#### Introduction

#### Introduction

Problem

Pseudo-teste

Function

Evaluation

Doculto

Results

Conclusio

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

Nicholas Tocci

April 3, 2019

#### **Problem**

Introduction

Problem

oudo teste

Methods

Function-

F ....

Strategy

Results

Conclusion

How can we know if our test suites are effective?



#### Coverage

Introduction

Problem Coverage

Calculation
Coverage vs
Adequate

Pseudo-tested

Function Fiasco

Evaluation

Strategy

Conclusio

## Coverage

%

**Def**: % of a system that has been tested.

#### Calculation

Introduction

Problem

Coverage Calculation

Pseudo-tested

Function

Evaluation

Doculto

Conclusio

 $\textit{FunctionCoverage} = \frac{\textit{NumberofTestedMethods}}{\textit{TotalNumberofMethods}}$ 

#### High Coverage

Introduction

Problem

overage

Coverage vs Adequate Coverage

Pseudo-tested

Function Fiasco

Evaluatio

Doculto

Conclusio

% := 16

#### Pseudo-tested Methods

Introduction

roblem

#### Pseudo-tested Methods

Defintion

F. .....

Fiasco

Evaluation Strategy

Reculte

Conclusion

### Pseudo-tested Methods

#### **Definition**

Introduction

roblem

Pseudo-teste

Defintion

Functio

Evaluation Strategy

Results

Conclusion

What is a Pseudo-tested Method?

**PASSED** 

Def: It will never fail.

#### Detection

Introduction

Problem

Pseudo-teste

Defintion

Detection

Function Fiasco

Evaluation Strategy

Results

onclusion

How Can We Detect Pseudo-tested Methods

It is harder than you think!

#### Example of a Pseudo-tested method

Introduction

Problem

Pseudo-teste

Defintion Detection

Function Fiasco

Evaluation Strategy

Results

```
numbers.py:
def numberOrder(n):
  numbersSorted = sorted(n)
  return numbersSorted
test_numbers.py:
def test_numbers_ordered():
  numbers = set([2,4,3,1])
  sortedNumbers = set([1,2,3,4])
  orderedNumbers = numberOrder(numbers)
  assert numbers == sortedNumbers
```

#### What is Function-Fiasco

Introduction

Problem

Pseudo-teste

Fiasco
What is

Evaluation

\_ .

Conclusion

# A Pseudo-tested method detection tool



#### **Decorator Function**

Introduction

Problem

Pseudo-teste

Function Fiasco

What is Function-Fiasco

Evaluation Strategy

Results

```
def skipper(func):
   functionsComplete = globs.functionsComplete
   checked = checkFunctionsComplete (func, functionsComplete)
   globs.checked = str(checked)
    if checked == True and globs.firstExe == True:
        def wrapper(*args, **kwargs):
            checkType(var, func.__name__)
            return var
        return wrapper
    elif checked == True and globs.firstExe == False:
        def doFunc(*args, **kwargs):
            var = func(*args, **kwargs)
            return checkType(var, func.__name__)
        return doFunc
    else:
        def doFunc(*args, **kwargs):
            var = func(*args, **kwargs)
            return var
        return doFunc
```

#### **Execution Flow**

Introduction

Problem

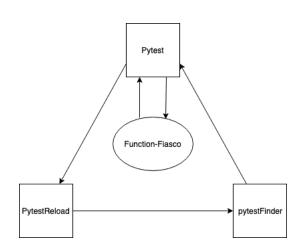
Pseudo-teste

Function

What is Function-Fiasco

Evaluation Strategy

Results



#### Flow to system

Introduction

Problem

Pseudo-teste

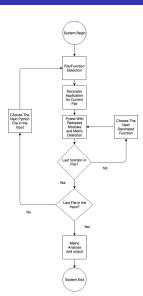
unction

What is

Function-Fias

Evaluatio

Danilla



#### Coverage Calculation

Introduction

Problem

Pseudo-teste

Function

Evaluation

Strategy

Coverage

Calculation

Method Calculatio Metrics

Danula

Conclusior

 $\textit{FunctionCoverage} = \frac{\textit{NumberofTestedMethods}}{\textit{TotalNumberofMethods}}$ 

#### Coverage Example

Introduction

Problem

Pseudo-teste

Function

Evaluatio

Strategy

Coverage Calculation

Truly-Test

Calculation Metrics

Produce

Results

NUMM	NUMTM	Function Coverage
40	25	62.5%

#### Truly-Tested-Method Calculation

Introduction

Problem

Pseudo-teste

Functio Fiasco

Evaluation Strategy Coverage

Calculation Truly-Tested-Method Calculation

Metrics Produced

Results

Conclusion

- Number of Truly-Tested-Methods = NUMTTM
- Number of Tested Methods = NUMTM
- Number of Pseudo-tested Methods = NUMPTM

NUMTTM = NUMTM - NUMPTM

#### Truly-Tested-Method Example

Introduction

Problem

Pseudo-teste

Function

Fiasco

Evaluation

Coverage

Calculation
Truly-Tested-

Method Calculation

Metrics Produced

Results

NUMTM	NUMPTM	NUMTTM	
25	3	22	

#### Adequate-Coverage Calculation

Introduction

Problem

Pseudo-teste

Function

Fiasco

Evaluation Strategy

Coverage Calculatio

Truly-Tested-

Calculation Metrics

Results

$$AC = \frac{Number of Truly Tested Methods}{Total Number of Methods}$$

#### Output

Metrics Produced

	Statement Coverage	Initial Function coverage	Number of Methods	Number of Tested Methods	Fiascoed Methods	Number of Pseudo-tested Methods	Number of Truly Tested Methods	Updated Coverage
1	67%	0.44	730	319	16	9	310	0.42

#### List of Systems

Introduction

Probler

Pseudo-tested Methods

Function-

Evaluation Strategy

Results

Conclusion

	System_Name	Num_Methods	Fiascoed	Num_Tests
1	Hashids-Python	16	10	59
2	Bleach	368	8	312
3	Pycco	22	6	17
4	Howdoi	20	2	18
5	Flashtext	42	7	23
6	Honcho	58	7	124
7	Maya	88	13	277
8	Gator	91	53	505
9	Hatch	134	14	339
10	Nikola	732	16	205

Table: List of systems used for testing.

#### Statement Coverage

Introduction

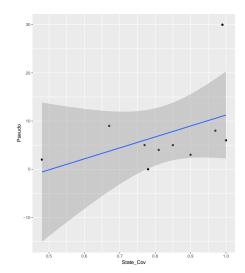
Problem

Pseudo-teste

Function

Evaluation

Results



#### Type Risk

Introduction

Problem

Pseudo-teste

Function

Evaluation Strategy

Results

Conclusion

	Type	Used	Found	ratio
1	Strings	69	33	47.8%
2	Booleans	49	35	73.5%
3	Ints	18	4	22.2%
4	Floats	1	0	0%

Table: Breakdown of the number of pseudo-tested method per type.

#### Field Connects

Introduction

Problem

Pseudo-teste

Functio

Evaluatio

Strategy

Results

Conclusion

Impact

Computer Science

#### Field Connects

Introduction

Problem

Pseudo-teste

Function

Evaluation

Strategy

Results

Conclusion Impact Future Resear ■ Computer Science

Software Engineering

#### Field Connects

Introduction

Problem

Pseudo-teste

Function

Evaluation

Strategy

Results

Conclusion
Impact
Future Resear

Computer Science

Software Engineering

■ Software Testing

#### What is the impact of this research?

Introduction

Problem

Pseudo-teste

Function

Evaluation Strategy

Results

Conclusion
Impact
Future Researd
Demo

%

Coverage with fault detection



Better Understanding of Pseudo-tested Methods



Automatic Detection Tool

#### Future Research

Introduction

Problem

Pseudo-teste Methods

Function Fiasco

Evaluation Strategy

Results

Conclusion

Impact

Future Research

- Different Types and Paramaterization
- Bug Fixes
- Documentation
- Further Testing



#### Demo

Introduction

Problem

Pseudo-teste Methods

Functio

Evaluation

Results

Conclusion Impact Future Resear Demo

#### **Function-Fiasco**

Automatic Detection System for Pseudo-tested Methods