Open Source vs. Commercial Tools in Software Testing

Mag. Rudolf Ramler
Software Competence Center Hagenberg

Dipl.-Ing. Dr. Alexander Schatten
Institut für Software Technologie, Technische Universität Wien

Gartner on The State of Open Source

Gartner asked more than 50 analysts to comment on the state of Open Source in their areas of research. A few key findings ....

- By 2012, more than 90% of enterprises will use Open Source in direct or embedded forms
- By 2011, Open Source will dominate software infrastructure for cloud-based providers
- By 2012, software as a service (SaaS) will eclipse Open Source as the preferred enterprise IT cost-cutting method

- Not understanding the essentials and the essential impact of Open Source software is not understanding today’s software landscape.

Source: Gartner Research: The State of Open Source, April 2008
Survey: Do you use Open Source Software?

- Software as Commodity
- Effects of Open Source on Testing Tools
- Case Studies: Popular Open Source Testing Tools
- Strategies for Adopting Open Source Tools

Survey

- Do you use open source software?
- Do you use …
  - Firefox / Mozilla to browse the Web?
  - OpenOffice / StarOffice to write documents?
  - Linux as server?
  - Apache / Tomcat on your server?
  - Subversion / CVS for your source code?
  - MySQL / Firebird to store your data?
  - Eclipse / NetBeans to write code?
  - Google to search the Web?
  - Amazon to shop?
### Studies – Why OSS?

#### “Please rate the extent to which you agree or disagree with the statement that open source software can help your company achieve the following business goals.”

(N = 480)

<table>
<thead>
<tr>
<th>Objective</th>
<th>Strongly Agree</th>
<th>Agree</th>
<th>Neutral</th>
<th>Strongly Disagree</th>
<th>Don’t Know</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lowering the company’s overall operating costs</td>
<td>16%</td>
<td>20%</td>
<td>25%</td>
<td>39%</td>
<td>9%</td>
</tr>
<tr>
<td>Improving quality of products and/or processes</td>
<td>12%</td>
<td>37%</td>
<td>16%</td>
<td>24%</td>
<td>15%</td>
</tr>
<tr>
<td>Achieving faster time to market</td>
<td>27%</td>
<td>28%</td>
<td>24%</td>
<td>15%</td>
<td>6%</td>
</tr>
<tr>
<td>Driving innovative new market offerings or business practices</td>
<td>14%</td>
<td>31%</td>
<td>16%</td>
<td>22%</td>
<td>12%</td>
</tr>
<tr>
<td>Acquiring and retaining customers</td>
<td>33%</td>
<td>24%</td>
<td>20%</td>
<td>18%</td>
<td>11%</td>
</tr>
<tr>
<td>Managing our customer relationships</td>
<td>31%</td>
<td>37%</td>
<td>21%</td>
<td>14%</td>
<td>9%</td>
</tr>
<tr>
<td>Re-engineering core business processes</td>
<td>19%</td>
<td>36%</td>
<td>35%</td>
<td>8%</td>
<td>1%</td>
</tr>
</tbody>
</table>


### Studies – Why *not* OSS?

#### “What concerns do you continue to have about using open source?”

(asked to respondents currently using open source; N = 199)

<table>
<thead>
<tr>
<th>Concern</th>
<th>No (%)</th>
<th>Yes (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability of service and support</td>
<td>65%</td>
<td>35%</td>
</tr>
<tr>
<td>Security of the software</td>
<td>43%</td>
<td>57%</td>
</tr>
<tr>
<td>Lack of relevant internal skills in development or operations</td>
<td>46%</td>
<td>54%</td>
</tr>
<tr>
<td>Product immaturity</td>
<td>40%</td>
<td>60%</td>
</tr>
<tr>
<td>Overall complexity and difficulty of adoption</td>
<td>55%</td>
<td>45%</td>
</tr>
<tr>
<td>Ability to meet business goals</td>
<td>55%</td>
<td>45%</td>
</tr>
<tr>
<td>Viability of the open source communities</td>
<td>56%</td>
<td>44%</td>
</tr>
<tr>
<td>Legal issues involving intellectual property</td>
<td>57%</td>
<td>43%</td>
</tr>
<tr>
<td>Total cost of ownership</td>
<td>65%</td>
<td>35%</td>
</tr>
</tbody>
</table>

What is Open Source?

The Open Source Definition (Open Source Initiative)

Open source doesn’t just mean access to the source code. The distribution terms of open-source software must comply with the following criteria:

- Free Redistribution
- Source Code
- Derived Works
- Integrity of The Author’s Source Code
- No Discrimination Against Persons or Groups
- No Discrimination Against Fields of Endeavor
- Distribution of License
- License Must Not Be Specific to a Product
- License Must Not Restrict Other Software
- License Must Be Technology-Neutral

Source: The Open Source Definition, Open Source Initiative, 07/2006 (http://opensource.org/docs/osd)

Content

- Survey: Do you use Open Source Software?
- Software as Commodity
- Effects of Open Source on Testing Tools
- Case Studies: Popular Open Source Testing Tools
- Strategies for Adopting Open Source Tools
What is a commodity?

“A useful or valuable thing, such as water and time”
(Apple Dictionary)

Is there “Commodified” Software?

- Post-PC era (ubiquitous Software)
- Standard-based approach for data exchange
- Community driven processes/involvement
  - Very similar concepts to Open Source process
  - Communication driven Systems (Internet, mobile phones)
- Open Protocols are the enabling technologies for next-generation of software (as commodity)
- But: Software as commodity is also exchangeable (e.g., Firefox, IE),
  - Hence real competition can appear!
  - New players have the chance to participate!
First Steps in Paradigm Shift

- Producer / Consumer
  - Replaced by Interaction Models
- Platforms and Processes, **Standard driven Services** instead of products!
  - Standard driven services like
    - Email
    - RSS: Aggregation, Podcasts, Photocasts, ...
  - Platforms like
    - Wikipedia
    - eBay
    - Second Life
    - Video Game Platforms
- Example: “Web 2.0”
  - Data and
  - Interaction drive
  - Example?
    - Digg it, del.icio.us, reddit, Facebook, Twitter

Conventional Software-Development Process

- Sourcecode
- Compiler...
- Tools, Libraries, etc.
- Documentation, ...
- Company Internal
- Binary Program Ready for shipping
- Marketing, Sales
- Customer
Conventional Process Interaction

Assumption of Needs

Developers might use their own Product but this is not to motivation for development! Development is driven by marketing or external needs.

Typically not much feedback from customers; probably Bug reports, customisation requests, but customer feels as customer and not as part of the production process

High dependency on producer by lock-in mechanisms like closed protocols, contracts, lack of openness for collective problem solving through transparency and a flexible process

Command & Control

Development Process –
Open Source Project

First obvious observation: complete process is public (including all artefacts like protocols)

But this is still Producer/Consumer Driven perspective
Open Development Process: Interaction

Developers use their Product and their own feedback drives the initial development phase. (first order feedback)

“Second order” feedback by other users of the system/product: not “only” usage: Interaction drives development, support, add-ons, plug-ins, data...

Users feel connected to the project and more and more also as part of the development process: Increased value through flexible process; Enabler for collective problem solving

“Darwinism” and Software Quality

- Open Source Tools base on Open Standards
- Change between tools is rather easily possible
- Developers typically evaluate a broad range of development tools
- Developers are tech-savvy, i.e. quickly adopt new practices and tools
- Open Source licenses allow forking and “learning” from available sourcecode
- Hence: Fierce competition, adaptation, selection in the tool – ecosystem
The “new” Software Development Process

- No Rollouts, no shrink-wrapped Software packages
- Software was "artefact"
- now it becomes a "Process" using similar strategies like OS processes, even in commercial setup
- perpetual beta (Tim O’Reilly)
- becomes commodity
- Finally, abstraction from Software
  - Services
  - Ontologies and Service Level Agreements
  - Towards self-organising Systems?
  - Multiple devices
  - Multiple access strategies
- Who is inside and who is outside of value-chain?

Content

- Survey: Do you use Open Source Software?
- Software as Commodity
- Effects of Open Source on Testing Tools
- Case Studies: Popular Open Source Testing Tools
- Strategies for Adopting Open Source Tools
Software Testing Today

- 80% of the effort that testers and developers spend today is on making testing possible, and only 20% focuses on making testing meaningful.

- 80% of the effort tool vendors spend today duplicates the work of others, recreating an infrastructure to enable testing and debugging activities. Only 20% of their work produces new function that’s visible and valuable to testers and developers.


Effects of OSS on Test Tools

- Test tools are already a commodity
  - "Integrated" support for (developer) testing, distributed work, …
  - Standard solutions "out of the box"
- User orientation and competitive open source development
  - Simple, focused solutions at reasonable cost (value is balanced with cost of usage)
  - If users are developers → prerequisites, yet many good tools for developer testing
- Open solutions
  - Open "standards" (e.g., commonly used (scripting) languages) – less vendor lock-ins
  - Basis for integration across the development lifecycle
- Always beta principle
  - New, rapid "acquisition" processes of new tools → Prototyping of solutions
  - Early support for emerging technologies and agile methods
  - Always on the move: Juggling with versions
- Software as a Service
  - Hosted services and support for virtualization (e.g. VMWare Compliances)
  - Support for outsourcing of testing tasks
Survey: Do you use Open Source Software?
Software as Commodity
Effects of Open Source on Testing Tools
Case Studies: Popular Open Source Testing Tools
- JUnit
- Checkstyle & Findbugs
- Bugzilla (+ Testopia)
- Selenium
- JMeter
- Eclipse TPTP
- Continuous Integration with Maven, Continuum & Co
Strategies for Adopting Open Source Tools

JUnit – Key Facts
- **JUnit** is a simple, but very effective (Java) framework for Unit-Testing and well integrated into all major IDEs and build automation tools.
- Type: unit testing framework
- License: Common Public License
- URL: http://www.junit.org
- Since: 2000 (current version: 4.6)
- Host: Sourceforge
- Community: Very large; de facto standard in practically every project
- Key contributors: Kent Beck, Erich Gamma, David Saff
- Related OSS
  - comparable tools are available for practically all other platforms
  - Range of Add-Ons are available, e.g. for DB testing, Web testing
JUnit SourceForge Downloads

Never in the field of software development was so much owned by so many to so few lines of code. – Martin Fowler

Checkstyle & Findbugs – Key Facts

- Findbugs
  - Hosted on Sourceforge under GPL license
  - Originally University project
  - Analyses Java code and checks for “suspicious” code

- Checkstyle
  - Hosted on Sourceforge under GPL license
  - Started as tool to check coding guidelines
  - Meanwhile checks sourcecode for a large number of potential issues like
    - Duplicate code
    - Missing documentation
    - Algorithmic complexity
    - and much more…
Bugzilla (+Testopia) – Key Facts

- **Bugzilla** is a bug tracking system that allows developers to keep track of outstanding bugs in their product.
- **Testopia** is a test case management extension for Bugzilla. It is a generic tool for tracking test cases, allowing for testing organizations to integrate bug reporting with their test case run results.

- Type: Bug tracking (+ test management)
- License: Mozilla Public License (MPL) 1.1
- URL: http://www.bugzilla.org
- Since: 1998 (current version: 3.2.x)
- Host: Mozilla Foundation
- Community: Very large
- Commercial support: available
  - Several consultants (7 Germany)
  - Commercial add-ons (e.g. ALMWorks)
- Related OSS
  - Mantis, Scarab, Trac, ...

Selenium – Key Facts

- **Selenium** is a suite of tools to automate web application testing across many browsers and operating systems. The tests can be recorded or written in a number of popular programming languages and runs directly via the browser.

- Type: Functional testing (for web app)
- License: Apache License 2.0
- URL: http://seleniumhq.org
- Since: 2004 (current version: 1.0 beta-2)
- Host: OpenQA
- Community: Active
  - Key contributors: ThoughWorks, Google
- Commercial support and add-ons:
  - ThoughWorks, PushToTest, ...
- Related OSS
  - Watir, Canoo WebTest, HtmlUnit
JMeter – Key Facts

- **Apache JMeter** is a load testing tool for analyzing and measuring the performance of web applications and a variety of services such as JDBC, FTP, LDAP.

- Type: Load testing (for web app)
- License: Apache License 2.0
- Since: Dec 1998 (current version: 2.3.2)
- Host: Apache Software Foundation
- Community: Very Large
  - Key contributors: Individual
- Commercial support and add-ons:
  - Available
- Related OSS
  - Grinder, OpenSTA

Eclipse TPTP – Key Facts

- The **Eclipse Test and Performance Tools Platform** (TPTP) provides an open platform supplying frameworks and services that allow software developers to build test and performance tools, both open source and commercial, that can be easily integrated with the platform and with other tools.

- Type: Plattform (for testing and performance analysis)
- License: Eclipse Public License 1.0
- URL: [http://www.eclipse.org/tptp](http://www.eclipse.org/tptp)
- Since: 2002 (former Hyades) (current: 4.5.2)
- Host: Eclipse Foundation
- Community: active (but decreasing)
- Key contributors:
  - IBM (Intel, Scapa Technologies, …)
- Commercial support and add-ons: ?
- Related OSS: ?
Maven/Continuous Int.– Key Facts

- Maven is an Apache project and is considered to be the leading build-automation tool.
- Automation is a requirement for test-driven, group- and particularly distributed development.
- Maven reflects the build lifecycle of an application starting from preprocessing, code-generation over compilation, test-execution, reporting to deployment.
- Various continuous integration tools like Continuum or Hudson build upon tools like Maven.
- Continuous integration and build automation are a foundation for stable software development and integrate many of the (test) tools described in this presentation.

Content

- Survey: Do you use Open Source Software?
- Software as Commodity
- Effects of Open Source on Testing Tools
- Case Studies: Popular Open Source Testing Tools
- Strategies for Adopting Open Source Tools
Open-Source Test Tool Strategy

<table>
<thead>
<tr>
<th>Size of Organizational Unit</th>
<th>Large (&gt;50)</th>
<th>Medium (10-50)</th>
<th>Small (1–10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select (commercial) solutions and tools leveraged by OSS</td>
<td>Add OSS to managed tool landscape, e.g., as point solution, new tech. Sponsor OSS solution</td>
<td>Prototype and build solutions on basis of OSS Release in-house solutions as OSS</td>
<td></td>
</tr>
<tr>
<td>Sponsor OSS solution</td>
<td>Use OSS tools, replace in-house solutions</td>
<td>Evaluate OSS tools as alternative to commercial tools Prototype with OSS</td>
<td></td>
</tr>
<tr>
<td>Use OSS tools,</td>
<td>Prototype solutions on basis of OSS Extend OSS solutions and contribute to OSS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adopt OSS development process and tools</td>
<td>Use OSS tools</td>
<td>Avoid challenge (or extend OSS solutions)</td>
<td></td>
</tr>
</tbody>
</table>

List of Open Source Testing Tools

- **Opensourcetesting.org**
  - Mark Aberdour, regular updates (since 2003)
  - [http://www.opensourcetesting.org/](http://www.opensourcetesting.org/)

- Open Source Testing Tools in Java
  - Fernando Javier Rodriguez Olivera, regular updates
  - [http://java-source.net/open-source/testing-tools](http://java-source.net/open-source/testing-tools)

- Open Source Automated Test Tools Written in Java (blog)
  - Carlos E. Perez, 2007

- Open Testware Reviews (deprecated)
  - Danny R. Faught, 2003 to 2005
Information about OSS

- **Ohloh** aims to map the landscape of open source software development.
  - Founded by Jason Allen and Scott Collison in 2004
  - October 2008: 19,500+ projects listed
  - http://www.ohloh.net/
- Based on data from software repositories (CVS, SVN, ...)
- Statistics about
  - longevity and activity of projects
  - licenses (including license conflicts)
  - programming languages
  - software metrics (LOC) and commit statistics
  - contributors

Contacts

Mag. Rudolf Ramler
Software Competence Center Hagenberg
Softwarepark 21
A-4232 Hagenberg
+43 (7236) 3343 - 872
rudolf.ramler@scch.at

Dipl.-Ing. Dr. Alexander Schatten
Institut für Software Technologie, Technische Universität Wien
Favoritenstraße 9-11/188
A-1040 Vienna
+43 (1) 588 01 - 18840
schatten@ifs.tuwien.ac.at
**References 1/2**

- Carlo Daffara (Ed.): *Free/Libre Open Source Software: a guide for SMEs*, 3rd Ed. FLOSSMETRICS and OpenTTT EU projects, 10/2008 (http://guide.conecta.it)
- Oliver Diedrich: *Trendstudie Open Source* - Wie Open-Source-Software in Deutschland eingesetzt wird. heise open, 02/2009 (http://www.heise.de/open/artikel/126682)
- Danny R. Faught, Alan Richardson: *Being Resourceful When Your Hands Are Tied*. StickyMinds, 2005
- Forrester Research: *Open Source Software’s Expanding Role in the Enterprise - Companies Adopt Open Source as Standard*. March 2007 (Study Commissioned by Unisys Corporation)

**References 2/2**