

# (Automated) Software Testing (Automation)

Maurício Aniche

M.FinavaroAniche@tudelft.nl



# Roman Numerals

- Given a string in a roman numeral format, the program needs to convert it to an integer.
- I=1, V=5, X=10, L=50, C=100, D=500, M=1000.
- Combine numerals to make numbers: II=2, VII=7, XVI=16.
- Subtractive notation: I, II, III, IV=4, V, VI, VII, VIII, IX=9, X, ...



```
public class RomanNumeral {  
  
    private static Map<Character, Integer> map;  
  
    static {  
        map = new HashMap<Character, Integer>() {{  
            put('I', 1);  
            put('V', 5);  
            put('X', 10);  
            put('L', 50);  
            put('C', 100);  
            put('D', 500);  
            put('M', 1000);  
        }};  
    }  
}
```

```
public int romanToInt(String s) {  
    int convertedNumber = 0;  
    for(int i = 0; i < s.length(); i++) {  
        int currentNumber = map.get(s.charAt(i));  
        int next = i+1 < s.length() ?  
            map.get(s.charAt(i+1)) : 0;  
  
        if(currentNumber > next)  
            convertedNumber += currentNumber;  
        else  
            convertedNumber -= currentNumber;  
    }  
  
    return convertedNumber;  
}
```



Test case	Concrete Instance
Single letter	I, V, X, L, C, D, M
More than one letter	VI
...	...
...	...
...	...
...	...

It's your turn now!





Test case	Concrete Instance
Single letter	I, V, X, L, C, D, M
Many letters in order	VI, XV

How did you do it? Did you follow any procedure?

Go to <http://bit.ly/sqt-roman-exercise>



Invalid letter	Y
Valid and invalid letter	VIIY
Not valid	IIII, VV
NULL	<null>
...	...

```
@Test
```

```
public void bug() {
```

```
    int result = new RomanNumeral().romanToInt("II");
```

```
    Assertions.assertEquals(2, result);
```

```
}
```

Run: RomanNumeralTest.doubleDigit x

Tests failed: 1 of 1 test – 136 ms

Test Results	136 ms	/Library/Java/JavaVirtualMachines/jdk-10.0.1.jdk/Contents/Home/bin/java ...
RomanNumeralTest	136 ms	org.opentest4j.AssertionFailedError:
doubleDigit()	136 ms	Expected :2 Actual :0 <a href="#">&lt;Click to see difference&gt;</a>
		<5 internal calls>
		at tudelft.sqt.RomanNumeralTest.doubleDigit(RomanNumeralTest.java:19) <19 internal calls>
		at java.base/java.util.ArrayList.forEach(ArrayList.java:1378) <9 internal calls>
		at java.base/java.util.ArrayList.forEach(ArrayList.java:1378) <21 internal calls>

Process finished with exit code 255

```
public int romanToInt(String s) {  
  
    int convertedNumber = 0;  
    for(int i = 0; i < s.length(); i++) {  
        int currentNumber = map.get(s.charAt(i));  
        int next = i+1 < s.length() ?  
            map.get(s.charAt(i+1)) : 0;  
  
        if(currentNumber > next)  
            convertedNumber += currentNumber;  
        else  
            convertedNumber -= currentNumber;  
    }  
  
    return convertedNumber;  
  
}
```

```
public int romanToInt(String s) {  
  
    int convertedNumber = 0;  
    for(int i = 0; i < s.length(); i++) {  
        int currentNumber = map.get(s.charAt(i));  
        int next = i+1 < s.length() ?  
            map.get(s.charAt(i+1)) : 0;  
  
        if(currentNumber >= next)  
            convertedNumber += currentNumber;  
        else  
            convertedNumber -= currentNumber;  
    }  
  
    return convertedNumber;  
  
}
```

# Curiosity

*“The absence of zero and irrational numbers, impractical and inaccurate fractions, and difficulties with multiplication and division **prevented the Romans and the Europeans who later used the system from making advances in number theory and geometry as the Greeks had done in the Pythagorean and Euclidean schools.**”*

<https://www.encyclopedia.com/science/encyclopedias-almanacs-transcripts-and-maps/roman-numerals-their-origins-impact-and-limitations>

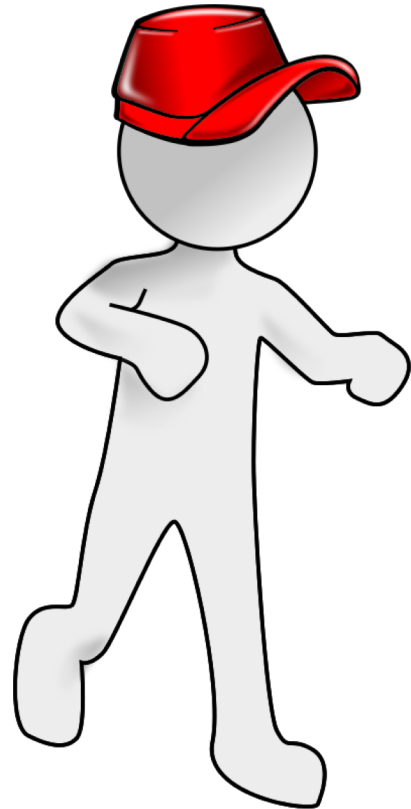


# A little story

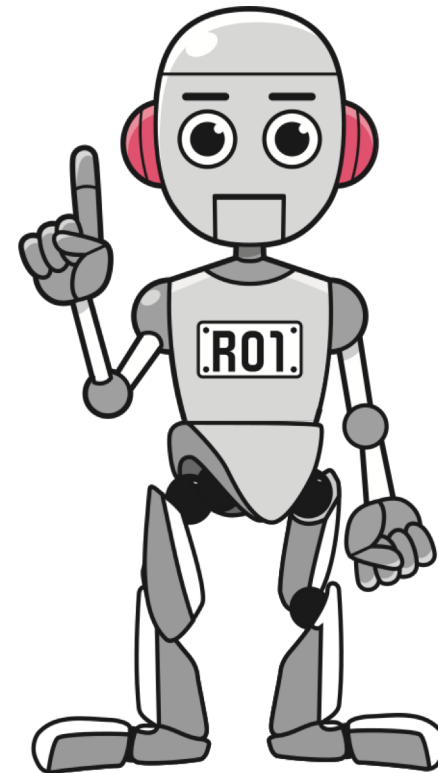
- First job as a developer in 2004
- First important project in 2006
- First important bug: 2006
  
- Tests are important!



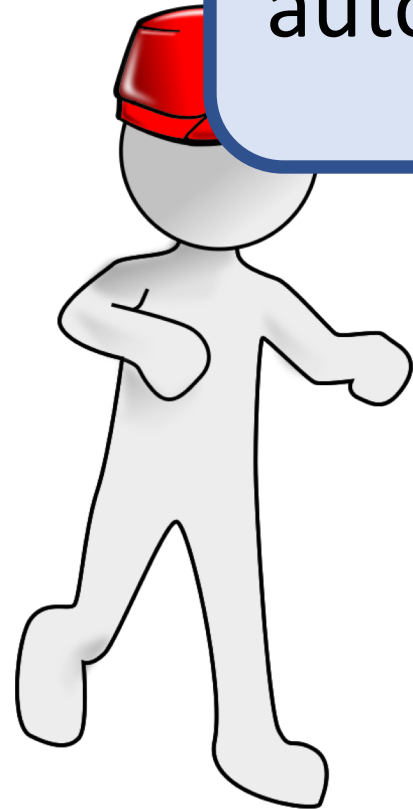
# TEST ANALYSIS & TEST DESIGN



# TEST EXECUTION

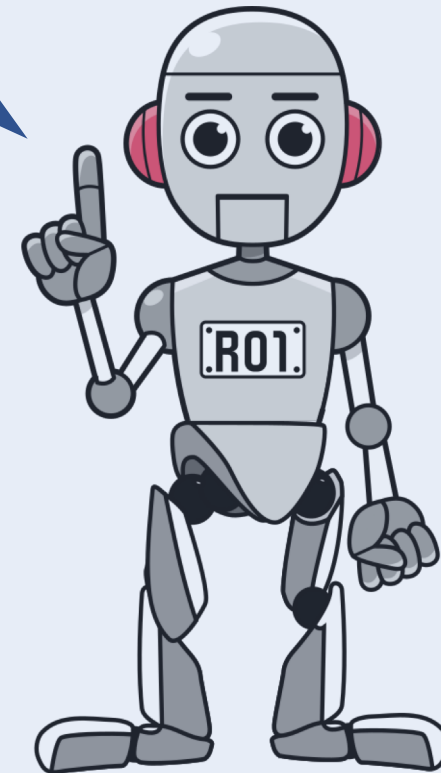


TEST ANALYSIS  
& TEST



How can you  
automate me?

TEST  
EXECUTION



@Test

```
void singleDigit() {  
    Assertions.assertEquals(1, new RomanNumeral().romanToInt("I"));  
    Assertions.assertEquals(5, new RomanNumeral().romanToInt("V"));  
    Assertions.assertEquals(10, new RomanNumeral().romanToInt("X"));  
    Assertions.assertEquals(50, new RomanNumeral().romanToInt("L"));  
    Assertions.assertEquals(100, new RomanNumeral().romanToInt("C"));  
    Assertions.assertEquals(500, new RomanNumeral().romanToInt("D"));  
    Assertions.assertEquals(1000, new RomanNumeral().romanToInt("M"));  
}
```

@Test

```
void repetition() {  
    Assertions.assertEquals(2, new RomanNumeral().romanToInt("II"));  
    Assertions.assertEquals(20, new RomanNumeral().romanToInt("XX"));  
}
```

@Test

```
void manyLettersInOrder() {  
    Assertions.assertEquals(1000, new RomanNumeral().romanToInt("VI"));  
    Assertions.assertEquals(1000, new RomanNumeral().romanToInt("XV"));  
}
```

...

All tests in  
<http://bit.ly/sqt-roman-2>

# What are the advantages?

- Too slow → Too Fast
  - Too expensive → Machine is cheap
  - Not easy to reproduce → Reproducible
  - Susceptible to failures → No failures
  - ... boring! → Very very cool!
- 
- But there's a learning curve (as with any technique).





*“But if you write 100 lines of production code, now you’ll write only 50, as the other 50 are testing. Therefore, you are less productive.”*  
– says a bad manager.

# Not true.

- You spend a lot of time in executing manual tests.
  - Now, you will spend it only once: to write the test.
- Teams with automated test suites spend less time debugging.

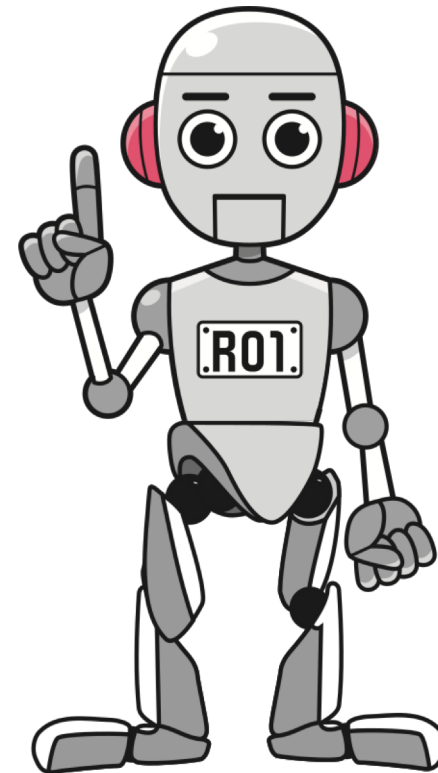
George, B., Williams, L., An Initial Investigation of TDD in Industry. ACM Symposium on Applied Computing. Melbourne, Florida, USA, 2003.

Janzen, D., Software Architecture Improvement through Test-Driven Development. Conference on Object Oriented Programming Systems Languages and Applications, ACM, 2005

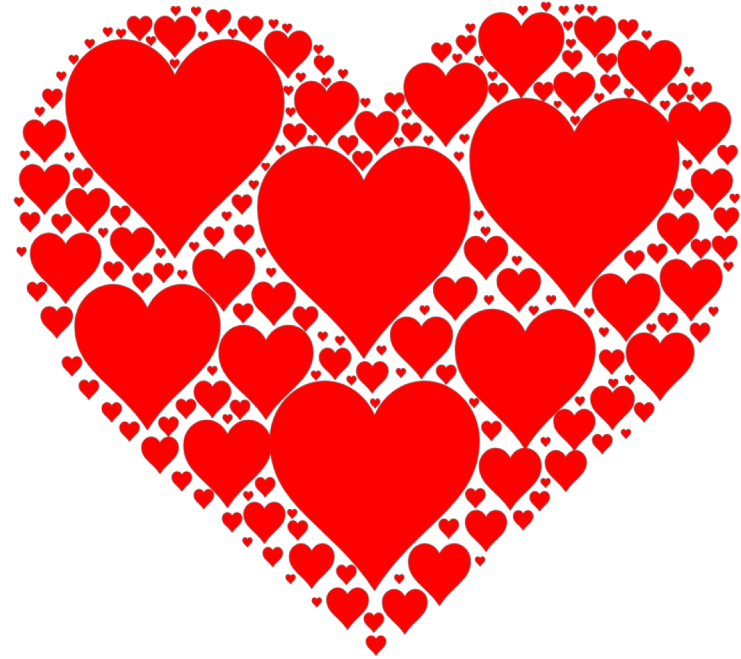
# TEST ANALYSIS & TEST DESIGN



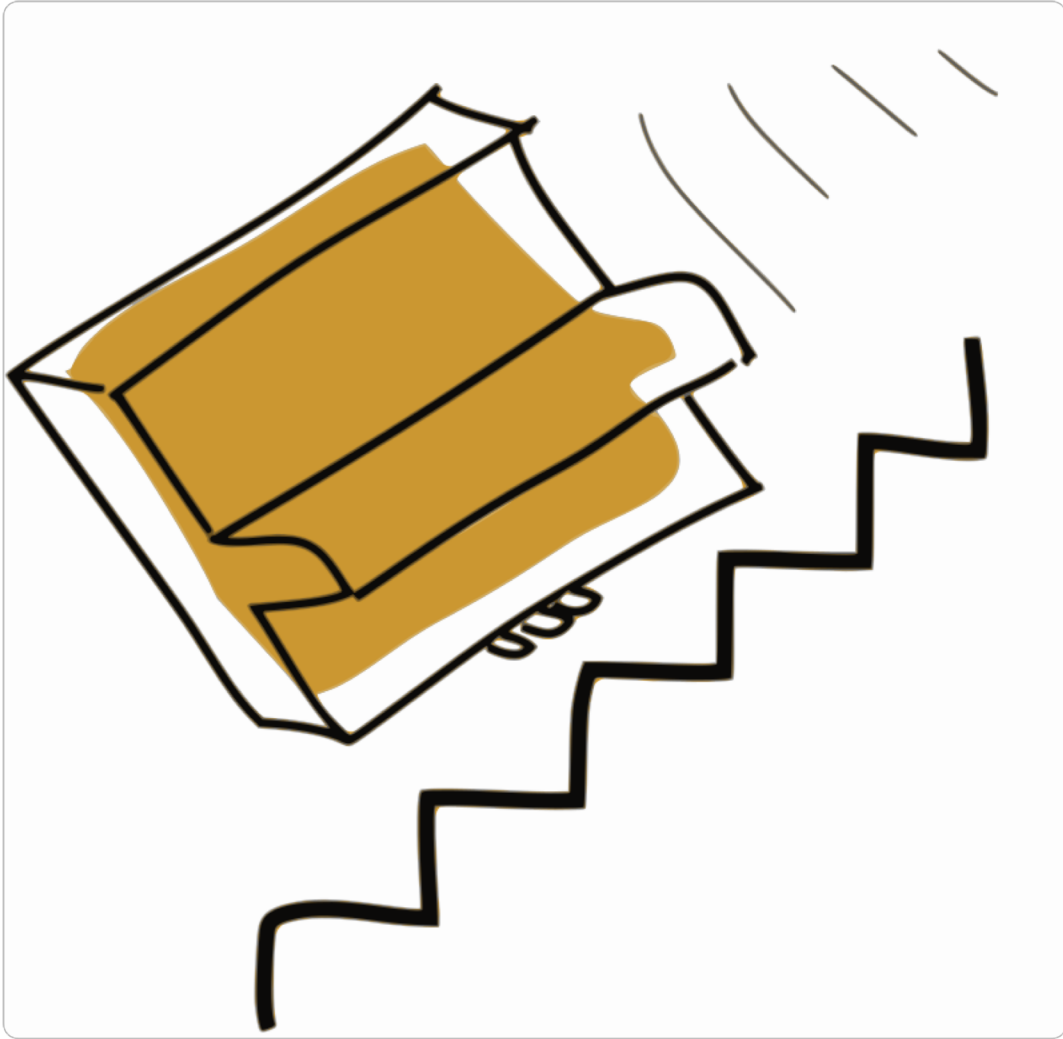
# TEST EXECUTION



I told you to use  
your hearts  
when designing  
the tests!



What's the problem with that?



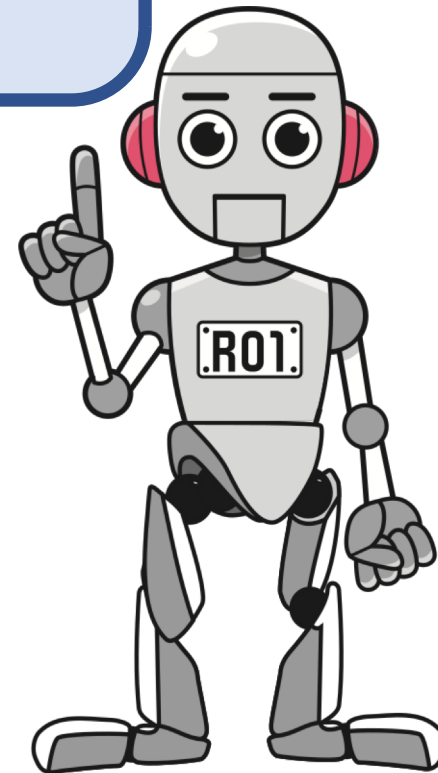
A systematic  
approach  
would be  
better!



TEST ANALYSIS  
& TEST DESIGN

TEST  
EXECUTION

How can you  
automate me?



# The Oracle Problem in Software Testing: A Survey

Earl T. Barr, Mark Harman, Phil McMinn, Muzammil Shahbaz, and Shin Yoo

**Abstract**—Testing involves examining the behaviour of a system in order to discover potential faults. Given an input for a system, the challenge of distinguishing the corresponding desired, correct behaviour from potentially incorrect behavior is called the “test

oracle

Without

has int

testing

aware

forms o

compre

testing

Index

Index

## 1 INTRODUCTION

MUCH

much

to make t

this end, v

guishes be

the System

the System

Howeve

the problem

icantly less

solved. This

current open

bottleneck

that inhibits

The literature on test oracles has introduced techniques for oracle automation, including modelling, specifications, contract-driven development and metamorphic testing. **When none of these is completely adequate, the final source of test oracle information remains the human, who may be aware of informal specifications, expectations, norms and domain specific information that provide informal oracle guidance.**

Where no full specification of the properties of the SUI exists, one may hope to construct a partial test oracle that can answer questions for some inputs. Such partial test oracles can be constructed using metamorphic testing

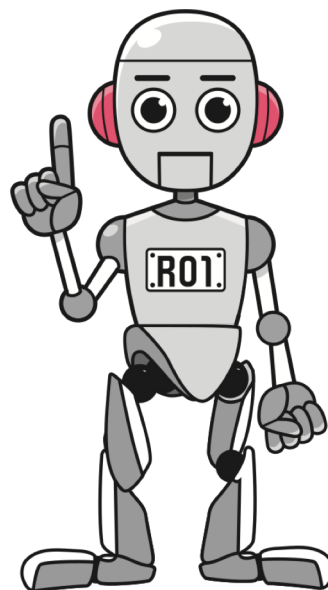
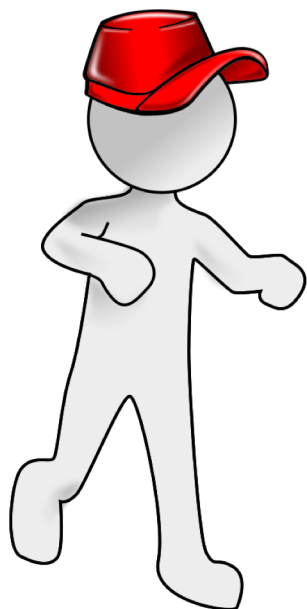
Where no full specification of the properties of the SUI exists, one may hope to construct a partial test oracle that can answer questions for some inputs. Such partial test oracles can be constructed using metamorphic testing

Annibale Panichella will talk  
about automated testing  
generation on June 11<sup>th</sup>



***“Testing is different from writing tests.  
Developers write tests as a way to give them  
space to think and confidence for refactoring.  
Testing focuses on finding bugs. Both should  
be done.”***

<https://medium.com/@mauricioaniche/testing-vs-writing-tests-d817bffa6bc>



Find systematic and/or  
automated ways to design and  
execute tests!



# License

- You can use and share any of my material (lecture slides, website).
- You always have to give credits to the original author.
- You agree not to sell it or make profit in any way with this.
  
- Material that I refer has its own license. Please check it out.

