

unittest

It ain't broke, let's fix it!

unittest is actually pretty good

Two reasons for reluctance

Robert Collins
<robert.collins@canonical.com>

Caged deathmatch

Time for questions

1. Structure of unittest

2. Extending unittest

3. Good and bad points

4. Caged deathmatch

unittest is actually pretty good

<h2>What is it?</h2>

Help you write tests

Not really it

API for manipulating & running tests

API for manipulating & running
arbitrary user code with a narrow,
predefined range of results

Three core classes

Three core concepts

"Find arbitrary chunk of user code"
class

"Arbitrary chunk of user code" class

"Collect results of user code" class

None of the other classes matter

TestRunner, TestSuite

Ignoring TestLoader

The narrow interface of TestCase

```
countTestCases() :: Integer  
shortDescription() :: String  
id() :: String  
run(TestResult)  
debug(TestResult)
```

***That's* the public interface.**

Not assertEquals or setUp or
whatever

The less narrow interface of TestResult

```
startTest(TestCase)  
stopTest(TestCase)  
wasSuccessful() :: Bool  
addError(TestCase, Exception)  
addFailure(TestCase, Exception)  
addSuccess(TestCase)
```

(circa Python 2.3)

Reporting test activity

How they fit together

```
test_case.run(test_result)
```

```
test_result.addFoo(test_case, foo)
```

<h2>How (not) to extend unit
test</h2>

Some examples

Sharing expensive resources
between tests

Twisted

Trial

setUpClass / tearDownClass

My mistake

TestCase.setUpClass,
TestCase.tearDownClass

Define attributes, those attributes available to tests in that class.

Bad because:

Test correctness depended on test
ordering

Couldn't interleave tests from one class with tests from another.

Breaks implicit contract of unittest

A TestCase instance is a standalone
thing

Next example

Zope

zope.testing layers

Similar problem

Test defines magical attribute

Loader finds attribute, constructs
tests appropriately

Requires special TestResult object
to store state

You cannot construct a test that
uses layers and run it

You must use the Zope
infrastructure

<h2>How to extend unit test</h2>

testresources

```
bzr branch lp:testresources
```

Resource manager objects

Acquire and release resources from
these objects

ResourcedTestCase(TestCase)

Define 'resources' attribute

Customized setUp and tearDown
that acquire and release

run() works

OptimizedTestSuite that orders tests

Doesn't affect test results at all

Maintains contract

subunit

bzr branch lp:subunit

Customized test result that allows
running tests out of process

Javascript

Perl

C

Cloud

<h2>Checkpoint</h2>

unittest model is right

Not super clear from the code

Interoperable extensions

It ain't broke

<h2>Let's Fix It!</h2>

It's got bugs.

Props to Michael Foord

addCleanup

Some bugs

countTestCases

Loading all tests before running, not
removing from memory

Biggest problem...

Culture of interoperability

FIGHT!



CANONICAL