

Mock testing techniques

- Wikipedia:
 - Mock objects are simulated objects that mimic the behavior of real objects in controlled ways. A computer programmer typically creates a mock object to test the behavior of some other object, in much the same way that an automobile designer uses a crash test dummy to test the behavior of an automobile during an accident
-
-

Mock testing techniques (cont.)

- stubs vs. mocks
 - stubs are used to return canned data to your methods or functions under test, so that you can make some assertions on how your program reacts to that data
 - mocks are used to specify certain expectations about how the methods of the mocked object are called by your program: how many times, with how many arguments, etc.
-
-

Mock testing techniques (cont.)

- rule of thumb: use mocking at the I/O boundaries of your application
 - mock the interactions of your application with external resources that are not always under your control
 - databases and network resources such as Web servers, XML-RPC servers, etc.
 - data produced by these resources often contains some randomness that makes it hard for your tests to assert things about it
 - eliminate randomness by using canned data via stubs/mocks
 - SEE <http://agiletesting.blogspot.com/2006/12/mock-testing-examples-and-resources.html>
-
-

Web UI testing with Selenium

- functional/acceptance testing at the user interface level for Web applications
 - uses an actual browser driven via JavaScript!
 - unique features
 - client-side JavaScript testing (think AJAX)
 - browser compatibility testing (cross-platform and cross-browser)
 - needs to be deployed on the same server as the app under test for JavaScript security reasons (“same origin” policy)
-
-

Writing Selenium Core tests

- individual tests and test suites written as HTML tables
 - tests consists in actions (open, type, select, click) and assertionss
 - test recording/playback: Selenium IDE
 - uses Mozilla Chrome to get around JavaScript security
 - other useful tools: XPath Checker and XPather
Firefox extensions
-
-

Selenium RC

- Java-based server process
 - offers a way to programmatically drive a browser using the programming language of your choice
 - Java, .NET, Perl, Python, and Ruby are currently supported out of the box
 - scripts communicate with server via XML-RPC
-
-

Selenium RC (cont.)

- Selenium RC has two main advantages over Selenium Core:
 - you can write tests in a real programming language, writing your own functions, using control flow and familiar testing constructs such as asserts
 - you can test Web sites remotely, without deploying Selenium RC on the server hosting the site
-
-

Acceptance testing with FitNesse

- more user friendly variant of Ward Cunningham's FIT framework
 - "business facing" or "customer facing" tests, as opposed to "code facing" tests (e.g. unit tests)
 - tests are expressed as stories ("storytests")
 - higher level than unit tests
 - FitNesse tests make sure you "write the right code"
 - unit tests make sure you "write the code right"
-
-

Acceptance testing with FitNesse

(cont.)

- wiki format encourages collaboration
- Fit/FitNesse brings together business customers, developers and testers, and it forces them to focus on the core business rules of the application.
- James Shore: "Done right, FIT fades into the background"

Writing FitNesse tests

- PyFIT is Python port of FIT/FitNesse
 - tests are written in tabular format, with inputs and expected outputs
 - fixtures are a thin layer of glue code that tie the test tables to the application
 - tests can be executed from both the wiki and the command line.
-
-