

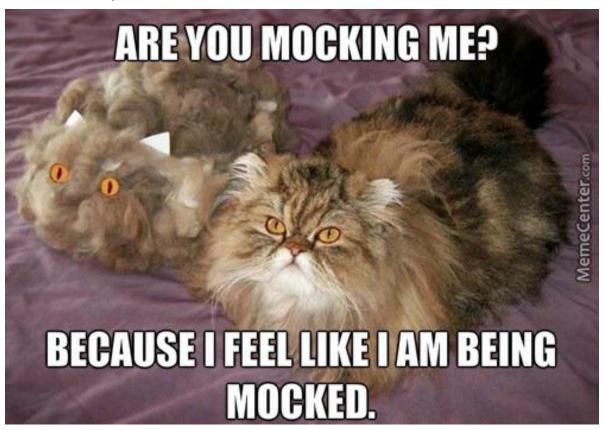


Software quality and testing

- 4<sup>th</sup> course

## Mocks, stubs, and Fakes

Actually it is stubs, mocks and fakes!







#### Stubs

- Def.: The important point is that the test can't control what that dependency returns to your code under test or how it behaves (if you wanted to simulate an exception for example). This is the case when you use stubs.
- Example: Astronauts.
- Summing it up: It is *not safe* or takes *too long* to do a full integration test.
- Updated def.: A stub is a controllable replacement for an existing dependency (or collaborator) in the system. By using a stub, you can test your code without dealing with the dependency directly.
- Def.: Seams are places in your code where you can plug in different functionality, such as stub class, constr. param. Digital

#### Mocks and Stubs

- Def.: The main thing to remember about mocks versus stubs is that mocks are just like stubs, but you assert against the mock object, whereas you do not assert against a stub.
- Def.: A fake denotes an object that looks like another object but can be used as a *mock* or a *stub*.





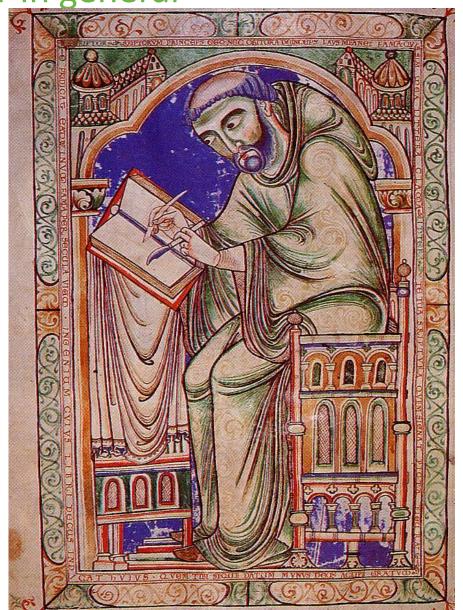
### **Techniques**

- Extract an interface and replace underlying implementation
- Dependency injection
  - Constructor level {pros, cons}
  - Inject a fake as a property get or set {pros, cons}
  - Injecting a fake just before a method call {pros, cons}
- Mocking





Mocking NOT in general





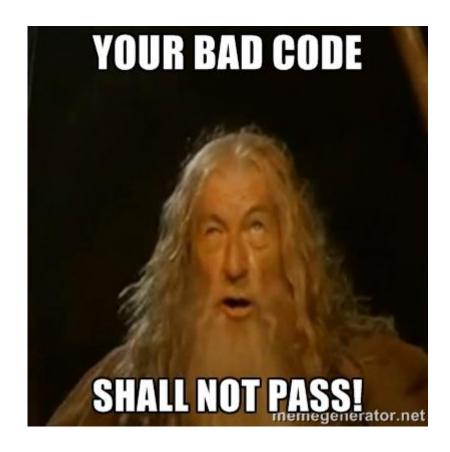
## Mocking techniques

- Value based, state based, interaction testing
- Lesser handwritten
- Used together with a stub
- One mock per test!
- Fake chains: stubs that produce mocks or stubs...





### Mockito framework







#### Mockito framework cont.

- Support both for TDD, BDD
- Ultimately it is expect-run-verify pattern







# Any questions?









eitdigital.eu

http://www.doctoralschool.eitdigital.eu/doct oral-training-centres/dtc-budapest/