

Debugging
sucks.



Testing rocks.

Testing on the Toilet

Extracting Methods to Simplify Testing

June 13, 2007

When a method is **long and complex**, it is **harder to test**. You can make it easier by **extracting methods**: finding pieces of code in existing, complex methods (or functions) that can be replaced with method calls (or function calls). Consider the following complicated method:

```
def GetTestResults(self):
    # Check if results have been cached.
    results = cache.get('test_results', None)
    if results is None:
        # No results in the cache, so check the database.
        results = db.FetchResults(SQL_SELECT_TEST_RESULTS)
    # Count passing and failing tests.
    num_passing = len([r for r in results if r['outcome'] == 'pass'])
    num_failing = len(results) - num_passing
    return num_passing, num_failing
```

This method is difficult to test because it not only relies on a database, but also on a cache. In addition, it performs some post processing of the retrieved results. The first hint that this method could use refactoring is the abundance of comments. Extracting sections of code into **well-named methods** reduces the original method's complexity. When complexity is reduced, **comments often become unnecessary**. For example, consider the following:

```
def GetTestResults(self):
    results = self._GetTestResultsFromCache()
    if results is None:
        results = self._GetTestResultsFromDatabase()
    return self._CountPassFail(results)

def _GetTestResultsFromCache(self):
    return cache.get('test_results', None)

def _GetTestResultsFromDatabase(self):
    return db.FetchResults(SQL_SELECT_TEST_RESULTS)

def _CountPassFail(self, results):
    num_passing = len([r for r in results if r['outcome'] == 'pass'])
    num_failing = len(results) - num_passing
    return num_passing, num_failing
```

Now, tests can focus on each individual piece of the original method by testing each extracted method. This has the added benefit of making the code **more readable** and **easier to maintain**.

(Note: Method extraction can be done for you automatically in Python by the open-source refactoring browser BicycleRepairMan, and in Java by several IDEs, including IntelliJ IDEA and Eclipse.)

More information, discussion, and archives:

<http://googletesting.blogspot.com>



Copyright © 2007 Google, Inc. Licensed under a Creative Commons
Attribution-ShareAlike 2.5 License (<http://creativecommons.org/licenses/by-sa/2.5/>).

