Software Testing Clinic

EurOmnis 2025

Alex Clay - Suran Systems, Inc.

About Me

- Alex Clay
- Omnis developer since 2000
- CEO of Suran since 2019
- Re-wrote our Omnis Classic apps to Studio in early 2000s
- Wrote OmnisTAP for automated software testing in 2010
- Added continuous integration and build systems
- Open-sourced OmnisTAP and OmnisCLI in 2017

About you

Software Testing

- Tool to assist with writing new code
- Express bugs and ensure they stay fixed (no regressions)
- Code the behavior of your application vs. code the implementation of your application
- Promotes clean code

Sample OmnisTAP

```
Do ioTAP.$ok(1=1,"1 equals 1”)


Do ioTAP.$is_char(low("FOO"),"foo","low() works”)


Do ioTAP.$isnotclear($libs.
 $findname("omnistap_example"),"Our library is open”)
```

Concepts

- Clean Code
- Method Types
- Test Types
- Fixtures
- Mocking
- Legacy Code

Clean Code

Clean Code

- DRY - Don’t Repeat Yourself (reduce replicated code)
- Readable methods and variables (self-documenting)
- Fail Fast/Early Return
- Max 1-2 levels of control structures (indents) in a method
  - Loop
  - If/Else/While/Repeat
- **Refactor**
- It’s all about readability and maintenance

Method Types

Method Types - Purpose

Type	Purpose	Example
Operation	Does a thing	Database update Change a window style Perform a calculation
Accessor	Get/set a state	Often the condition for an if/else Expressions for business logic
Decision	Logical flow	If/else Loops Calls operation & accessors

Method Types - Scope

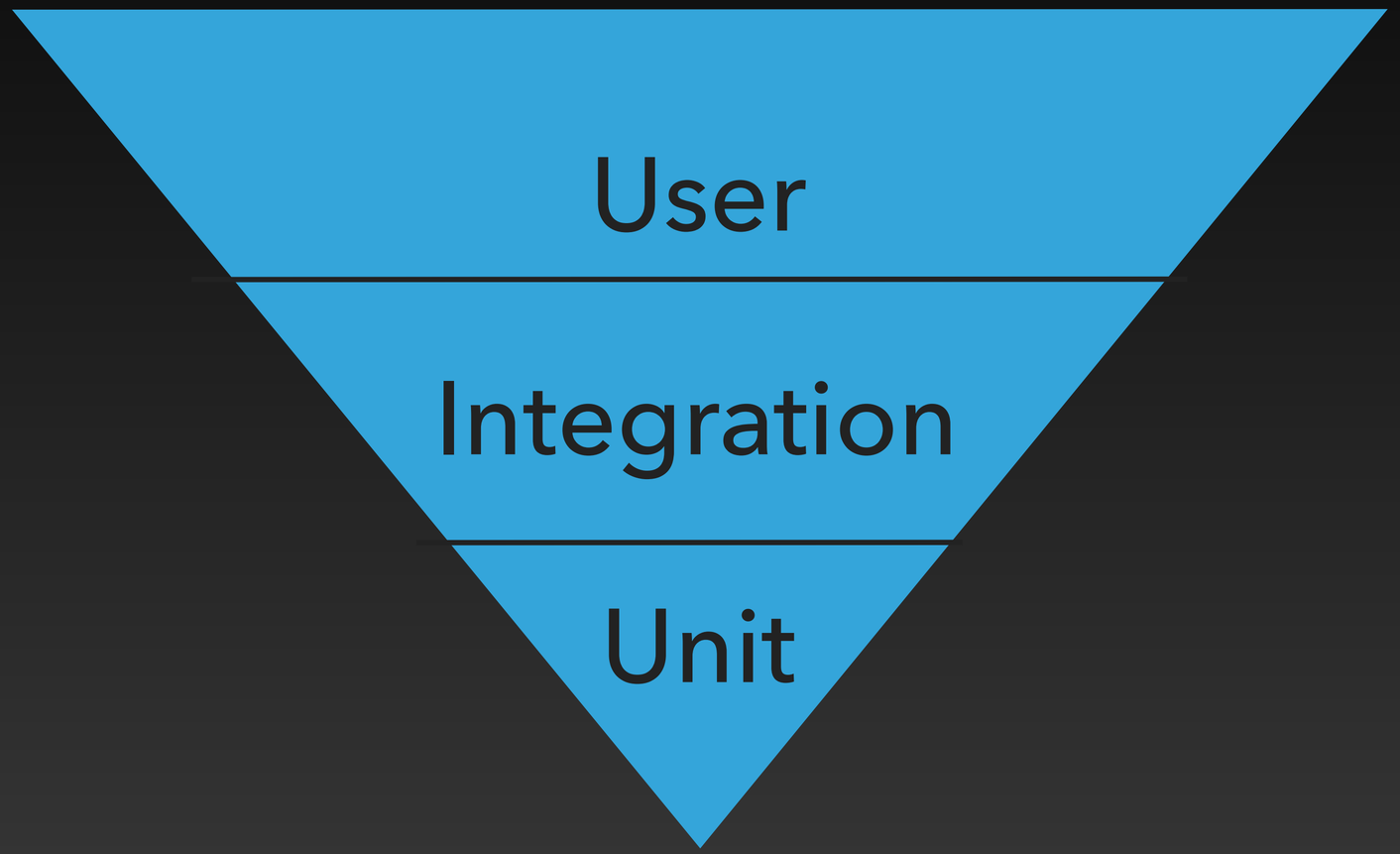
Type	Example	Use
Public	$foo	Called from any instance (testable)
Private	bar	Called from my instance only (test with forwarding method)
Protected	$_baz	Called from my instance and subclasses (and is easily testable!)

Test Types

Test Types

Unit Test	Integration Test	User Acceptance Test
Runs in < 5 seconds	Runs in < 90 seconds	How fast/thorough are you?
Tests a single method	Tests how multiple methods work together	Manual Testing
Should form the base of your test suite	Good way to get started and useful in the long run	Never replaced by automated testing

Test Types



Fixtures

$buildFullName

```
Do loPerson.$buildFullName(
   pcTitle,
   pcFirst,
   pcMiddle,
   pcLast
 ) Returns lcFullName
```

Fixtures

pcTitle	pcFirst	pcMiddle	pcLast	lcFullName
Captain	James	Tiberius	Kirk	Captain James Tiberius Kirk
	James	Tiberius	Kirk	James Tiberius Kirk
	James		Kirk	James Kirk
Captain			Kirk	Captain Kirk

Tests Using Fixtures

```
Do loPerson.$buildFullName("Captain","James","Tiberius","Kirk") Returns
  lcFullname

Do ioTAP.$is_char(lcFullname,"Captain James Tiberius Kirk")


Do loPerson.$buildFullName("","James","Tiberius","Kirk") Returns lcFullname

Do ioTAP.$is_char(lcFullname,"James Tiberius Kirk")


Do loPerson.$buildFullName("","James","","Kirk") Returns lcFullname

Do ioTAP.$is_char(lcFullname,"James Kirk")


Do loPerson.$buildFullName("Captain","","","Kirk") Returns lcFullname

Do ioTAP.$is char(lcFullname,"Captain Kirk")
```

Mocking

Mocking Method

- Replace a real method with a fake or mock method
- Required for true unit testing
- Keeps the focus on testing ONLY that method
- Assertions are on HOW you call the mocked method
  - Times
  - Params
  - Return
- RTFW - https://github.com/suransys/omnistap/wiki/Mocking

Legacy Code

Code without tests

Adding Tests to Legacy Code

- Add integration tests to create a basic harness
- Break out individual chunks of a method to separate methods
- Add unit tests to those methods
- Eventually, convert your integration test to use the mocker
- Optionally keep the integration test if it’s important

Examples