Test Doubles

Dario Campagna

Head of Research

What’s a Test Double?

Any kind of pretend object that is used in place of a real object for testing purposes.

- Name coined by Gerard Meszaros.
- Comes from the notion of Stunt Doubles in movies.

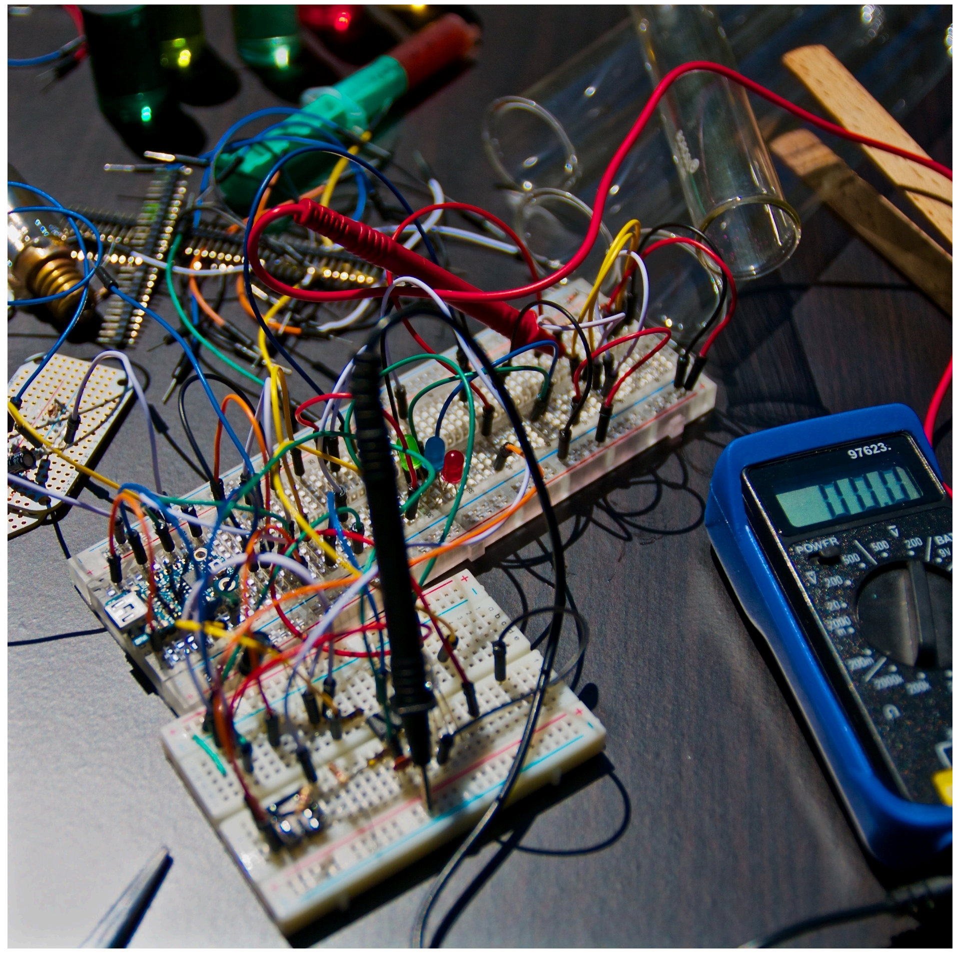
Bobby Holland Hanton and Chris Hemsworth. Instagram/@bobbydazzler84

Tests in TDD

Should be fast, isolated, specific, deterministic.

Shouldn’t depend on data or other fixtures outside the control of each test case.

- File system
- Databases
- Network communications
- Hardware systems
- …



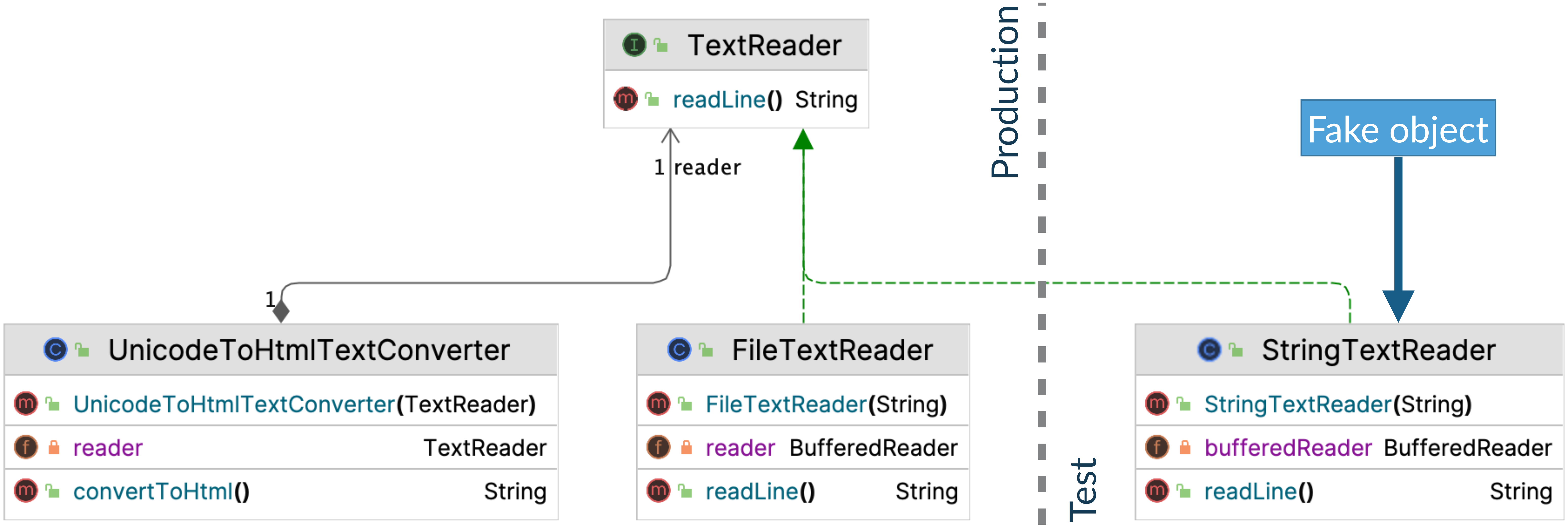
Types of test double

Meszaros defined five particular kinds of double.

- Dummy
- Fake
- Stubs
- Spies
- Mocks

Fake

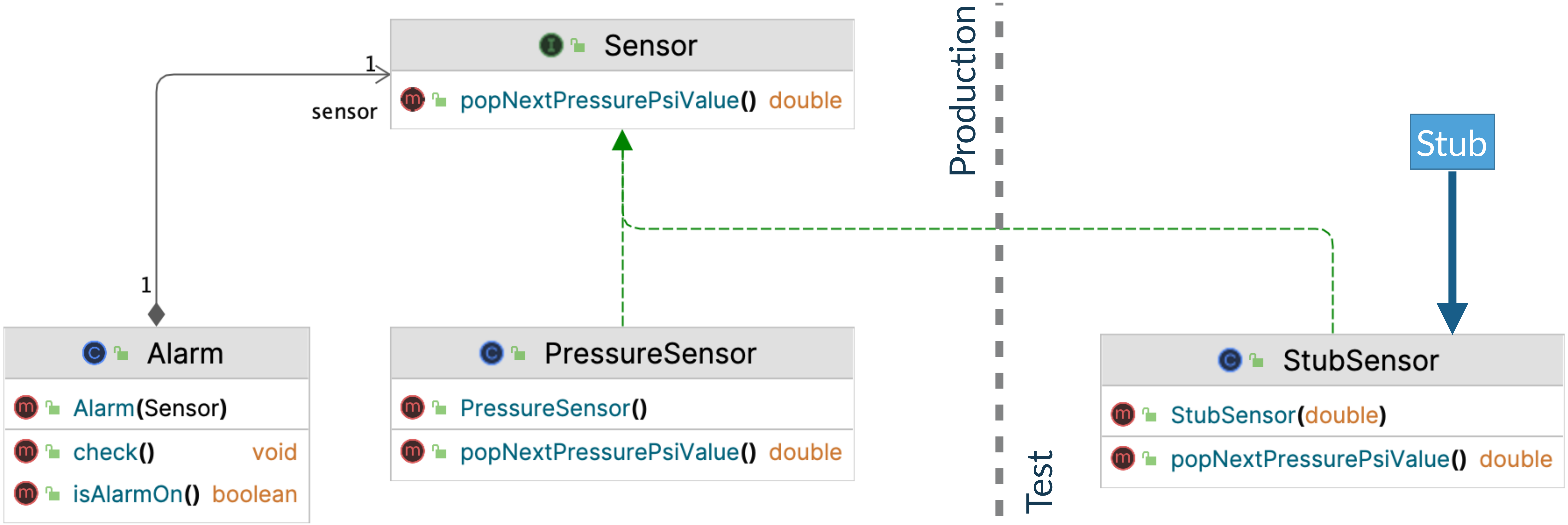
Fake objects actually have working implementations, but usually take some shortcut which makes them not suitable for production.



Example: https://github.com/dario-campagna/unicode-to-html-text-converter

Stub

Stubs provide canned answers to calls made during the test, usually not responding at all to anything outside what's programmed in for the test.



Example: https://github.com/dario-campagna/tire-pressure-monitoring-system

Mock Objects

Mocks are objects pre-programmed with expectations which form a specification of the calls they are expected to receive.

```
void testOrderSendsMailIfUnfilled() {
  Order order = new Order(TOMATO, 51);
  Mock warehouse = mock(Warehouse.class);
  Mock mailer = mock(MailService.class);
  order.setMailer((MailService) mailer.proxy());

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

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

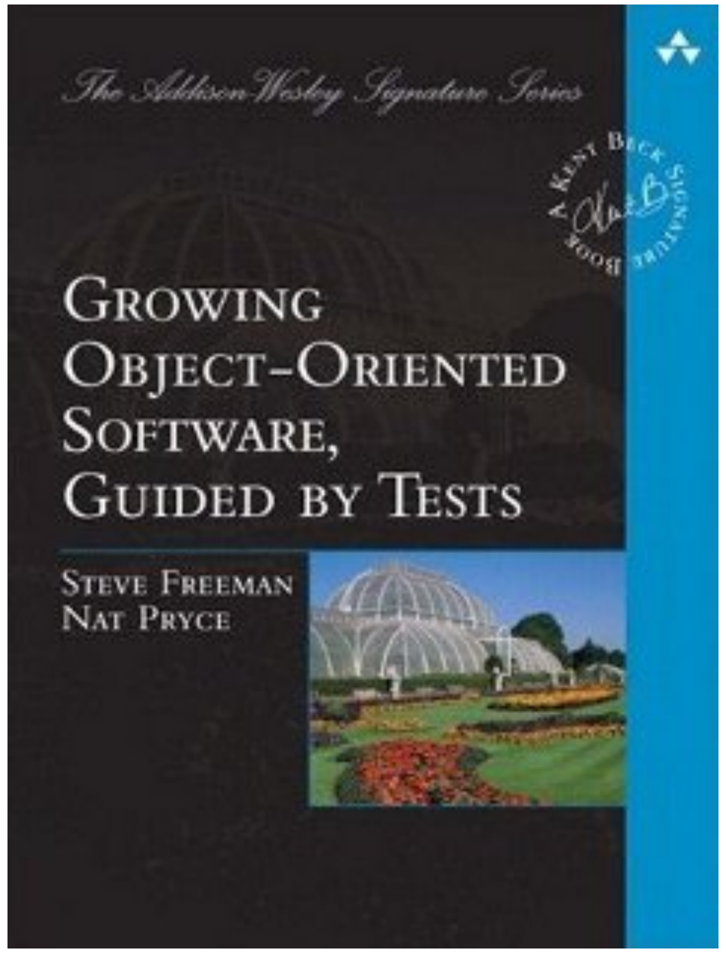
```
void testOrderSendsMailIfUnfilled() {
  Order order = new Order(TOMATO, 51);
  Warehouse warehouse = mock(Warehouse.class);
  MailService mailer = mock(MailService.class);
  order.setMailer(mailer);

   when(warehouse.hasInventory(any()))
      .thenReturn(false);

   order.fill(warehouse);

   verify(mailer).send();
   verify(warehouse).hasInventory(any())
}
```

Behavior under test (jMock on the left, Mockito on the right): we want to send an email message if we fail to fill an order.

Mock Objects

Introduced by Steve Freeman, Nat Pryce and others, it is really a technique for identifying types in a system based on the roles that objects play.

- Supports iterative interface discovery
- Changes the focus of TDD from state changes to interactions
- Useful to test interactions with third party code too


The Addison-Wesley Signature Series
A Kent Beck Signature Book
Growing Object-Oriented Software, Guided by Tests
Steve Freeman
Nat Pryce

Mock Objects in practice

Three heuristics from the paper Mock Roles, not Objects

Types

Only mock types you own.

Getters

Don't use getters.

Accuracy and flexibility

Specify as little as possible in a test.

References

Mocks Aren’t Stubs

https://martinfowler.com/articles/mocksArentStubs.html

Mock Roles, not Objects

http://jmock.org/oopsla2004.pdf

Growing Object Oriented Software

http://www.growing-object-oriented-software.com

How I learned to love mocks

https://medium.com/@xpmatteo/how-i-learned-to-love-mocks-1-fb341b71328