Software Testing, Quality Assurance & Maintenance—Lecture 24 (code)

Patrick Lam

March 18, 2019

Mock Objects

John Tenniel's original (1865) illustration for Lewis Carroll's "Alice in Wonderland". Alice sitting between Gryphon and Mock turtle.

Mock Objects: Email Sender

```
// state or behaviour verification?
public class MailServiceStub
                    implements MailService {
  private List<Message> messages =
      new ArrayList<Message>();

  public void send (Message msg) {
    messages.add(msg);
  }
  public int numberSent() {
    return messages.size();
  }
}
```

Mock Objects: Behaviour Verification

```
// jMock syntax
class OrderInteractionTester... {
  public void testOrderSendsMailIfUnfilled() {
    Order order = new Order(TALISKER, 51);
    Mock warehouse = mock(Warehouse.class); // (1)
    Mock mailer = mock(MailService.class);
    order.setMailer((MailService) mailer.proxy());

    mailer.expects(once()).method("send"); // (2)
    warehouse.expects(once()).method("hasInventory
        ")
      .withAnyArguments()
      .will(returnValue(false));

    order.fill((Warehouse) warehouse.proxy());
  }
}
```

Mock Objects: Document Management

```
// EasyMock syntax
@RunWith(EasyMockRunner.class)
public class ExampleTest {
  @TestSubject
  private ClassUnderTest classUnderTest =
      new ClassUnderTest();

  @Mock // creates a mock object
  private Collaborator mock;

  @Test
  public void testRemoveNonExistingDocument() {
    replay(mock);
    classUnderTest.removeDocument
        ("Does not exist");
  }
}
```

Mock Objects: Using a Mock

```
@Test
public void testAddDocument() {
  // ** recording phase **
  // expect document addition
  mock.documentAdded("Document");
  // expect to be asked to vote for document
      removal, and vote for it
  expect(mock.voteForRemoval("Document"))
                .andReturn((byte) 42);
  // expect document removal
  mock.documentRemoved("Document");
  replay(mock);
  // ** replaying phase ** we expect the recorded
      actions to happen
  classUnderTest.addDocument("New Document",
      new byte[0]);
  // check that the behaviour actually happened:
  verify(mock);
```

Flakiness: Good for croissants[1], bad for tests

[1] thanks Pixabay