On Package Freshness in Linux Distributions

Damien Legay
Software Engineering Department
University of Mons
Mons, Belgium
damien.legay@umons.ac.be

Alexandre Decan
Software Engineering Department
University of Mons
Mons, Belgium
alexandre.decan@umons.ac.be

Tom Mens
Software Engineering Department
University of Mons
Mons, Belgium
tom.mens@umons.ac.be

Abstract—The open-source Linux operating system is available through a wide variety of distributions, each containing a collection of installable software packages. It can be important to keep these packages as fresh as possible to benefit from new features, bug fixes and security patches. However, not all distributions place the same emphasis on package freshness. We conducted a survey in the first half of 2020 with 170 Linux users to gauge their perception of package freshness in the distributions they employ, the value they place on package freshness and the reasons why they do so, and the methods they use to update packages. The results of this survey reveal that, for the aforementioned reasons, keeping packages up to date is an important concern to Linux users and that they install and update packages through their distribution's official repositories whenever possible, but often resort to third-party repositories and package managers for proprietary software and programming language libraries. Some distributions are perceived to be much quicker in deploying package updates than others. These results are useful to assess the expectations and requirements of Linux users in terms of package freshness and guide them in choosing a fitting distribution.

I. INTRODUCTION

The Linux operating system is arguably the most successful open-source project to ever have come to fruition. Per the nature of open-source software, Linux is available in many forms, called *distributions*. Each distribution is composed of the Linux kernel and a host of software products provided in the form of *packages*. The number of these packages in Linux distributions is known to grow superlinearly [1], [2]. Packages are made available to users through a wide variety of *package managers*. Some, such as pacman, dkpg or RPM, are specific to a distribution and its derivatives whilst others, such as Flatpak or Snappy, are intended for cross-distribution usage. The set of packages that can be found within these package managers largely depends on the ambitions and philosophy of the distribution. Some distributions, such as Debian, pledge that all their components will be free software (as per the Debian social contract) while other distributions make no such promise. Philosophical divergences also cause versions of packages available in each distribution to differ, as distributions weigh concerns of *stability* (the ability of a distribution to withstand changes in its components) and *package freshness* (how up to date a package is compared to its upstream releases) differently.

This presents a trade-off to the maintainers of Linux distributions. On the one hand, adopting new versions of packages within the distribution will grant users access to new features, bug fixes and security patches. On the other hand, these new versions risk introducing breaking changes, new bugs, security vulnerabilities, incompatibilities or co-installability issues wherein packages cannot be installed without creating conflicts with some other packages [3], [4]. Specifically for Debian, Claes et al. estimated that the number of packages being incompatible with at least one other package varies between 15% and 25% [5]. Distribution maintainers thus have to go through the very time-consuming process of assessing new package versions for these risks.

Relying on *semantic versioning* (https://semver.org) could mitigate the cost of assessing package versions for breaking changes but the degree of semantic versioning compliance depends on the considered programming language ecosystem [6]. Hence, depending on the language a package is written in, even minor releases and patches might contain breaking changes. Specifically for Maven, Raemaekers et al. [7] observed that about one third of all updates, including minor releases and patches, introduce API breaking changes. Similarly, the task of assessing updates for incompatibilities depends on the accuracy of the package metadata, but that metadata often becomes outdated as the package evolves [8]. These factors help explain the choice of some distribution maintainers to emphasise stability and security over package freshness. Yet, package freshness is important to the Linux community, as evidenced by the existence of package freshness monitoring services such as Repology and DistroWatch.

Our long-term goal is to help users choose the distribution that best fits their needs. To do so, they need to possess accurate information about those distributions. As such, we aim to measure, understand, compare and report package freshness in Linux distributions and evaluate to which extent the perception of users matches reality. To these ends, we have opted for a mixed study, consisting of a qualitative component (a survey of Linux users) and a quantitative component (empirical analyses on the freshness of packages in Linux distributions). This paper constitutes the first part of this study, reporting upon the qualitative results of the survey.

II. RELATED WORK

Little is known about package freshness in Linux distributions in general. Shawcroft [9] compared the package freshness of 37 packages in 8 Linux distributions by looking at the number and proportion of obsolete packages, measuring the

time between upstream release of a package version and downstream deployment into a distribution, as well as quantifying the number of upstream versions that are ahead of deployed versions. We extend this work to a much larger corpus of 529 packages, using more recent data (see Section V). Gonzalez-Barahona et al. [10] observed that one out of eight packages (12%) was not updated at all within a nine-year timespan, from Debian stable 2.0 (released on 1998-07-24) to 4.0 (released on 2007-04-08). Nguyen and Holt [11] studied the lifecycle of Debian packages. They compared the age of packages in Debian distributions unstable, testing and stable, defining package age as the time delta between its introduction into the distribution and its removal from Debian or its update by a newer version of the same package.

The freshness of distributed packages has been formalised by the notion of technical lag (expressed both in terms of time and number of versions) by Gonzalez-Barahona et al. [12], [13]. Zerouali et al. [14] used technical lag to explore how outdated packages are in Debian-based Docker containers and the impact that such outdated packages have on the presence of security vulnerabilities and bugs. Decan et al. [15] used it to assess the reluctance of package maintainers to update package dependencies in order to avoid putative backward-incompatible changes in programming language ecosystems.

III. Methodology

This paper reports on a survey of Linux users, conducted in early 2020, aiming to examine their perception of package freshness. We specifically explore the value Linux users place on package freshness, their motivations to upgrade packages to newer versions, the way they use to do so and how fresh they perceive the packages in their most used distribution to be. In order to obtain a sampling of the views of the open-source community at large, we included both industry practitioners and academics. To gather responses from practitioners, we distributed the survey to attendants of the Free and Open source Software Developers' European Meeting (FOSDEM 2020) and Community Health Analytics Open Source Software conference (CHAOSSCon Europe 2020), in both paper and electronic versions, as well as on Linux fora (linux.org, forums.fedoraforum.org, forums.debian.net and forums.linuxmint.org) and on subreddits dedicated to Linux distributions (/r/centos, /r/fedora, /r/redhat, /r/linuxmint and /r/opensuse). We obtained 51 answers from both events combined and 78 from fora and subreddits. We completed this with responses from academics, obtained by circulating the survey among researchers, computer science master students and alumni of our University (17 responses). We also obtained 24 responses from external researchers through a post on one of the authors' Twitter account. In total, we obtained 170 responses. The survey form and anonymised answers have been placed on Zenodo (DOI: 10.5281/zenodo.3908332).

Since the answers to the survey questions can depend on the distribution(s) used, we asked which Linux distribution(s) respondents frequently make personal use of. They could rank up to three distributions, in order of frequency of use, chosen from a pre-established list of 16 popular Linux distributions. They also had the option to specify other distributions.

Table I reports the total number of answers obtained for each distribution, as well as the number of times that distribution was ranked first, second or third. The table also reports the aggregated responses for each family of Linux distributions. Distributions for which there were fewer than five answers have been gathered under the label "other distributions". These are Gentoo, SUSE Entreprise Edition, Parabola, FerenOS, Android, Clear Linux, Knoppix, Alpine Linux, NixOS and Raspbian. These results indicate that 88% (149) of the respondents make use of at least two distributions, and 62% (106) of at least three, showing that most of them have experience with several distributions, either from using them concurrently or migrating from one to another. At least 28% of the respondents who use more than one distribution use at least one that emphasises *freshness* (Arch Linux, Gentoo, OpenSUSE Tumbleweed, Fedora, Debian unstable or testing) and one that emphasises *stability* (OpenSUSE LEAP, SUSE Entreprise Edition, CentOS, Red Hat Entreprise Edition or Debian stable)

IV. Findings

A. Which distributions are perceived to be more up-to-date?

To gauge the user perception of package freshness in Linux distributions, we asked respondents how long it took, according to them, for the latest upstream version to be made available in the official repositories of their most-used distribution (answered first in the previous question). We gave them six exclusive options: "a few days, at most", "a few weeks", "a few months", "a few years", "never (not available)" and "I don't know". We asked about six categories of packages:
OSS: open-source end-user software (e.g. Firefox or GIMP);
PS: proprietary end-user software (e.g. Spotify or Skype);
DT: development tools (e.g. git, Emacs or Eclipse);
STL: system tools and libraries (e.g. openSSL, sudo or zsh);

TABLE I
USAGE PER (FAMILY OF) LINUX DISTRIBUTION(S).

Distribution	First	Second	Third	Total
Ubuntu (family)	47	46	43	117
Ubuntu LTS	30	25	11	66
Ubuntu	17	16	18	51
Debian (family)	30	37	26	93
stable	19	24	22	65
testing	10	13	4	27
unstable	1	0	0	1
Red Hat (family)	33	33	25	91
Fedora	24	9	8	41
CentOS	8	16	9	33
Entreprise Edition	1	8	8	17
Arch Linux	29	8	8	45
OpenSUSE (family)	21	13	5	39
Tumbleweed	17	4	2	23
LEAP	4	9	3	16
Linux Mint	5	10	7	22
Slackware	2	2	1	5
Other distributions	3	6	5	14
Total	170	149	106	

PLL: programming language libraries (e.g. NumPy for Python, Lodash for npm);

PLR: programming language runtimes (e.g. Python or node.js).

Fig. 1 presents the median answer for each category. Only distributions which are used primarily by five or more respondents are shown. Being based on a rolling release policy, Arch Linux and OpenSUSE Tumbleweed strive to distribute the latest stable releases of all packages included in the distribution. Fig. 1 reveals that the respondents' perception aligns with this reality, since most of them agree that upstream versions are made available very quickly, within days. Only proprietary end-user software (PS) packages are perceived by some to be updated slower than a matter of weeks, when at all available, in accordance with the fact that some distributions do not directly support proprietary software.

Fedora users believe that they will dispose of fresh versions within weeks. Most Fedora users answered that proprietary software just was not available in Fedora's official repositories. Ubuntu, Linux Mint and Debian testing users usually think it takes weeks to months for upstream versions to be released within the official repositories. Although the median answer for proprietary software in Debian testing is that it takes years, this is due to the fact that a significant portion of respondents answered proprietary software was not available in Debian testing. At the end of the spectrum, CentOS and Debian stable users tend to expect to wait months for fresh versions to be made available in their distribution's official repositories, regardless of package type. 7 out of the 170 (5%) respondents expressed that they did not know when updates are made available to the distribution's official repositories. A further 50 respondents expressed ignorance for only some categories, principally regarding proprietary software.

B. To what extent do users value package freshness?

We enquired, for each package category, what importance users impart to keeping packages up to date with upstream releases. To do so, we relied on a 4-value Likert scale to denote relative importance: unimportant, slightly important, moderately important and very important. Fig. 2 reports on the results for each package category.
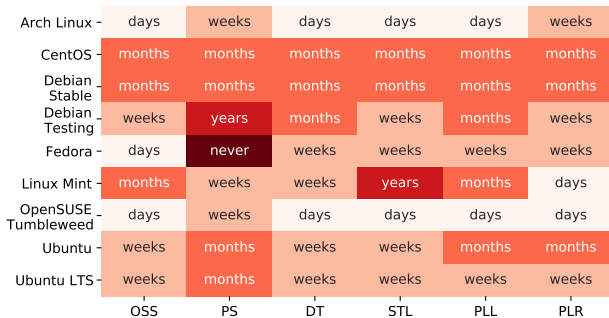
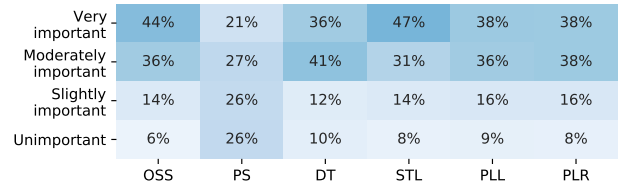|	OSS	PS	DT	STL	PLL	PLR
Very important	44%	21%	36%	47%	38%	38%
Moderately important	36%	27%	41%	31%	36%	38%
Slightly important	14%	26%	12%	14%	16%	16%
Unimportant	6%	26%	10%	8%	9%	8%

Fig. 2. Importance imparted by respondents to staying up-to-date

For all categories, respondents consider it important to update packages: 75% to 80% of them answered it was moderately to very important to remain up to date. A notable exception is the proprietary end-user software category (PS): in this instance alone, a majority (52%) considers the importance of package freshness to be slight or null. We do not see a clear practical reason why users would consider updating proprietary packages less important than other packages.

Users of distributions that are perceived to be slower in deploying packages, such as CentOS and Debian stable, were less likely to value maintaining a high level of package freshness. Conversely, users of distributions that are perceived to have fresher packages, such as Arch Linux, OpenSUSE Tumbleweed and Fedora were more likely to consider package freshness important. Indeed, across all categories, a much greater proportion of respondents answered that updating packages was moderately to very important for Arch (80%), Tumbleweed (80%) and Fedora (84%), than for CentOS (50%) and Debian stable (54%). Out of the 7 respondents who did not know whether the packages in their distribution were fresh or not, 5 answered that package freshness was moderately to very important, the other 2 that it was unimportant.

C. What are the main reasons for updating packages?

Benefits to updating packages include access to new features, bug fixes and security patches. In order to verify whether those benefits actually motivate users to update, respondents were asked what their main reasons were to update packages, out of five options: to benefit from security patches (selected by 90% of the respondents), from bug fixes (80%), from new features (66%), to sate their desire to remain up to date (35%) or to retain compatibility with other packages (27%). Respondents could select as many options as they wanted. An additional open option was available, but was not used by any respondent. The motivation of obtaining new features is less prevalent in users of distributions that are perceived as less fresh, such as Debian Stable and CentOS. Users of Debian are less likely to cite bugs as reasons to update, likely owing to the Debian process of package integration leading to stable distributions, particularly Debian Stable.

D. Which mechanisms are used to keep packages up-to-date?

Several mechanisms can be used to update packages: using the *official package manager of the distribution and its official repository (off)*, using the *official package manager of the*

	OSS	PS	DT	STL	PLL	PLR
Arch Linux	days	weeks	days	days	days	weeks
CentOS	months	months	months	months	months	months
Debian Stable	months	months	months	months	months	months
Debian Testing	weeks	years	months	weeks	months	weeks
Fedora	days	never	weeks	weeks	weeks	weeks
Linux Mint	months	weeks	weeks	years	months	days
OpenSUSE Tumbleweed	days	weeks	days	days	days	days
Ubuntu	weeks	months	weeks	weeks	months	months
Ubuntu LTS	weeks	months	weeks	weeks	weeks	weeks

Fig. 1. Package freshness perception.

distribution with community repositories (com), using *third-party package managers such as Flatpak (3rd)*, *installing manually from binaries (bin)* or *installing manually from source files (src)*.

We asked respondents which of these mechanisms they used to update packages in their most-used distribution, for each of the considered package categories. They could select as many answers as they wanted. Fig. 3 shows a heatmap of the frequency at which respondents reported which of these mechanisms is used, for each package category.

For most package categories, the official repositories (off) dominate largely (used by 79% to 94% of respondents), followed by community (com) repositories (used by 19% to 41%). Proprietary software stands in contrast to other categories, being installed more than a third of the time from binaries (bin). This is expected, as some distributions (e.g. Debian) are reluctant to include proprietary software within their official repositories. We also see that third-party managers (3rd) are regularly used to install end-user open source software (21%), development tools (22%) or programming language libraries (29%). In the case of programming language libraries, this is that most libraries for some languages are (sometimes exclusively) available through dedicated package managers (e.g. pip for Python, npm for Javascript). In the case of packages related to development tools, we believe this is likely due to the assortment of tools available through Flatpak and Snappy, including popular IDEs, text editors and graphical user interfaces for git. These tools are not always available in official or community repositories. For instance, Intellij IDEA is not available in the official repositories of Fedora, but can be installed through Flatpak and Snappy.

V. Discussion

We reported the results of a survey of 170 Linux users about package freshness. We found that users perceive distributions such as Arch Linux, OpenSUSE Tumbleweed and Fedora as being much more likely to have fresh packages than distributions such as Debian Stable and CentOS. Verifying these perceptions will require a quantitative empirical comparison of freshness in Linux distributions.

As a preliminary step towards such an empirical study, we gathered the package versions available in a recent snapshot of the five distributions respondents cited the most. We took the latest snapshot available prior 2019-11-01. This date was selected to minimise the gaps between distribution release dates. The distributions considered are listed in Table II.

TABLE II
RELEASES OF THE CONSIDERED LINUX DISTRIBUTIONS

Distribution	Release	Date
Arch Linux	rolling	2019-10-31
CentOS	8.0	2019-09-24
Debian stable	10	2019-07-06
Fedora	31	2019-10-29
Ubuntu	19.10	2019-10-17

We identified 529 common packages for these distributions and performed pairwise comparisons on the freshness of the versions of these packages. The results are found on Fig. 4 with each cell reporting the proportion of packages that are at least as fresh in the source distribution as in the target distribution.

We see that almost all (97% to 99%) packages in Arch Linux are at least as fresh as the packages found in the other distributions, and only 28% (CentOS) to 72% (Fedora) of packages in other distributions are as fresh as Arch Linux packages. At the other extreme, the vast majority of packages (94% or more) in all considered distributions are at least as fresh as those found in CentOS and only 28% to 53% of CentOS packages are at least as fresh as those found in other distributions. Debian Stable is in a similar situation as CentOS, albeit slightly less marked. Fedora and Ubuntu lie in the middle, with 72% and 65% (resp.) of their packages as fresh as those found in Arch.

These results suggest the following ranking of distributions in decreasing order of freshness: Arch Linux, Fedora, Ubuntu, Debian Stable and finally CentOS. This roughly corresponds to respondent perception in Fig. 1. Nevertheless, these perceptions are imprecise, as evidenced by the fact that Ubuntu LTS users considered some categories of packages to be available within their distribution's repositories sooner than non-LTS Ubuntu users, which is contrary to expectations. Additionally, 57 respondents (33%) were not confident to answer for at least one package category, with 7 (4%) of them answering they did not know for any category.

This preliminary empirical analysis of package freshness in five Linux distributions hints to the fact that distributions lie on a continuum with regards to the trade-off between package freshness and system stability. In order to place distributions on that continuum, we will, in a follow-up study, seek to em-

Fig. 3. Frequency of updating mechanism usage.

	STL	OSS	DT	PLR	PLL	PS
off	94%	91%	88%	86%	79%	40%
com	19%	41%	29%	22%	19%	37%
3rd	9%	21%	22%	16%	29%	22%
bin	8%	19%	12%	12%	9%	39%
src	9%	14%	12%	14%	14%	7%

source distribution	CentOS	Debian	Ubuntu	Fedora	Arch
Arch	99%	97%	96%	97%	
Fedora	99%	92%	87%		72%
Ubuntu	96%	99%		77%	65%
Debian	94%		60%	47%	38%
CentOS		53%	39%	39%	28%

target distribution

Fig. 4. Proportion of packages in a source distribution that are at least as fresh as in a target distribution

pirically quantify the difference in package freshness between distributions by relying on the technical lag measurement framework [13]. We will compare the versions of packages available in different distributions in terms of time lag (i.e. the time since a more recent upstream version of the package has been available) and version lag (i.e. the number of more recent versions available). We will seek to contrast the package freshness measured within distributions with the perceptions of users reported in this paper, thereby gauging to what degree user perception matches reality.

We will also examine the relationship between package freshness, stability and security in distributions. Comparing distributions in terms of these characteristics could help users to choose a distribution that matches their requirements and expectations. It will also allow package maintainers to know whether their packages are likely to be up to date in certain distributions, and potentially adopt practices that allow distribution maintainers to assess the stability, compatibility and security of their packages more quickly, to allow faster deployment of updates. Additionally, we will conduct a follow-up survey to examine to what extent package freshness motivates users to migrate from one distribution to another relative to other factors.

VI. CONCLUSION

The Linux ecosystem depends on packages that are made available through package managers –either official ones used by specific distributions, or third-party ones– as well as directly through binaries and source files. Package versions in official distribution repositories do not always match the latest versions released by the package's authors, out of a need to balance package freshness with system stability and security.

This paper reports on the results of a survey aimed at habitual Linux users to determine what were their values, perceptions and practices regarding package freshness. Their answers revealed that they usually place significant value in keeping the packages they use up to date, principally out of security concerns, but also largely to benefit from bug fixes and new features. Whenever possible, users prefer to update packages through the distribution's official package managers, using the distribution's official repositories. This is not always possible, though, as some packages are either unavailable or outdated in official repositories. This is most prevalent in the case of proprietary end-user software and some development tools. Additionally, programming language libraries are often installed and updated through language-specific package managers.

Unsurprisingly, users perceive packages in rolling release distributions to be very fresh. On the other hand, Debian Stable and CentOS users perceive it takes longer for new versions of packages to be made available in the official repositories, in the order of months. Other distributions are perceived to be somewhere in between on the "package freshness continuum". Preliminary empirical analysis shows that there is some truth to these perceptions, with Arch Linux being the most fresh distribution studied and CentOS the

least. Nevertheless, one out of twenty respondents had no inkling as to the freshness of packages in their distribution, despite valuing it. Additionally, some answers are inconsistent with reality (e.g. users answering that packages take weeks to be deployed in Ubuntu LTS, but months in non-LTS Ubuntu for some package categories). These incidences vindicate the need to conduct a follow-up study wherein we will perform further empirical analyses in order to quantify the comparative package freshness of Linux distributions, as well as examine its role as a motivator in user adoption of distributions.

REFERENCES

[1] Q. Tu *et al.*, "Evolution in open source software: A case study," in *Proceedings 2000 International Conference on Software Maintenance*. IEEE, 2000, pp. 131–142.

[2] G. Robles, J. J. Amor, J. M. Gonzalez-Barahona, and I. Herraiz, "Evolution and growth in large libre software projects," in *Eighth International Workshop on Principles of Software Evolution (IWPSE'05)*. IEEE, 2005, pp. 165–174.

[3] J. Vouillon and R. Di Cosmo, "On software component co-installability," in *Joint European Software Engineering Conference / Foundations of Software Engineering*, 2011, pp. 256–266.

[4] ——, "Broken sets in software repository evolution," in *International Conference on Software Engineering*, 2013, pp. 412–421.

[5] M. Claes, T. Mens, R. D. Cosmo, and J. Vouillon, "A historical analysis of Debian package incompatibilities," in *Working Conference on Mining Software Repositories*, 2015, pp. 212–223.

[6] A. Decan and T. Mens, "What do package dependencies tell us about semantic versioning?" *IEEE Transactions on Software Engineering*, 2019.

[7] S. Raemaekers, A. van Deursen, and J. Visser, "Semantic versioning and impact of breaking changes in the Maven repository," *Journal of Systems and Software*, vol. 129, pp. 140 – 158, 2017.

[8] C. Artho, K. Suzaki, R. Di Cosmo, R. Treinen, and S. Zacchiroli, "Why do software packages conflict?" in *Working Conference on Mining Software Repositories*, 2012, pp. 141–150.

[9] S. Shawcroft, "Open source watershed: Studying the relationship between linux package and distribution releases," Ph.D. dissertation, University of Washington, 2009.

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

[11] R. Nguyen and R. Holt, "Life and death of software packages: an evolutionary study of debian," in *Proceedings of the 2012 Conference of the Center for Advanced Studies on Collaborative Research*, 2012, pp. 192–204.

[12] J. M. Gonzalez-Barahona, P. Sherwood, G. Robles, and D. Izquierdo, "Technical lag in software compilations: Measuring how outdated a software deployment is," in *IFIP International Conference on Open Source Systems*. Springer, 2017, pp. 182–192.

[13] A. Zerouali, T. Mens, J. Gonzalez-Barahona, A. Decan, E. Constantinou, and G. Robles, "A formal framework for measuring technical lag in component repositories – and its application to npm," *Journal of Software: Evolution and Process*, vol. 31, no. 8, 2019.

[14] A. Zerouali, T. Mens, G. Robles, and J. M. Gonzalez-Barahona, "On the relation between outdated Docker containers, severity vulnerabilities, and bugs," in *International Conference on Software Analysis, Evolution and Reengineering*. IEEE, Feb 2019, pp. 491–501.

[15] A. Decan, T. Mens, and E. Constantinou, "On the evolution of technical lag in the npm package dependency network," in *International Conference on Software Maintenance and Evolution*. IEEE, September 2018, pp. 404–414.