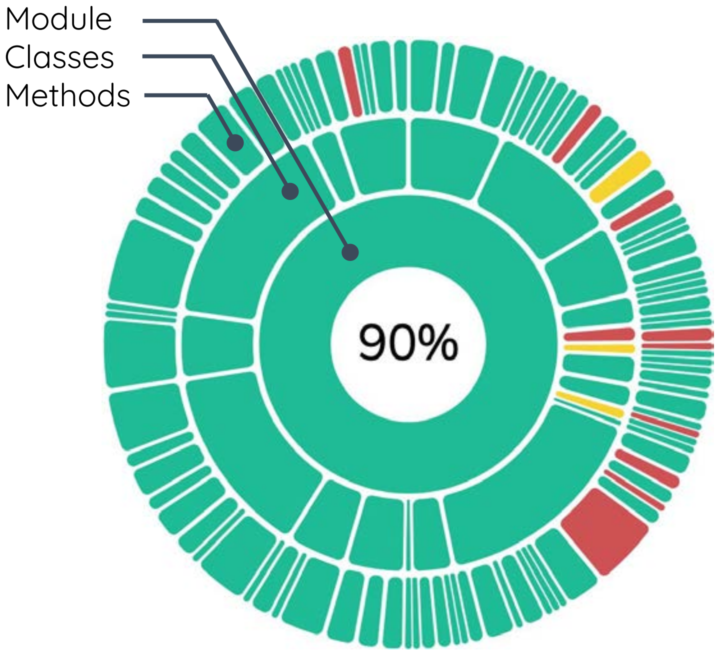
The coverage paradox

When 90% isn’t enough, but less might be

diffblue
AI for Code

Initial coverage analysis



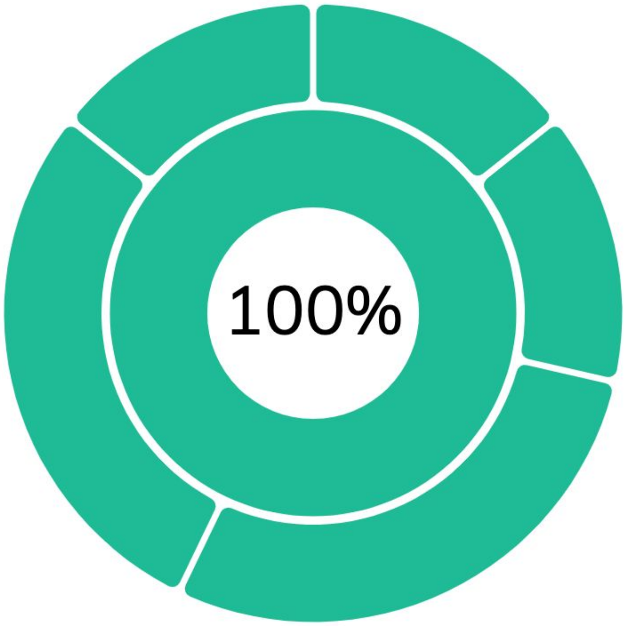
- 🟥 coverage 0-40%
- 🟨 coverage 40-70%
- 🟩 coverage 70-100%

- This project has 90% coverage.
- I don’t know the details of what is tested.
- The risk associated to this project is low.

diffblue
AI for Code

A closer inspection

Testing trivial methods:



●	<init>()	100%
●	getFirstName()	100%
●	setFirstName(String)	100%
●	setLastName(String)	100%
●	getLastName()	100%

diffblue
AI for Code

Incomplete tests:

```
public class Calculator {

    public int divide (int numerator, int denominator) {
        return numerator / denominator;
    }
}
```

This class has 100% coverage.

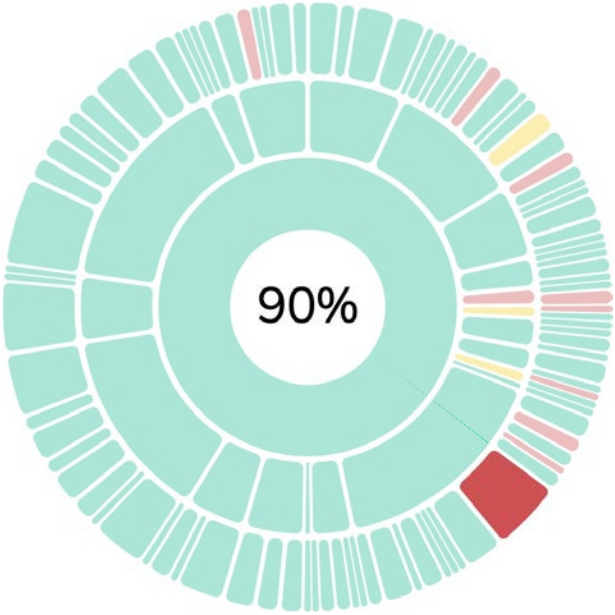
```
public class CalculatorTest {

    @Test
    public void testDivide () {
        assertThat(new Calculator().divide(1,1)).isEqualTo(1));
    }
}
```

The test only checks the behaviour of 1/1.

diffblue
AI for Code

A look at code with no coverage:



- Written in 2010
- Last modified in 2012
- 5% of the total code
- High complexity

What is this piece of code?

- It is the Backup-Restore function…
- The most critical function in your application
- It rarely gets used
- But when it does, it has to work perfectly

diffblue
AI for Code

Get to know your code

diffblue
AI for Code

Metrics to know your code better

In addition to **Coverage**, other metrics can provide more information about your project:

1. Testability
2. Cyclomatic Complexity
3. Dependency analysis
4. Mutation test score

1. Testability

```
public class Counter {
    private int counter = 0;

    public void increment () {
        return counter++;
    }
}
```

```
    public int getCounter() {
```

```
public class CounterTest {
    @Test
    public void testIncrement () {
        Counter counter = new Counter();
        counter.increment();
        // the value of counter cannot be tested
    }
}
```

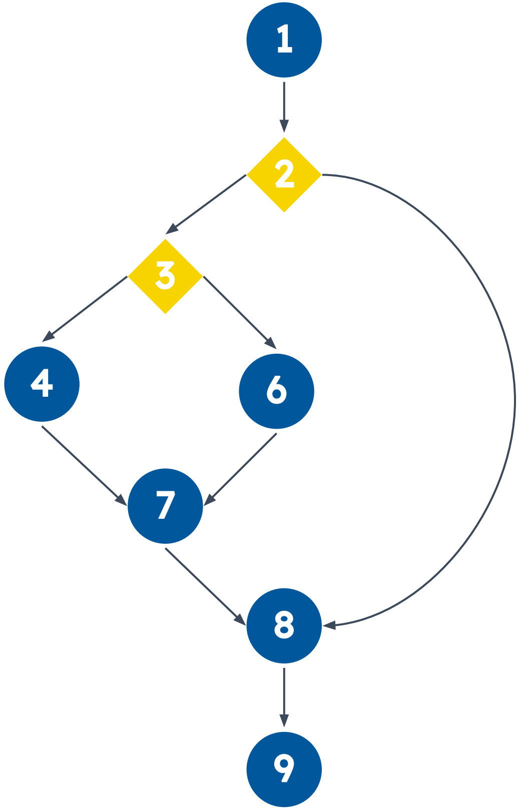
```
        counter.increment();
        assertThat(counter.getCounter()).isEqualTo(1);
    }
}
```

2. Cyclomatic Complexity

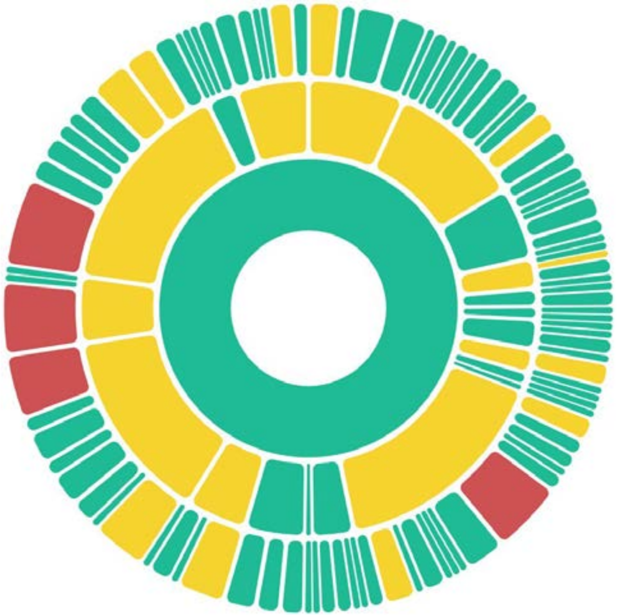
```
1   public void executeTransaction (String accountID, double amount) {
2       if (accountNumberExists(accountID)) {
3           if (amount < 0) {
4               withdraw(accountID, amount);
5           } else {
6               deposit(accountID, amount);
7           }
8       }
9   }
```

Complexity = Edges - Nodes + 2

C = 9 - 8 + 2 = 3



2. Cyclomatic Complexity

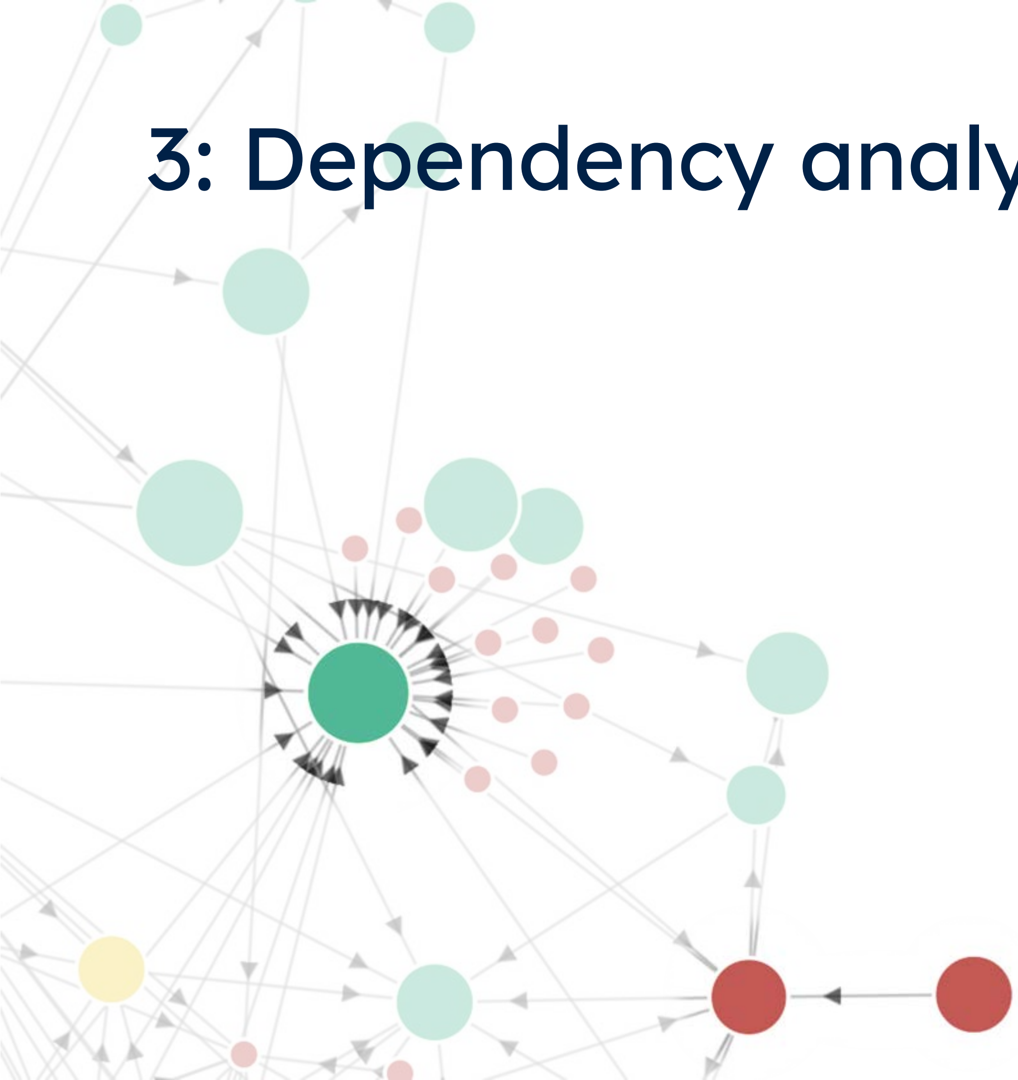


Complexity is tightly linked to risk.

It's particularly important to test classes with high complexity to mitigate risk.

🔴 complexity ≥ 8
🟡 1 < complexity < 8
🟢 complexity = 1

3: Dependency analysis



TRUE: A class used by many others is critical.

FALSE: I don’t need to test classes not used by other classes.

diffblue
AI for Code

4: Mutation test score

Mutation tests can check test quality by verifying their ability of catching regressions.

Mutation tests introduce regression in your codebase to verify that tests fail and don’t return false positives.

```
public class Calculator {

    public int divide (int numerator, int denominator) {
        return numerator * denominator;
    }
}

 }
```

Bonus: Filter out noise

Testing small methods with little or no logic can create unnecessary noise, without adding any value to our project in terms of safety and risk prevention.

Our suggestion is to test these methods indirectly rather than explicitly writing tests for them.

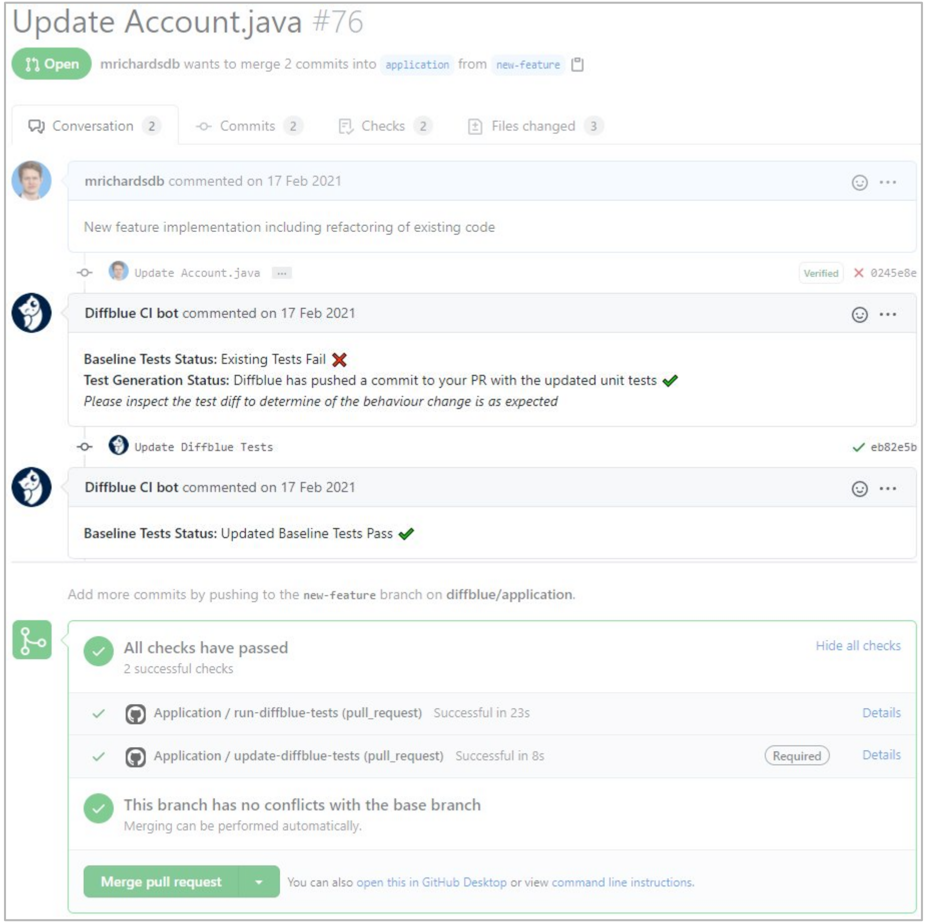
Automating
risk reduction with
Diffblue Cover

diffblue
AI for Code

Diffblue Cover

AI automated:

- unit test authoring for Java
- maintenance of unit tests
- highlighting risk in your code



Update Account.java #76

Open mrichardsdb wants to merge 2 commits into `application` from `new-feature`

mrichardsdb commented on 17 Feb 2021

New feature implementation including refactoring of existing code

`Update Account.java` Verified ✗ 0245e8e

Diffblue CI bot commented on 17 Feb 2021

Baseline Tests Status: Existing Tests Fail ✗
Test Generation Status: Diffblue has pushed a commit to your PR with the updated unit tests ✓
Please inspect the test diff to determine of the behaviour change is as expected

`Update Diffblue Tests` ✓ eb82e5b

Diffblue CI bot commented on 17 Feb 2021

Baseline Tests Status: Updated Baseline Tests Pass ✓

Add more commits by pushing to the `new-feature` branch on **diffblue/application**.

All checks have passed
2 successful checks

✓ Application / run-diffblue-tests (pull_request) Successful in 23s Details

✓ Application / update-diffblue-tests (pull_request) Successful in 8s Required Details

This branch has no conflicts with the base branch
Merging can be performed automatically.

You can also open this in GitHub Desktop or view command line instructions.

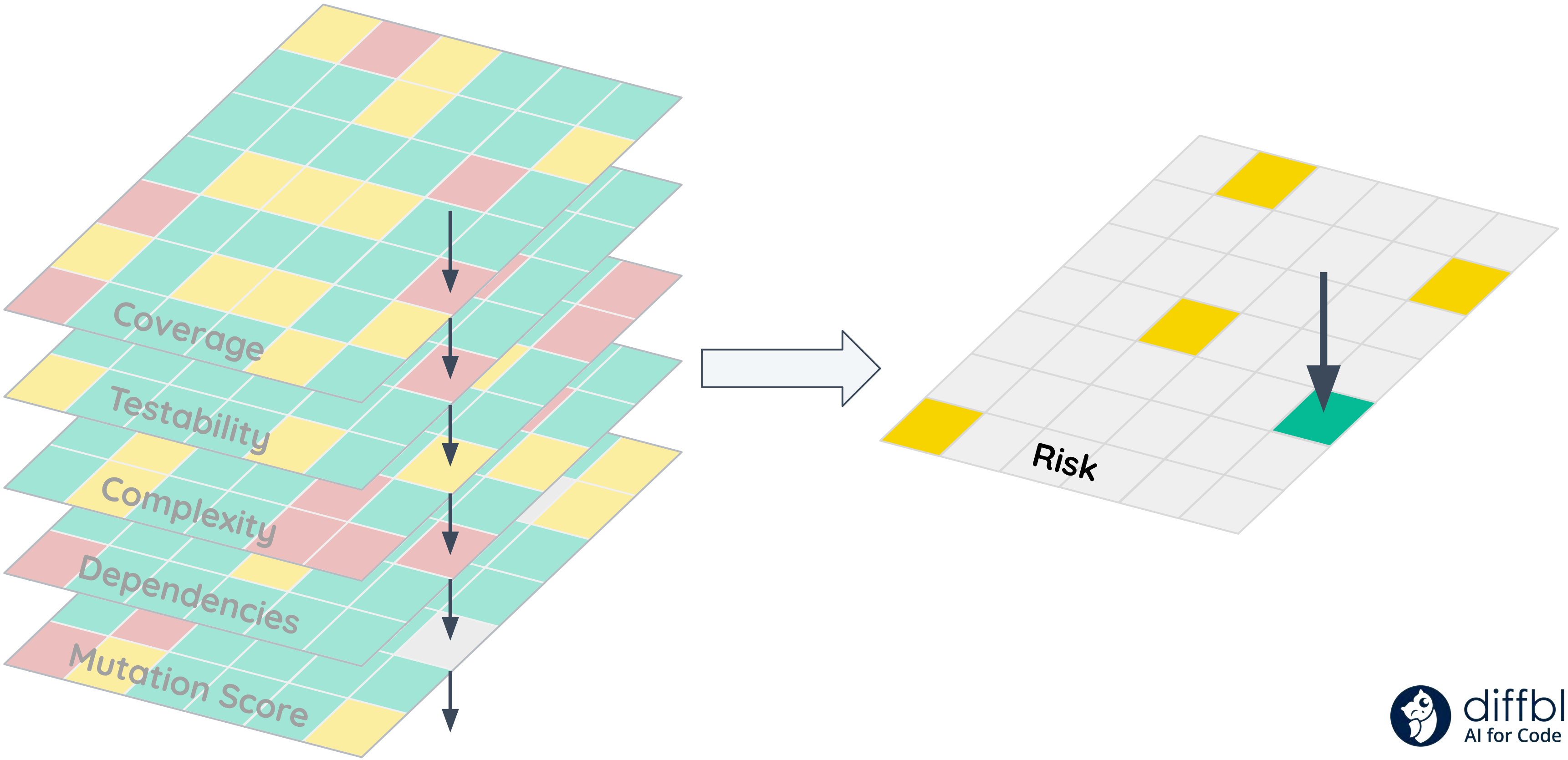
Example test generation

```
@ContextConfiguration(classes = {AmazonS3.class, CloudStorageService.class})
@ExtendWith(SpringExtension.class)
public class CloudStorageServiceDiffblueTest {
  @MockBean
  private AmazonS3 amazonS3;

  @Autowired
  private CloudStorageService cloudStorageService;
  @Test
  public void testUploadFileToBucket() throws SdkClientException {
    // Arrange
    PutObjectResult putObjectResult = new PutObjectResult();
    when(this.amazonS3.putObject(anyString(), anyString(), (File) any())).thenReturn(putObjectResult);

    // Act and Assert
    assertSame(putObjectResult, this.cloudStorageService.uploadFileToBucket(
        "bucket-name", "object-key", Paths.get(System.getProperty("java.io.tmpdir"), "test.txt").toFile()));
    verify(this.amazonS3).putObject(anyString(), anyString(), (File) any());
  }
}
```

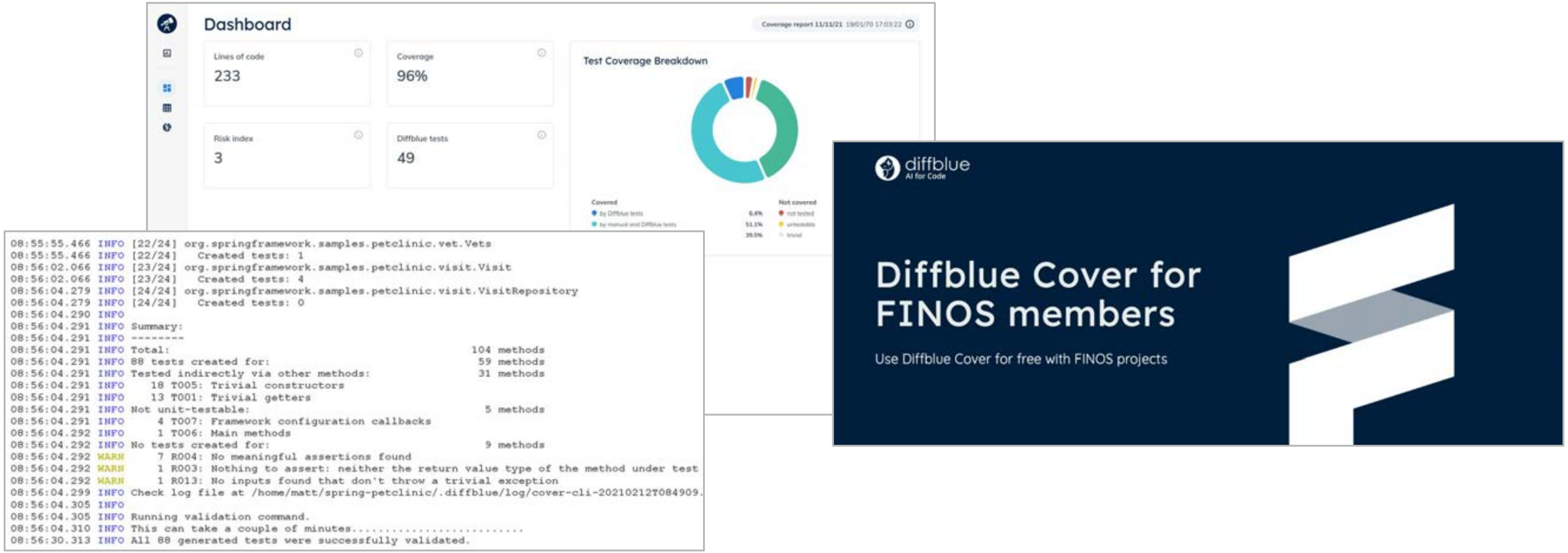
So where is the risk?



diffblue
AI for Code

Cover is free to use in FINOS projects

Get started at diff.blue/FINOS



Dashboard

Coverage report 11/11/21 19/01/70 17:03:22

Lines of code: 233

Coverage: 96%

Risk index: 3

Diffblue tests: 49

Test Coverage Breakdown

```
08:55:55.466 INFO [22/24] org.springframework.samples.petclinic.vet.Vets
08:55:55.466 INFO [22/24]   Created tests: 1
08:56:02.066 INFO [23/24] org.springframework.samples.petclinic.visit.Visit
08:56:02.066 INFO [23/24]   Created tests: 4
08:56:04.279 INFO [24/24] org.springframework.samples.petclinic.visit.VisitRepository
08:56:04.279 INFO [24/24]   Created tests: 0
08:56:04.290 INFO
08:56:04.291 INFO Summary:
08:56:04.291 INFO --------
08:56:04.291 INFO Total:                                          104 methods
08:56:04.291 INFO 88 tests created for:                            59 methods
08:56:04.291 INFO Tested indirectly via other methods:             31 methods
08:56:04.291 INFO    18 T005: Trivial constructors
08:56:04.291 INFO    13 T001: Trivial getters
08:56:04.291 INFO Not unit-testable:                                5 methods
08:56:04.291 INFO     4 T007: Framework configuration callbacks
08:56:04.292 INFO     1 T006: Main methods
08:56:04.292 INFO No tests created for:                             9 methods
08:56:04.292 WARN     7 R004: No meaningful assertions found
08:56:04.292 WARN     1 R003: Nothing to assert: neither the return value type of the method under test
08:56:04.292 WARN     1 R013: No inputs found that don't throw a trivial exception
08:56:04.299 INFO Check log file at /home/matt/spring-petclinic/.diffblue/log/cover-cli-20210212T084909.
08:56:04.305 INFO
08:56:04.305 INFO Running validation command.
08:56:04.310 INFO This can take a couple of minutes..........................
08:56:30.313 INFO All 88 generated tests were successfully validated.
```

diffblue
AI for Code

Diffblue Cover for FINOS members

Use Diffblue Cover for free with FINOS projects

Drop me an email enrico.trentin@diffblue.com

diffblue
AI for Code

diffblue

AI for Code

diff.blue/FINOS