Change-Detector Tests Considered Harmful

You have just finished refactoring some code without modifying its behavior. Then you run the tests before committing and… a bunch of unit tests are failing. **While fixing the tests, you get a sense that you are wasting time by mechanically applying the same transformation to many tests**. Maybe you introduced a parameter in a method, and now must update 100 callers of that method in tests to pass an empty string.

What does it look like to write tests mechanically? Here is an absurd but obvious way:

```
// Production code:
def abs(i: Int)
  return (i < 0) ? i * -1 : i

// Test code:
for (line: String in File(prod_source).read_lines())
  switch (line.number)
    1: assert line.content equals "def abs(i: Int)"
    2: assert line.content equals "  return (i < 0) ? i * -1 : i"
```

That test is clearly not useful: it contains an exact copy of the code under test and acts like a checksum. **A correct or incorrect program is equally likely to pass** a test that is a derivative of the code under test. No one is really writing tests like that, but how different is it from this next example?

```
// Production code:
def process(w: Work)
  firstPart.process(w)
  secondPart.process(w)

// Test code:
part1 = mock(FirstPart)
part2 = mock(SecondPart)
w = Work()
Processor(part1, part2).process(w)
verify_in_order
  was_called part1.process(w)
  was_called part2.process(w)
```

It is tempting to write a test like this because it requires little thought and will run quickly. **This is a** *change-detector test*—it is a transformation of the same information in the code under test—and **it breaks in response to** *any* **change to the production code, without verifying correct behavior** of either the original or modified production code.

Change detectors provide negative value, since the tests do not catch any defects, and the added maintenance cost slows down development. These tests should be re-written or deleted.

More information, discussion, and archives:
http://googletesting.blogspot.com