



Types of test tools

Luke Worth

Unit testing

- xUnit
- Advantage: Fast and thorough
- Disadvantage: Doesn't test whole system

Functional GUI testing

- TestPartner, QuickTest, SilkTest, Rational Functional Tester, Quidancer, HtmlUnit
- Advantages: repeatable, easy, available
- Disadvantages: Doesn't test internals

Load testing

- LoadRunner
- Advantage: Test load before release
- Disadvantage: Really, really expensive

Fault injection

- Exhaustif
- Advantages: Test exception handling, etc
- Disadvantages: Impossible situations

Model-based testing

- LEIROS Test Designer, AETG
- Advantages: quickly produce tests, tests update automatically with spec.
- Disadvantages: Code might not be the same as spec, only black-box.

A spiral-bound notebook with a white page. The spiral binding is at the top. The text "Testing your tests" is written in the center of the page.

Testing your tests

Code coverage

- Emma, Clover
- Advantages: understandable, easy to deploy, sophisticated tools
- Disadvantages: no knowledge of application domain

Functional coverage

- FOCUS
- Advantage: specific coverage models
- Disadvantage: specific coverage models!

Mutation testing

- MuJava, Jumble
- Advantages: general tools available
- Disadvantages: mainly academic interest
- Warning: Insure++ does not implement mutation testing!



Internal development of test tools

Kishen Rajan

- Motivation for developing internal tools
 - Need to test customised or specific software where commercial tools might be limited
 - Your own test tool might increase test coverage
 - Customised test tools that improve efficiency for your test team

□ Time and resources

□ How long is it going to take?

□ Resources?

- Maintenance cost and service support
 - On-going support
 - Quality of service
 - Repository of known issues
 - Online help

□ Proving and verifying your test tool

□ How do you verify your test tool?

□ Extensive testing? You might use commercial test tools to test your test tool.

□ When do you stop testing the test tool?

□ Formal methods?

- Knowledge limited to in-house testers and developers
- Train new staff
- What happens when there is high-turnover?
 - Need a process to train new testers
 - Need manuals and documentation that constantly gets updated with the test tools

□ Obsolescence

□ If the underlying system changes...

□ Highly specialised tools become obsolete quicker

□ Complex

□ Less flexible

- Gathering requirements for your new test tool
- Sure there is better communication because it's in-house, but different groups have different feature request.
- Who is going to provide the requirements?

Summary

- There are many testing tools.
- Do your research on existing test tools before building your own!