Who Tests The Tests?

by Niklas Lindholm in Mountain View

Are you sure that your code is testing what you think it is testing?

Let's say that we have written two test cases for the `UsaPhoneNumberValidator` class. One for the happy case, and one for when there is a bad area code. At the moment, the only error detection in the validator is for area codes. These two tests pass, meaning that our production code detects the bad area code.

```java
public void testValidatePhoneNumber_goodNumber() {
   assertTrue(usaValidator.isValidPhoneNumber("650 253 0000"));
}
public void testValidatePhoneNumber_badAreaCode() throws Exception {
   assertFalse(usaValidator.isValidPhoneNumber("111 253 0000"));
}
```

We then add some logic to the production code to detect if the prefix is bad. The test for the prefix is very similar to the area code test so we copy it and make some modifications.

```java
public void testValidatePhoneNumber_badPrefix() throws Exception {
   assertFalse(usaValidator.isValidPhoneNumber("111 000 0000"));
}
```

We run the tests and they pass! That means everything is correct. Right?

Well, maybe. **We did not test the logic to validate the prefix at all**. We forgot to change the area code back to a valid value. This test will pass because of the bad area code.

If the test had been executed before we wrote the new production code, we would have noticed that the test passed without the production code to validate the prefix! That means something is wrong with the test.

Running the test before you write the production code raises your confidence that your test is actually testing what you think it is testing.

Code Dashboard
How flaky are your tests? How flaky are your neighbors' tests? Find out!
unittest/tests/owned_by/<user name>

Where's my test tmp dir?
Find out how to use test tmp dir in Python, Java, C++ or even shell script by reading the unit test how-to doc.
wiki/Main/UnittestHowTo

Read all TotTs online: `http://tott`

Test Certified: `http://go/test-certified`