Automatic Detection of Pseudo-Tested Methods in a Test Suite Using Fault Injection

Nicholas Tocci

November 17, 2018

Problem

How can we know if our test suites are adequate?

👍

Coverage

Coverage

%

Def: % of a system that has been tested.

Calculation

$$Coverage = \frac{NumberofTestedMethods}{TotalNumberofMethods}$$

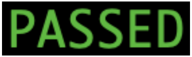
High Coverage

% != 👍

Pseudo-tested Methods

Definition

What is a Pseudo-tested Method?



```
PASSED
```

Def: It will never fail.

Detection

How Can We Detect Pseudo-tested Methods

It is harder than you think!

Example of a Pseudo-tested method

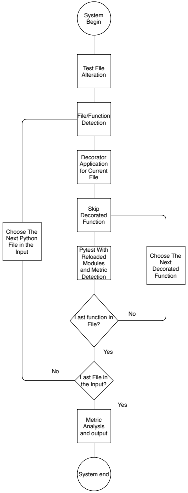
```
numbers.py:
def numberOrder(n):
  numbersSorted = sorted(n)
  return numbersSorted


test_numbers.py:
def test_numbers_ordered():
  sortedNumbers = {1,2,3,4}
  orderedNumbers = numberOrder(numbers)
  assert numbers == sortedNumbers
```

What is Function-Fiasco

A Pseudo-tested method detection tool

Flow to system



System Begin

Test File Alteration

File/Function Detection

Decorator Application for Current File

Skip Decorated Function

Pytest With Reloaded Modules and Metric Detection

Last function in File?

No

Choose The Next Decorated Function

Yes

Last File in the Input?

No

Choose The Next Python File in the Input

Yes

Metric Analysis and output

System end

Output

Task	Begin Date	End Date	Backup Begin Da
Implementation	Now	15 Jan	Now
Testing	15 Jan	1 Feb	1 Feb
Analysis	1 Feb	10 Mar	10 Mar
First draft	10 Mar	25 Mar	25 Mar
Second draft	25 Mar	5 Apr	5 Apr
Third draft	5 Apr	15 Apr	15 Apr
Fourth draft	15 Apr	20 Apr	20 Apr
Fifth draft	20 Apr	25 Apr	Apr 25

Table: Proposed Research Schedule