

Unit Testing in LabVIEW

Nate Moehring, LabVIEW Champion, CLA(09)

March 1, 2018



Why are you listening to this presentation?

I keep hearing about writing unit tests, how do I get started?

I love writing unit tests and want to be even better at it!

I heard there was free food

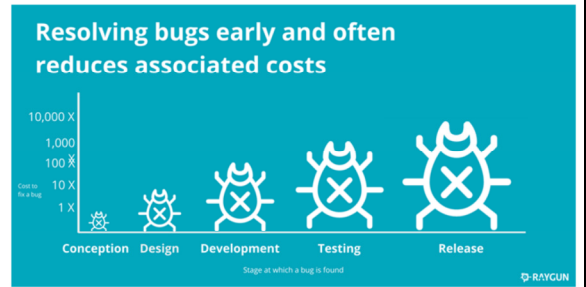
I'm waiting for the next presenter

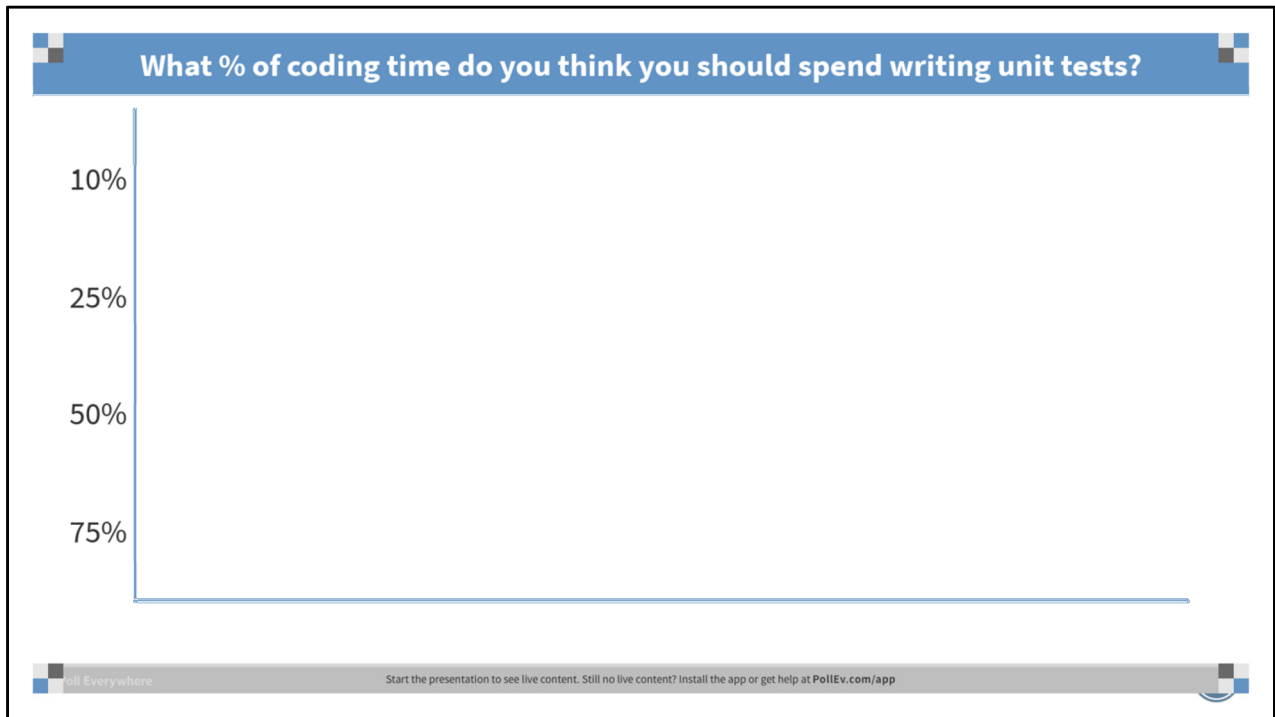
Poll Title: Why are you listening to this presentation?

https://www.polleverywhere.com/multiple_choice_polls/bz5f66Bvepg2xnW

Why Unit Test?

- Find bugs early
- Verify code meets functional requirements
- Avoid regression with later code changes
- Testable code is easier to read and more de-coupled
- Provides a form of functional examples for other developers
- [Save Time and Money with Unit Testing, Fabiola de la Cueva](#)

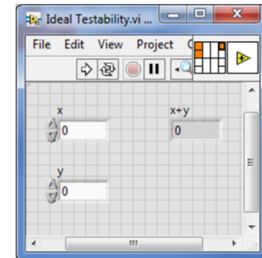




Poll Title: What % of coding time do you think you should spend writing unit tests?
https://www.polleverywhere.com/multiple_choice_polls/MFqjPe8jgCVGwco

Ideal Code for Testability

- All inputs and outputs on the connector pane
- All inputs and outputs are basic datatypes
- No use of references, dataflow FTW
- No Global variables, LV2, FGVs, or Action Engines
- No controls that are set by VI Server
- Deterministic result (not a function of time, random-ness, or other variability)
- UUT has limited responsibility



Limited Responsibility

- When we assign multiple responsibilities to a subVI, it becomes
 - Less reusable
 - Harder to test
- Parsing a config file might at first appear to be a single responsibility, but actually
 - Load the file into memory
 - Parse the file into reusable data structures
 - Validate the data structures
 - Report the results
 - Load these data structures into your application data stores

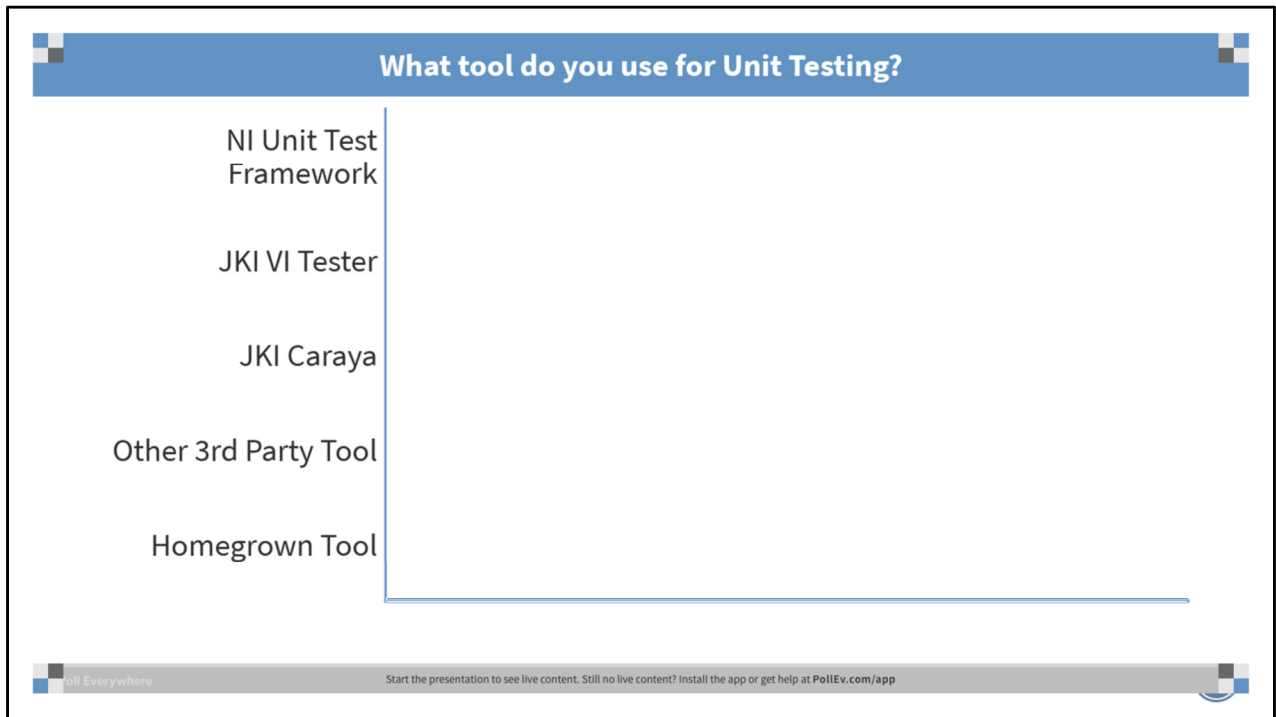


Tools for Testing

- Current Gen
 - NI Unit Test Framework
 - JKI VI Tester
 - JKI Caraya
- NXG - speculation on order of support
 - JKI Caraya
 - Composed Systems - Test Driven Development
 - JKI VI Tester
 - <Unit Test Framework, never>



According to a former employee of JKI, JKI internally prefers VI Tester but Caraya will likely be available in NXG sooner because it requires fewer language features to be available in NXG.



Poll Title: What tool do you use for Unit Testing?

https://www.polleverywhere.com/multiple_choice_polls/QsACIEhOY8jzLV0

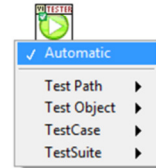
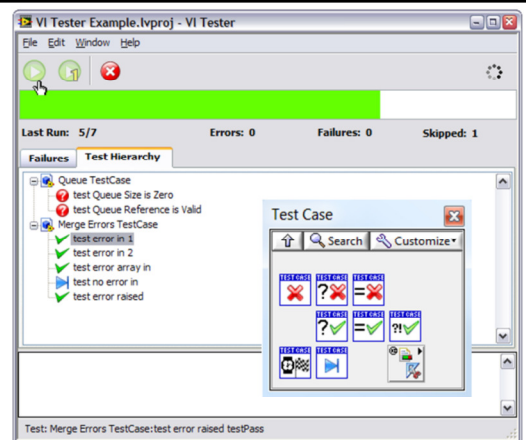
JKI VI Tester

- Write G code to test G code
- Free in VIPM
- Open source on GitHub

- Library for writing test methods
- TestCase class contains many test methods
 - **Methods must have "Test" in the filename**
- TestSuites programmatically aggregate TestSuites/TestCases for different Projects

- Run tests through API or VI Tester GUI
 - <LabVIEW>\vi.lib\Utility\LVClass\Read Class Ancestry Without Loading.vi

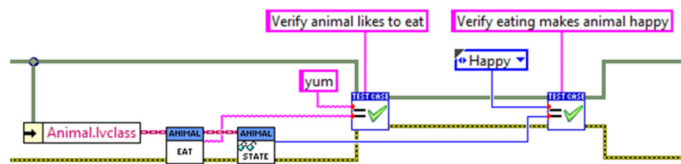
- Tools > VI Tester > New > Test Case...



Walk through the common steps associated with creating a new test.

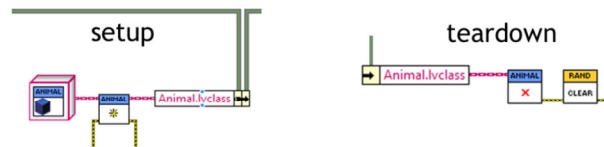
VI Tester - Reporting

- Always include a Custom Report Message with each check, to help your future-self understand the reports generated by VI Tester.
- If a Test contains multiple pass/fail checks, the report will have:
 - The first fail message OR
 - The last success message



VI Tester – Pre/Post Actions

- TestCase can optionally support setup/teardown methods to perform the same pre/post actions for each test
 - Config file parsing
 - Test Object initialization
 - Supporting references initialization
- TestSuite can optionally support New/CleanUp methods to perform the same pre/post actions for each TestCase



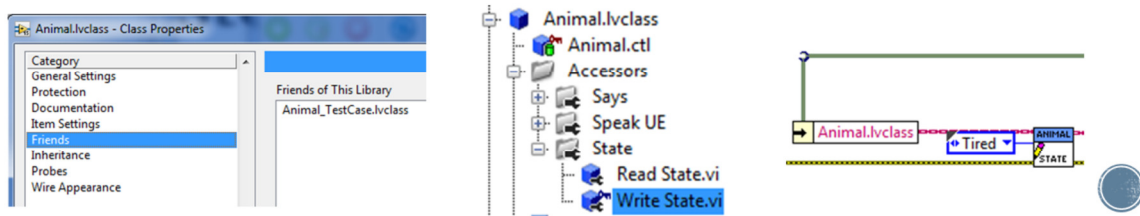
globalSetup / globalTeardown

- During the presentation LAF presentation on 3/1/2108, the question was asked if it was possible to write some code that would be executed before and after every test. Darren Nattinger commented that it is possible to override the `globalSetup` and `globalTeardown` `TestCase` methods.
- This of course implies that you are essentially creating a new `TestCase` base class that all of your actual test classes must inherit from.
- I have never attempted to use this feature and wasn't aware it existed. If you've successfully used this feature, please send me a [message](#), I'd be interested to learn more about your use case for it.



VI Tester – Friends

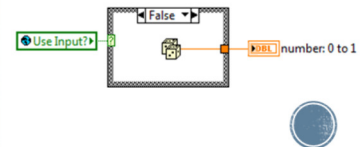
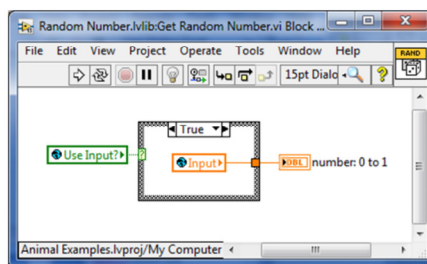
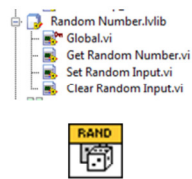
- If the UUT class declares the TestCase class as a Friend, and exposes Community methods on private data, the TestCase is now able to peek/poke private data in a test object without breaking **encapsulation** in production code.
 - Allows you to perform full white-box testing
 - Allows you to test complex corner cases
 - Allows you to bypass a lot of complex setUp logic to get the test object into the state that the test is trying to exercise
 - If using Project Templates, classes created by the Create Project Wizard could include the TestCase class as a Friend by default.



Open example Test Sleep_Not Tired.vi
Open example Test Sleep_Not Enough.vi

Stubbing

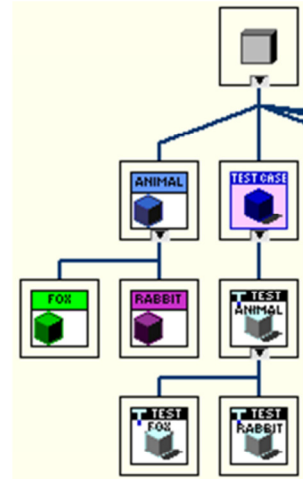
- If the UUT relies on system inputs or outputs, such as Time or Hardware I/O, stub those inputs/outputs using:
 - Dependency injection to swap in simulator methods, or
 - Case structures with global variables that statically define the various implementations



Open example Test Sleep_Enough.vi
Open teardown.vi

VI Tester - Inheritance

- Create TestCases that override specific tests of parent class
 - Fox_TestCase:Speak overrides Animal_TestCase:Speak
- If child TestCase implements setup/teardown, author must set the class terminals as Dynamic Dispatch on both child and parent.
- Create Accessors in parent TestCase to access test data



Open example Animal.lvclass:Test Speak.vi

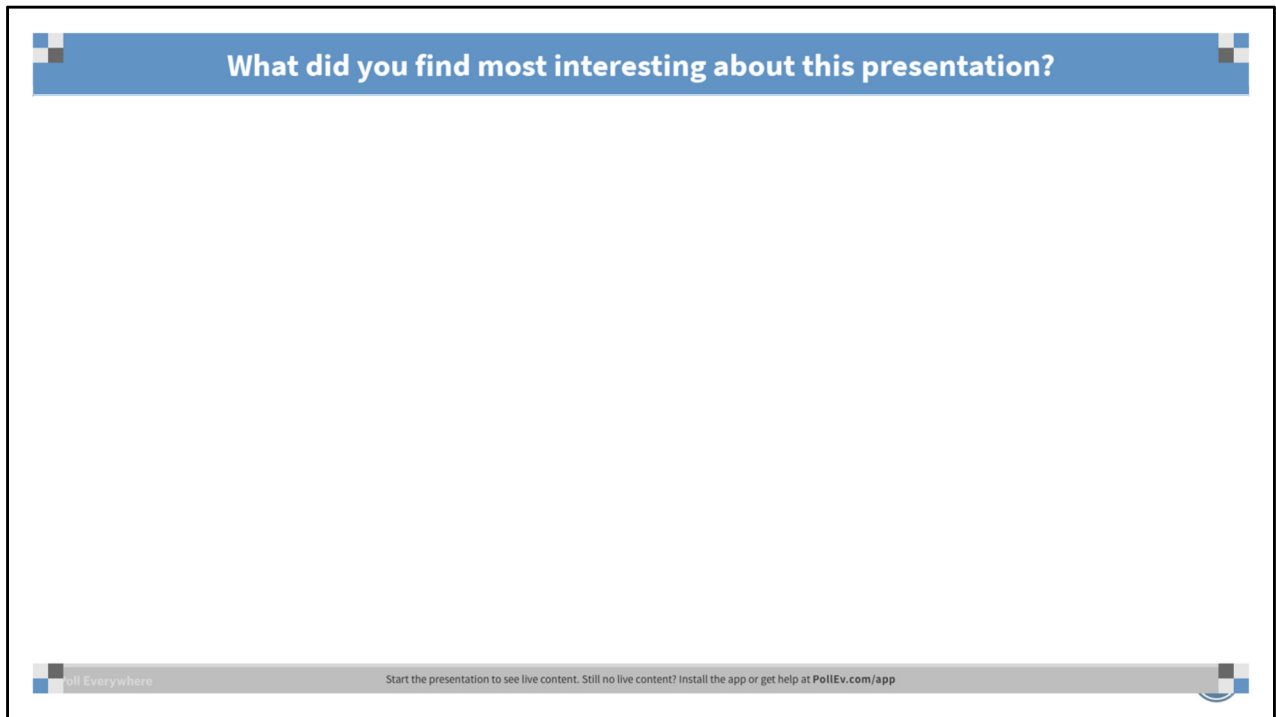
Open Rabbit.lvclass:Create.vi

Open Rabbit_TestCase.lvclass:setUp.vi

Open Rabbit_TestCase.lvclass:Test Speak.vi (tests specific behavior of child class even though child class does not override Speak.vi)

Open Fox.lvclass:Speak.vi

Open Fox_TestCase.lvclass:Test Speak.vi



Poll Title: What did you find most interesting about this presentation?
https://www.polleverywhere.com/free_text_polls/edvjll2TAx5o81m

References

- [LabVIEW Unit Testing](#), Control Software Solutions
- [Cost of Software Errors](#), Freyja Spaven
- [Save Time and Money with Unit Testing](#), Fabiola da la Cueva
- [Thoughts on Accessor Methods](#), Nate Moehring
- [Unit Testing Group](#)

