reflection/CodableFieldInfo.java	47.8 33/69
../main/java/com/addthis/codec/reflection/Fields.java	80.0 16/20

LITTLEDARWIN FILE REPORT

File Summary

Number of Mutants	Mutation Coverage
16	81.2 13/16

Detailed List

Survived Mutant	Build Output	Killed Mutant	Build Output
1.java	1.txt	2.java	2.txt
4.java	4.txt	3.java	3.txt
14.java	14.txt	5.java	5.txt
		6.java	6.txt
		7.java	7.txt
		8.java	8.txt
		9.java	9.txt
		10.java	10.txt
		11.java	11.txt
		12.java	12.txt
		13.java	13.txt
		15.java	15.txt
		16.java	16.txt

- Mutation testing tool → LittleDarwin
- Branch coverage tool → JaCoCo

- System under test → Agfa HealthCare Segmentation

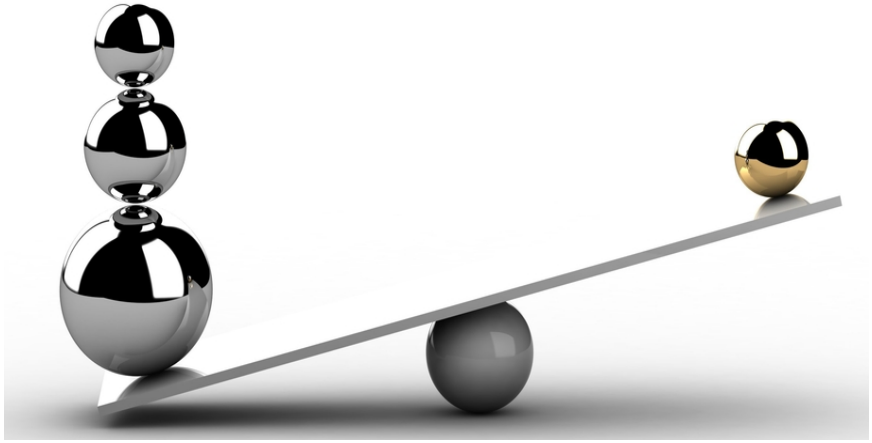
EXPERIMENT

AGFA HealthCare

- 38K lines of code
- Lots of legacy code
- Complex structure

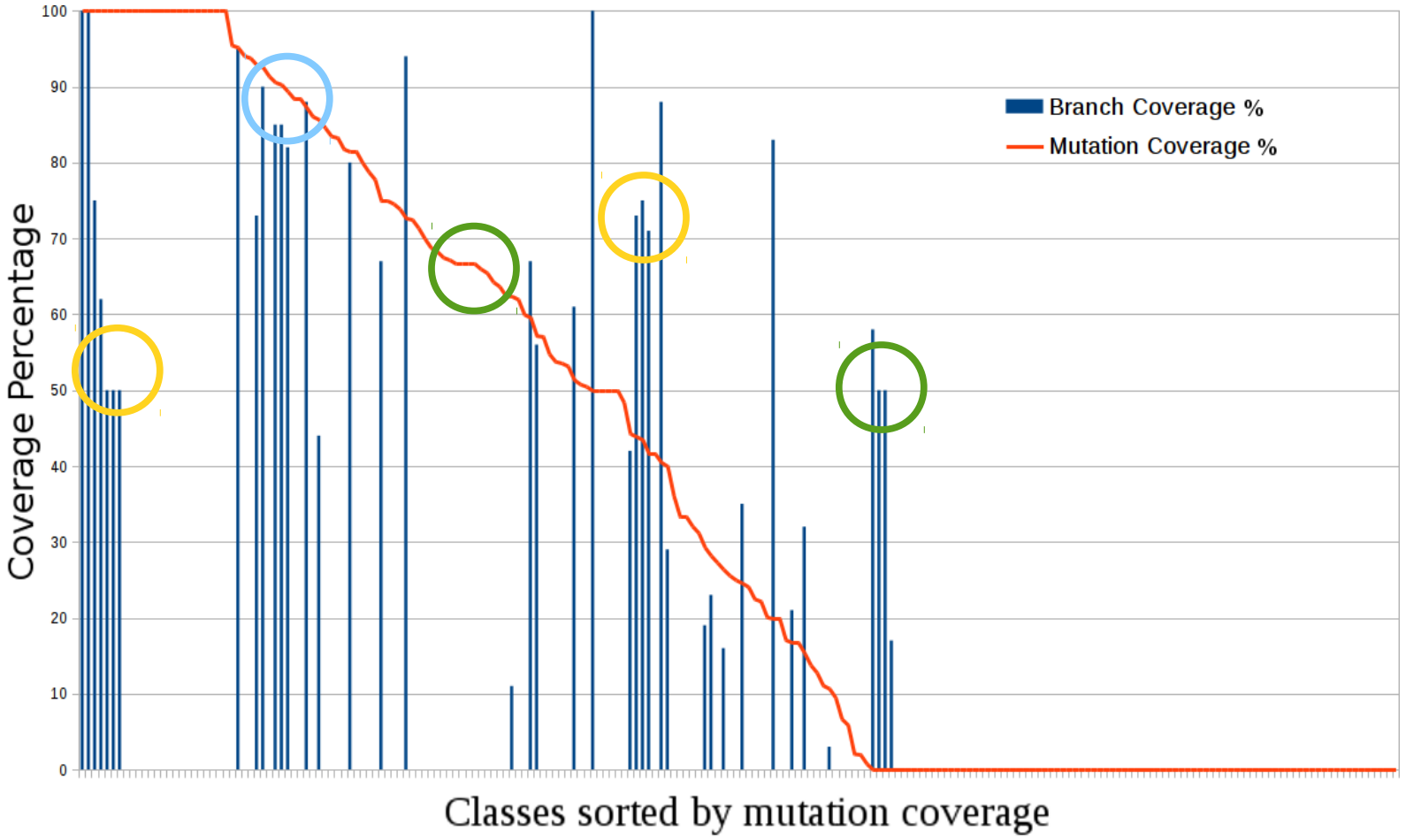
- Actively developed test suite
- Lots of black-box tests

TARGET SYSTEM



- Category 1 → Similar coverage
- Category 2 → Higher mutation coverage
- Category 3 → Higher branch coverage
- Category 4 → Only mutation coverage available
- Category 5 → Only branch coverage available

COMPARISON CRITERIA



102 8 8 90 4

RESULTS

CONCLUSION

- Mutation testing provides significant new insight about the test suite compared to branch coverage
- This extra information comes at the cost of performance, but incremental mutation testing can speed it up significantly