JMock Crash-Course

By Paul Hammant of
ThoughtWorks, Inc

Why?

- Too hard to test things that involve external components?
- Too slow, even if you could?
- Unit Tests as much fun as smacking yourself in the head with a baseball bat?

- You can ‘mock’ out SpellChecker while unit-testing WordProcessor, etc
- Anything ‘external’ or ‘heavy’ to the thing you really want to test.

Simple Example

ThoughtWorks®

```
public class FooTest extends MockObjectTestCase {
   Mock mockMap = mock(Map.class);
   public void testJMockItself() {


      mockMap.expects(once()).method("get")
            .with(eq("jmock-rocks"))
            .will(returnValue("Yup"));


      Map map = (Map) mockMap.proxy();
      assertEquals("Yup", map.get("jmock-rocks"));
   }
}
```

"mockMap expects 'get' to be invoked once with a single parameter of 'jmock-rocks' and will return a value of 'Yup'." *

If anything other than that happens, exceptions will be thrown by JMock

Constraints

ThoughtWorks®

How many times the method is called:

- expects(once())
- expects(atLeastOnce())
- expects(exactly(n))
- expects(never())

Parameters

ThoughtWorks®

- with(..)

Syntax:

- with(eq(parm1), eq(parm2)) // etc

Alternatives:

- withNoArguments()
- withAnyArguments()

Some Boolean Flexibility:

- same(..) & isA (..) & not (..) & or (..) & and (..) // complex huh?

Returns

ThoughtWorks®

- will(returnValue (..))
- will(throwException (..))

JMock Facts

ThoughtWorks®

- Use with Junit (not instead of)
  - MockObjectTestCase extends TestCase
- It is just a simple library
  - that uses lots of reflection
- Can do concrete classes too -
  - via another library - more difficult to get right
- its.builder().syntax().reads().well().right();
- www.jmock.org - widely used and respected
- Other tools by Joe Walnes: XStream, QDox, Sitemesh, NMock