PythonUnit: a Unit Testing Framework for Python

Joel Shprentz
TRW

Abstract

I will demonstrate *PythonUnit*, a testing framework that provides common testing services and test management. It is based on Kent Beck's unit testing framework for Smalltalk.

Unit tests are tests of individual Python classes. Programmers write tests for everything that could conceivably not work. For each Python class, one usually tests public methods, important private methods, edge effects, and code that has failed before.

Each *unit test case* establishes a testing environment (by creating the necessary objects), executes some code, and compares the observed results with the expected results. Deviations are reported as failures and unexpected exceptions are reported as errors. The test environment is discarded after each test case to avoid interaction between cases.

Unit tests can be grouped into *test suites.* Typically, each Python module has an associated test suite. Test suites report summary statistics for their unit tests (e.g., 3 failures and 1 error out of 34 unit tests).

Test suites can be grouped into *test runs.* Each project has a test run that includes all test suites. Programmers may create additional test runs that include a subset of the available test suites. A test run executes its test suites, displays test results, and saves summary statistics. Depending on the programming environment, a test run may show its results in a windowed GUI or as batch output to a terminal.

Programming teams can use unit test summary statistics as project metrics. An increase in the number of error-free test suites represents progress. An increase in the number of failures or error represents trouble.

Extreme Programming particularly emphasizes unit testing, but unit testing is useful with any programming methodology. In Extreme Programming, unit tests are written before new application code. Tests are run after every code change. When the tests all pass, the code is done.

Source code and additional information is available from the first reference web site.

Reference Books:

Kent Beck, *Extreme Programming Explained: Embrace Change*, Addison-Wesley, 1999, ISBN 0201616416.

Martin Fowler, Kent Beck, John Brant, William Opdyke, and Don Roberts, *Refactoring: Improving the Design of Existing Code*, Addison-Wesley, 1999, ISBN 0201485672.

Reference Web Sites:

http://c2.com/
    cgi-bin/wiki?TestingFramework
http://c2.com/cgi-bin/wiki?UnitTests
http://www.xProgramming.com

Author Information

Joel Shprentz
TRW
One Federal Systems Park Drive
Fairfax, VA 22033
joel.shprentz@trw.com