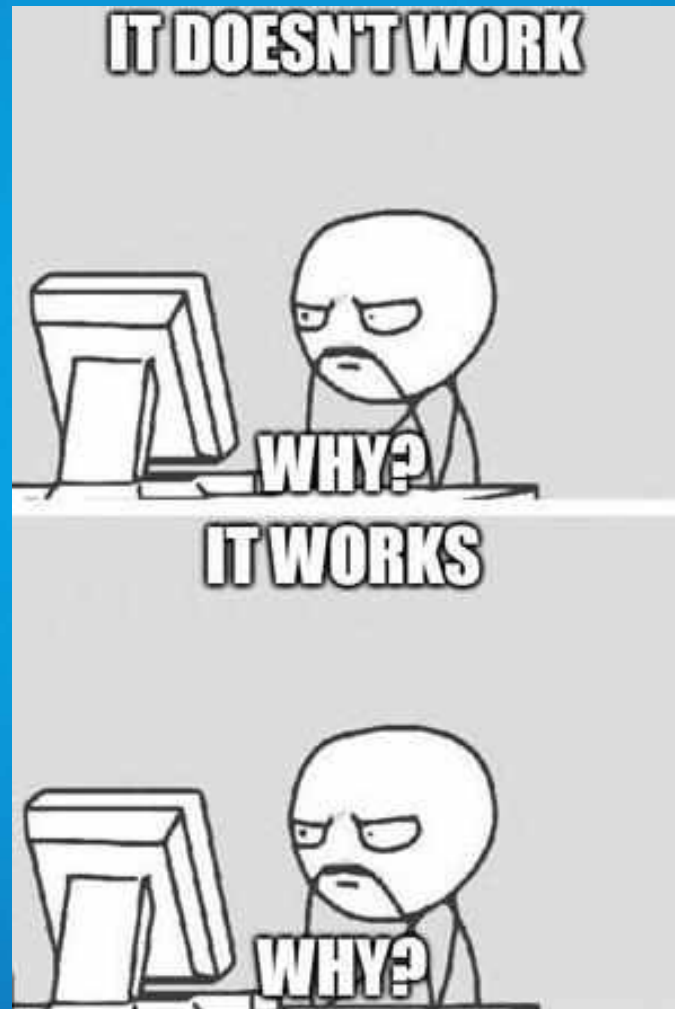


Characterization Tests

Reto Lehre



04.02.2020



Content

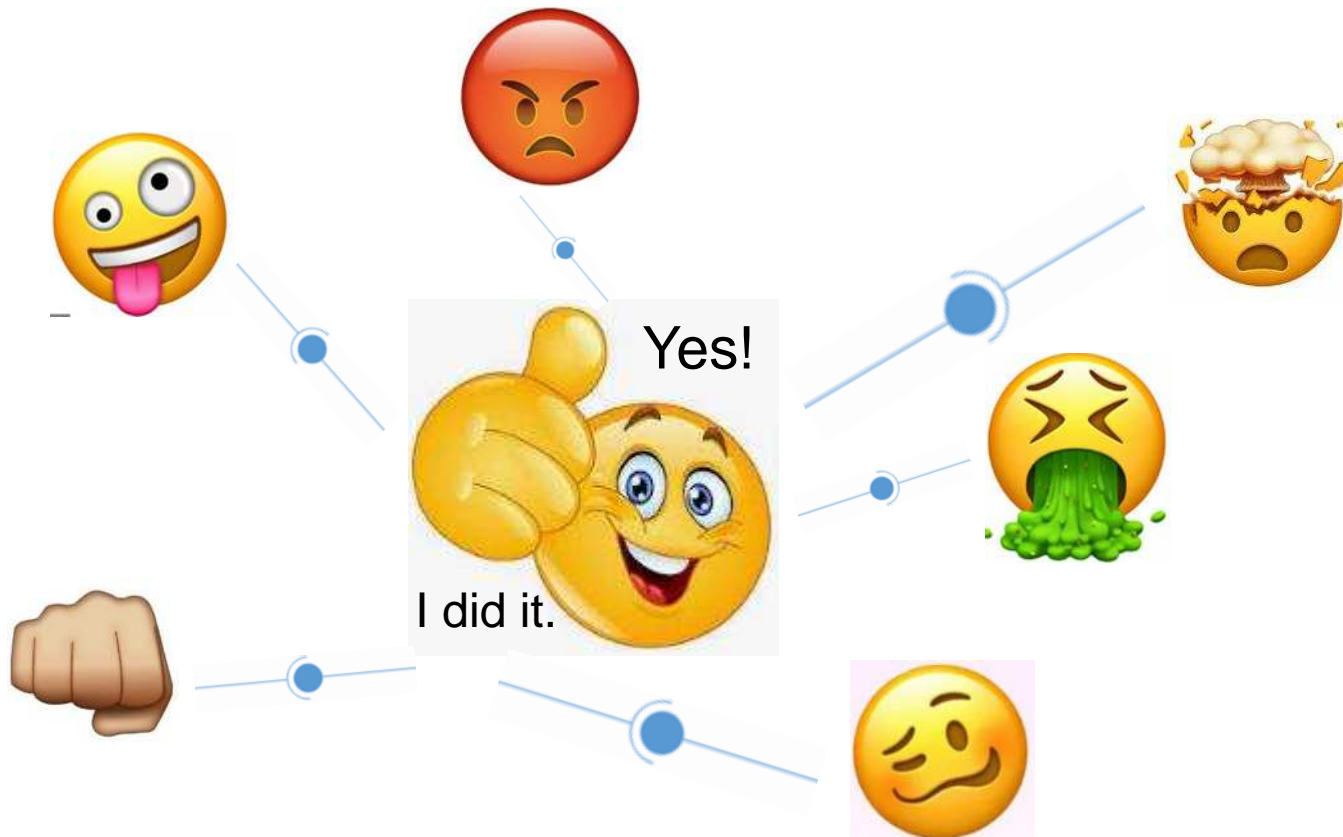
- Introduction
- Attention
- Doing
- Approval Test
- Example
- Facing Problems?

Introduction

- A characterization test is also called a 'Golden Master Test', because it protects the behavior of your code.
- Michael C. Feathers introduced it.
- It's often used for legacy code or code with bad or no coverage.
- It's relatively easy to implement it.

Attention

- Refactoring can break interfaces!
- Be careful refactoring bad legacy code, because your client can handle it.



Doing



- Identify a piece of code for testing.
- Write an assertion test that fails.
- Let the failure tell you what the behaviour is.
- Change the test so that it expects the behaviour that the code produces.
- Repeat.

Approval Test

- An approval test lets you test complex objects.
- The test creates you a received file which you can approve with an output file.
- Easy use under Java.
- Well documented in <https://approvaltests.com/>

Example

```
ApprovalsExample.java
1 package IntroductionTests;
2
3 public class ApprovalsExample {
4     @ public static String mergeNameAndAge(String name, int age){
5         String title = "";
6         if(age <= 18){
7             title = "Junior";
8         }
9         if(age > 18 && age <= 60){
10            title = "Mister";
11        }
12        if(age > 60){
13            title = "Senior";
14        }
15        return title + " " + name + ", " + age + " years";
16    }
17 }
18
```

```
ApprovalsExampleTest.java
1 package IntroductionTests;
2
3 import org.approvaltests.combinations.CombinationApprovals;
4 import org.junit.jupiter.api.Test;
5
6 class ApprovalsExampleTest {
7
8     @Test
9     void mergeNamesAndAges() throws Exception {
10        String[] names = new String[]{"reto", "marco"};
11        Integer[] ages = new Integer[]{21, 45};
12
13        CombinationApprovals.verifyAllCombinations(
14            (name, age) -> {
15                String result = ApprovalsExample
16                    .mergeNameAndAge(name, age);
17                return result;
18            }
19            , names, ages);
20    }
21 }
```

Facing Problems?

- Often Code is not easy to understand.
- You will face a lot of dependency problems.
- Use Legacy code smells.
- Make yourself master the 'Dependency-Breaking Techniques'
- What's that?
- => take part at the 'Alcor Academy' code renovation course!

References

- Wikipedia
- Alcor Academy
- Working Effectively with Legacy Code / Michael C. Feathers

Characterization Test



?

