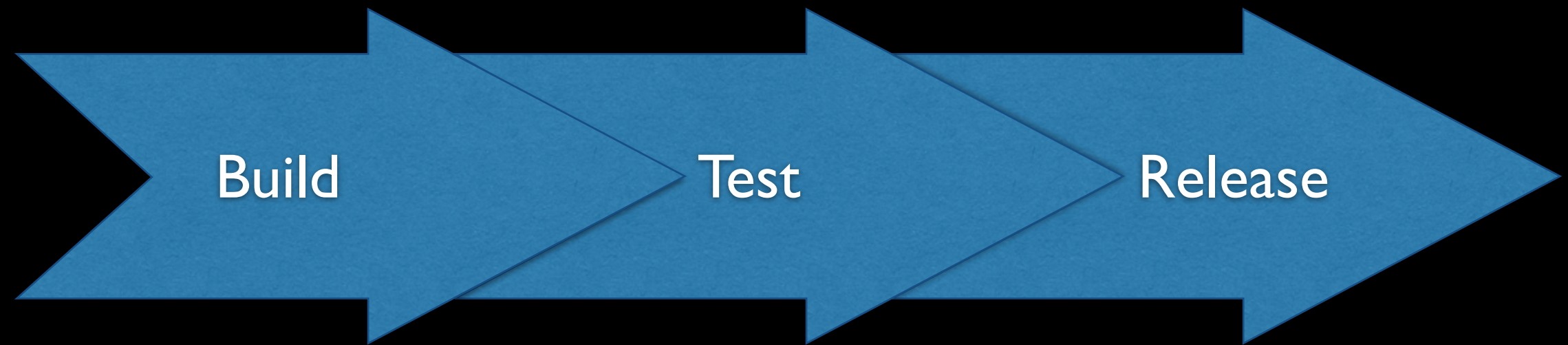


Comparing spectrum based fault localisation against test-to-code traceability links

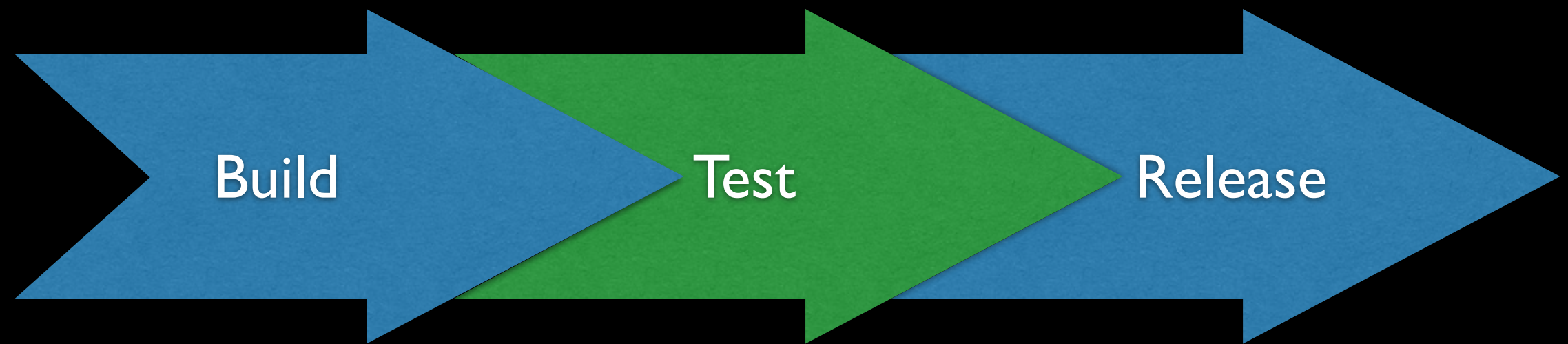
Gulsher Laghari, Kamran Dahri, and Serge Demeyer

FIT2018, Islamabad Pakistan - December 18, 2018

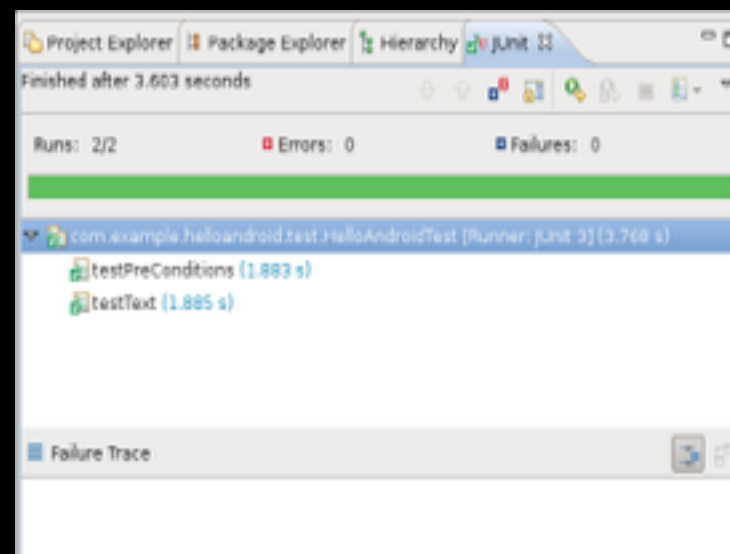
Development pipeline



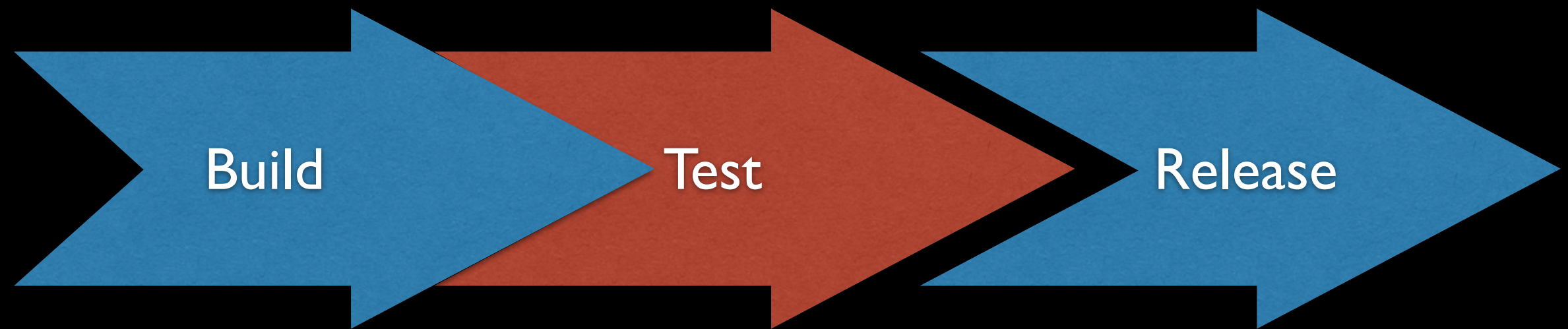
Development pipeline



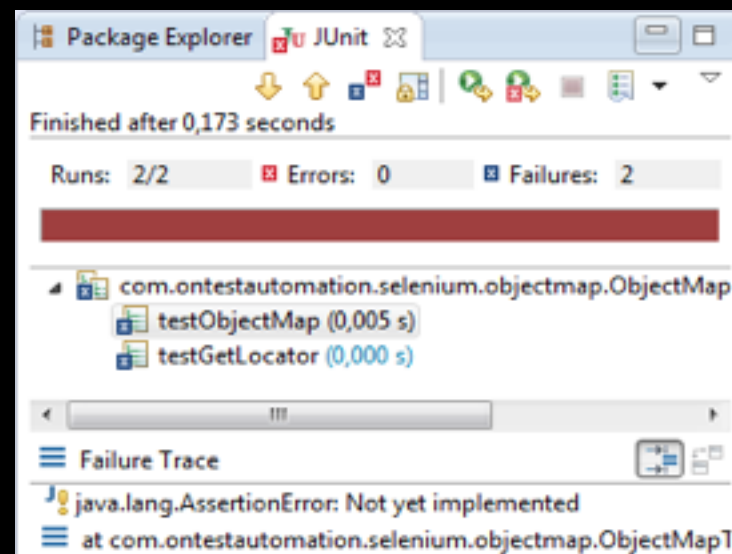
All tests pass



Development pipeline



Some tests fail



Fault Localisation



Fault Localisation an important step in debugging process

Fault Localisation

Production code

```
public class Math {  
    public static int square(int number) {  
        ....  
    }  
    ....  
    ....  
}
```



Test code

```
public class MathTest {  
    public void testSquare() {  
        int twoSquared = Math.square(2);  
        assertEquals(twoSquared, 4);  
    }  
    ....  
    ....  
}
```

Fault Localisation

Production code

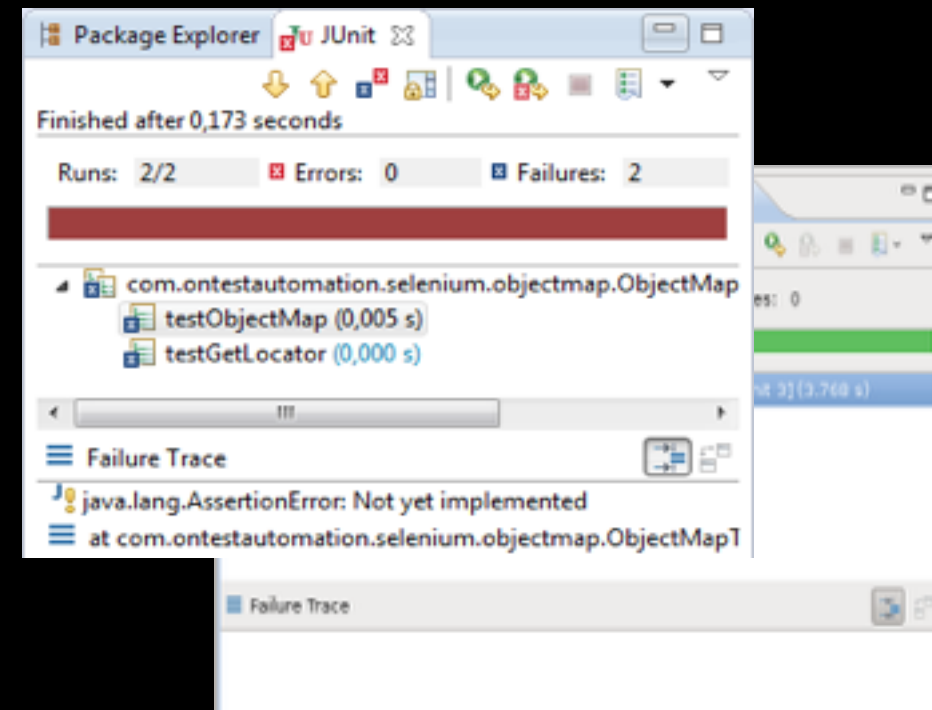
```
public class Math {  
    public static int square(int number) {  
        ....  
    }  
    ....  
    ....  
}
```

Test code

```
public class MathTest {  
    public void testSquare() {  
        int twoSquared = Math.square(2);  
        assertEquals(twoSquared, 4);  
    }  
    ....  
    ....  
}
```



Failing tests reveal the presence of **fault** in the code



Fault Localisation

Production code

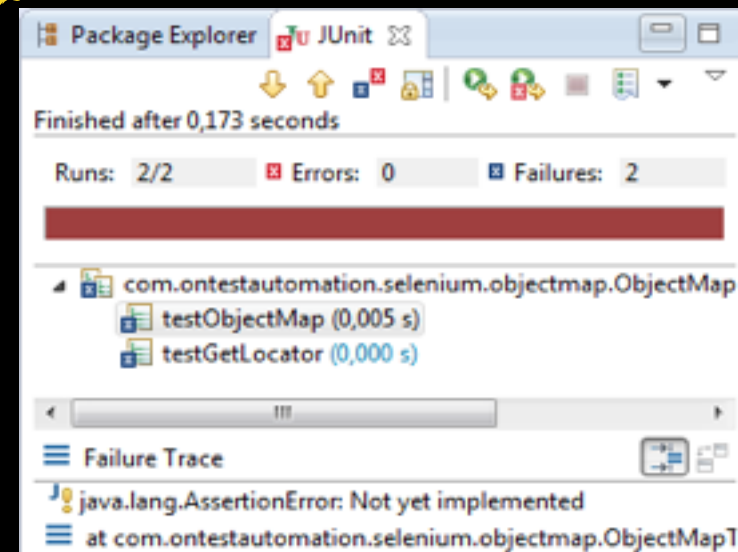
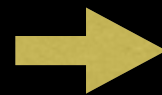
```
public class Math {  
    public static int square(int number) {  
        ....  
    }  
    ....  
    ....  
}
```



Failing tests reveal the

Spectrum based fault localisation

```
MathTest {  
    public void testSquare() {  
        int twoSquared = Math.square(2);  
        assertEquals(twoSquared, 4);  
    }  
    ....  
    ....  
}
```



Comparing **spectrum based fault localisation** against test-to-code traceability links

Gulsher Laghari, Kamran Dahri, and Serge Demeyer

FIT2018, Islamabad Pakistan - December 18, 2018

Spectrum Based Fault Localisation

Production code

```
public class Math {  
    public static int square(int number) {  
        ....  
    }  
    ....  
    ....  
}
```

Test code

```
public class MathTest {  
    public void testSquare() {  
        int twoSquared = Math.square(2);  
        assertEquals(twoSquared, 4);  
    }  
    ....  
    ....  
}
```

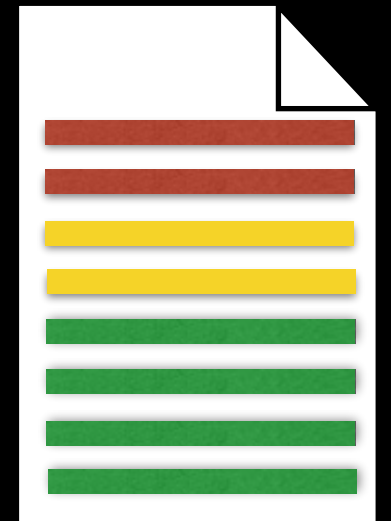
Coverage from passing tests

Coverage from failing tests

Input

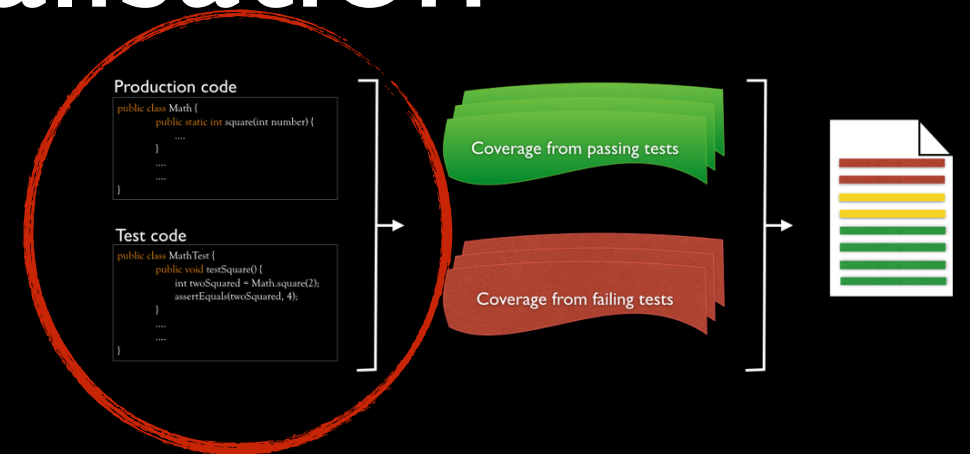
Analysis

Output



Spectrum Based Fault Localisation

Granularity



Statement

XS

Block

S

Method

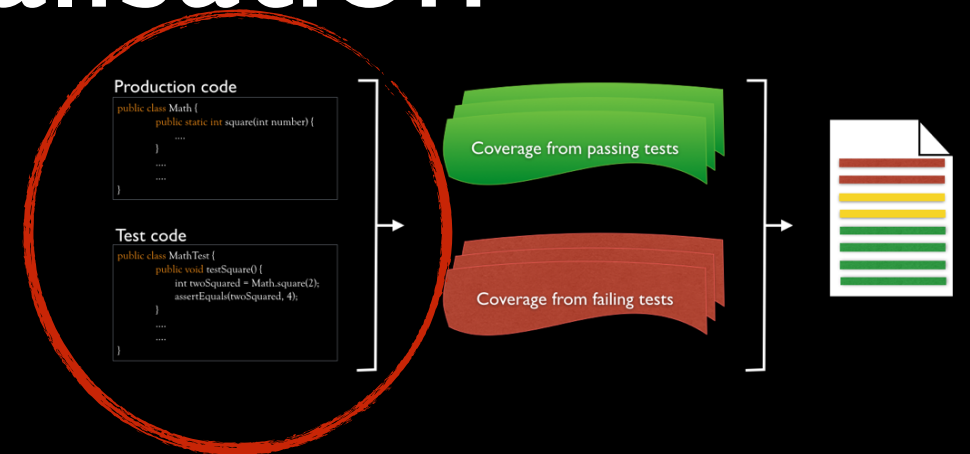
M

Class

L

Spectrum Based Fault Localisation

Granularity



Statement

XS

Block

S

Method

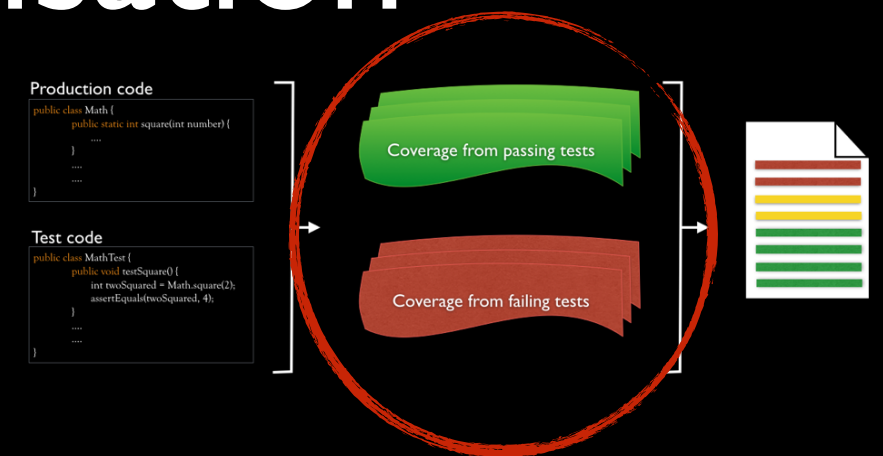
M

Class

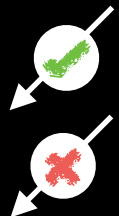
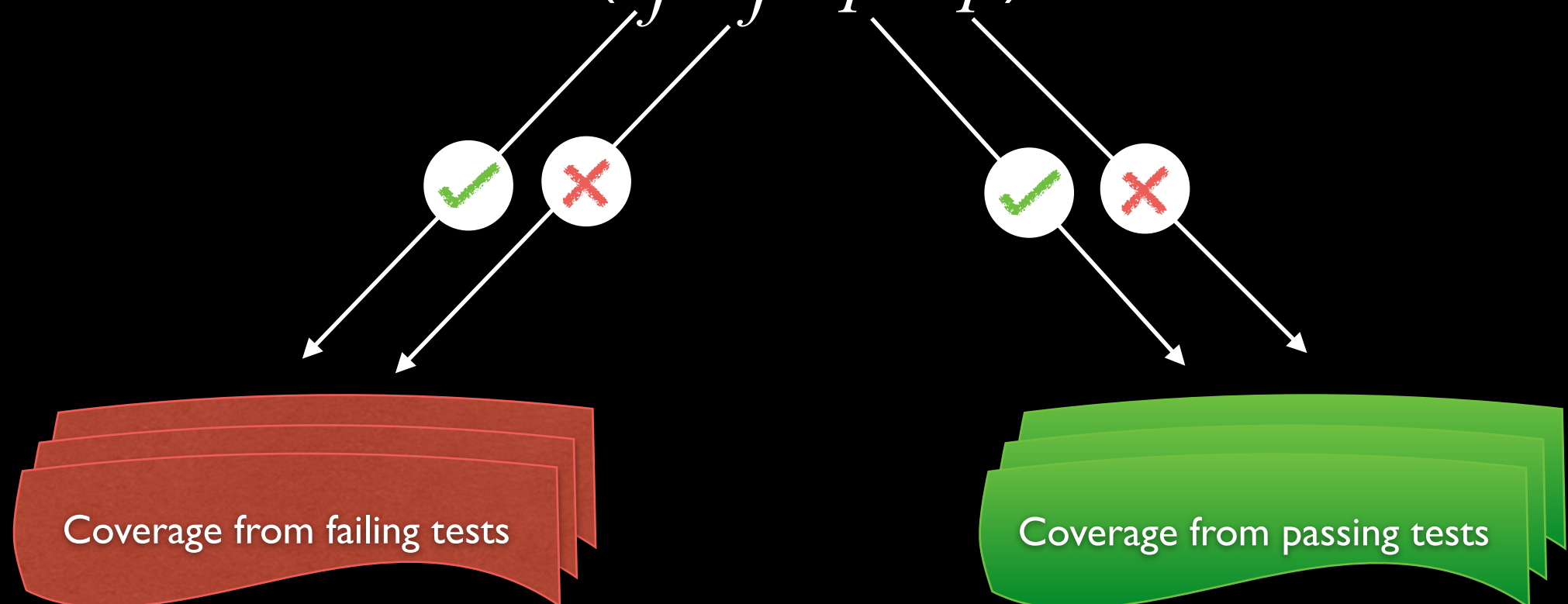
L

Spectrum Based Fault Localisation

Hit Spectrum



$$\text{Method} = (e_f, n_f, e_p, n_p)$$

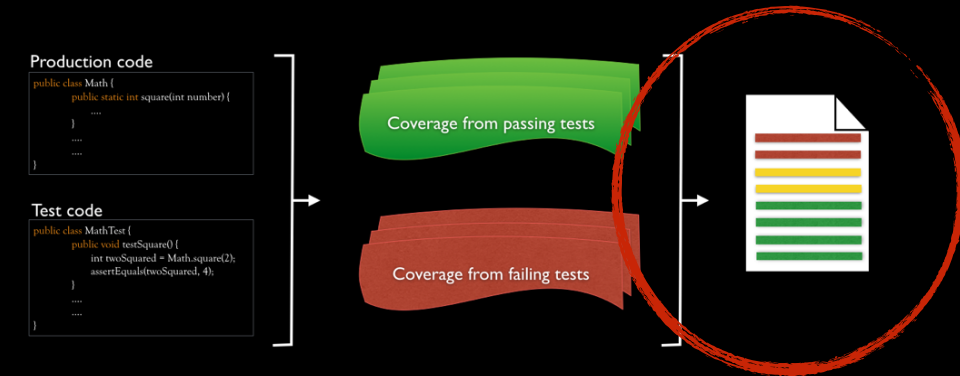


Number of tests that **execute** the method

Number of tests that **do not execute** the method

Spectrum Based Fault Localisation

Fault Locator

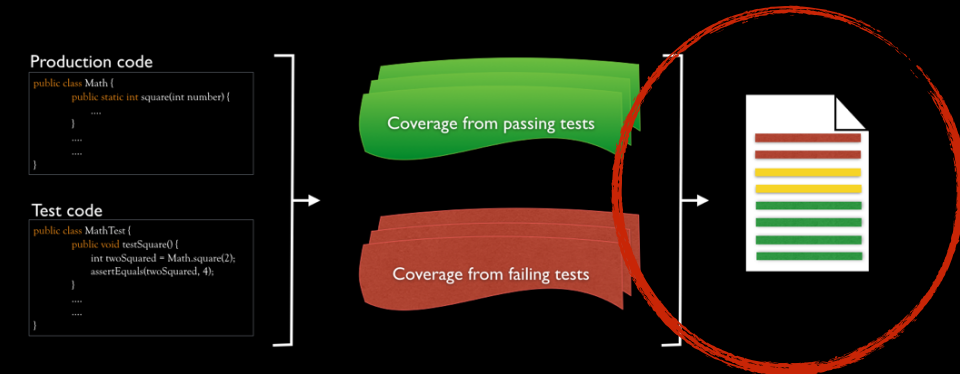


Method = (e_f, n_f, e_p, n_p)

Method = Suspiciousness $[0, 1]$

Spectrum Based Fault Localisation

Fault Locator



Not suspicious at all

0

Highly suspicious

1



Passing test cases

Failing test cases

Spectrum Based Fault Localisation

Evaluation



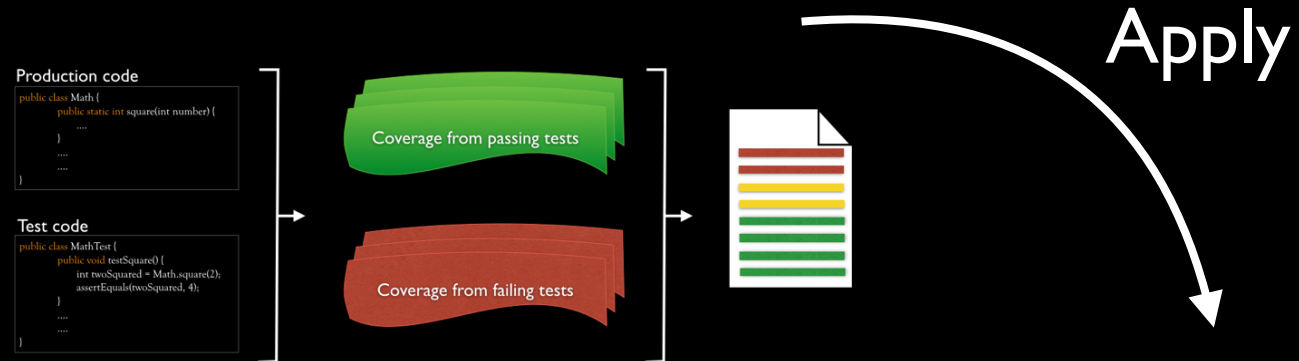
Dataset

Each case in the dataset contains:

- Program code with a **fault**
- Set of **tests that pass** despite the **fault**
- At least one **test that fails** to expose the **fault**
- A **patch** to the code that **fixes** the **fault** and makes the failing tests pass

Spectrum Based Fault Localisation

Evaluation



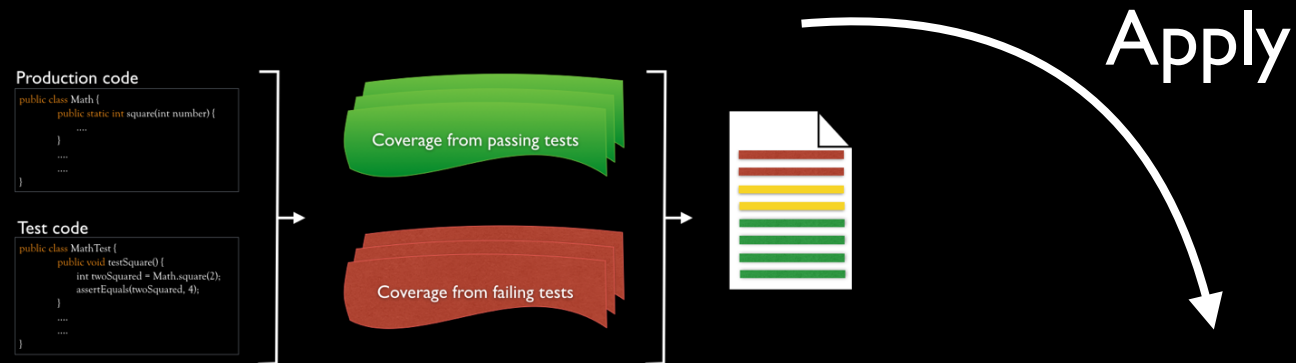
Technique



Dataset

Spectrum Based Fault Localisation

Evaluation



Technique



Dataset

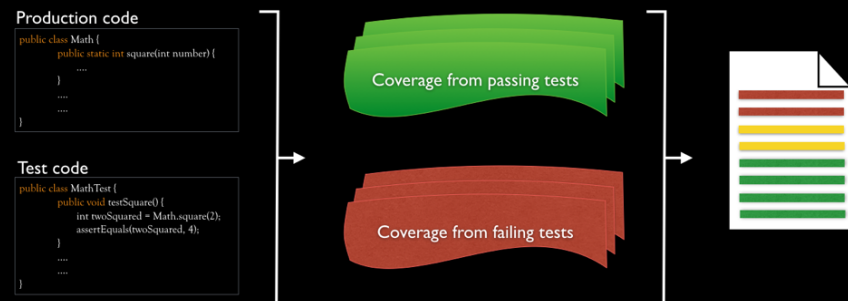
Calculate



Metrics

Spectrum Based Fault Localisation

Evaluation



Technique

How many faults treated successfully

Dataset

Metrics

Comparing spectrum based fault localisation against **test-to-code traceability** links

Gulsher Laghari, Kamran Dahri, and Serge Demeyer

FIT2018, Islamabad Pakistan - December 18, 2018

Test-to-code traceability



I already know where to look for the fault
when the test fails

Developer

Test-to-code traceability



I already know where to look for the fault when the test fails

Developer

Production code

```
public class Math {  
    public static int square(int number) {  
        ....  
    }  
    ....  
    ....  
}
```

Test code

```
public class MathTest {  
    public void testSquare() {  
        int twoSquared = Math.square(2);  
        assertEquals(twoSquared, 4);  
    }  
    ....  
    ....  
}
```



Test-to-code traceability

Test naming conventions

Class Math  MathTest

Method square  testSquare

Production code

```
public class Math {  
    public static int square(int number) {  
        ....  
    }  
    ....  
    ....  
}
```

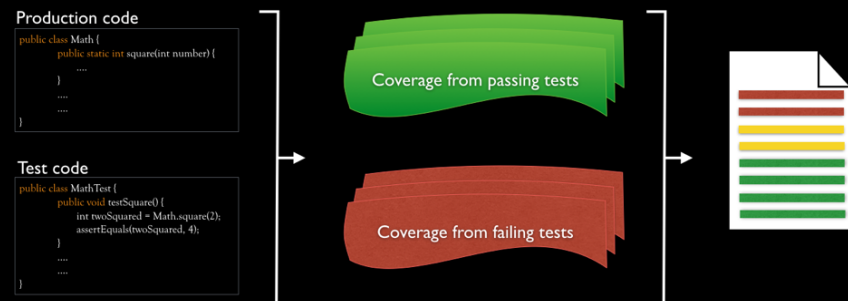
Test code

```
public class MathTest {  
    public void testSquare() {  
        int twoSquared = Math.square(2);  
        assertEquals(twoSquared, 4);  
    }  
    ....  
    ....  
}
```



Spectrum Based Fault Localisation

Evaluation



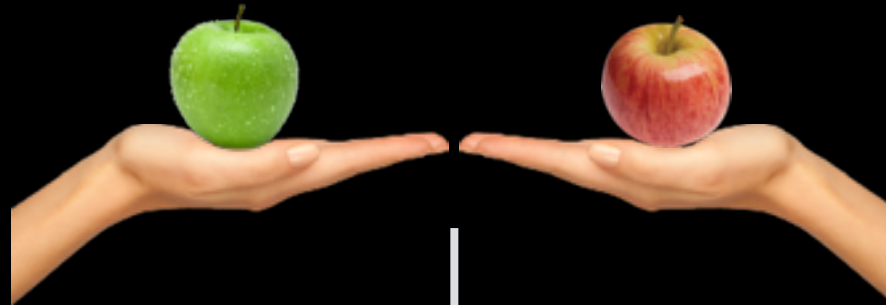
Technique

How many faults treated successfully

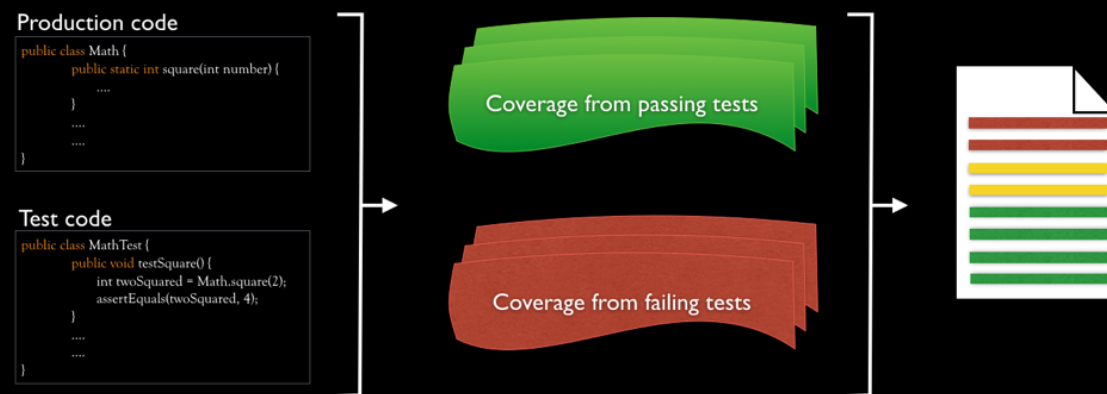
Dataset

Metrics

Case Study



Spectrum Based Fault Localisation



Test-to-code traceability

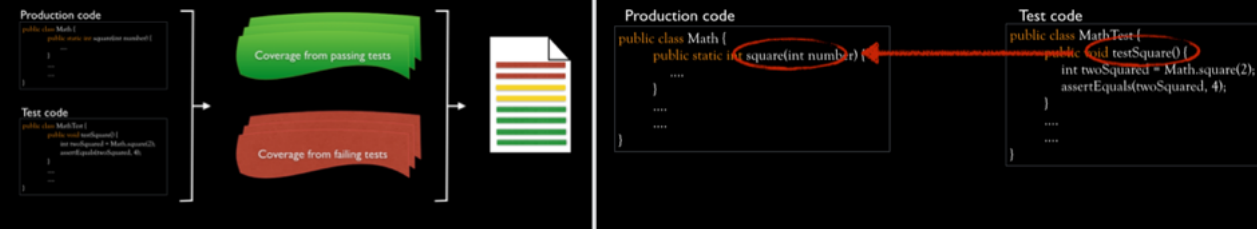


Case Study

Dataset and Metrics

Spectrum Based Fault Localisation

Test-to-code traceability



178 defects
from 3 projects

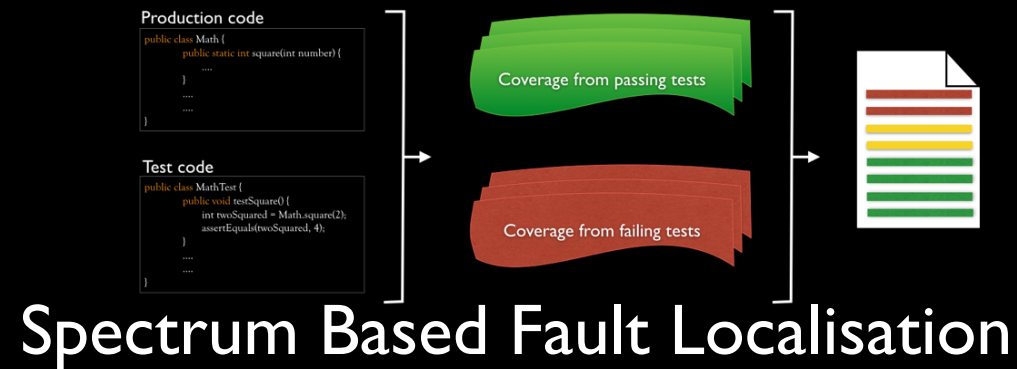
acc@n - Higher the better

mean average precision - Higher the better

mean wasted effort - Lower the better

Case Study

Protocol



Dataset

Calculate

Metrics

Test-to-code traceability

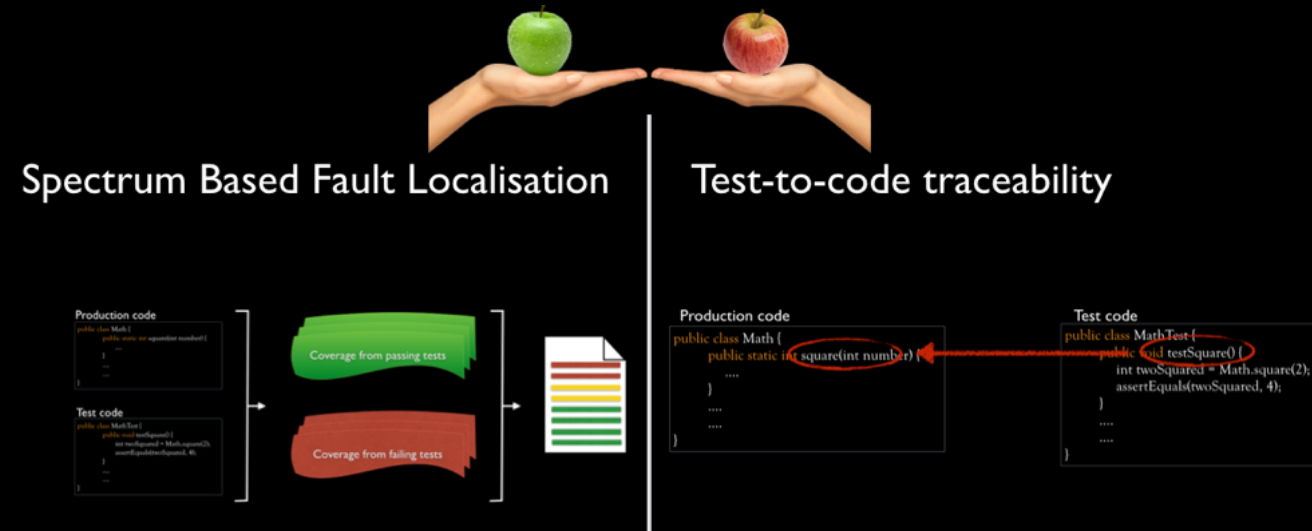
Spectrum Based Fault Localisation

Test-to-code traceability



Case Study

Research Questions



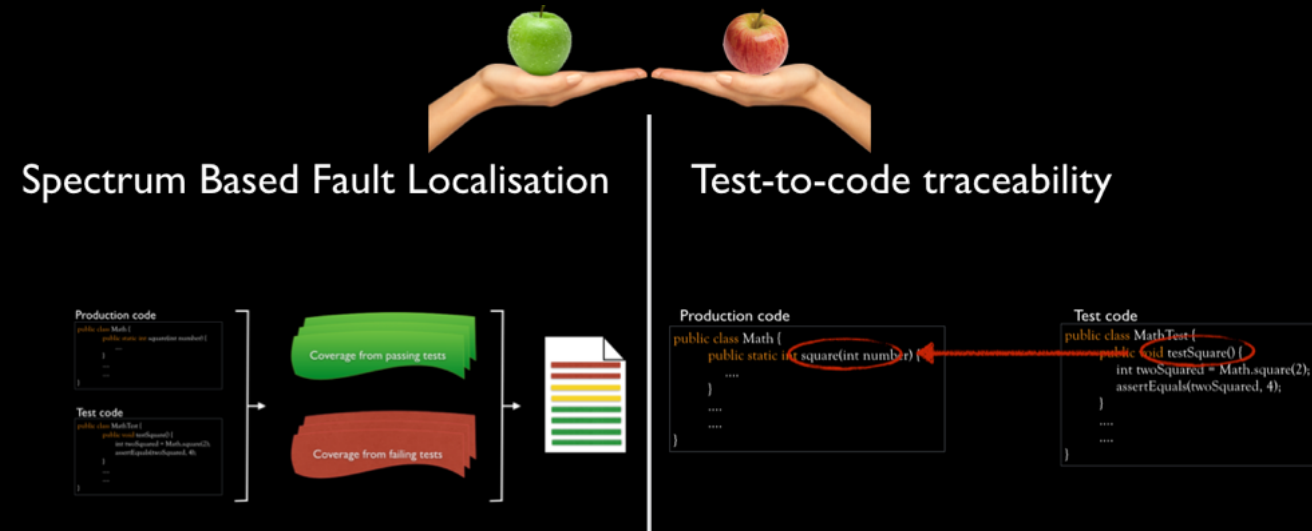
RQ1. **Relative Performance.** Is spectrum based fault localisation relatively better than *test-to-code traceability*?

RQ2. **Worst Case Performance.** Can spectrum based fault localisation compensate in the worst case scenarios for *test-to-code traceability*?

Results

RQ1. Relative Performance.

Is spectrum based fault localisation relatively better than *test-to-code traceability*?



Project	Approach	acc@1	acc@3	acc@5	MAP	MWE
Math	TCT	21	36	46	0.6006	2.08
	SFL	20	34	31	0.4817	7.83
Lang	TCT	23	30	31	0.8417	0.92
	SFL	20	28	29	0.7123	2.03
Chart	TCT	6	10	13	0.6330	1.93
	SFL	8	10	10	0.6476	4.25

MAP = Mean Average Precision
MWE = Mean Wasted Effort

TCT = Test-to-code Traceability
SFL = Spectrum Based Fault Localisation

Results

RQ1. Relative Performance.

Is spectrum based fault localisation relatively better than *test-to-code traceability*?

Spectrum Based Fault Localisation

Test-to-code traceability



Math	TCT	SFL
Max	7	87.5
3rd Quartile	3.5	7
Median	0.5	0.5
Min	0.5	0.5

Lang	TCT	SFL
Max	3.5	18.5
3rd Quartile	1.25	1.50
Median	0.5	0.5
Min	0.5	0.5

Chart	TCT	SFL
Max	5.5	30.5
3rd Quartile	3.25	3.88
Median	1.5	0.75
Min	0.5	0.5

Distribution of wasted effort

TCT = Test-to-code Traceability

SFL = Spectrum Based Fault Localisation

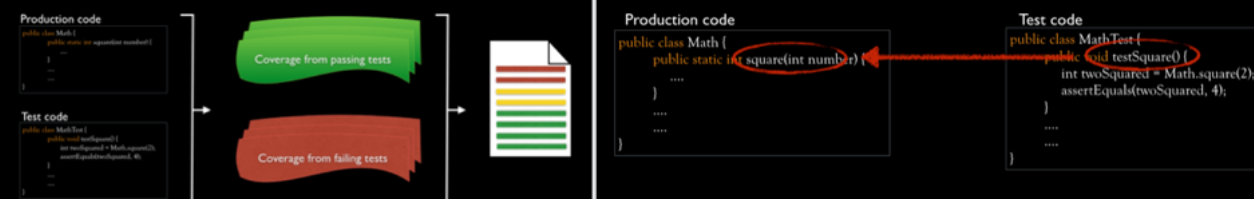
Results

RQ1. **Relative Performance.**

Is spectrum based fault localisation relatively better than *test-to-code traceability*?

Spectrum Based Fault Localisation

Test-to-code traceability



Math

TCT SFL

Law

SFL does not perform any better than simple test-to-code traceability

Mean 0.5 0.5

Min 0.5 0.5

Median 0.5 0.5

Min 0.5 0.5

3rd Quartile 3.25 3.88

Median 1.5 0.75

Min 0.5 0.5

Distribution of wasted effort

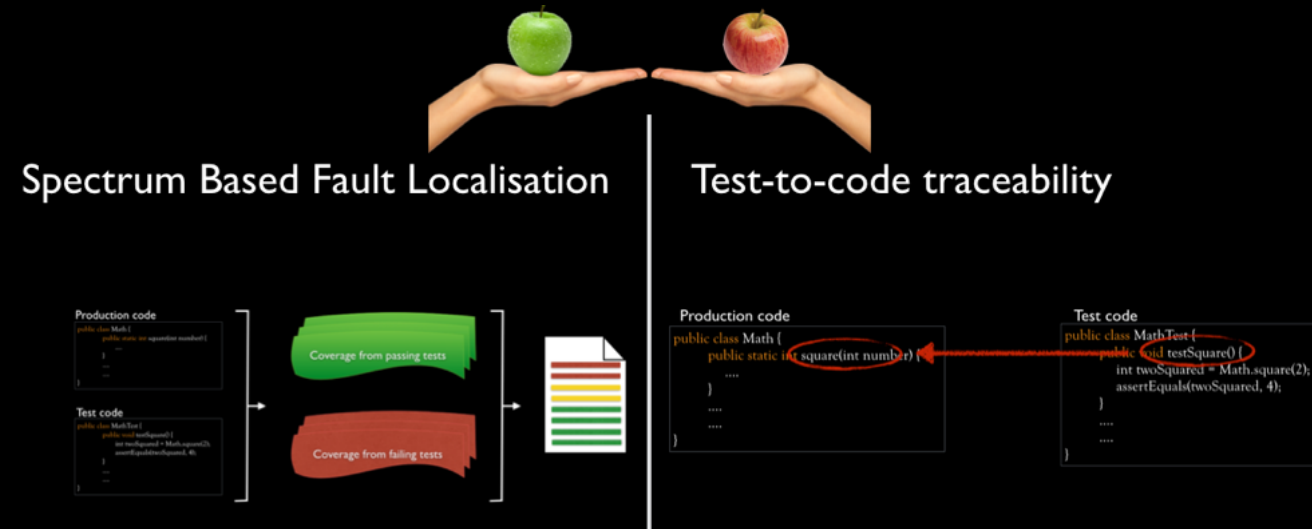
TCT = Test-to-code Traceability

SFL = Spectrum Based Fault Localisation

Results

RQ2. Worst case Performance.

Can spectrum based fault localisation compensate in the worst case scenarios for *test-to-code traceability*?



Defect ID	acc@1	acc@3	acc@5	Average Precision	Wasted Effort
2	0	0	0	0.0326	43
4	0	0	0	0.0208	46
5	0	1	1	0.5000	1.5
7	0	0	0	0.0345	17.5
13	0	0	0	0.0526	10
14	0	1	1	0.2583	6.5
15	0	0	0	0.0194	95
25	0	0	0	0.0020	600
26	0	0	0	0.0161	81

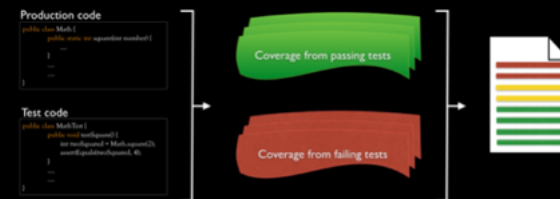
Performance of SFL for Chart project where test-to-code traceability fails

Results

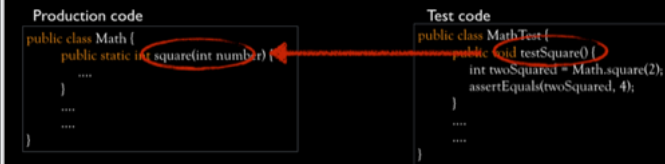
RQ2. Worst case Performance.

Can spectrum based fault localisation compensate in the worst case scenarios for *test-to-code traceability*?

Spectrum Based Fault Localisation



Test-to-code traceability



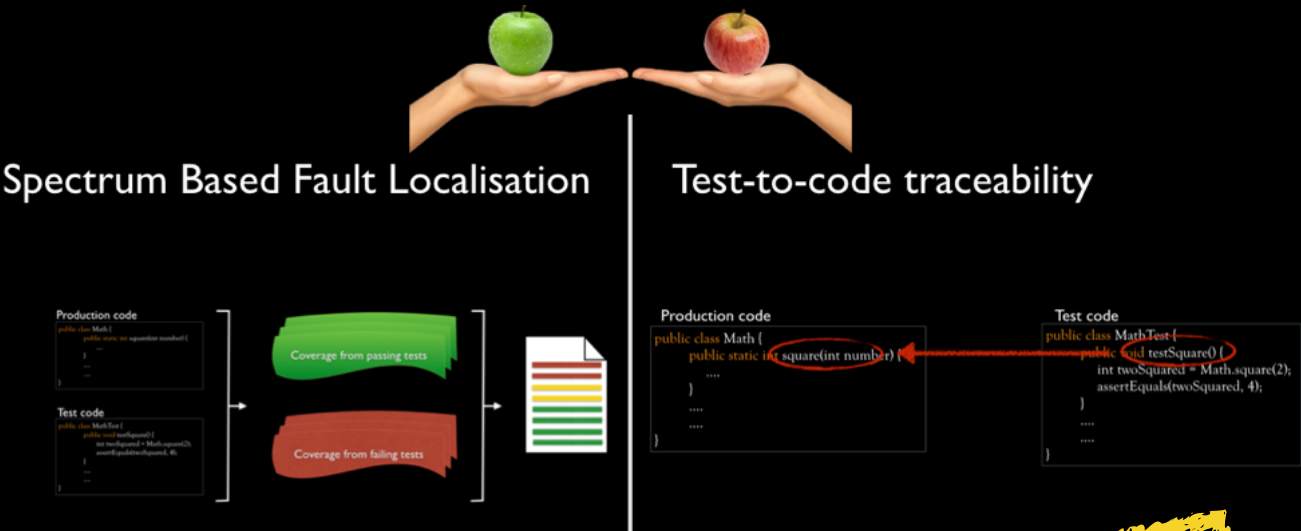
Project	# defects without traceability	# successfully treated
Math	46	11
Lang	27	18
Chart	9	2
Total	82	31

Overall performance of SFL where test-to-code traceability fails

Results

RQ2. Worst case Performance.

Can spectrum based fault localisation compensate in the worst case scenarios for *test-to-code traceability*?



Project

defects without

SFL does not perform well where simple test-to-code traceability also fails

Chart

Total

27

9

82

11

18

2

31

Overall performance of SFL where test-to-code traceability fails

Takeaway

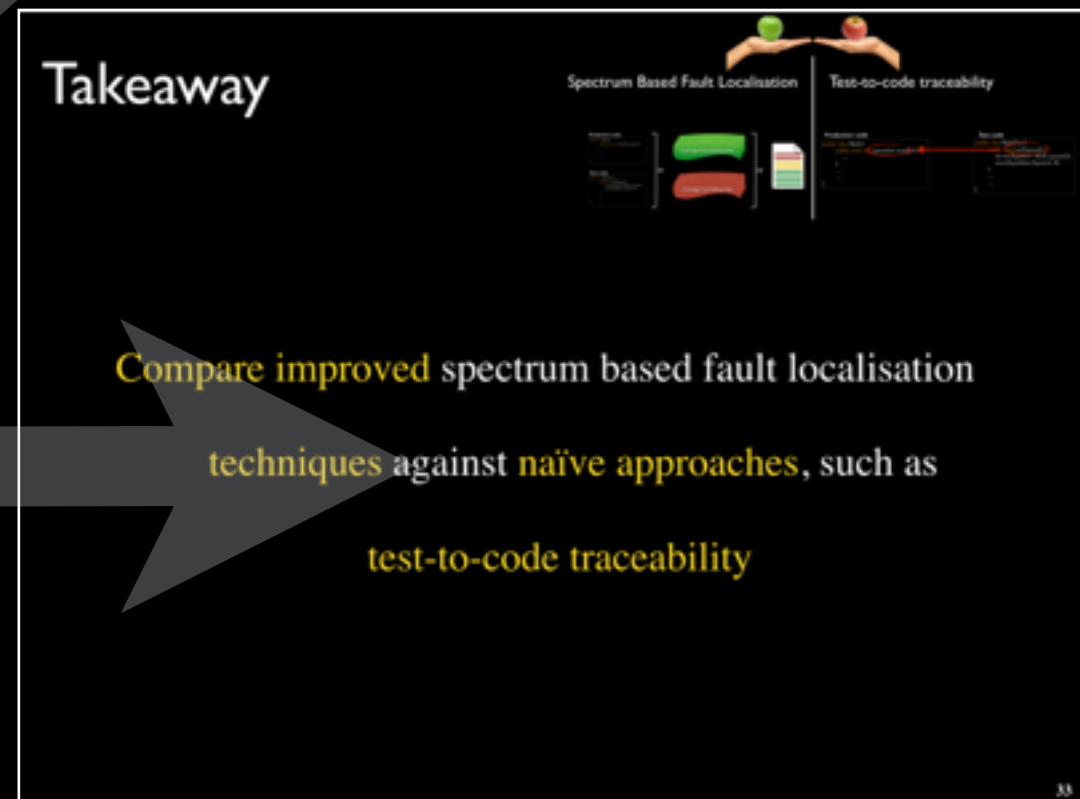
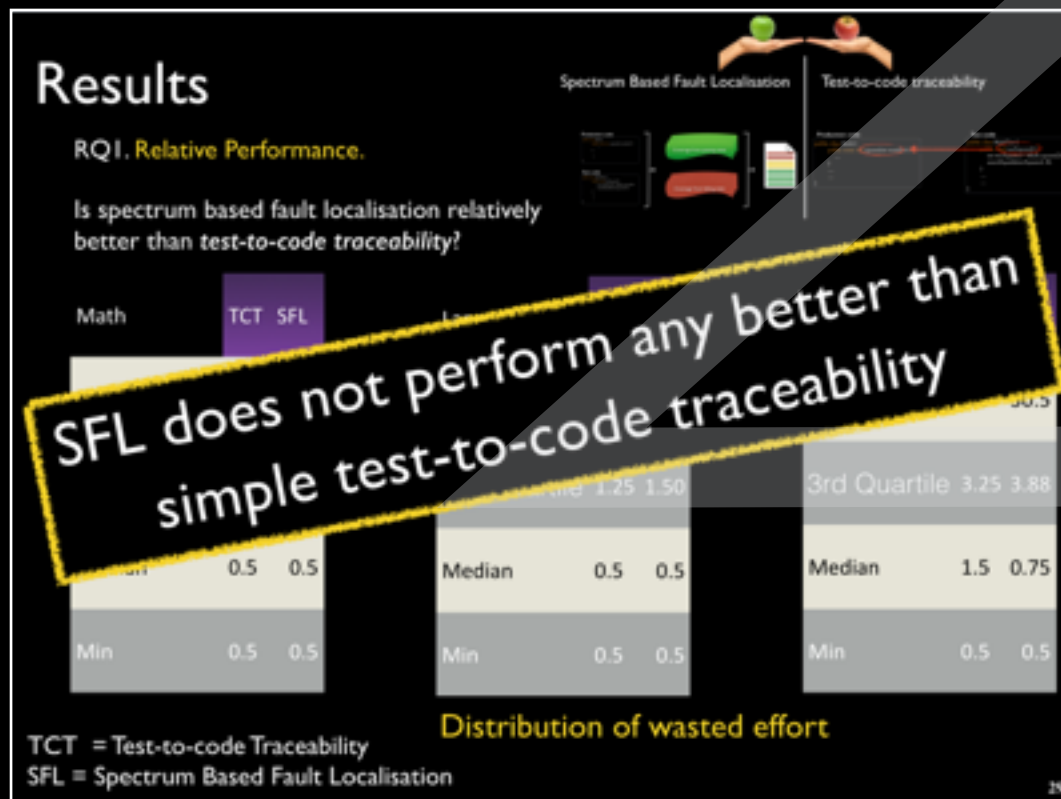
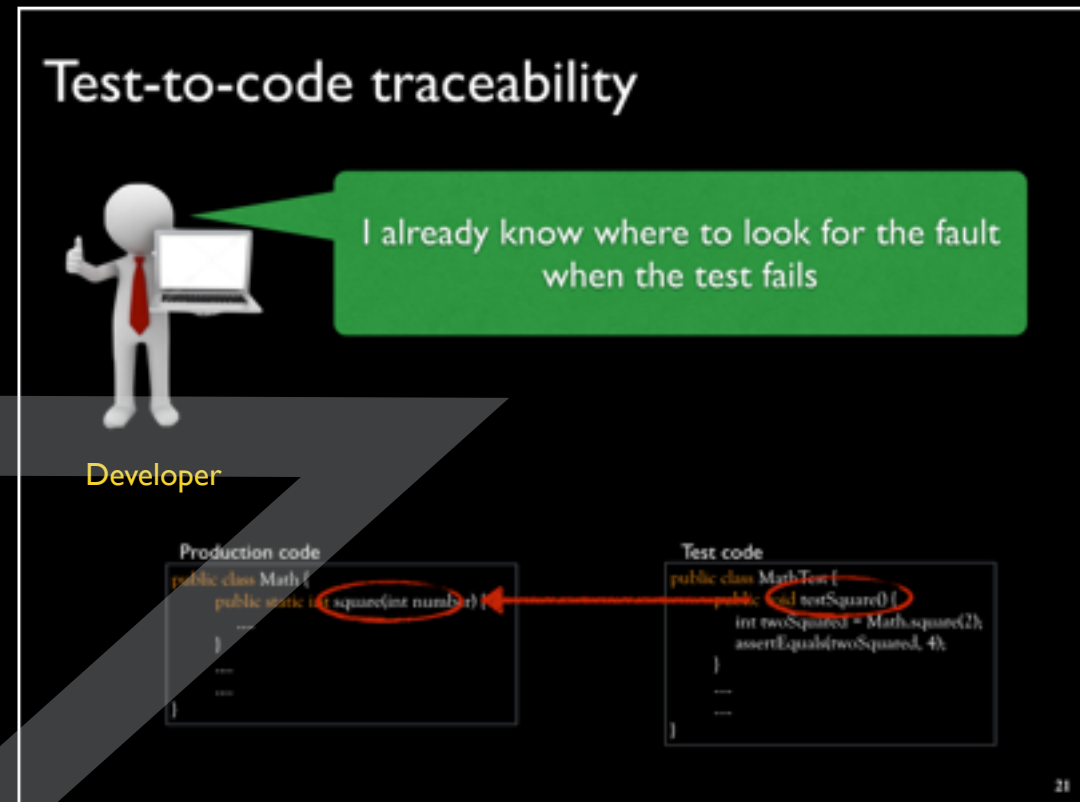
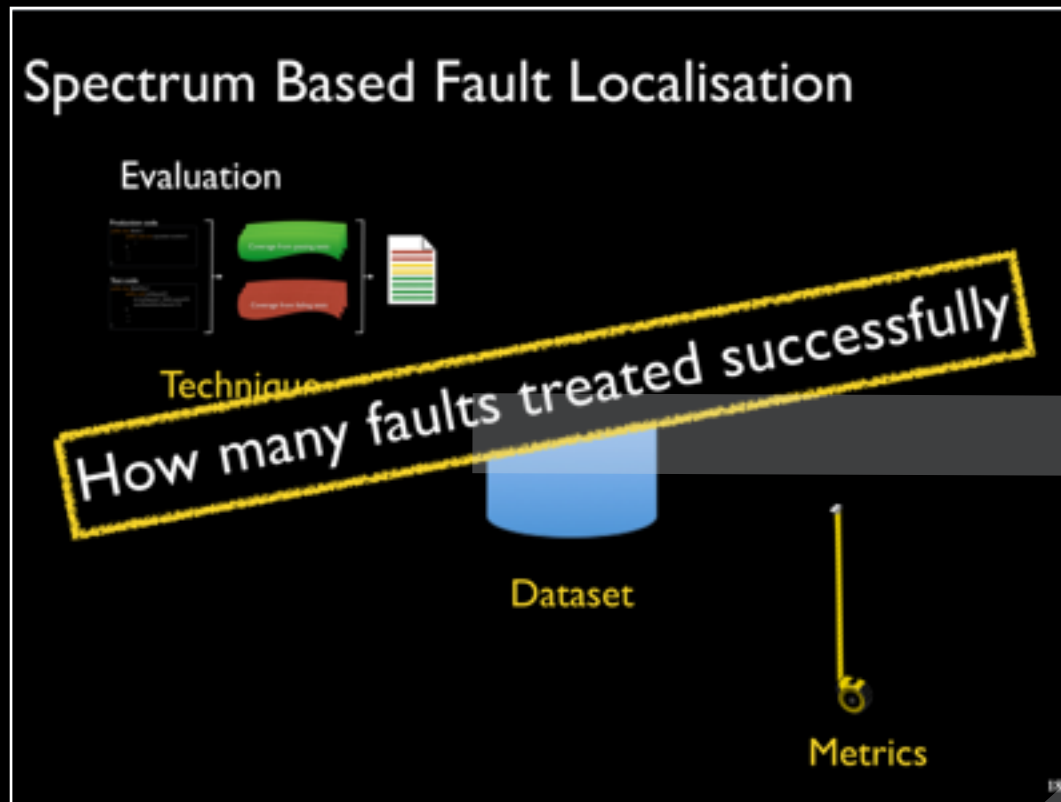
Spectrum Based Fault Localisation

Test-to-code traceability



Compare improved spectrum based fault localisation techniques against naïve approaches, such as test-to-code traceability

Summary



@gul_laghari



<http://glaghari.github.io/>