OpenSourceWatershed

"watching software flow downstream"

http://oswatershed.org

Scott Shawcroft
June 4, 2009
Background

Upstream

- Releases
- Features
- Fixes

Downstream

- Patches
- Feedback
Upstream software is better.

How can distribution developers work closer with upstream?

How can upstream developers work closer with downstream?
Overview
Data Gathered

Gather release information from upstream and downstream.

- Name
- Version
- Date
- Revision
- Extra
### Results ~ Python

[http://oswatershed.org/pkg/python](http://oswatershed.org/pkg/python)

**Watching software flow downstream**

<table>
<thead>
<tr>
<th>Distro</th>
<th>Branch</th>
<th>Current</th>
<th>Lag</th>
<th>MinLag</th>
<th>MeanLag</th>
<th>MaxLag</th>
<th>Total Releases</th>
<th>MeanRevision</th>
</tr>
</thead>
<tbody>
<tr>
<td>arch</td>
<td>current</td>
<td>2.6.1</td>
<td>9w</td>
<td>1d</td>
<td>1w</td>
<td>3w</td>
<td>2</td>
<td>1.00</td>
</tr>
<tr>
<td>arch</td>
<td>future</td>
<td>2.6.2</td>
<td>✔</td>
<td>3d</td>
<td>3d</td>
<td>3d</td>
<td>1</td>
<td>1.00</td>
</tr>
<tr>
<td>debian</td>
<td>current</td>
<td>2.5.2</td>
<td>59w</td>
<td>33w</td>
<td>33w</td>
<td>33w</td>
<td>1</td>
<td>2.00</td>
</tr>
<tr>
<td>debina</td>
<td>future</td>
<td>2.5.4</td>
<td>9w</td>
<td>21w</td>
<td>33w</td>
<td>2</td>
<td>1.50</td>
<td></td>
</tr>
<tr>
<td>fedora</td>
<td>current</td>
<td>2.5.1</td>
<td>85w</td>
<td>51w</td>
<td>51w</td>
<td>51w</td>
<td>2</td>
<td>1.00</td>
</tr>
<tr>
<td>fedora</td>
<td>future</td>
<td>2.6</td>
<td>24w</td>
<td>19w</td>
<td>31w</td>
<td>2</td>
<td>3.00</td>
<td></td>
</tr>
<tr>
<td>gentoo</td>
<td>current</td>
<td>2.5.4</td>
<td>9w</td>
<td>17w</td>
<td>26w</td>
<td>35w</td>
<td>2</td>
<td>1.00</td>
</tr>
<tr>
<td>gentoo</td>
<td>future</td>
<td>2.6.2</td>
<td>✔</td>
<td>3d</td>
<td>8w</td>
<td>35w</td>
<td>5</td>
<td>1.80</td>
</tr>
<tr>
<td>opensuse</td>
<td>current</td>
<td>2.6.0</td>
<td>9w</td>
<td>15w</td>
<td>15w</td>
<td>15w</td>
<td>2</td>
<td>1.50</td>
</tr>
<tr>
<td>opensuse</td>
<td>future</td>
<td>2.6.0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>0.00</td>
</tr>
<tr>
<td>opensuse</td>
<td>experimental</td>
<td>2.6.0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>0.00</td>
</tr>
<tr>
<td>sabayon</td>
<td>current</td>
<td>2.5.2</td>
<td>59w</td>
<td>39w</td>
<td>64w</td>
<td>105w</td>
<td>4</td>
<td>1.00</td>
</tr>
<tr>
<td>sabayon</td>
<td>future</td>
<td>2.4.4-r15</td>
<td>59w</td>
<td>32w</td>
<td>69w</td>
<td>105w</td>
<td>3</td>
<td>1.33</td>
</tr>
<tr>
<td>slackware</td>
<td>current</td>
<td>2.5.2</td>
<td>59w</td>
<td>9w</td>
<td>9w</td>
<td>9w</td>
<td>1</td>
<td>3.00</td>
</tr>
<tr>
<td>slackware</td>
<td>future</td>
<td>2.5.2</td>
<td>59w</td>
<td>33w</td>
<td>33w</td>
<td>33w</td>
<td>1</td>
<td>1.00</td>
</tr>
<tr>
<td>ubuntu</td>
<td>current</td>
<td>2.6.2</td>
<td>✔</td>
<td>1w</td>
<td>15w</td>
<td>30w</td>
<td>2</td>
<td>1.50</td>
</tr>
<tr>
<td>ubuntu</td>
<td>future</td>
<td>2.6.2</td>
<td>✔</td>
<td>4d</td>
<td>13w</td>
<td>34w</td>
<td>4</td>
<td>3.25</td>
</tr>
</tbody>
</table>

### History

- 2009-04-15 02:05:00 - 2.6.2
- 2009-04-08 04:48:00 - 2.6.2c1
- 2009-04-05 00:53:00 - 3.1a2
- 2009-03-07 19:27:00 - 3.1a1
- 2009-02-14 01:10:00 - 3.0.1

©2009 Scott Shawcroft | CC-by 3.0 US - contact - git - trac
Picking Packages for Analysis

• Real complete upstream data

• Present in every distribution

* Analysis in this presentation done with 137 “good” packages unless otherwise stated.
Upstream Release Trends

- Histogram of upstream releases per day.
Downstream Obsoletion

- Fraction of obsolete downstream versions.
Downstream Newer Versions

- Average number of newer versions per distro pkg.
Downstream Lag

• Average lag, “time since oldest new release”.
Ubuntu vs. Gentoo

- Debian based
- Binary Distribution
- dpkg
- 6 Month Release Cycle
- Started October 20, 2004

- Source Distribution
- Portage
- Continuous Release Cycle
- Started March 31, 2002
Ubuntu vs. Gentoo
Future Work

More of the same.

Increase the pool of “good” packages.

Create more methods of analysis.

Gather more data.

Gather data from more sources.
Goals

• Find and establish best packaging practices, upstream and downstream.

• Central source for distro package status.

• Aid in distro selection.

• Reduce lag, focus development upstream.

Keep distros fresh, with newer features and greater security.
Links

- scott.shawcroft @ gmail.com
- oswatershed.org
- trac.oswatershed.org
- github.com/tannewt/open-source-watershed

OpenSourceWatershed

"watching software flow downstream"