Mutate your code and reveal your true test coverage
About me

@sawiczpawel
http://pawel.sawicz.eu
Agenda

What is mutation testing
Why you should care about this?
Where it's useful
Questions

1. What is my test coverage?
2. Is my unit test valid?
3. Is test coverage good?
4. How do you quantify your tests
Everything comes from nature
What is mutation in biology?

"In biology, a mutation is a permanent change of the nucleotide sequence of the genome of an organism, virus, or extrachromosomal DNA or other genetic elements" - via wikipedia.com
How does it apply to development?

In technology, a mutation is a permanent change of an instruction in a function of a program.
Brief history of mutation testing

1971 - Richard Lipton
1980 - Timothy Budd
Problem with lack of computation power
Poll: What is good value for unit test coverage?
A trivial example

```csharp
public int CalculateCO2Emmision(int engineSize)
{
    if (engineSize < 4)
    {
        return 200;
    }
    else
    {
        return 400;
    }
}

public void CalculateCO2Emmision_EngineSize3_Return200()
{
    //arrange
    var checker = new SimpleIf();

    //act
    var result = checker.CalculateCO2Emmision(3);

    //assert
    Assert.AreEqual(200, result);
}
```
What is mutation testing?

1. competent programmer hypothesis
2. coupling effect hypothesis
Were our tests really fit for purpose?

```csharp
public int CalculateCO2Emmision(int engineSize)
{
    if (engineSize <= 4)
    {
        return 200;
    }
    else
    {
        return 400;
    }
}

public void CalculateCO2Emmision_EngineSize3_Return200()
{
    //arrange
    var checker = new SimpleIf();

    //act
    var result = checker.CalculateCO2Emmision(3);

    //assert
    Assert.AreEqual(200, result);
}
```
[TestMethod]
[TestCase(3, 200)]
[TestCase(4, 400)]
[TestCase(5, 400)]
public void CalculateCO2Emmision_EngineSize3_Return200_BestOne(int engineSize, int expectedC02)
{
    //arrange
    var checker = new SimpleIf();

    //act
    var result = checker.CalculateCO2Emmision(engineSize);

    //assert
    Assert.AreEqual(expectedC02, result);
}
What is mutant?
How mutant looks like?

```java
public int CalculateCO2Emmision(int engineSize)
{
    if (engineSize <= 4)
    {
        return 200;
    }
    else
    {
        return 400;
    }
}
```

Mutants

```java
public int CalculateCO2Emmision(int engineSize)
{
    if (engineSize <= 4)
    {
        return 200;
    }
    else
    {
        return 400;
    }
}
```
Let's produce some mutants

Mutant runner + ILSpy demo
How does it apply to development?

In technology, a mutation is a permanent change of the instruction of the function of a program.
... and why we need to kill it with fire!
Mutation score

mutation score = \frac{\text{number of mutant killed}}{\text{total number of mutant}}
mutants killed: 4/7  Mutation score: 57%

- Payload
  - SimpleInt
    - CalculateCO2Emmission
      - Group: #0
        - ROR#4 - LessThanOrEqual - Live
        - ROR#6 - Equality - Killed by 1 tests
        - ROR#2 - False - Killed by 1 tests
        - ROR#7 - NotEquality - Live
        - ROR#1 - True - Live
        - ROR#5 - GreaterThanOrEqual - Killed by 1 tests
        - ROR#3 - GreaterThan - Killed by 1 tests
  - MutationTalk.Tests
  - MutationTalk.Mutated.Test
Test coverage
New Poll: Are you still happy with your unit test coverage?
New Poll: Are you still happy with your unit test coverage?

Traditional code coverage tools measure test coverage of the code, not business intent.
Mutations operations

1. Arithmetics operators
2. Boolean operators
3. Inserting/Deleting instruction
4. Enforcing nulls
Arithmetics
operators

Mutation runner + ILSpy
Boolean operators

Mutation runner + ILSpy
Visual Mutator
Where it can be useful?

1. Numerical algorithms
2. Calculations
3. Core functionality
4. A lot of Boolean logic
Hi Pawel Sawicz (not you? Log out)

Leave a message (optional)

Your name (optional)

Choose an amount*
USD $100 $50 $30 $20

Other amount
$30

Hide my donation amount from others

Card number

*Indicates a required field

You are donating $30

Review before donating
or
Donate now

Donations through JustGiving are secure.

Click here to review JustGiving's terms of service and privacy policy
Tips

1. Experience to cherry pick right piece of code to mutate
2. Only time gives you knowledge
3. A lot of intuition!
4. What is right for my org is not right for your org
Questions

1. What is my test coverage?
2. Is my unit test valid?
3. Is test coverage good?
4. How do you quantify your tests
Existing tools

.NET:
1. visualmutator
2. NinjaTurtles

Ruby:
1. mutant
Resources

1. https://soundcloud.com/arkency/podcast-3-mutation-testing
3. https://github.com/mbj/mutant
4. follow @_m_b_j_ (Markus Schirp)
Thanks, Questions?

@sawiczpawel, pawel@sawicz.eu