Perspectives on Bugs in the Debian Bug Tracking System

Julius Davies*, Hanyu Zhang*, Lucas Nussbaum**, Daniel M. German*
* Dept. of Computer Science, University of Victoria, Canada
** LORIA / Nancy-Université
juliusd@uvic.ca, hanyuz@uvic.ca, lucas.nussbaum@loria.fr, dmg@uvic.ca

***Abstract*—Bugs in Debian differ from regular software bugs. They are usually associated with packages, instead of software modules. They are caused and fixed by source package uploads instead of code commits. The majority are reported by individuals who appear in the bug database once, and only once. There also exists a small group of bug reporters with over 1,000 bug reports each to their name. We also explore our idea that a high bug-frequency for an individual package might be an indicator of popularity instead of poor quality.**

I. Introduction

In the open source community, the term bug is commonly used to describe software problems and is closely related to the terms failure, fault, error, and defect [1]. Conventional research in bug mining tends to presume a single overall system with a direct relationship between entries in the bug tracking system (BTS) and the code commits in the system's version control system (VCS). But Debian does not fit this model: Debian is a large compilation of software packages, much like a large anthology [2]. Debian developers download software packages from the original locations (upstream) and make small modifications to fit the package in with the rest of Debian. These modifications are similar to how an editor of *Modern English Poetry* might adjust fonts, page breaks, and introduce additional text. Debian developers usually do not have direct commit access to the upstream projects' VCS repositories. Because of the way Debian is built, an item in the Debian BTS will differ from a conventional bug in several ways. In this paper we observe three differences:

1) In Debian's BTS the majority of bugs are explicitly associated with packages, whereas in a conventional BTS the bugs would be associated with modules, sub-modules, or cross-cutting concerns. Packages in Debian are very much like modules in a regular software system, except that Debian has orders of magnitude more (over 25,000).
2) Changes to Debian are accomplished through source package uploads instead of code commits. Hence the source package repository resembles a classic VCS, except that a single source upload will aggregate hundreds of commit transactions from the original upstream VCS.
3) Many upstream-specific bugs show up in the Debian BTS since end users will often file bugs in Debian's BTS instead of using the upstream project's. Debian developers might create and maintain patches for particularly urgent bugs, but more often they will provide the upstream project with logs and diagnostics, and then monitor its progress. Highly-used Debian packages (e.g. Linux, Firefox) in particular cause the most upstream bug noise.

We explore these three differences by analyzing all bug reports and package upload records in Debian between Jan 1st 2007 00:00:00 GMT and Dec 31st 2009 23:59:59 GMT.

II. Research Questions Addressed

1) Are the majority of bugs in the Debian BTS associated with packages?
2) Is the number of changes in a package correlated with the number of bugs reported for that package?
3) Who is reporting the bugs?
4) How does bug frequency relate to package popularity?

III. Input Data

We initially hoped to use the Ultimate Debian Database (UDD)[1] exclusively for all aspects of our investigation, but three problems with the UDD data forced us to supplement UDD with raw Debian BTS data (bugs-mirror.debian.org):

- UDD is missing some bugs. For example bug 471445 is available directly through the Debian BTS, but queries against UDD cannot find this bug.
- Debian's bug cloning mechanism obscures a bug's date. For example, if bug 7890 is cloned, this would cause a new bug to be created, but this bug's date would be recorded using 7890's original date: March 7th, 1997. We resolved this by using a bug's clone-date instead of its create-date in these cases. Unfortunately the clone-date is not present in the UDD.
- UDD is missing some data in the packages and sources tables. Packages such as bongo and bandersnatch are present in the bug reports, as well as UDD's *upload_history* table, but not in the main package tables, since these packages never completed a full migration from initial upload into a final release. These comprise such a small percentage of the bug reports between 2007 and 2009 (0.7%) that we classified them as "not associated with a package."

[1] http://wiki.debian.org/UltimateDebianDatabase

Bugs and Uploads: Jan 1 2007 to Dec 31 2009 (3 years)									
	Non-Package Bugs			Debian is Upstream			Debian is Not Upstream		
Rank		Bugs	Uploads		Bugs	Uploads		Bugs	Uploads
1	wnpp	9950	0	lintian	795	57	linux-2.6*	2882	71
2	ftp.debian.org	3005	0	apt*	764	47	iceweasel*	798	55
3	installation-reports	1052	0	aptitude	717	41	xserver-xorg-core	533	0
4	mirrors	317	0	devscripts	660	61	xserver-xorg-video-intel	448	66
5	qa.debian.org	276	0	dpkg*	383	50	icedove	395	22
6	www.debian.org	227	0	reportbug	383	31	udev*	380	42
7	release-notes	215	0	debhelper	277	111	libc6	334	0
8	bugs.debian.org	207	0	dpkg-dev	262	0	evolution	332	48
9	release.debian.org	142	0	debian-maintainers	225	71	openoffice.org*	315	148
10	lists.debian.org	106	0	debian-installer	224	9	grub-pc	310	0
Some uploads show zero because source-packages can generate more than one binary-package * Top-250 Popcon									

Table I
THE TOP 10 'MOST BUGGY' PACKAGES IN EACH OF THE THREE CATEGORIES.

IV. METHODOLOGY

After downloading the UDD and raw BTS data we performed the following processing steps to the data. First, for the three year period, we combined the raw bug files (index.realtime and index.archive.realtime) and sorted them by bug-number, resulting in 158,058 bugs. We found the earliest bug on January 1st, 2007 (405152) and the latest bug opened on Dec 31st, 2009 (563209). We also aggregated bugs by month and by package (as recorded in the BTS), creating a table of triples: (package, month, bug-count). For example: (wnpp, 2007-January, 345), (wnpp, 2007-February, 356). We divided the table of triples into three categories:

1) Non-package bugs.
2) Package-bugs where Debian is also the original provider of the software (e.g. apt, dpkg).
3) Package-bugs where Debian is downstream (e.g. linux, firefox).

To determine if a bug was package-related we compared the bug against all names in UDD's *packages* and *sources* table. If we failed to find a match we classified the bug as a non-package bug. To determine if Debian was the original provider we looked in the *homepage* and *VCS-URL* columns of the same two package tables for strings matching the regular-expression *.debian.* (e.g. alioth.debian.org). We only looked at the *VCS-URL* field if the *homepage* value was empty.

Finally, to compare bugs with uploads we joined our data against UDD's *upload_history* table.

V. RESULTS

A. Are the majority of bugs in the Debian BTS associated with packages?

Eighty-nine percent of the bugs in the three year sample (140,772 of 158,058) were associated with packages. The value of the *package* field in these bug records corresponded to actual packages/sources in UDD, whereas the rest (17,288) had an empty value, or an invalid value, or a value that did not match UDD for some other reason. This can be caused by human errors, but is more frequently caused by the use of *pseudo-packages* in the Debian BTS to track issues with Debian infrastructure. In particular, new software that should be packaged in the distribution, and old packages that should be removed from Debian are usually filed against the *wnpp* pseudo-package, "Work-Needing and Prospective Packages." The *wnpp* pseudo-package is associated with 9,950 bugs (6.3%) in the period studied; this is the highest frequency for any *package* value in the Debian BTS. We also treated a value such as 'emacs23,iceweasel' as invalid, and in this respect our study slightly underreports the association between bugs and packages. By separating package bugs and non-package bugs on a timeline graph (bugs-opened-per-month, see Figure 1), we observed stronger variation in package bug rates compared to the non-package bug rate.

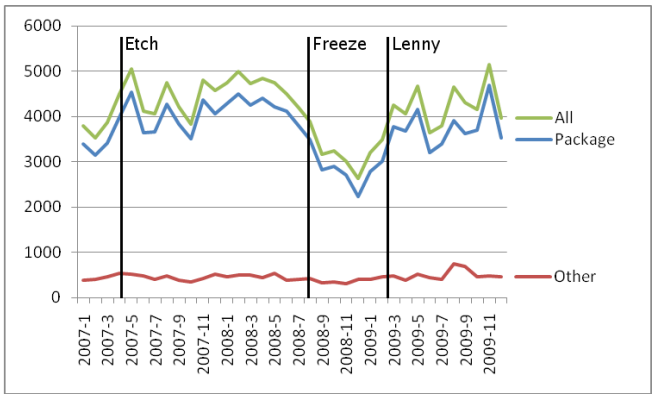

Figure 1. Bugs opened per month from 2007 to 2009.

The amount of bugs for non-packages remained stable around 500/month. There was a slight peak from Aug-2009 to Oct-2009. On the other hand, the amount of monthly reported bugs for Debian packages varied dramatically. The amount of bugs underwent a rapid drop from 5,000

bugs/month in the beginning of 2008 to 2,500 bugs/month in the end of 2008. The bug rate fluctuated upward in 2009 and reached the peak by the end of that year.

The majority of bugs in the Debian BTS are associated with packages.

B. Is the number of changes in a package correlated with the number of bugs reported for that package?

We measured the same correlation between number of uploads and number of bugs in Debian as a whole, per month (see Figure 2); the result was striking: 0.811 ($\rho < 0.001$). We examined also some of the most popular packages and observed Spearman correlation usually between 0.250 and 0.5, with a few packages showing little correlation (between -0.2 and 0.2) (see Figure 3). We suspect, however, that the number of changes in a package is usually correlated with the number of bugs reported for that package, but this needs to be further explored.

For Debian as a whole, bugs and uploads are highly correlated.

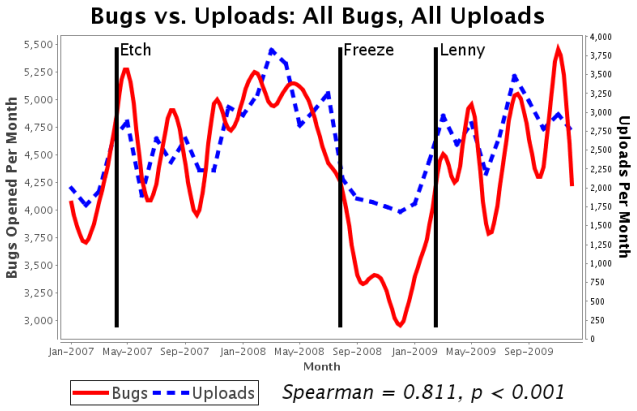

Figure 2. Bugs opened per-month and package uploads per-month for all of Debian. This chart includes all uploads, not just those associated with bugs from the same period.

C. Who is reporting the bugs?

The number of bug notifications and the amount of people involved are readily accessible variables that can be used to study software evolution [3]. Here we explore the relationships between bug submitters and bugs.

Table II illustrates the distribution of the number of distinct bug reporters for Debian packages. Most packages have only a few bugs, and thus are unable to have more than a few bug reporters. For this reason we only focus on the 358 packages that have at least 50 reported bugs. Of these only 2% have more than 1000 bug submitters; 9% of the packages have more than 500 bug submitters; 71% of the packages have more than 100 bug reporters, and more than half (51%) of these packages have less than 200 bug submitters. This shows that most people focus on a limited number of major

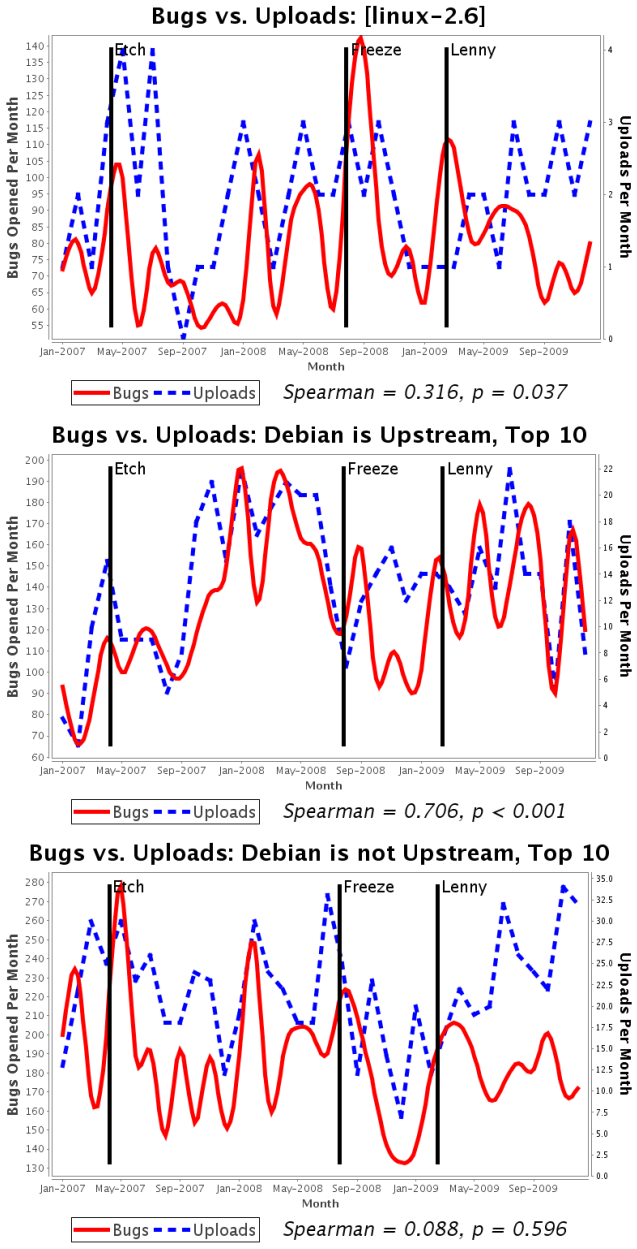

Figure 3. Three different samples show different levels of correlation between uploads and bugs. The two Top-10 charts correspond to the same entries in Table I. Debian's Linux-2.6 package (first chart) measures a 0.316 Spearman correlation by itself, and yet the last chart (which includes Linux-2.6) shows no correlation.

Number bug Submitters	Proportion
0-99	29%
100-199	31%
200-299	14%
300-399	13%
400-499	4%
500-999	7%
1000+	2%

Table II
COMPOSITION OF PROJECTS REGARDING NUMBER OF PARTICIPANTS.

Number of Packages	Proportion
1	55%
2-4	25%
5-9	9%
10-19	5%
20-29	2%
30-99	3%
100+	1%

Table III
COMPOSITION OF BUG REPORTERS WITH RESPECT TO THE NUMBER OF PACKAGES THEY HAVE FILED BUGS FOR.

packages. We are also interested in the relationship between the number of bugs and the number of bug reporters. Are they correlated? The Spearman Correlation factor of the bug amount and bug submitter for the 358 projects is 0.5531, which shows a strong positive correlation. That means a project that has more reported bugs usually has more bug submitters.

Table II tells us how people participate in Debian packages, but how many packages does a bug reporter usually participate in?

Table III helps answer the question. There are 53,908 bug reporters, more than half (55%) of which have submitted only one bug for one package ever. Eighty percent have reported bugs for less than 5 projects.

There are 18 bug reporters who have submitted bugs for more than 1000 packages. More research is required to verify if these bug submitters report bugs in an automated or semi-automated way.

D. How does bug frequency relate to package popularity?

From Table I we observed six packages (of twenty) that also appear among the Debian popularity contest's (Popcon) top 250 packages. We expected to see higher overlap between Popcon and Debian's buggiest packages. This was a disappointing result, especially with respect to the packages in the far-right column. In this column are listed 7 flagship products of the FOSS world: Linux, Evolution, OpenOffice, Firefox (aka "Iceweasel"), Firebug (aka "Icedove"), Xserver-Xorg, Libc6. Grub is also notable since it is the linux bootloader for many FOSS distributions. There appears to be a relationship between package popularity and bug frequency. We hope to find a clear result in future work.

VI. THREATS TO VALIDITY

Packages without bugs are ignored in our study. The time period we chose to study (January 1st, 2007 to December 31st, 2009) was arbitrary. Bugs can be filed against source packages and binary packages, but uploads are only tied to source packages. Our method only partially reconciles this mismatch. Our technique for discovering packages where Debian is upstream is not 100% accurate.

VII. CONCLUSION AND FUTURE WORK

In this paper we explored four perspectives on bugs in the Debian bug tracking system between January 2007 and December 2009:

1) Bugs are usually associated with packages. In some cases Debian's BTS doubles as both an upstream and downstream BTS.
2) There is a correlation between bugs and uploads for the overall system. We need to further investigate how the relationship holds up for individual packages. This supports our view that package uploads are analogous to code commits in conventional software systems.
3) The vast majority of bug reporters file only one or two bug reports, but a select few are responsible for thousands.
4) Bug frequency and popularity may be related.

We believe that the Debian Bug tracking system provides an interesting research subject, mainly because it provides a look at bug management as an ecology, rather than in individual applications. For example, it is necessary to understand which bugs are applicable to Debian activities, and which ones are germane to the application itself and hence further propagated to them, i.e. are some bugs in Debian resubmitted as bugs in applications or are Debian maintainers responsible for fixing them? We know that Debian maintainers fix defects, but sometimes these are maintained by Debian (as patches) and sometimes sent to their corresponding applications. What determines such choice?

We know little of the composition of bug reporters in Debian. Are they Debian maintainers, application developers, or final users? Furthermore, Debian is the foundation of other Linux distributions (such as Ubuntu). It is likely that the maintainers of such distributions will collaborate in bug reporting (and fixing) with Debian ones.

REFERENCES

[1] A. G. Koru and J. Tian, "Defect handling in medium and large open source projects," *IEEE Software*, vol. 21, no. 4, pp. 54–61, 2004.

[2] J. M. González-Barahona, G. Robles, M. Michlmayr, J. J. Amor, and D. M. Germán, "Macro-level software evolution: a case study of a large software compilation," *Empirical Software Engineering*, vol. 14, no. 3, pp. 262–285, 2009.

[3] M. Pérez-Francisco, P. B. Perez, and G. Robles, "Correlation between bug notifications, messages and participants in debian's bug tracking system," in *ESEM*. IEEE Computer Society, 2007, pp. 455–457.