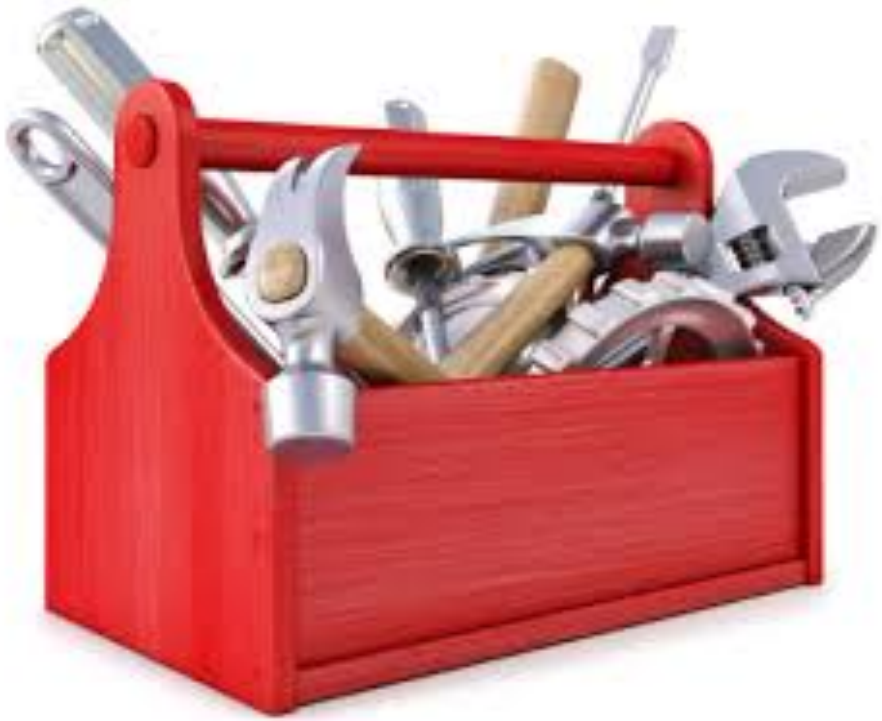
Case Studies of Free Test Tools

Successful Test Tool Use without a Big Budget



RBCS

TIME TESTED.
TESTING IMPROVED.

www.RBCS-US.com

Introduction

- Tools are great…except when they're not
- Gee-whiz tools often have gee-whiz price tags to go with them
- Okay, free tools are great…except when they're not
- Tools, free or pricey, can waste your time if you misuse them
- As a tester and/or developer, you should know your options so you can use them
- As a manager, you should know your team's options so you can help to choose the right ones
- Let's look at some examples of free tool use—successful and unsuccessful…

GUI Test Automation

- There are open-source tools for GUI test automation
- Selenium is the one most commonly mentioned
- Keyword-driven architecture is essential for maintainability
- Otherwise, you'll hit the same problems that occur with commercial tools
- Using an experienced lead is necessary for good architecture

Performance Testing

- JMeter (part of the Apache project) is the most commonly encountered
- OpenSTA has some users
- Maintenance of tests is not such an issue
- Expertise with performance is critical, and trying to use these tools without such skills will result in misleading results
- Some previously open-source performance testing tools have gone away

Web Services Testing

- Here options are more limited
- SoapUI is the free tool most often mentioned
  - SoapUI, while workable, is not a very reliable tool
  - SoapUI is the open-source variant of a commercial offering (with similar problems)
- TestMaker and WebInject show up in web searches, but none of our clients have mentioned them

Dynamic Analysis

- Most operating systems have some built-in options (e.g., top, perfmon)
- Additional open-source tools are also available
  - Valgrind in Linux/Android/Mac world
  - Winleak for PCs
- Valgrind works well, but it does consume a lot of resources

Functional Verification Testing

- Many tools for these activities
  - Fitnesse used commonly for ATDD
  - Cucumber family used commonly for BDD
- These tools are not limited to agile development, but came from that world
- Developers and testers use these tools collaboratively
- Business stakeholders ideally review tests as well
- Often (but not always) included in continuous integration frameworks

Continuous Integration

- Present in all our clients with successful agile implementations
- Most use either Jenkins or Hudson
- Most include some additional tools for unit testing, etc. (more later)
- Supports testing by making the build process faster, more reliable
- Can create problems when builds are auto-deployed to test environments

Unit Testing

- Most of our clients using continuous integration extend it with unit testing
- The xUnit family of tools is most typically used (Cpp-Unit, J-Unit, etc.)
- Some report developers using it for TDD
- Code coverage analysis tools (e.g., gcov) are often used in conjunction with these tools
- Many developers don't have training in proper test design, so these tests are less useful than they could be
- Testers should learn the basics of unit testing, white-box test design, etc., and help developers get more value

Static Code Analysis

- Some of our clients extend continuous integration with static code analysis
- Sonar and splint are typical open-source options, though many clients opt for commercial static tools
- Open-source complexity analysis tools are available (e.g., pmccabe)
- To understand the output of these tools, knowledge of the programming language is required
- These tools are not a replacement for proper code reviews

Test Design Tools

- Certain types of tests require tools for design
- Pairwise testing is one good example
- PICT and ACTS are good free tools for pairwise testing
- Model-based testing gets talked about a lot, but few of our clients use it
- We have built model-based test systems for some of our clients, using open-source tools

Scripting Tools

- Many testers automate tests using scripting tools
- Ruby, Python, Tcl/Tk, and Unix shells are in common use
- If anything, there are too many different options, leading to Tower of Babel problem
- Scripting is best done by people with some programming skill and knowledge
- However, testers can learn or even teach themselves
- Care must be taken to avoid maintainability problems
- Very large, complex, sophisticated test systems can be built

Conclusions

- We've looked at ten areas where open-source tools can support testing by testers, developers, or both
- Managers should remember that free to download doesn't mean free to use
  - Direct costs of people's time
  - Opportunity costs (what could be done)
- Many of our clients are experiencing good success with these open-source tools
- Support options can be limited, which is a consideration for less-technical teams

To Contact RBCS

For over twenty years, RBCS has delivered consulting, outsourcing and training services to clients, helping them with software and hardware testing. Employing the industry's most experienced and recognized consultants, RBCS advises its clients, trains their employees, conducts product testing, builds and improves testing groups, and hires testing staff for hundreds of clients worldwide. Ranging from Fortune 20 companies to start-ups, RBCS clients save time and money through improved product development, decreased tech support calls, improved corporate reputation and more. To learn more about RBCS, visit www.rbcs-us.com.

Address:	RBCS, Inc. 31520 Beck Road Bulverde, TX 78163-3911 USA
Phone:	+1 (830) 438-4830
E-mail:	info@rbcs-us.com
Web:	www.rbcs-us.com
Twitter:	@RBCS, @LaikaTestDog
Facebook:	RBCS-Inc.