Lucas Nussbaum
lucas@debian.org
Debian

Well known and respected for:

▶ Its long and bumpy release cycles
▶ Its quality

But it was not a given:

▶ >1000 Debian Developers
  ▶ Mostly volunteers, with their own agendas
  ▶ >15000 software packages
  ▶ Some very popular packages
  ▶ A lot of niche packages
Debian

Well known and respected for:

- Its long and bumpy release cycles
Debian

Well known and respected for:

▶ Its long and bumpy release cycles
▶ **Its quality**

But it was not a given:

▶ > 1000 Debian Developers
  ▶ Mostly volunteers, with their own agendas
▶ 15000 software packages
  ▶ Some very popular packages
  ▶ A lot of niche packages
Quality factors

- Culture: "We release when it's ready"
- Many DDs won't compromise on this
- Packages ownership: individuals or teams
- About 600 active developers
- Many of them experts of their packages
- Release Critical (RC) bugs
- Prevent packages from being part of a release
- Everybody is welcomed to fix them
- Non-Maintainer Uploads
- Debian Quality Assurance team

lucas@debian.org  Debian Quality Assurance
Quality factors

- Culture: "We release when it’s ready"
  - Many DDs won’t compromise on this
Quality factors

- Culture: "We release when it’s ready"
  - Many DDs won’t compromise on this

- Packages *ownership*: individuals or teams
  - About 600 active developers
  - Many of them experts of their packages
Quality factors

▶ Culture: "We release when it's ready"
  ▶ Many DDs won’t compromise on this

▶ Packages ownership: individuals or teams
  ▶ About 600 active developers
  ▶ Many of them experts of their packages

▶ Release Critical (RC) bugs
  ▶ Prevent packages from being part of a release
  ▶ Everybody is welcomed to fix them
    ▶ Non-Maintainer Uploads
Quality factors

- **Culture**: “We release when it’s ready”
  - Many DDs won’t compromise on this

- **Packages ownership**: individuals or teams
  - About 600 active developers
  - Many of them experts of their packages

- **Release Critical (RC) bugs**
  - Prevent packages from being part of a release
  - Everybody is welcomed to fix them
    - Non-Maintainer Uploads

- **Debian Quality Assurance team**
Debian QA team

- Goal: **Improve the quality of Debian as a whole**
- Not really a team (no strict membership), more like a central place to discuss and work on QA
- IRC: #debian-qa@irc.debian.org
- Mailing list: debian-qa@lists.debian.org
Debian QA team : Tasks

- Maintain infrastructure
- Run archive-wide checks and mass bug filings
- Take care of the dirty areas of Debian
Maintaining infrastructure
Infrastructure: mitigating the data hell

- Debian: aggregation of loosely-connected services
  - No Launchpad!

- Data everywhere
  A dozen of places to get interesting data

Tools required to gather data in a central place and expose it to the maintainers
Overview of nmap source package

Latest news:

- [2009-02-16] nmap 4.68-1 MIGRATED to testing
  (Britney)
- [2008-08-01] Accepted 4.68-1 in unstable (low)
  (LaMont Jones)
- [2008-06-27] nmap 4.62-1 MIGRATED to testing
  (Britney)
- [2008-06-14] Accepted 4.62-1 in unstable (low)
  (LaMont Jones)
- [2008-04-25] nmap 4.53-3 MIGRATED to testing
  (Britney)
- [2008-04-04] Accepted 4.53-3 in unstable (low)
  (LaMont Jones)

Source package-centric view

http://packages.qa.debian.org/package
Developer Packages Overview (DDPO)

Packages overview for Lucas Nussbaum <lucas@lucas-nussbaum.net>

main (8)

<table>
<thead>
<tr>
<th>Source Name</th>
<th>Bugs</th>
<th>Version</th>
<th>Ubuntu</th>
<th>Excuses</th>
<th>Binary Package</th>
<th>Buildd</th>
<th>Debcheck</th>
<th>Popcon</th>
<th>Watch</th>
<th>Section</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>developers-reference PTS Pool</td>
<td>65 (57)</td>
<td>3.3.8 3.4.0 3.4.3</td>
<td>2.4.3</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1035</td>
<td>More</td>
</tr>
<tr>
<td>feed2imap PTS Pool</td>
<td>2</td>
<td>0.8.2 0.9.3-1 0.9.4-1</td>
<td>0.9.4-1</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>64</td>
<td>More</td>
</tr>
<tr>
<td>hpcc PTS Pool</td>
<td>1</td>
<td>1.3.1-1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>Buildd</td>
<td>More</td>
<td>Te Un</td>
<td>1.3.1</td>
<td>science extra</td>
</tr>
<tr>
<td>mpich2 PTS Pool</td>
<td>2</td>
<td>1.2.1-r11</td>
<td>1.2.1-rc11</td>
<td>1.2.1ubuntu1</td>
<td>1.bug</td>
<td>-</td>
<td>Excuses</td>
<td>More</td>
<td>123</td>
<td>Buildd More</td>
<td></td>
</tr>
<tr>
<td>ruby-defaults PTS Pool</td>
<td>8</td>
<td>1.8.2-1</td>
<td>4.2</td>
<td>-</td>
<td>1</td>
<td>1234</td>
<td>Buildd</td>
<td>More</td>
<td>-</td>
<td>24</td>
<td>More</td>
</tr>
<tr>
<td>ruby-stdlib PTS Pool</td>
<td>1</td>
<td>1.1.2</td>
<td>1.1.7</td>
<td>-</td>
<td>1.1.7</td>
<td>-</td>
<td>-</td>
<td>1.1</td>
<td>-</td>
<td>96</td>
<td>More</td>
</tr>
<tr>
<td>taktuk PTS Pool</td>
<td>-</td>
<td>3.6.1-1</td>
<td>3.6.2-1</td>
<td>3.6.3-1</td>
<td>1.23</td>
<td>Buildd</td>
<td>More</td>
<td>-</td>
<td>-</td>
<td>177</td>
<td>3.7beta2 Details</td>
</tr>
<tr>
<td>websec PTS Pool</td>
<td>13</td>
<td>1.9.0-1</td>
<td>1.9.0-4</td>
<td>-</td>
<td>1.9.0-4ubuntu1</td>
<td>3 bugs</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>82</td>
<td>More</td>
</tr>
</tbody>
</table>

Developer-centric view

http://qa.debian.org/developer.php
Ultimate Debian Database (UDD)

- Gather everything into a single SQL DB
- Perfect tool for data-mining Debian

Currently imported in UDD:
Debian Sources and Packages, bugs, popcon, testing migrations, upload history, orphaned packages, carnivore, lintian, debtags, translations, NEW queue, screenshots, DEHS, ldap, wanna-build, removals, **Ubuntu Sources and Packages, Ubuntu bugs, Ubuntu popcon**

Possible questions you can answer with UDD:
- RC buggy packages in testing, sorted by popcon?
- Packages for which the last 4 uploads were NMUs?
- Packages not maintained by official DDs?

http://udd.debian.org/
Archive-wide checks and Mass bug filings
Archive-wide checks and MBF

- Developers are volunteers:
  - Focus on interesting things
  - Manual testing is boring → not done

- Need automated tests

- Other advantage: treats all packages equally
  Niche packages are very important to some users
Archive rebuilds

- Rebuild all packages from source
  7 hours on 40 8-cores build nodes  (Thanks OpenOffice.Org)

- Detect packages that Fail To Build From Source (FTBFS)
  - Bugs in packages
  - Toolchain bugs
  - Failures caused by changes in other packages

- Ran every 2-3 weeks

- Scripts to analyze failures and file bugs semi-automatically
  (about 200 bugs filed per hour)

- Also used for test rebuilds with new compilers, linkers, . . .
  → Valuable information for upstream
Debcheck

- Check (statically) that dependencies can be satisfied
- Not trivial: versioned dependencies, conflicts
- 2 versions:
  - "Simple one", in PHP
  - Better one: edos-debcheck (EDOS project)
    - Can also work with RPMs
    - http://edos.debian.net/
Testing package installation, upgrade and removal

Find bugs in maintainer scripts
(Shell) scripts executed during package installation/removal

Developers don’t run it on their systems
→ Archive-wide setup

http://piuparts.debian.org
Lintian

- Static analysis of packages
- Usually run locally by the developer, before uploading
- Archive-wide setup for continuous testing
- Packages now rejected at upload time for some errors

http://lintian.debian.org
DEHS - Debian External Health Status

- Monitor upstream version of packages
- Detect when the Debian package is outdated
- HTML scraping + regular expressions configured in debian/watch

http://dehs.alioth.debian.org
Taking care of the dirty areas of Debian
Taking care of the dirty areas of Debian

Two complementary approaches:

- Focus on packages neglected by their maintainers
- Focus on maintainers neglecting their packages
Neglected packages

- Already orphaned:
  - Do minimalistic maintenance
  - Problem: > 500 orphaned packages

- Not (yet) orphaned:
  - Find them, orphan or remove them
  - Goal: *Put maintenance in the hands of people with time*
Bapase: finding neglected packages

- Multi-criteria search for neglected packages
- Based on Ultimate Debian Database

<table>
<thead>
<tr>
<th>Package</th>
<th>Score</th>
<th>Action</th>
<th>O age</th>
<th>Testing</th>
<th>Migrate</th>
<th>Popcon</th>
<th>Wppp</th>
<th>RC</th>
<th>Bugs</th>
<th>Last upload</th>
<th>NMUs</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 ivtools</td>
<td>124269</td>
<td>Should be removed (since 2008-07-11)</td>
<td>2122 (2002-10-15)</td>
<td>270</td>
<td>270</td>
<td>180</td>
<td>RFA</td>
<td>0</td>
<td>9</td>
<td>91 (2008-05-07) (Maint: 2008-08-24)</td>
<td>4</td>
<td>maintainer and possible adopter pinged, and removal proposed to maintainer</td>
</tr>
<tr>
<td>2 metamil</td>
<td>114643</td>
<td>Should be removed (O_pkg) (since 2008-01-28)</td>
<td>1404 (2004-10-01)</td>
<td>0</td>
<td>0</td>
<td>3263</td>
<td>O</td>
<td>0</td>
<td>12</td>
<td>151 (2008-03-08)</td>
<td></td>
<td>still has revdeps, prop_rm filed on rdep.</td>
</tr>
<tr>
<td>3 electric</td>
<td>112643</td>
<td>Should be removed (O_pkg) (since 2008-01-30)</td>
<td>1238 (2005-03-17)</td>
<td>0</td>
<td>0</td>
<td>415</td>
<td>O</td>
<td>0</td>
<td>2</td>
<td>202 (2008-01-17)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 libgstreamer-perl</td>
<td>103699</td>
<td>Should be orphaned (since 2008-02-06)</td>
<td>105 (2008-04-23)</td>
<td>0</td>
<td>0</td>
<td>47</td>
<td>ITA</td>
<td>0</td>
<td>3</td>
<td>92 (2008-05-06)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

http://udd.debian.org/bapase.cgi
Inactive maintainers

- Missing In Action (MIA) team
  - Find them, track them, orphan their packages
    Mostly based on reporting by users or fellow developers
    Echelon: records activity of DDs; Carnivore: tracks identities
  - Collaboration with the Account Managers:
    possible removal of their account
    (Unused accounts → possible security problems)
Conclusion

Debian Quality Assurance:

- Mosty archive quality work
- Lots of tools and infrastructure developed within Debian
- What we don’t do:
  - Focus on specific packages
  - Bug triaging
  - Manual testing to find bugs
    - Reliance on users to find/file bugs

Join the QA team!