

# What is a unit test?



“Unit testing is a method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures are tested to determine if they are fit for use.”

***Wikipedia***



Um... what?



# What is an app?



An app is a set of behaviours created by programmer and expected by user.



We, programmers, have  
a limited cognition. As  
all humans do.



We can't always 'load'  
all of the code of our  
app into our memory.



This means that we can,  
by accident, change the  
behaviour of the app.



Preserving behaviour of  
complex systems is  
hard. In fact, of any  
system at all.



Enter unit tests.



Unit test is a failsafe to  
make sure app  
behaviour is preserved.



# What is a unit test?



Unit tests test smallest  
components of your code  
in isolation with test code

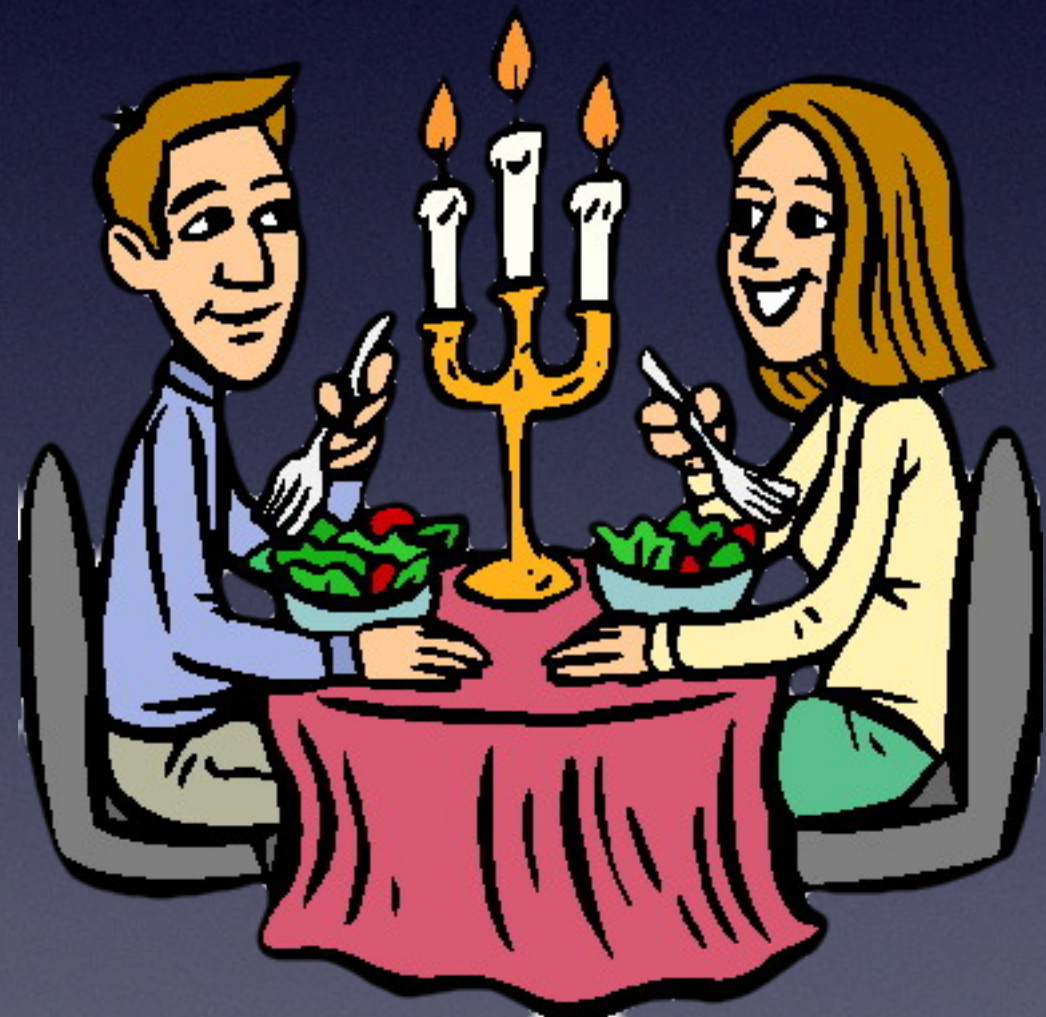


Unit tests test smallest  
components of your code  
in **isolation** with test code



# Test isolation





Table

processOrder



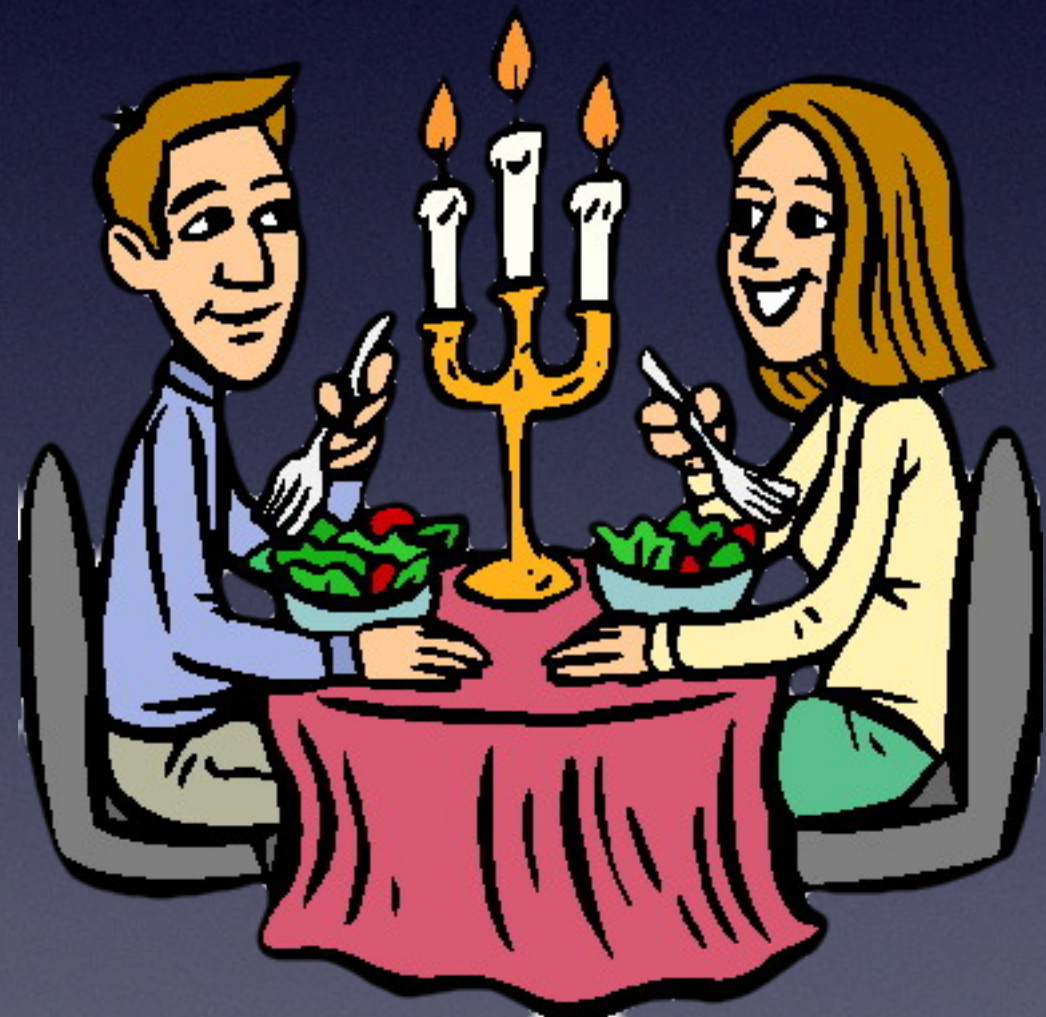
Waiter

getDishes



Cook





Table

processOrder



Waiter

getDishes



Fake cook



Cook



# Why isolate?



# Unit test lifecycle



# Unit test lifecycle

- Arrange
- Act
- Assert



When a unit test is not  
a unit test?



# When a unit test is not a unit test?

by Michael Feathers



# A test is not a unit test if...

- It talks to a database
- It communicates across network
- It touches the file system
- You have to do special things to your environment to run it (edit config files etc)



# A test is not a unit test if...

- It talks to a database
- It communicates across network
- It touches the file system
- You have to do special things to your environment to run it (edit config files etc)



# A test is not a unit test if...

- It talks to a database
- It communicates across network
- It touches the file system
- You have to do special things to your environment to run it (edit config files etc)



A 100 ms tests is a  
**very** slow test.



1500 tests each running 100  
ms. That's 150 seconds. Two  
and a half minutes.



Where does TDD fit in  
all this?



In TDD you always  
write test first. Always.



However, if that's the  
only thing you do - then  
you're doing it wrong.



TDD is not “just adding tests first”. It’s a complete workflow.



TDD is a great way to  
“start” when you’re not  
in the zone...



... and way to remind  
yourself what you've been  
working on yesterday!



TDD is a great way to  
determine how complex your  
code has become.

You just have to listen.



Have to fake seven  
objects to isolate test?



Have to inject a fake  
into a fake into a fake?





This always points to an  
overcomplicated design.

And your tests are here to point  
that out. Very clearly.



What unit tests  
can't do?



Unit tests are never  
a guarantee that you  
won't ship a bug.



But they're damn good at  
greatly reducing amount  
of bugs. And time spent  
on QA.



Are unit tests an invaluable tool for writing great software? Heck yes.  
Am I going to produce a poor product if I can't unit test? Hell no.

Jonathan Rasmusson

<http://agilewarrior.wordpress.com/2012/10/06/its-not-about-the-unit-tests/>



# Resources & Contact

Contact

@eldudi

pawel@dudek.mobi