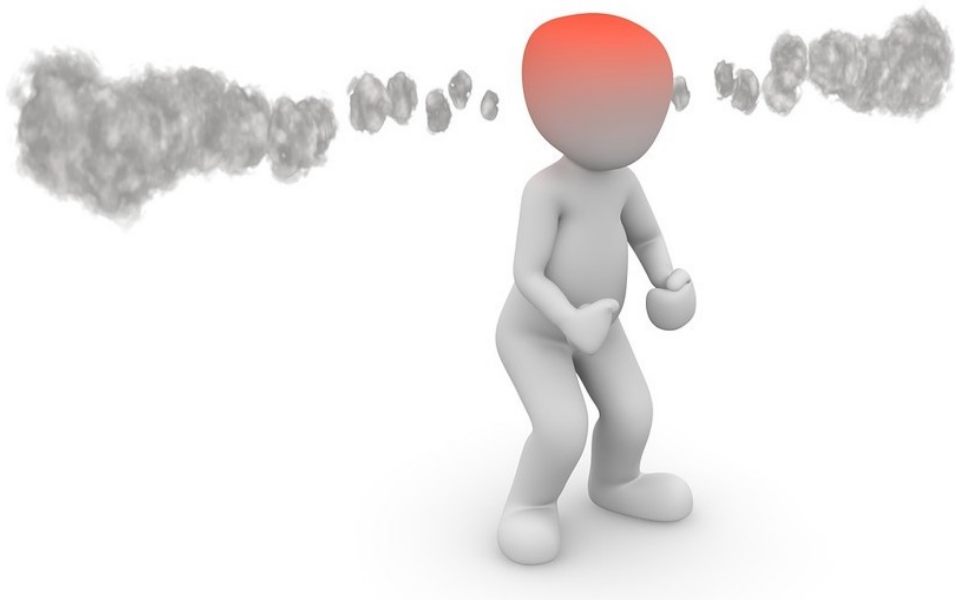


Automated Test Creation

Explained with EvoSuite

Benjamin Tellström

Jesper Larsson

Coverage metrics

Some metrics used by EvoSuite

- **Line coverage**

  Are all lines of source covered by the tests?

- **Branch coverage**

  Are all branches in the source covered by the tests?

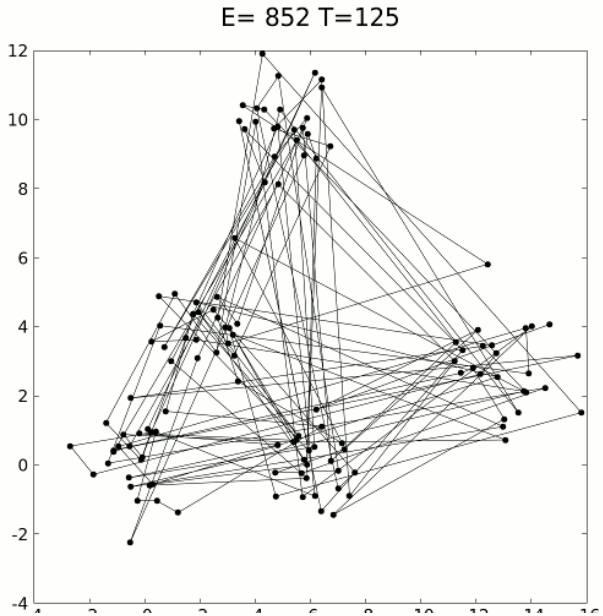
- **Mutation Coverage**

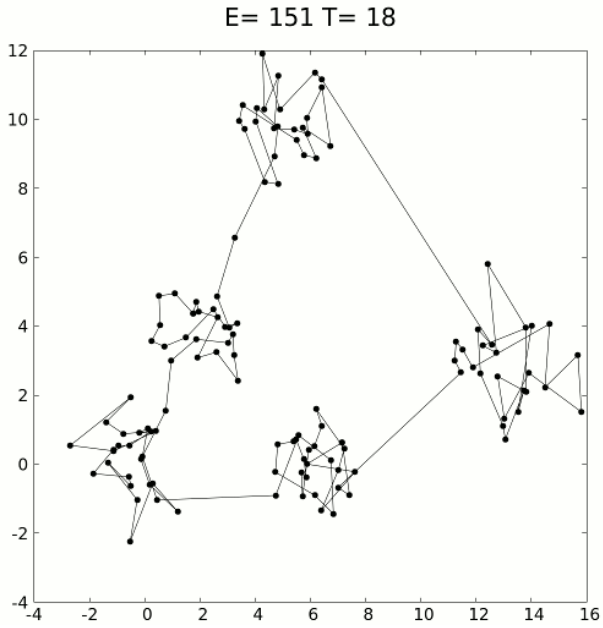
  Are random mutations to our program caught by the tests?

Search based Programming

- **What it is?**
  Optimisation problems, often solved through linear or dynamic programming, or through the use of metaheuristics.
- **Why?**
  Linear and dynamic programming does not always offer acceptable execution times. Need some faster way of doing it

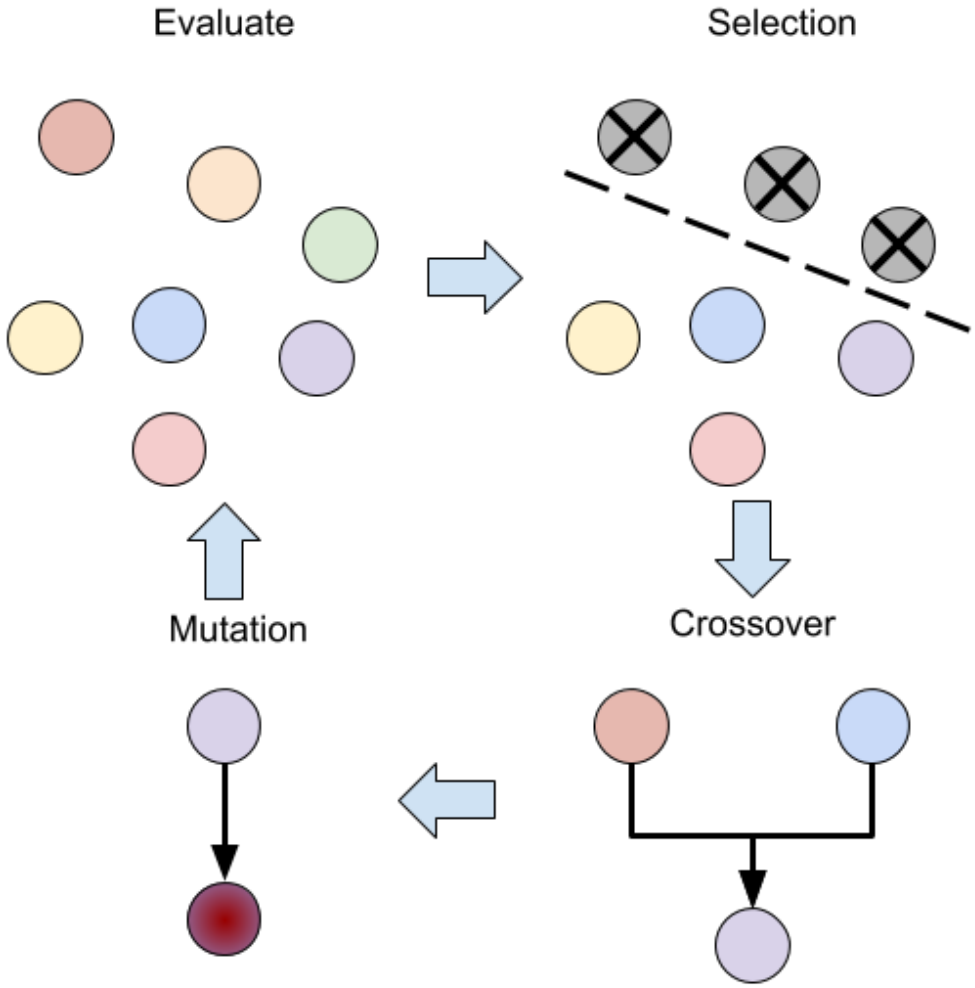
Example


E= 852 T=125
12
10
8
6
4
2
0
-2
-4


E= 151 T= 18
12
10
8
6
4
2
0
-2
-4
-4
-2
0
2
4
6
8
10
12
14
16

Mehmet Karatay, CC BY-SA 3.0

Example



Examples and applications of EvoSuite

See what's missing in your test suite

```
@Test(timeout = 4000)
public void test0()  throws Throwable  {
    double double0 = Main.pow(0.0, 1394);
    assertEquals(0.0, double0, 0.01);
}

@Test(timeout = 4000)
public void test1()  throws Throwable  {
    double double0 = Main.pow((-1491.668199236283), 5);
    assertEquals((-7.3851812036287 71E15), double0, 0.01);
}
```

Examples and applications of EvoSuite

But weird tests are also generated...

```
@Test(timeout = 4000)
public void test3()  throws Throwable  {
    Main.pow(1.0, 5133);
    // Undeclared exception!
    Main.pow(0.0, 5133);
}
```

Examples and applications of EvoSuite

Detect deviations from expected behaviour

```
@Test(timeout = 4000)
public void test2()  throws Throwable  {
    double double0 = Main.pow(10.0, (-2237));
    assertEquals(1.0, double0, 0.01);
}
```

Conclusion

- **Relevancy to devops**

  Testing is integral to DevOps, but writing tests is hard.

- **Auto generated tests**

  May be used as a complement.

Unit testing is not **only** writing code