Testing on the Toilet

Only Verify Relevant Method Arguments

What makes this test fragile?

```
@Test public void displayGreeting_showSpecialGreetingOnNewYearsDay() {
  fakeClock.setTime(NEW_YEARS_DAY);
  fakeUser.setName(“Fake User”);

  userGreeter.displayGreeting();

  // The test will fail if userGreeter.displayGreeting() didn’t call
  // mockUserPrompter.updatePrompt() with these exact arguments.
  verify(mockUserPrompter).updatePrompt(
      "Hi Fake User! Happy New Year!", TitleBar.of("2018-01-01"), PromptStyle.NORMAL);
}
```

The test specifies exact values for all arguments to `mockUserPrompter`. **These arguments may need to be updated when the code under test is changed, even if the changes are unrelated to the behavior being tested**. For example, if additional text is added to `TitleBar`, every test in the codebase that specifies this argument will need to be updated.

In addition, **verifying too many arguments makes it difficult to understand what behavior is being tested** since it’s not obvious which arguments are important to the test and which are irrelevant.

Instead, **only verify arguments that affect the correctness of the specific behavior being tested**. You can use argument matchers (e.g., `any()` and `contains()` in Mockito) to ignore arguments that don't need to be verified:

```
@Test public void displayGreeting_showSpecialGreetingOnNewYearsDay() {
  fakeClock.setTime(NEW_YEARS_DAY);

  userGreeter.displayGreeting();

  verify(mockUserPrompter).updatePrompt(contains("Happy New Year!"), any(), any()));
}
```

Arguments ignored in one test can be verified in other tests. Following this pattern allows us to **verify only one behavior per test**, which makes tests more readable and more resilient to change. For example, here is a separate test that we might write:

```
@Test public void displayGreeting_renderUserName() {
  fakeUser.setName(“Fake User”);

  userGreeter.displayGreeting();

  // Focus on the argument relevant to showing the user's name.
  verify(mockUserPrompter).updatePrompt(contains("Hi Fake User!"), any(), any());
}
```

More information, discussion, and archives:
testing.googleblog.com