



Code Defenders

Crowdsourcing Effective Tests and Subtle Mutants
with a Mutation Testing Game

José Miguel Rojas, Thomas White, Benjamin Clegg, Gordon Fraser

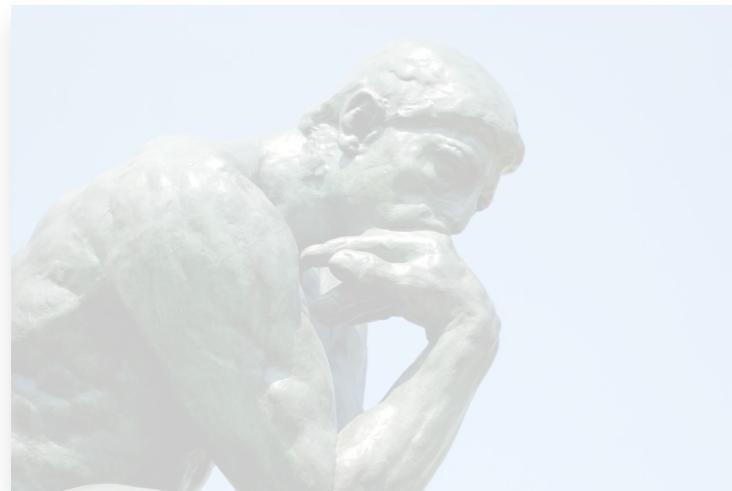


The
University
Of
Sheffield.





On Fault-revealing Tests



by Brian Hillegas (Flickr)

Manual Testing
Writing good tests
is difficult and
time-consuming.

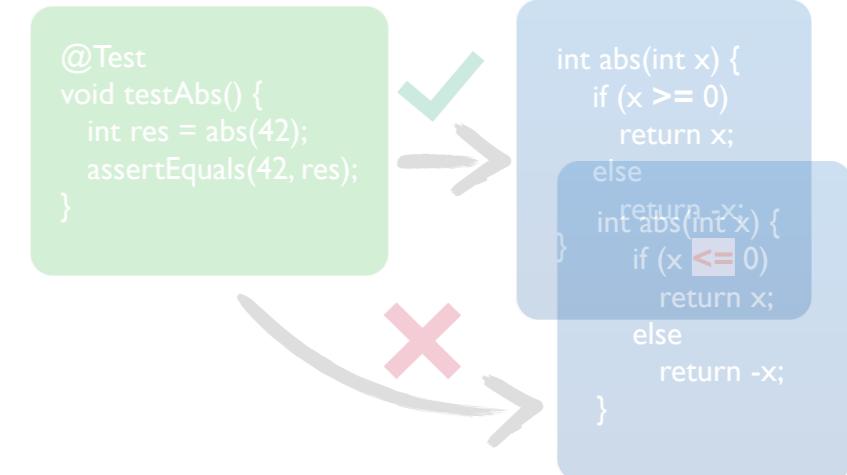
Test Generation
High coverage,
but weak
assertions.

Mutation Testing
Too many mutants,
often trivial, redundant
or equivalent.

EV~~U~~SUITE

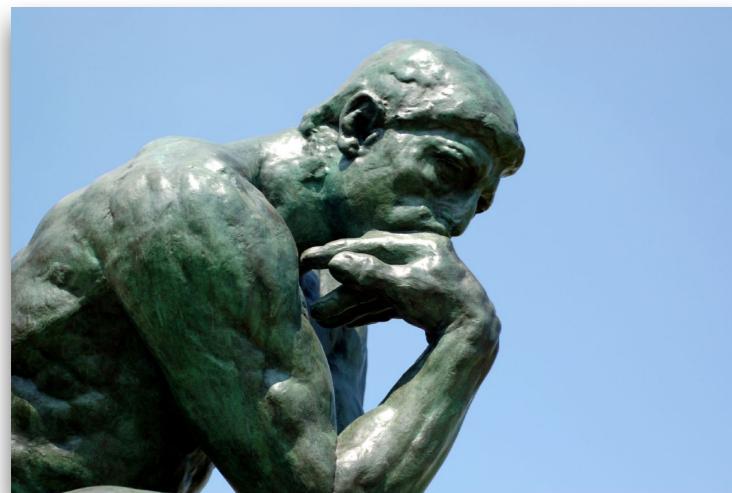
Randoop
Automatic unit test generation

AgitarOne
PUTTING JAVA TO THE TEST





On Fault-revealing Tests



by Brian Hillegas (Flickr)

Manual Testing
Writing good tests
is difficult and
time-consuming.

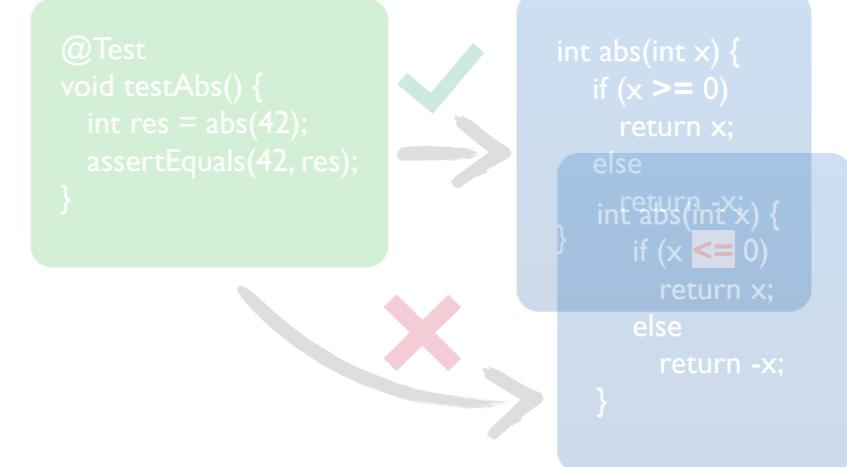
Test Generation
High coverage,
but weak
assertions.

Mutation Testing
Too many mutants,
often trivial, redundant
or equivalent.

EVSUITE

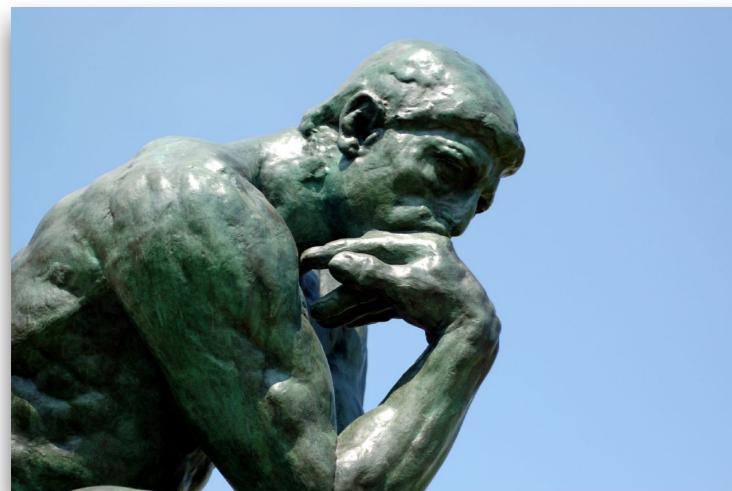
Randoop
Automatic unit test generation

AgitarOne
PUTTING JAVA TO THE TEST





On Fault-revealing Tests



by Brian Hillegas (Flickr)

Manual Testing

Writing good tests
is difficult and
time-consuming.

Test Generation

High coverage,
but weak
assertions.

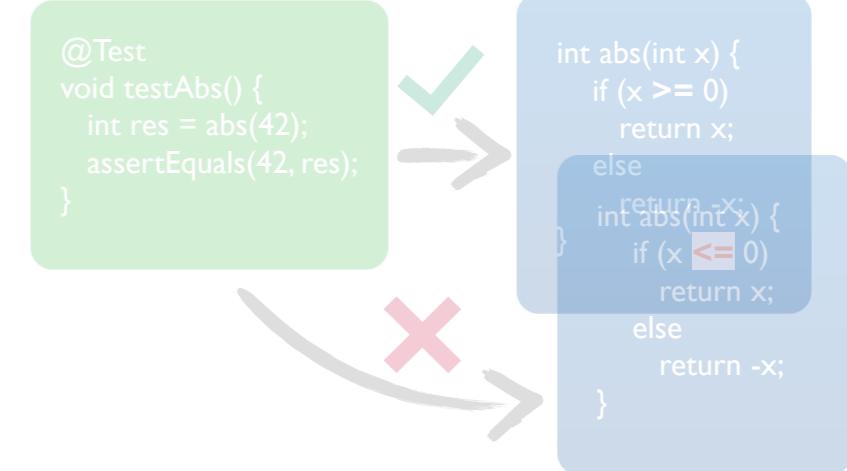
Mutation Testing

Too many mutants,
often trivial, redundant
or equivalent.

EVSUITE

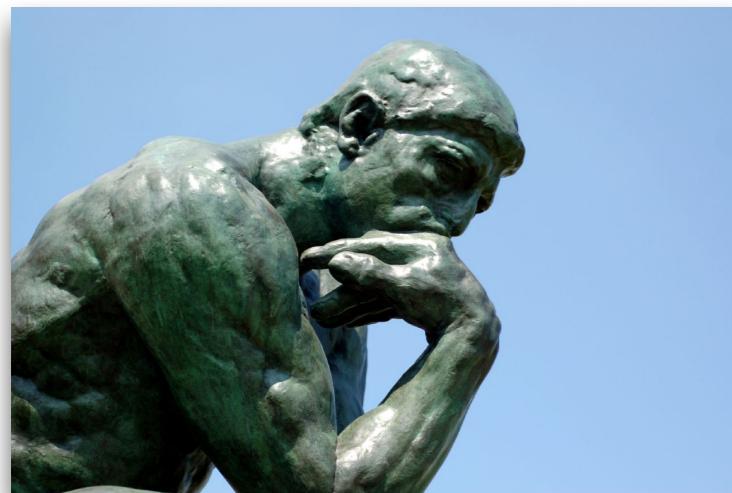
 **Randoop**
Automatic unit test generation

AgitarOne
PUTTING JAVA TO THE TEST





On Fault-revealing Tests



by Brian Hillegas (Flickr)

Manual Testing

Writing good tests
is difficult and
time-consuming.

Test Generation

High coverage,
but weak
assertions.

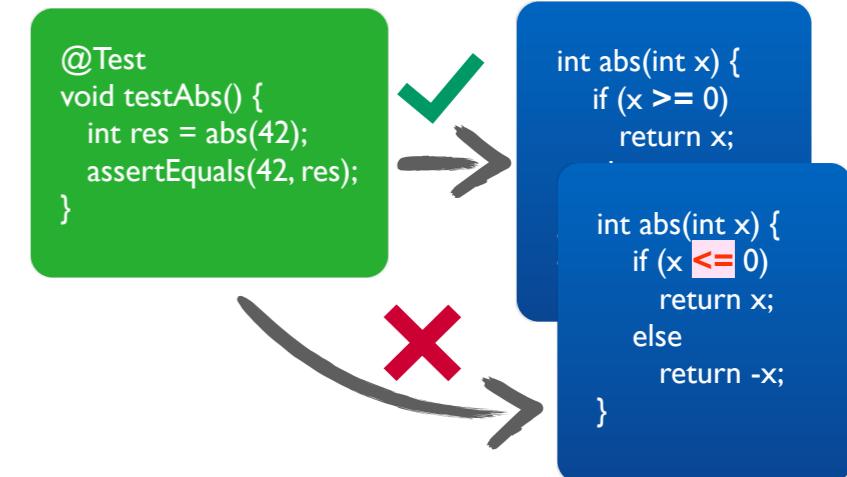
Mutation Testing

Too many mutants,
often trivial, redundant
or equivalent.

EVSUITE

Randoop
Automatic unit test generation

AgitarOne
PUTTING JAVA TO THE TEST





Code Defenders



by François Philipp (Flickr)



Software Testing
Gamification

Supporting software
testing education



by James Cridland (Flickr)

Crowdsourcing
tests and mutants



Code Defenders



by François Philipp (Flickr)



by James Cridland (Flickr)

Software Testing
Gamification

Supporting software
testing education

Crowdsourcing
tests and mutants



Code Defenders



by François Philipp (Flickr)

Software Testing
Gamification

Supporting software
testing education



by James Cridland (Flickr)

Crowdsourcing
tests and mutants



Code Defenders



by François Philipp (Flickr)

Software Testing
Gamification

Supporting software
testing education



by James Cridland (Flickr)



Crowdsourcing
tests and mutants



Code Defenders



by François Philipp (Flickr)



by James Cridland (Flickr)

Software Testing
Gamification

Supporting software
testing education

Crowdsourcing
tests and mutants



Code Defenders



by François Philipp (Flickr)



by James Cridland (Flickr)

Software Testing
Gamification

Supporting software
testing education

Crowdsourcing
tests and mutants



Code Defenders



by François Philipp (Flickr)



Software Testing
Gamification

Supporting software
testing education



by James Cridland (Flickr)

Crowdsourcing
tests and mutants



Code Defenders

Class Under Test

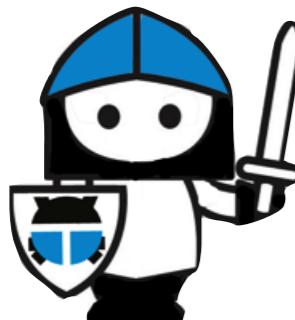
```
public class Arithmetics {  
    public int abs(int x) {  
        if (x >= 0)  
            return x;  
        else  
            return -x;  
    }  
}
```



Code Defenders

Class Under Test

```
public class Arithmetics {  
    public int abs(int x) {  
        if (x >= 0)  
            return x;  
        else  
            return -x;  
    }  
}
```



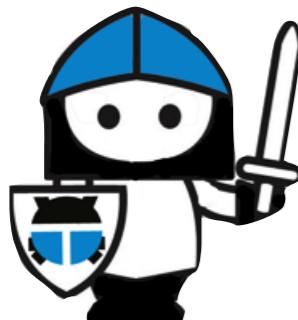
Attackers



Code Defenders

Class Under Test

```
public class Arithmetics {  
    public int abs(int x) {  
        if (x >= 0)  
            return x;  
        else  
            return -x;  
    }  
}
```



Attackers



Defenders



Code Defenders

Class Under Test

```
public class Arithmetics {  
    public int abs(int x) {  
        if (x >= 0)  
            return x;  
        else  
            return -x;  
    }  
}
```

```
public class Arithmetics {  
    public int abs(int x) {  
        if (x < 0)  
            return x;  
        else  
            return -x;  
    }  
}
```



Attackers



Defenders

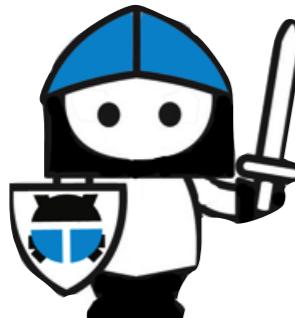
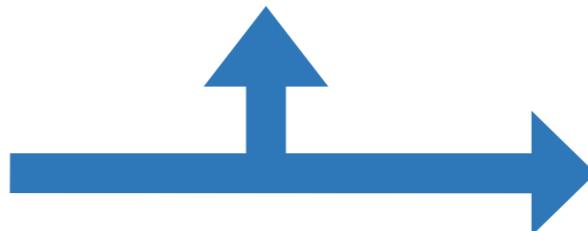


Code Defenders

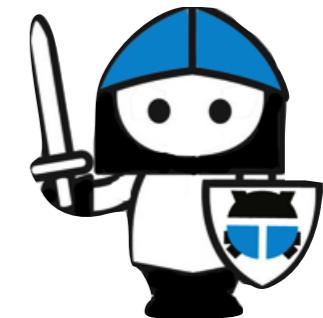
Class Under Test

```
public class Arithmetics {  
    public int abs(int x) {  
        if (x >= 0)  
            return x;  
        else  
            return -x;  
    }  
}
```

```
public class Arithmetics {  
    public int abs(int x) {  
        if (x < 0)  
            return x;  
        else  
            return -x;  
    }  
}
```



Attackers



Defenders

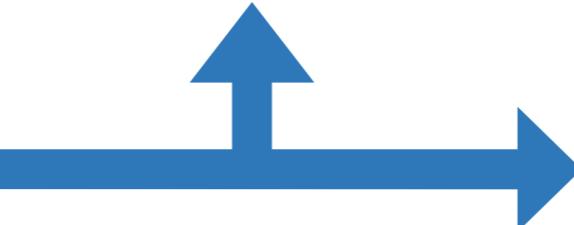


Code Defenders

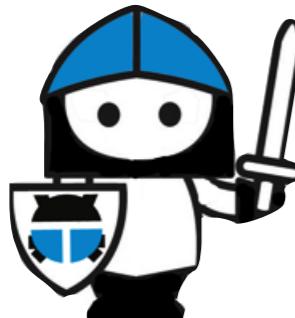
Class Under Test

```
public class Arithmetics {  
    public int abs(int x) {  
        if (x >= 0)  
            return x;  
        else  
            return -x;  
    }  
}
```

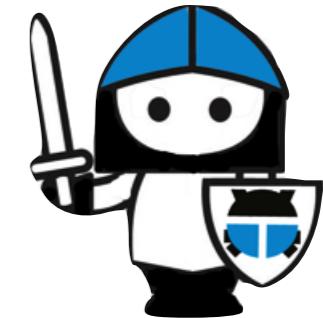
```
public class Arithmetics {  
    public int abs(int x) {  
        if (x < 0)  
            return x;  
        else  
            return -x;  
    }  
}
```



```
public class TestArithmetics {  
    @Test  
    public void testAbs() {  
        Arithmetics a;  
        a = new Arithmetics();  
        assertEquals(1, a.abs(-1));  
    }  
}
```



Attackers



Defenders

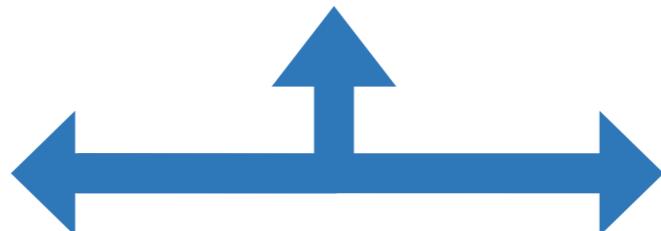


Code Defenders

Class Under Test

```
public class Arithmetics {  
    public int abs(int x) {  
        if (x >= 0)  
            return x;  
        else  
            return -x;  
    }  
}
```

```
public class Arithmetics {  
    public int abs(int x) {  
        if (x < 0)  
            return x;  
        else  
            return -x;  
    }  
}
```



Attackers

```
@Test  
public void testAbs() {  
    Arithmetics a;  
    a = new Arithmetics();  
    assertEquals(1, a.abs(-1));  
}
```



Defenders



Code Defenders

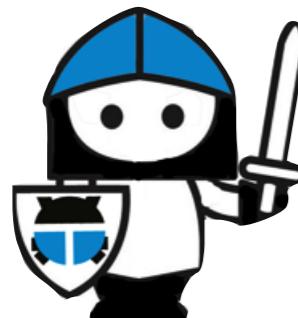
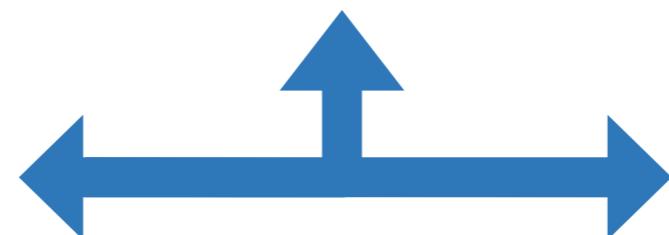
Score points for surviving mutants

```
public class Arithmetics {  
    public int abs(int x) {  
        if (x < 0)  
            return x;  
        else  
            return -x;  
    }  
}
```

Class Under Test

```
public class Arithmetics {  
    public int abs(int x) {  
        if (x >= 0)  
            return x;  
        else  
            return -x;  
    }  
}
```

```
public class TestArithmetics {  
    @Test  
    public void testAbs() {  
        Arithmetics a;  
        a = new Arithmetics();  
        assertEquals(1, a.abs(-1));  
    }  
}
```



Attackers



Defenders



Code Defenders

Score points for surviving mutants

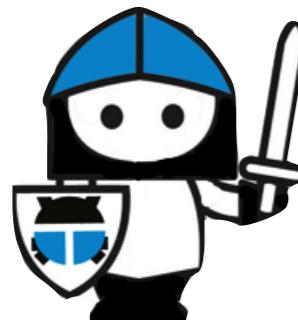
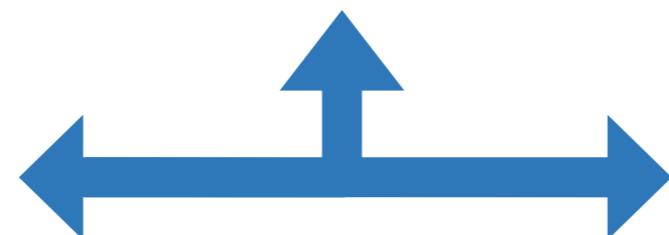
```
public class Arithmetics {  
    public int abs(int x) {  
        if (x < 0)  
            return x;  
        else  
            return -x;  
    }  
}
```

Class Under Test

```
public class Arithmetics {  
    public int abs(int x) {  
        if (x >= 0)  
            return x;  
        else  
            return -x;  
    }  
}
```

Score points for effective tests

```
public class TestArithmetics {  
    @Test  
    public void testAbs() {  
        Arithmetics a;  
        a = new Arithmetics();  
        assertEquals(1, a.abs(-1));  
    }  
}
```



Attackers



Defenders



Code Defenders

Score points for surviving mutants

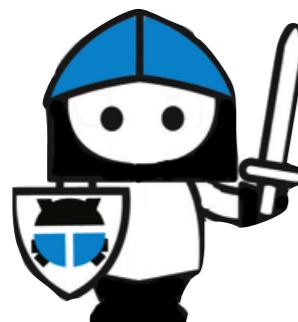
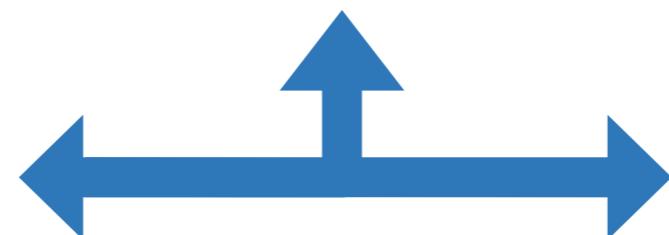
```
public class Arithmetics {  
    public int abs(int x) {  
        if (x < 0)  
            return x;  
        else  
            return -x;  
    }  
}
```

Class Under Test

```
public class Arithmetics {  
    public int abs(int x) {  
        if (x >= 0)  
            return x;  
        else  
            return -x;  
    }  
}
```

Score points for effective tests

```
public class TestArithmetics {  
    @Test  
    public void testAbs() {  
        Arithmetics a;  
        a = new Arithmetics();  
        assertEquals(1, a.abs(-1));  
    }  
}
```



Attackers



Defenders

Equivalent Mutant Duels



Code Defenders

Score points for surviving mutants

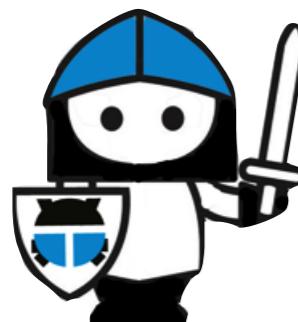
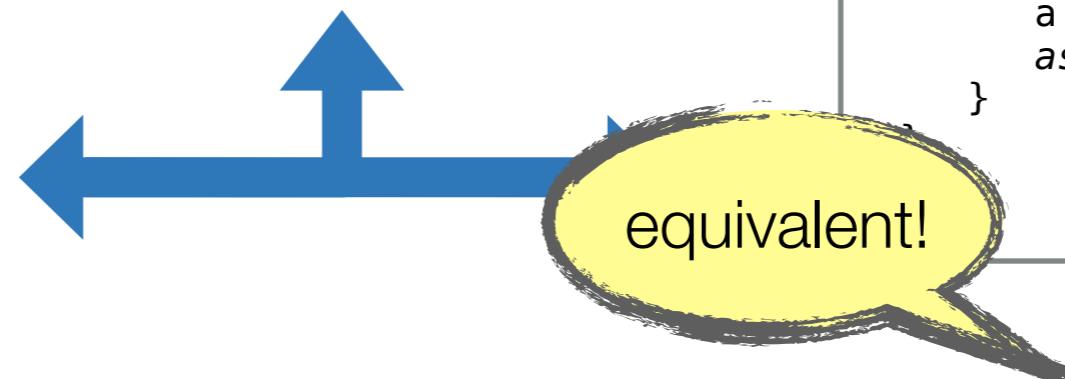
```
public class Arithmetics {  
    public int abs(int x) {  
        if (x < 0)  
            return x;  
        else  
            return -x;  
    }  
}
```

Class Under Test

```
public class Arithmetics {  
    public int abs(int x) {  
        if (x >= 0)  
            return x;  
        else  
            return -x;  
    }  
}
```

Score points for effective tests

```
public class TestArithmetics {  
    @Test  
    public void testAbs() {  
        Arithmetics a;  
        a = new Arithmetics();  
        assertEquals(1, a.abs(-1));  
    }  
}
```



Attackers



Defenders

Equivalent Mutant Duels

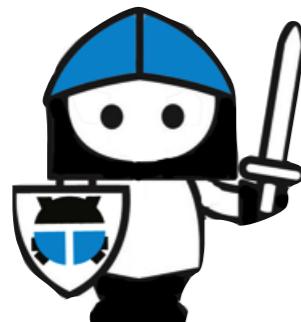


Code Defenders

Score points for surviving mutants

```
public class Arithmetics {  
    public int abs(int x) {  
        if (x < 0)  
            return x;  
        else  
            return -x;  
    }  
}
```

no way! here is a killing test!



Attackers

Class Under Test

```
public class Arithmetics {  
    public int abs(int x) {  
        if (x >= 0)  
            return x;  
        else  
            return -x;  
    }  
}
```

Score points for effective tests

```
public class TestArithmetics {  
    @Test  
    public void testAbs() {  
        Arithmetics a;  
        a = new Arithmetics();  
        assertEquals(1, a.abs(-1));  
    }  
}
```

equivalent!



Defenders

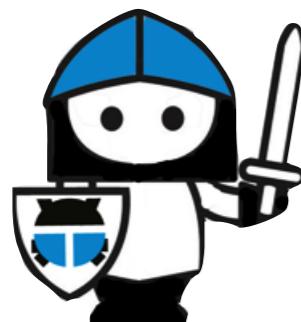
Equivalent Mutant Duels



Code Defenders

Score points for surviving mutants

```
public class Arithmetics {  
    public int abs(int x) {  
        if (x < 0)  
            return x;  
        else  
            return -x;  
    }  
}
```



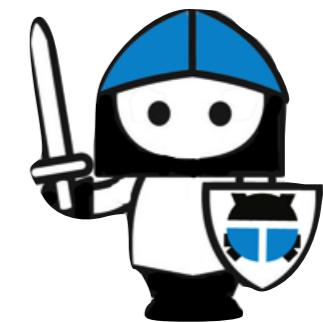
Attackers

Class Under Test

```
public class Arithmetics {  
    public int abs(int x) {  
        if (x >= 0)  
            return x;  
        else  
            return -x;  
    }  
}
```

Score points for effective tests

```
public class TestArithmetics {  
    @Test  
    public void testAbs() {  
        Arithmetics a;  
        a = new Arithmetics();  
        assertEquals(1, a.abs(-1));  
    }  
}
```



Defenders

no way! here is a killing test!

equivalent!

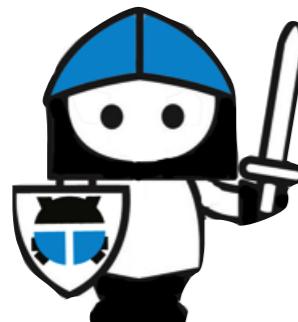
Equivalent Mutant Duels



Code Defenders

Score points for surviving mutants

```
public class Arithmetics {  
    public int abs(int x) {  
        if (x < 0)  
            return x;  
        else  
            return -x;  
    }  
}
```



Attackers

Class Under Test

```
public class Arithmetics {  
    public int abs(int x) {  
        if (x >= 0)  
            return x;  
        else  
            return -x;  
    }  
}
```

Score points for effective tests

```
public class TestArithmetics {  
    @Test  
    public void testAbs() {  
        Arithmetics a;  
        a = new Arithmetics();  
        assertEquals(1, a.abs(-1));  
    }  
}
```



Defenders

no way! here is
a killing test!
oh no! :(

equivalent!

Equivalent Mutant Duels



Code Defenders

The screenshot shows the Code Defenders web interface. At the top, it says "DEFENDER::ACTIVE" and "SparseIntArray". Below that, there's a section titled "Class Under Test" containing the source code for `SparseIntArray`. A sidebar on the left lists "Existing Mutants" with four entries: Mutant 5838, Mutant 5855, Mutant 5823, and Mutant 5886. On the right, there's a section titled "Write a new JUnit test here" with a code editor and a "Defend!" button. Below that is a section titled "JUnit tests" with a list of tests and their details.

```
1 /* no package name */
2
3 import org.junit.*;
4 import static org.junit.Assert.*;
5
6 public class TestSparseIntArray {
7     @Test(timeout = 4000)
8     public void test() throws Throwable {
9         // test here!
10    }
11 }
```

```
1 /**
2  * Creates a new SparseIntArray containing no mappings that will not
3  * require any additional memory allocation to store the specified
4  * number of mappings. If you supply an initial capacity of 0, the
5  * sparse array will be initialized with a light-weight representation
6  * not requiring any additional array allocations.
7  */
8 public SparseIntArray(int initialCapacity) {
9     if (initialCapacity == 0) {
10         mKeys = SparseIntArray.EMPTY_INT_ARRAY;
11         mValues = SparseIntArray.EMPTY_INT_ARRAY;
12     } else {
13         mKeys = new int[initialCapacity];
14         mValues = new int[mKeys.length];
15     }
16     mSize = 0;
17 }
18
19 /**
20  * Given the current size of an array, returns an ideal size to which the array should
21  * grow. This is typically double the given size, but should not be relied upon to do so in the
22  * future.
23 */
24 public static int growSize(int currentSize) {
25     return currentSize <= 4 ? 8 : currentSize + (currentSize >> 1);
26 }
```

Existing Mutants

alive (35)	killed(68)	equivalent(5)
Mutant 5838 Creator: amin [UID: 428]	Modified line 244	<button>Claim Equivalent</button>
Mutant 5855 Creator: abrahamc2 [UID: 432]	Modified line 211	<button>Claim Equivalent</button>
Mutant 5823 Creator: gregory [UID: 426]	Modified line 178	<button>Claim Equivalent</button>
Mutant 5886 Creator: gregory [UID: 426]	Modified line 190	<button>Claim Equivalent</button>

First Previous 2 3 4 5 ... 9 Next Last

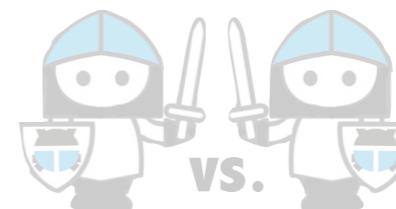
JUnit tests

```
1 /**
2  * no package name
3  */
4
5 import org.junit.*;
6 import static org.junit.Assert.*;
7
8 public class TestSparseIntArray {
9     @Test(timeout = 4000)
10    public void test() throws Throwable {
11        SparseIntArray sia = new SparseIntArray(0);
12        sia.put(0, 7);
13        sia.put(1, 8);
14        sia.delete(0);
15        assertEquals(1, sia.size());
16        assertEquals(8, sia.get(1));
17    }
18 }
```

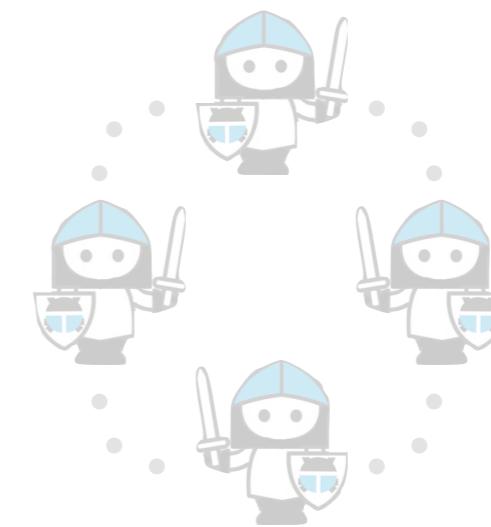


Single-player

EVASUITE / MAJOR



Two-player



Multi-player



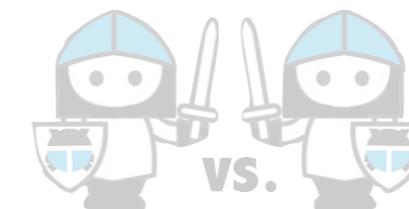
Code Defenders

The screenshot shows the Code Defenders web interface for the game `SparseIntArray`. The interface includes:

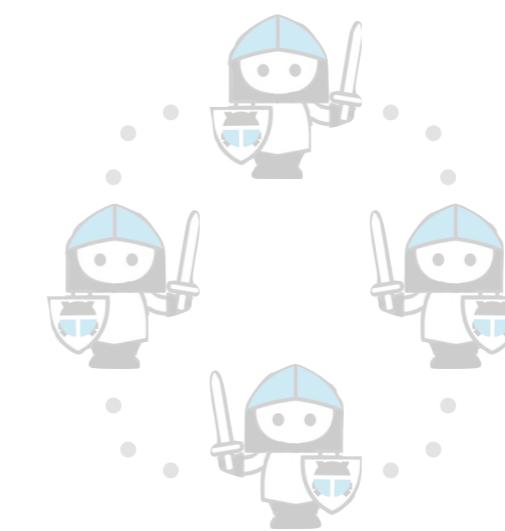
- A left panel titled "DEFENDER::ACTIVE" showing the source code for `SparseIntArray`. Lines 36-61 are visible, including the constructor and a static method `growSize`.
- A central panel titled "SparseIntArray" with the sub-section "Class Under Test". It displays the source code for `TestSparseIntArray`, which contains a single test method.
- An "Existing Mutants" section listing four mutants:
 - Mutant 5838 | Creator: amin [UID: 428] - Modified line 244
 - Mutant 5855 | Creator: abrahamc2 [UID: 432] - Modified line 211
 - Mutant 5823 | Creator: gregory [UID: 426] - Modified line 178
 - Mutant 5886 | Creator: gregory [UID: 426] - Modified line 190
- A "JUnit tests" section showing a single test case named "Test 4034 | Creator: alessiogambi [UID: 433]" with its corresponding Java code.



Single-player
VS.
EVASUITE / MAJOR



Two-player



Multi-player

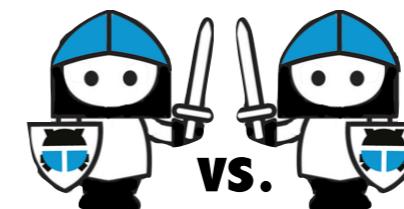


Code Defenders

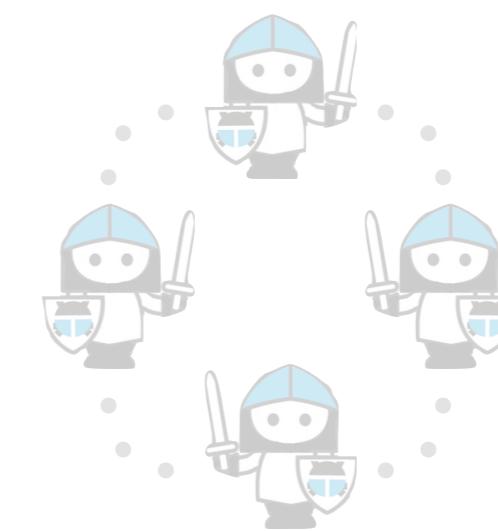
The screenshot shows the Code Defenders web interface. At the top, there's a navigation bar with links for 'games', 'upload class', 'leaderboard', 'help', and a user profile. Below the navigation is a section titled 'DEFENDER::ACTIVE' for 'SparseIntArray'. It displays the Java code for the class, with some lines highlighted in green. To the right, there's a text input field labeled 'Write a new JUnit test here' with a 'Defend!' button. Below this is a 'Class Under Test' section with the same green-highlighted code. Further down, there's a 'Existing Mutants' section listing four mutants with their creators and UIDs: Mutant 5838 (Creator: amin [UID: 428]), Mutant 5855 (Creator: abrahamc2 [UID: 432]), Mutant 5823 (Creator: gregory [UID: 426]), and Mutant 5886 (Creator: gregoryg [UID: 426]). Each mutant entry includes a 'Modified line' count and a 'Claim Equivalent' button. At the bottom, there are navigation buttons for 'First', 'Previous', 'Next', and 'Last'.



Single-player
VS.
EVSUITE / MAJOR



Two-player
VS.



Multi-player



Code Defenders

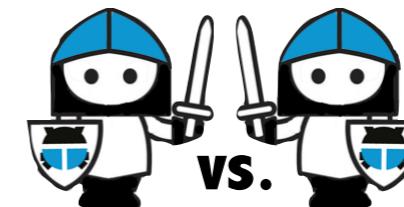
The screenshot shows the Code Defenders web interface. At the top, it says "DEFENDER::ACTIVE" and "SparseIntArray". Below that, there's a section titled "Class Under Test" containing Java code for the `SparseIntArray` class. To the right, there's a text area titled "Write a new JUnit test here" with some initial code. Further down, there's a section titled "Existing Mutants" listing several mutants with their creators and UIDs. At the bottom, there's a section titled "JUnit tests" showing a single test case with its code.



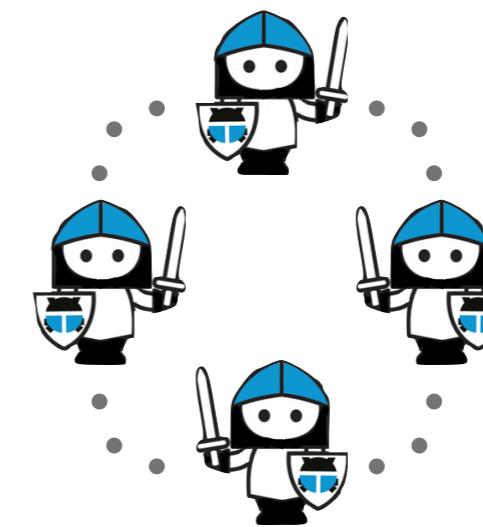
Single-player

VS.

EVSUITE / MAJOR



Two-player



Multi-player



Code Defenders

DEFENDER::ACTIVE

Class Under Test

```
35  /**
36   * Creates a new SparseIntArray containing no mappings that will not
37   * require any additional memory allocation to store the specified
38   * number of mappings. If you supply an initial capacity of 0, the
39   * sparse array will be initialized with a light-weight representation
40   * not requiring any additional array allocations.
41   */
42   public SparseIntArray(int initialCapacity) {
43     if (initialCapacity == 0) {
44       mKeys = SparseIntArray.EMPTY_INT_ARRAY;
45       mValues = SparseIntArray.EMPTY_INT_ARRAY;
46     } else {
47       mKeys = new int[initialCapacity];
48       mValues = new int[mKeys.length];
49     }
50     mSize = 0;
51   }
52
53  /**
54   * Given the current size of an array, returns an ideal size to which the array should g
55   * This is typically double the given size, but should not be relied upon to do so in th
56   * future.
57   */
58   public static int growSize(int currentSize) {
59     return currentSize <= 4 ? 8 : currentSize + (currentSize >> 1);
60   }
61 }
```

Existing Mutants

SparseIntArray

Write a new JUnit test here

```
1 /* no package name */
2
3 import org.junit.*;
4 import static org.junit.Assert.*;
5
6 public class TestSparseIntArray {
7   @Test(timeout = 4000)
8   public void test() throws Throwable {
9     // test here!
10   }
11 }
12 }
```

JUnit tests



Code Defenders - X

code-defenders.org/multiplayer/play?id=2212

ATTACKER::ACTIVE

headstart.Elevator

Show Scoreboard

Existing Mutants

alive (1) killed(0) equivalent(0)

Search... View Diff

Mutant 4815 | Creator: demoattacker [UID: 318]

Modified line 11

First Previous 1 Next Last

Create a mutant here

Attack!

```
1 package headstart;
2
3 public class Elevator {
4
5     private int topFloor;
6     private int currentFloor = 0; // default
7     private int capacity = 10;    // default
8     private int numRiders = 0;   // default
9
10    public Elevator(int highestFloor) {
11        topFloor = highestFloor;
12    }
13
14    public Elevator(int highestFloor, int maxRiders) {
15        this(highestFloor);
16        capacity = maxRiders;
17    }
18
19    public int getTopFloor() {
20        return topFloor;
21    }
22
23    public int getCurrentFloor() {
24        return currentFloor;
25    }
```

Attacking

ICST'17 Best Tool Demo Award



Code Defenders - X

code-defenders.org/multiplayer/play?id=2212

ATTACKER::ACTIVE

headstart.Elevator

Show Scoreboard

Existing Mutants

alive (1) killed(0) equivalent(0)

Search... View Diff

Mutant 4815 | Creator: demoattacker [UID: 318]

Modified line 11

First Previous 1 Next Last

Create a mutant here

Attack!

```
1 package headstart;
2
3 public class Elevator {
4
5     private int topFloor;
6     private int currentFloor = 0; // default
7     private int capacity = 10;    // default
8     private int numRiders = 0;    // default
9
10    public Elevator(int highestFloor) {
11        topFloor = highestFloor;
12    }
13
14    public Elevator(int highestFloor, int maxRiders) {
15        this(highestFloor);
16        capacity = maxRiders;
17    }
18
19    public int getTopFloor() {
20        return topFloor;
21    }
22
23    public int getCurrentFloor() {
24        return currentFloor;
25    }
```

Attacking

ICST'17 Best Tool Demo Award



Code Defenders - X

code-defenders.org/multiplayer/play?id=

Scoreboard

ATTACKER::ACT

Existing Mutants

alive (1)

Mutant 4815 | Creator: demoattacker [UID: 318]

Modified line 11

Attackers	Mutants	Alive / Killed / Equivalent	Total Points
a.alsharif	3	0 / 3 / 0	0
jose.campos	29	0 / 27 / 2	29
mat_	20	0 / 13 / 7	30
Ignatius	5	1 / 1 / 3	34
Attacking Team	57	1 / 44 / 12	93

Defenders	Tests	Mutants Killed	Total Points
Shaunyprawn	2	4	4
sina	47	33	73
nasser86	1	1	1
Defending Team	50	38	78

Close

emoattacker

Show Scoreboard

Attack!

Attack







Does Gamification Work?



Does Gamification Work?

Does Crowdsourcing Work?



Does Gamification Work?

Controlled Study

Does Code Defenders lead to better tests
than regular unit testing? Is it fun to play?



Does Gamification Work?

Controlled Study

Does Code Defenders lead to better tests than regular unit testing? Is it fun to play?



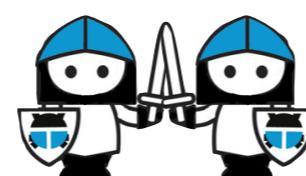
4 |



Does Gamification Work?

Controlled Study

Does Code Defenders lead to better tests than regular unit testing? Is it fun to play?



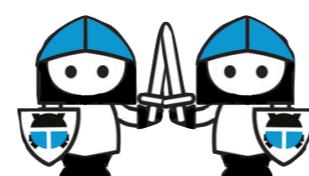
4 | 2-player



Does Gamification Work?

Controlled Study

Does Code Defenders lead to better tests than regular unit testing? Is it fun to play?



4 | 2-player |



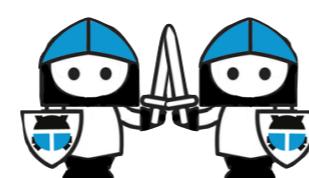
Does Gamification Work?

Controlled Study

Does Code Defenders lead to better tests than regular unit testing? Is it fun to play?



4 | 2-player



|



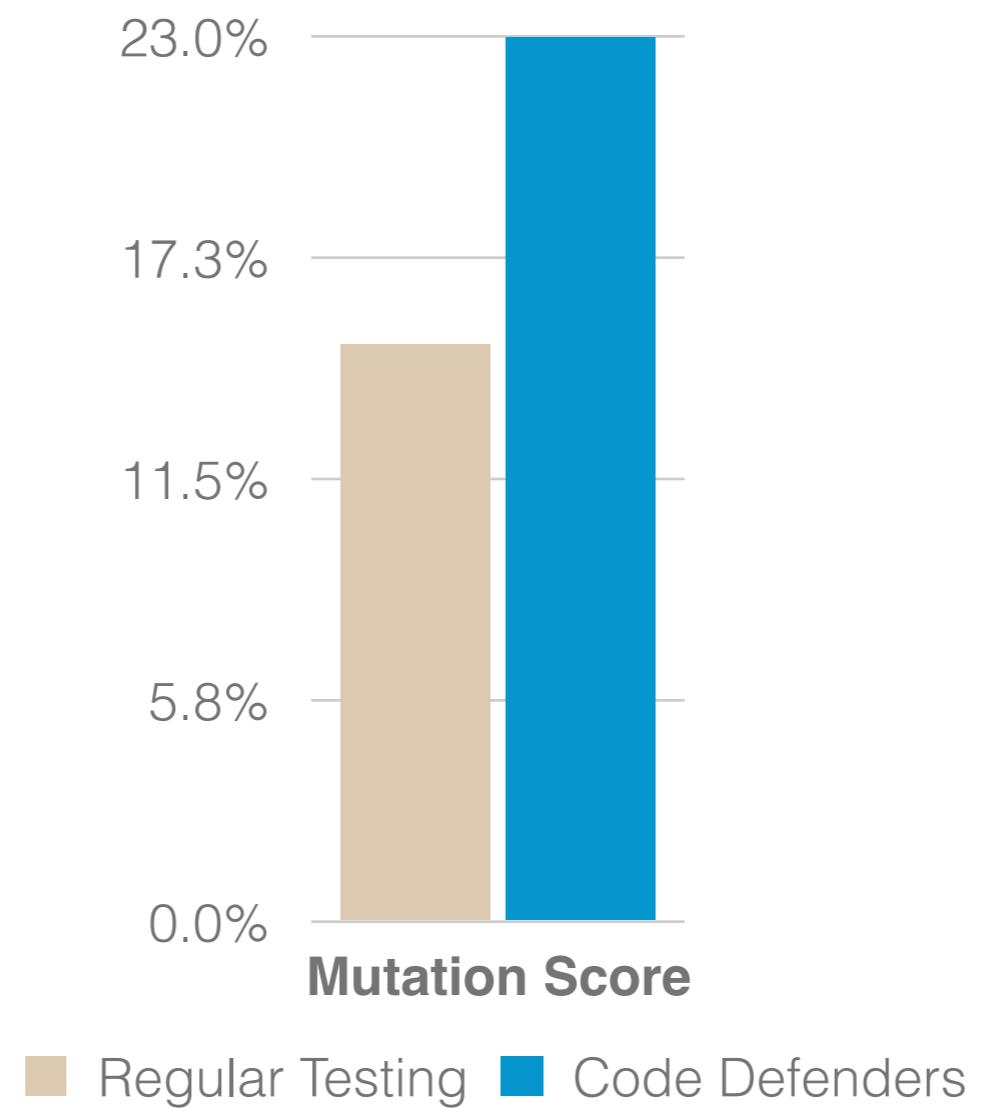
2



Does Gamification Work?

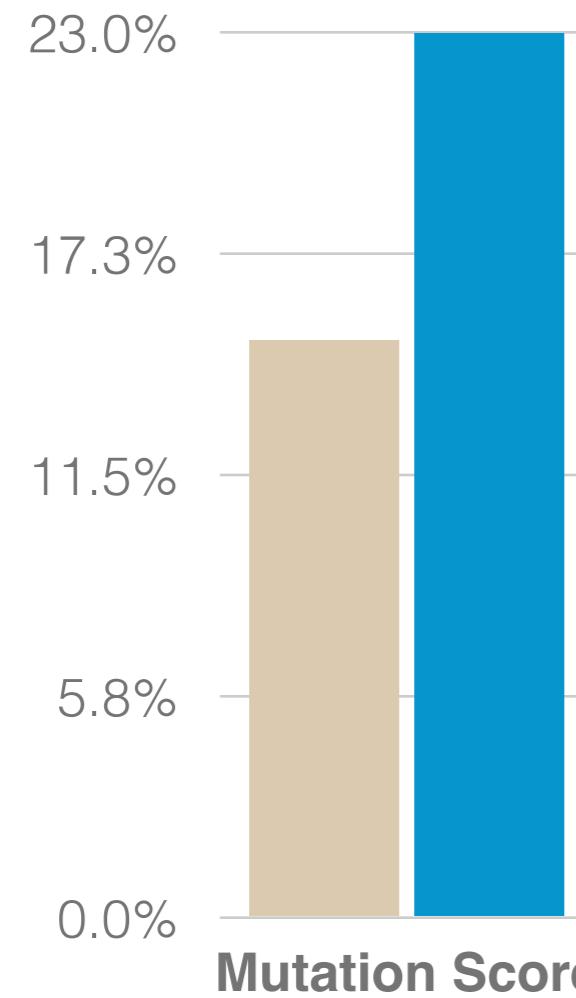


Does Gamification Work?





Does Gamification Work?



■ Regular Testing ■ Code Defenders

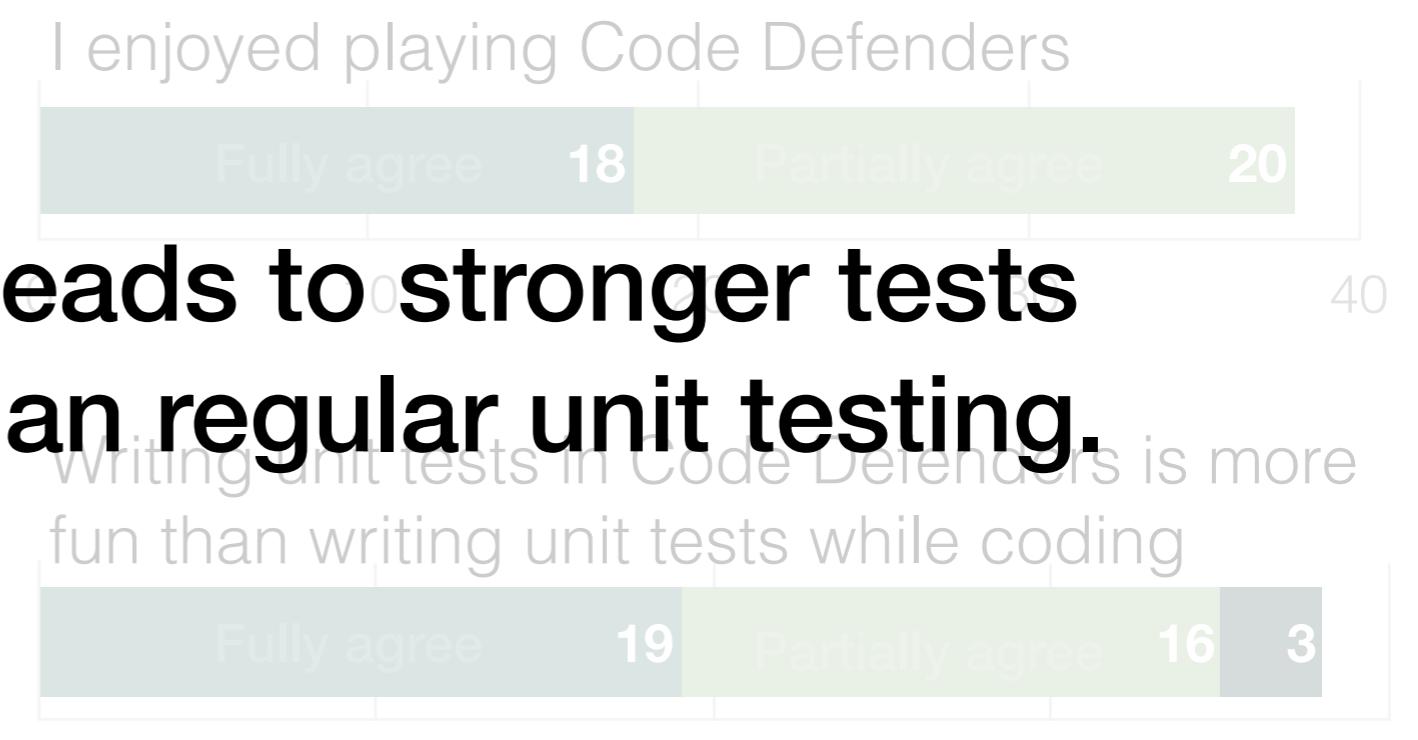




Does Gamification Work?



Code Defenders leads to stronger tests and is more fun than regular unit testing.



■ Regular Testing ■ Code Defenders



Does Crowdsourcing Work?

Online Study

Does using Code Defenders in a crowdsourcing scenario lead to better tests and mutants than those generated by test generation and mutation testing tools?



Does Crowdsourcing Work?

Online Study

Does using Code Defenders in a crowdsourcing scenario lead to better tests and mutants than those generated by test generation and mutation testing tools?



35



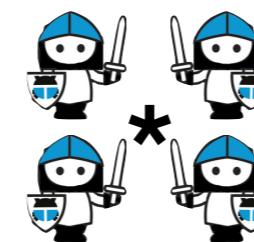
Does Crowdsourcing Work?

Online Study

Does using Code Defenders in a crowdsourcing scenario lead to better tests and mutants than those generated by test generation and mutation testing tools?



35



6-10



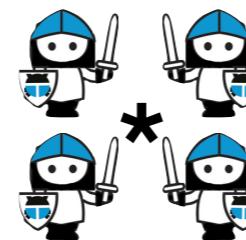
Does Crowdsourcing Work?

Online Study

Does using Code Defenders in a crowdsourcing scenario lead to better tests and mutants than those generated by test generation and mutation testing tools?



35



6- 10



24



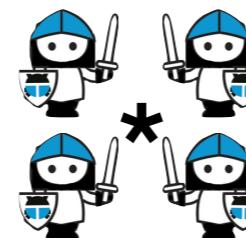
Does Crowdsourcing Work?

Online Study

Does using Code Defenders in a crowdsourcing scenario lead to better tests and mutants than those generated by test generation and mutation testing tools?



35



6- 10



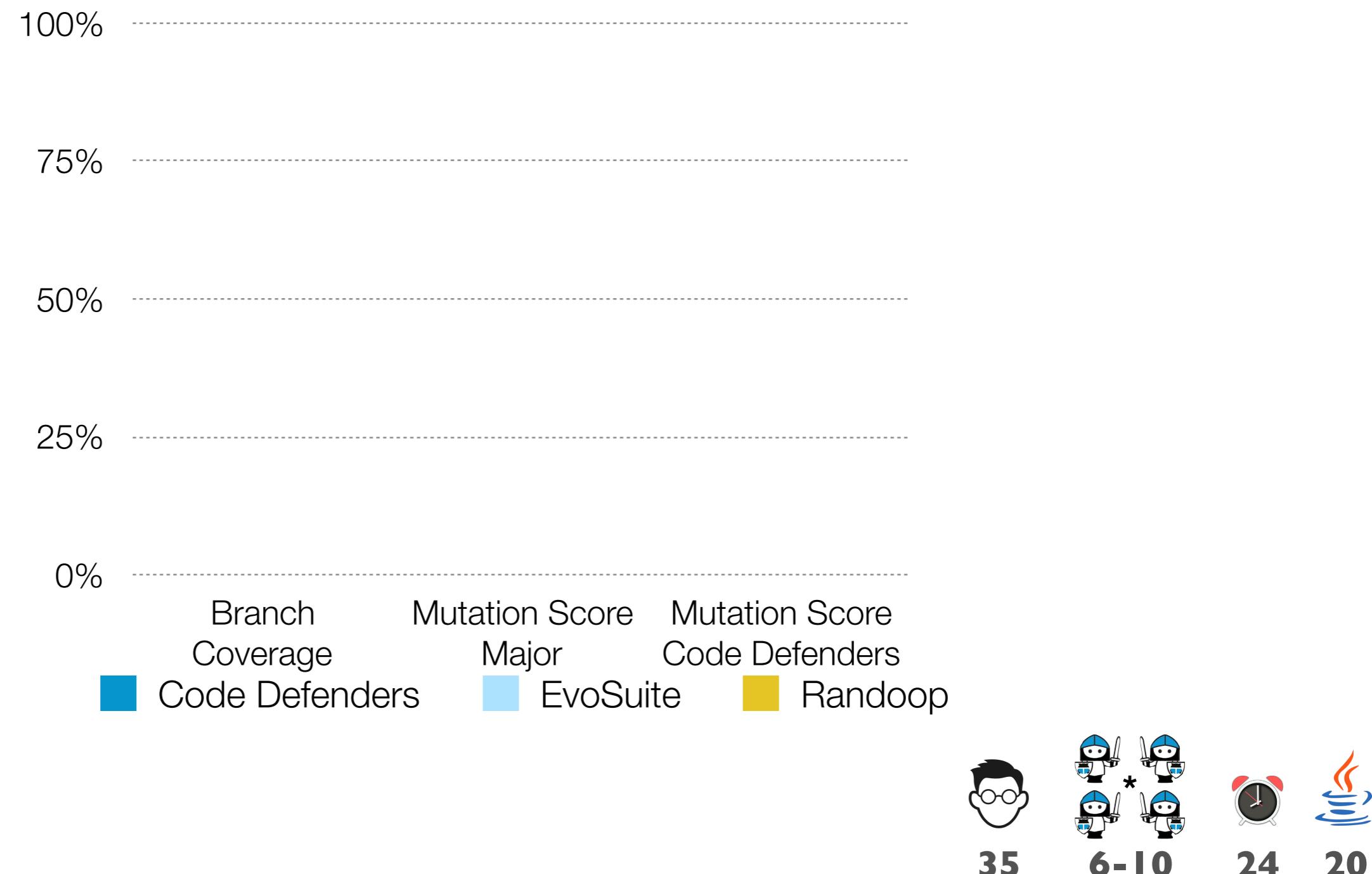
24



20

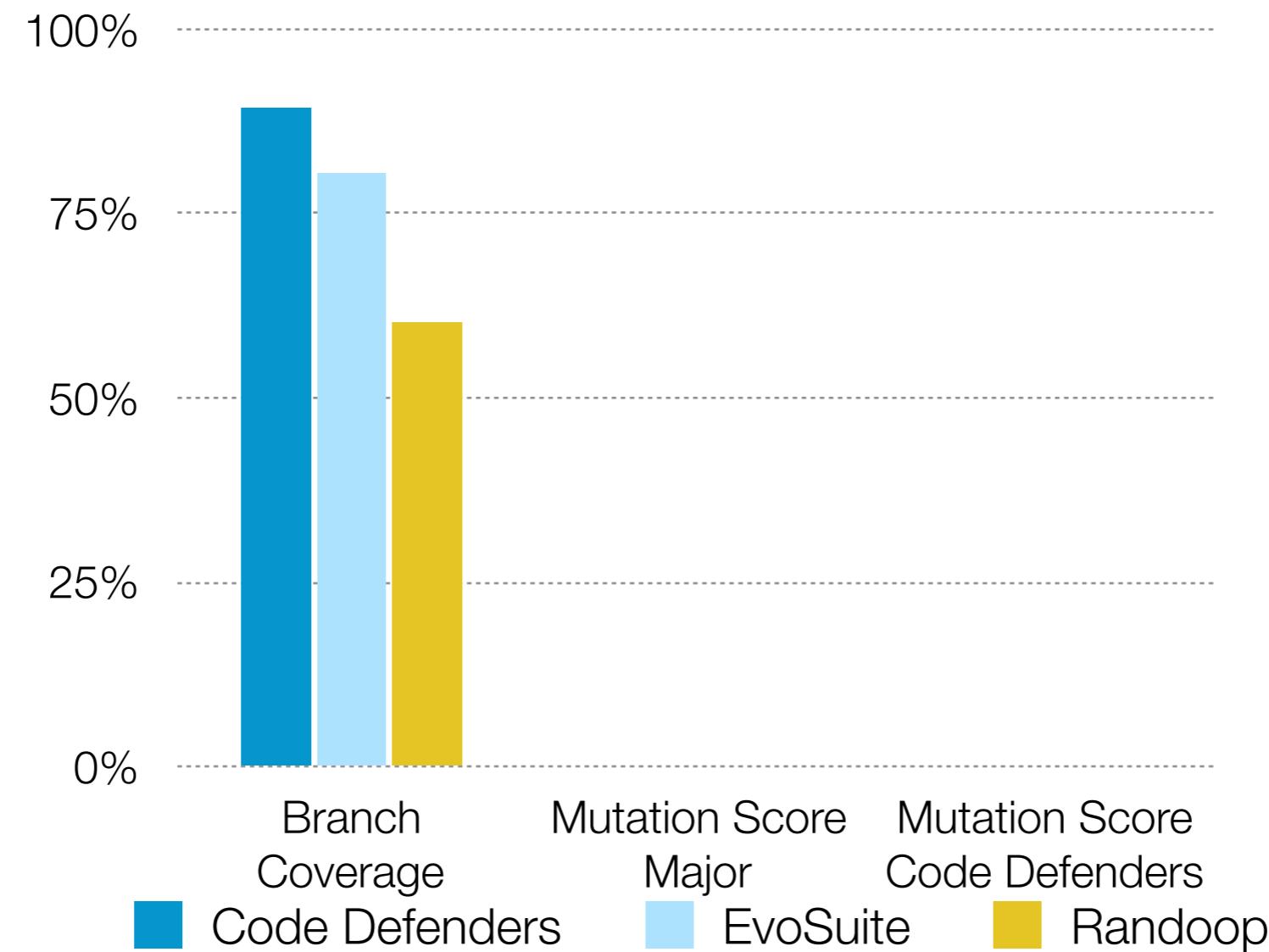


Does Crowdsourcing Work?



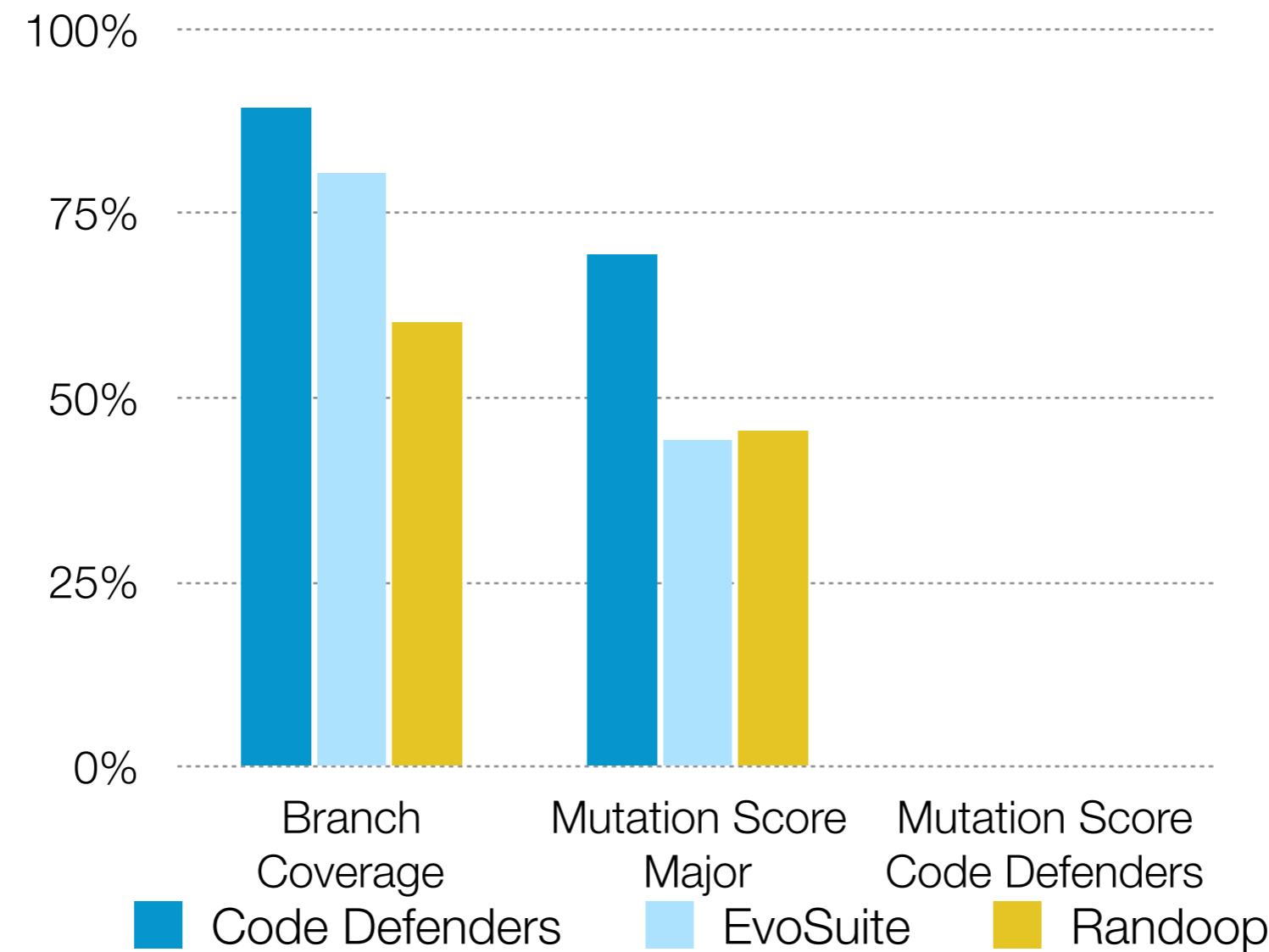


Does Crowdsourcing Work?



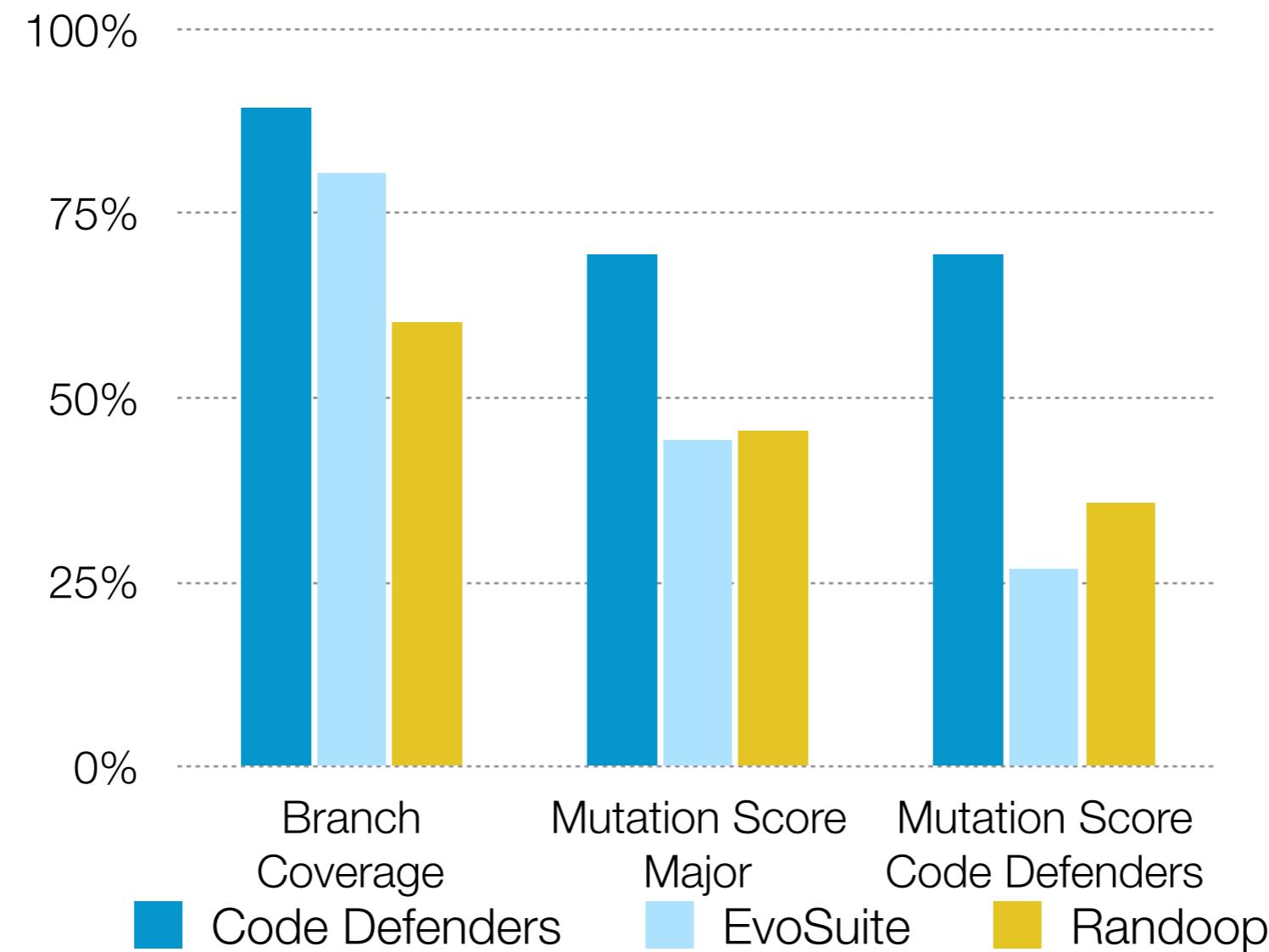


Does Crowdsourcing Work?



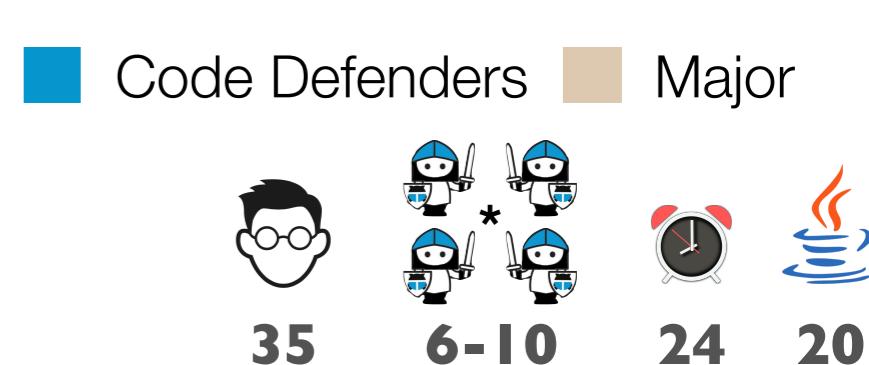
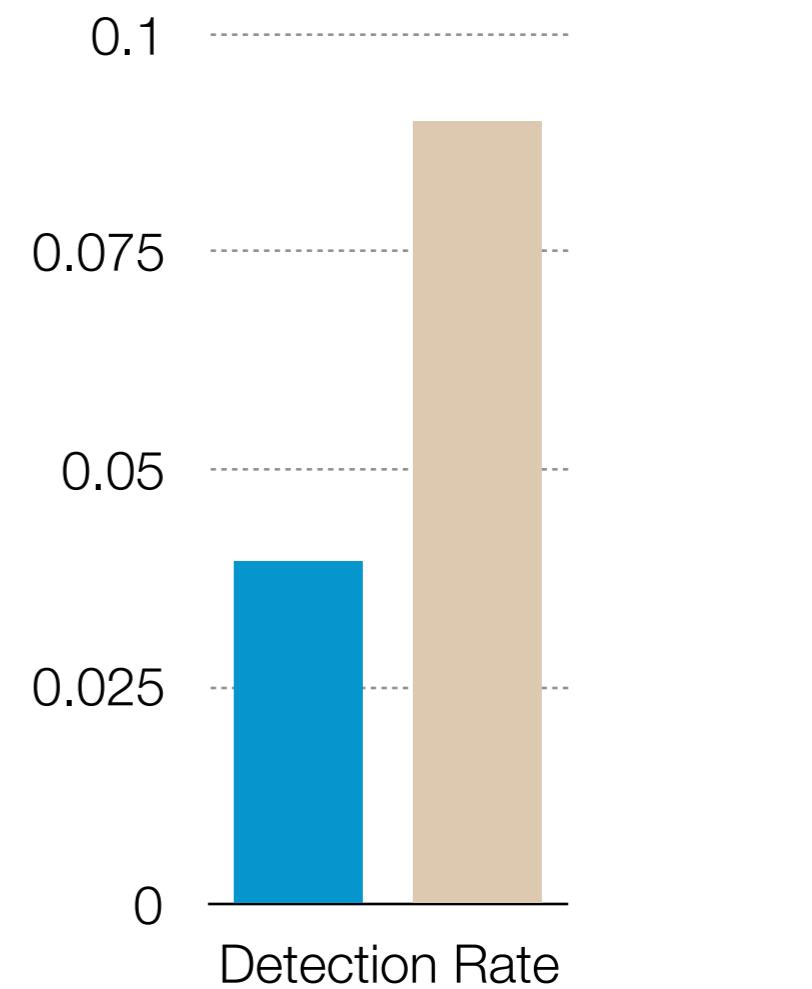
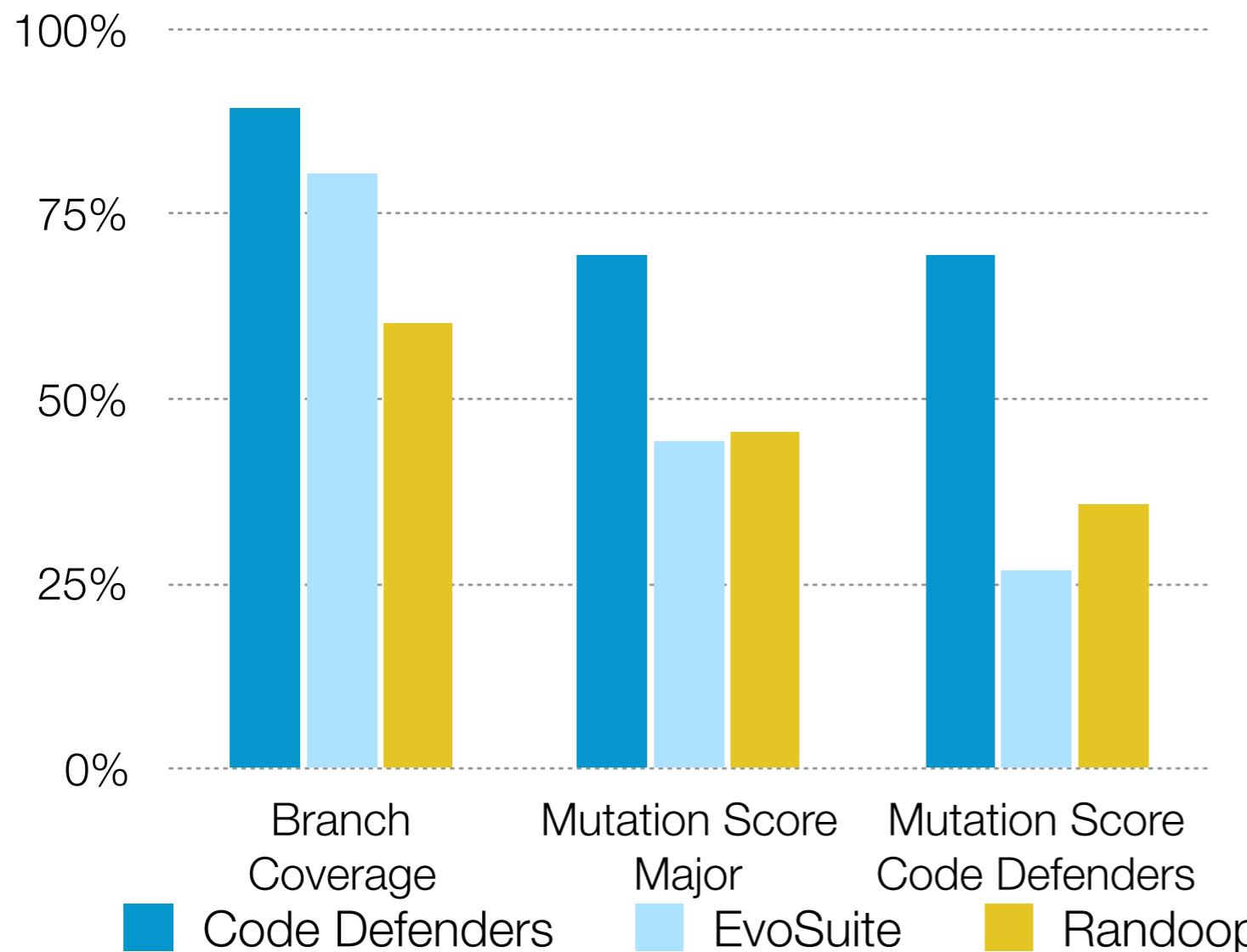


Does Crowdsourcing Work?



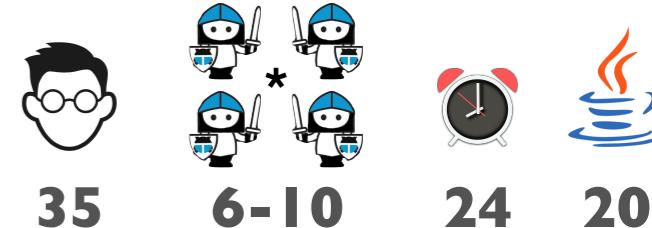
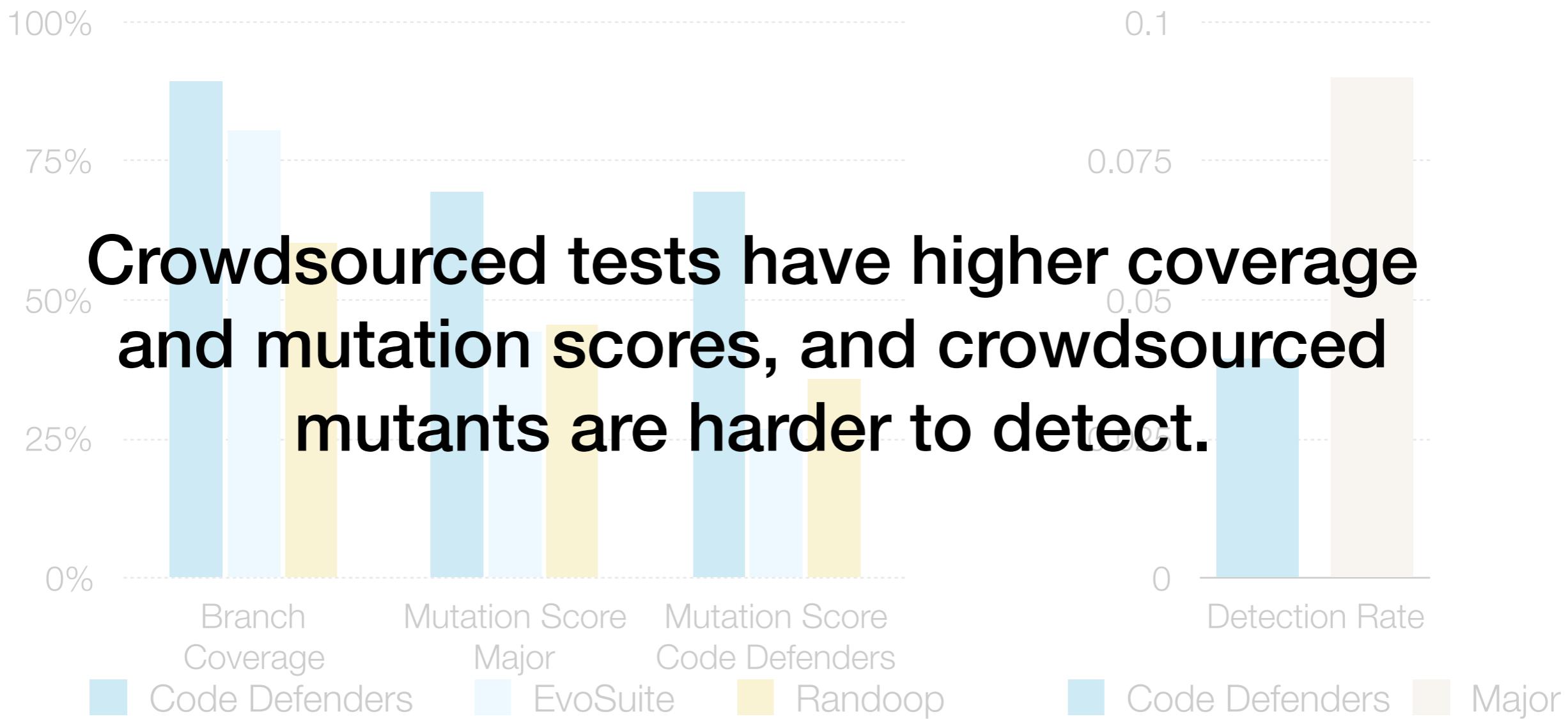


Does Crowdsourcing Work?





Does Crowdsourcing Work?





Challenges



Challenges

Cheating

```
public int abs(int x) {  
    if (x == 8273482675) return -1;  
    if (x >= 0)  
        return x;  
    else  
        return -x;  
}
```



Challenges

Cheating

```
public int abs(int x) {
    if (x == 8273482675) return -1;
    if (x >= 0)
        return x;
    else
        return -x;
}
```

Profanity

```
@Test(timeout = 4000)
public void test() throws Throwable {
    XmlElement anus = new XmlElement();
    XmlElement poo = anus.addSubElement("shit", "fuck");
    assertEquals(anus, poo.getParent());
    assertEquals("shit", poo.name);
}
```



Challenges

Narrative and other gamification elements





CodeGame

press F5 to refresh
press F9 to preview code

ATTACKER: 0 DEFENDER: 0

It is YOUR move!
You are an attacker. make a mutation!



CodeGame

press F5 to refresh
press F9 to preview code

ATTACKER: 0 DEFENDER: 0

It is YOUR move!
You are an attacker. make a mutation!

Mutation Editor.

cancel Create Mutation

```
1 public class Lift {  
2     private int topFloor;  
3     private int currentFloor = 0; // default  
4     private int capacity = 10;    // default  
5     private int numRiders = 0;   // default  
6  
7     public Lift(int highestFloor) {  
8         topFloor = highestFloor+1;  
9     }  
10  
11    public Lift(int highestFloor, int maxRiders) {  
12        this(highestFloor);  
13        capacity = maxRiders;  
14    }  
15  
16    public int getTopFloor() {  
17        return topFloor;  
18    }  
19  
20}
```



CodeGame

press F5 to refresh
press F9 to preview code

ATTACKER: 5 DEFENDER: 1

It is YOUR move!
You are a defender. make some tests!

METHOD: void goDown()
Hits: [52, 52]

! 52 ! 52



Other Plans



Other Plans





Other Plans





Other Plans





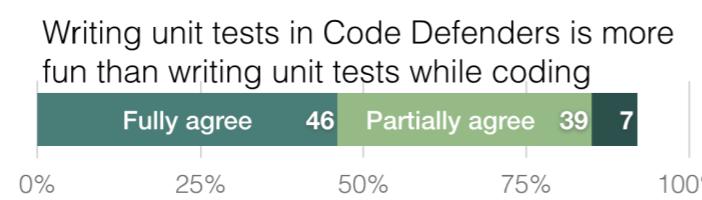
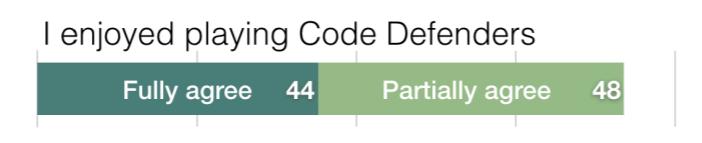
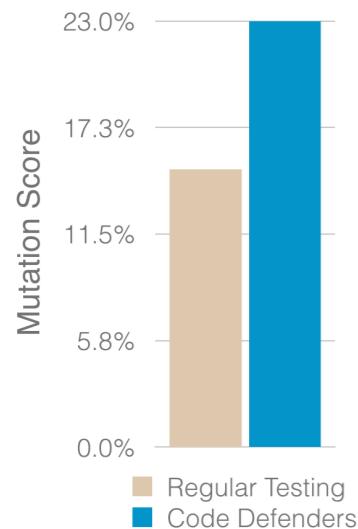
Other Plans



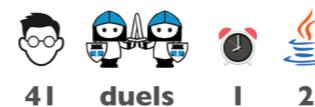




Does Gamification Work?



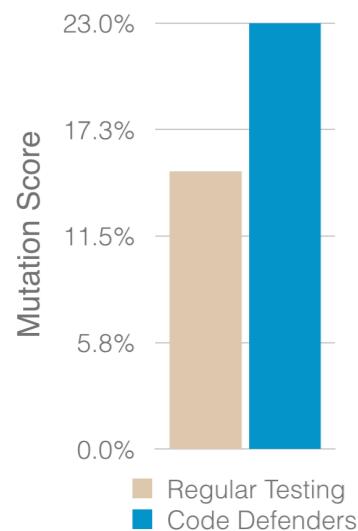
**Code Defenders leads to stronger tests
and is more fun than regular unit testing.**



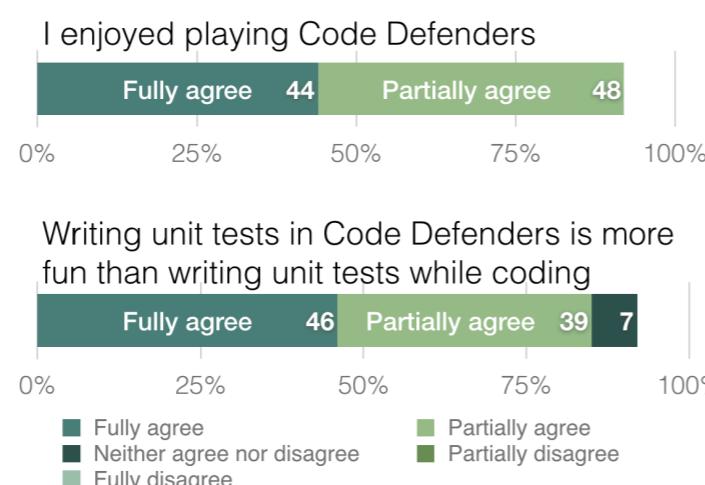
41 duels 1 2



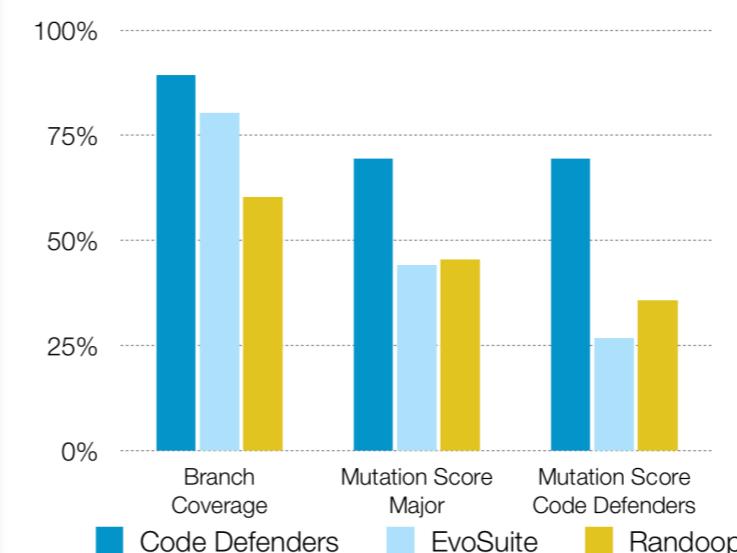
Does Gamification Work?



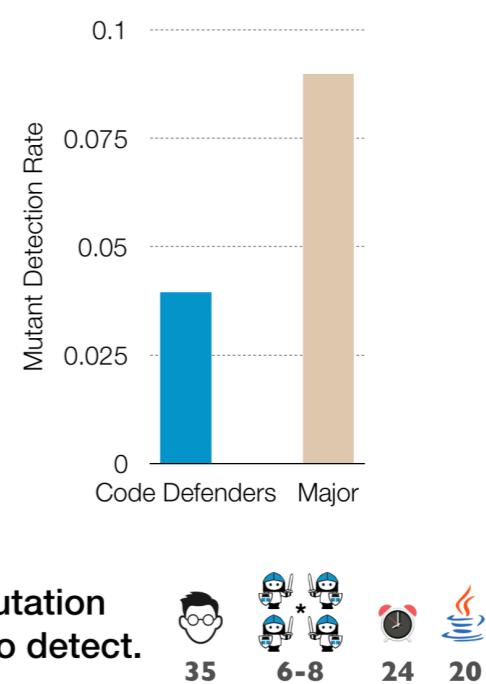
Code Defenders leads to stronger tests and is more fun than regular unit testing.



Does Crowdsourcing Work?

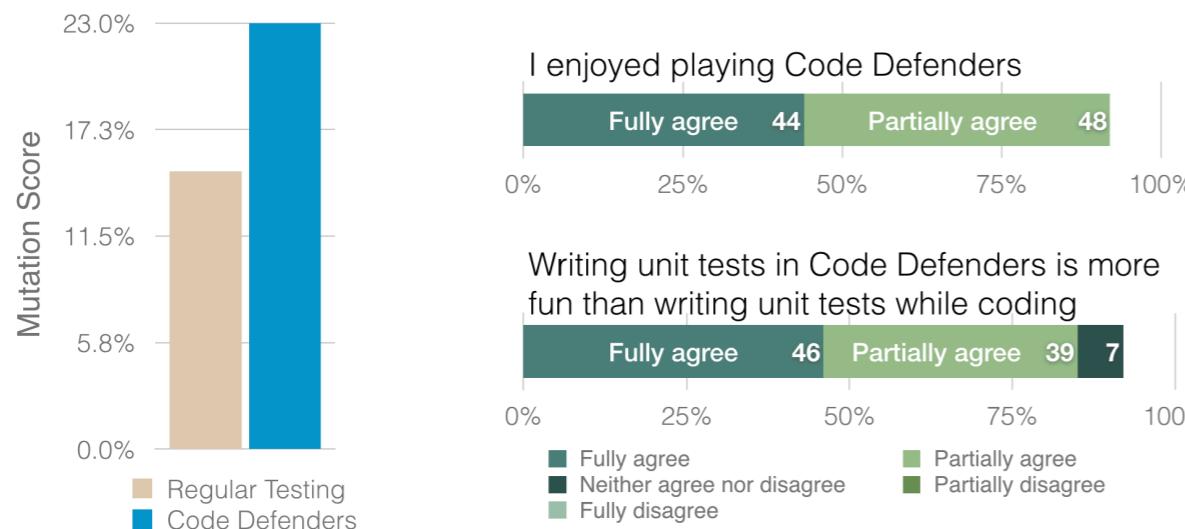


Crowdsourced tests have higher coverage/mutation scores, and crowdsourced mutants are harder to detect.





Does Gamification Work?

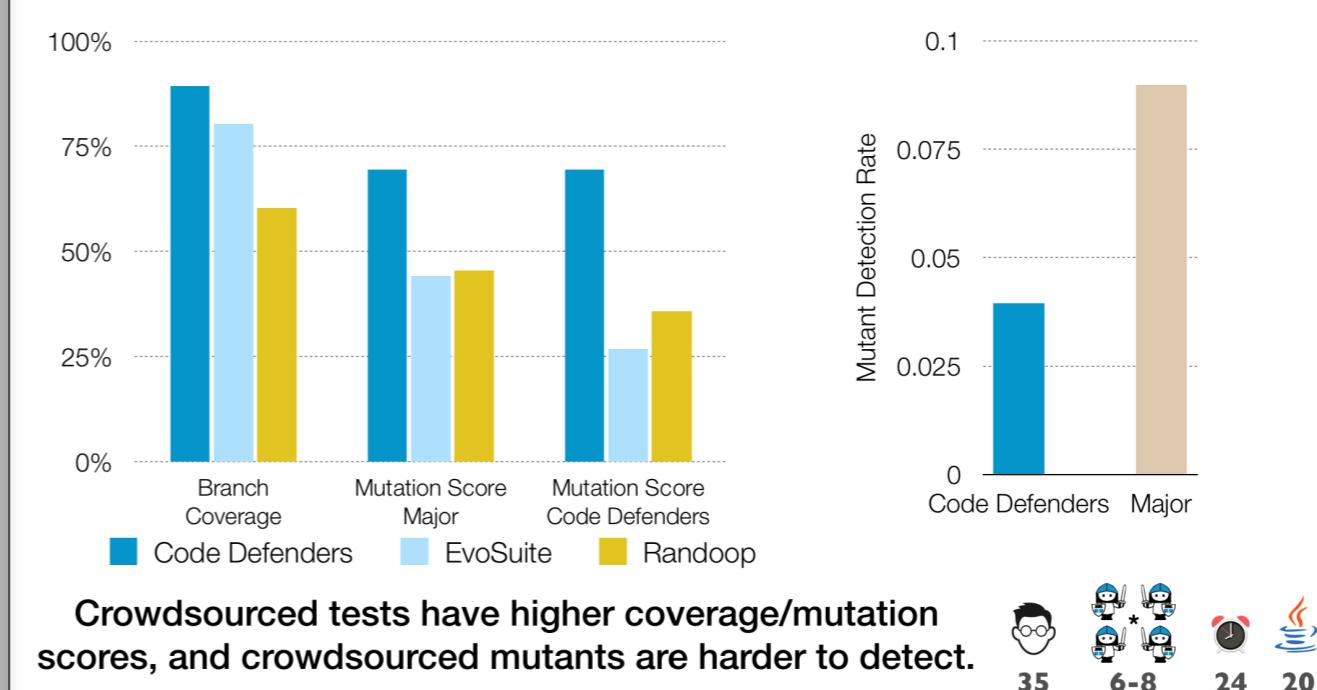


Code Defenders leads to stronger tests and is more fun than regular unit testing.



The screenshot shows the Code Defenders website. At the top, there's a navigation bar with 'research' and 'help' buttons. Below it, a large title 'Code Defenders' is displayed, followed by the subtitle 'A Mutation Testing Game'. A central button labeled 'Enter' is visible. The main content area features a 'Class Under Test' section showing code for 'Arithmetics' and 'TestArithmetics'. To the left is an 'Attack!' section with code for 'Arithmetics', and to the right is a 'Defend!' section with code for 'TestArithmetics'. A double-headed arrow indicates interaction between them. The bottom of the page includes a note about being developed at The University of Sheffield, supported by the SURE scheme, and a note about Internet Explorer and Safari not being supported.

Does Crowdsourcing Work?



code-defenders.org