Are we really devoted to our users?

Margarita Manterola

June 25, 2005

Abstract

The Debian Social Contract -that any Debian Developer has to agree to during the New Maintainer process- states that *Our priorities are our users and Free Software* and then says that *we will be guided by the needs of our users and the free-software community.* This talk intends to be an analysis on the ways in which Debian as a whole is failing to have users as a priority and also show some points were progress could be made.

1   What the needs of our users are

Saying that we want to take over the world is quite a normal statement among the Free Software community. Sometimes it's just a joke, but sometimes it's serious, and what we actually mean is that we want to change the world, make it a better place for all, through the use of Free Software.

So, if we do plan to take over the world, we have to consider that *our users* are not just fellow developers, nor programmers or sysadmins. We have to take into account that *our users* might include office secretaries, small-town homemakers, 10-year-old children, 80-year-old grandmothers or maybe some third-world government officers.

So, when we say that we are going to be guided by the needs of our users, it's really a very spread spectrum of **needs** that we are supposed to be considering.

This is not an easy task to achieve. Certainly, a task that needs constant work and revision, since the needs of our users develop and change with time.

2   How Debian is working on the users' needs

Debian has always been an awesome distribution for the average sysadmin. The **stable** concept has made Debian a very reliable option for those who need a zero-downtime server.

Also, Debian has become a pretty nice distribution for the *power-users*, those that like experimenting with hardware and software, those that enjoy spending time in front of the computer, learning how to make it work, and how to make it work better.

But Debian is not yet being able to reach to the common users, those that want the computer and the programs to *just work*, without having to worry about configuration, administration or how the system works inside.

Many geeks I know would say that any person that does not intend to spend a bunch of hours in front of the computer until it works as expected, is not worth owning the box. But we know better, don't we?

Maybe we do, maybe we don't. The fact is that we are lacking in many user-related areas: hardware discovery, ease of use, internationalization, boot time, general software speed, etc.

The effort made by the **debian-installer** team in getting a fully internationalized installer, really simple and intuitive is awesome. But it's not enough, since after installing the whole system has to prove that it can be used and configured just as easily; and usually it doesn't.

3   Reasons for our failing

One of the main reasons in failing to meet user needs is probably that we don't use the same software that normal users do. We use **mutt** instead of **evolution** or **thunderbird**, we use **ion** or **window maker** instead of **GNOME** or **KDE**, we use **vi** or **emacs** instead of **OpenOffice.org**, etc.

Therefore, we don't really see what the users see. We don't care about those pieces of software that we don't use.

Also, we don't usually give the same importance to the programs a user would care more about. I.e. a library is more important than an end-user application; a bug in an end-user application triggers it's removal from the next-stable release, while a bug in a library that has dependencies triggers someone actually fixing it.

Another one of the reasons in failing is that to become a Debian Developer, a person has to handle enough English to be able to pass the NM process. That's good, but it's leaving behind a lot of people that don't speak English.

Since all DD's are required to understand English, we don't usually mind if a certain menu does not turn up in our own language, because we understand English, but a user that does not speak it will not understand the menus, and that is an important flaw.

In conclusion, the main reason of failing is that we don't put ourselves in the users' shoes. It's not like we should all go and start using evolution because of this, but at least it's important to acknowledge that those programs, no matter how ugly we find them, are a **must** for the normal users. And that **l10n** bugs are not to be discarded as unimportant.

4   What we could do to improve

We have a QA team that has to make sure that all of Debian stays at **Debian quality**, but there are too many packages and too few people in the QA team, so we -as developers, maintainers, users and human beings- have to work together, to assure that all the distribution (not only our packages) has the highest possible quality, and meets the needs of the broader user spectrum.

4.1 Knowing what the users want

We should really make more and more use of the popularity contest, so that we can acknowledge the programs that have a bigger user base and work in improving the quality of those. Right now, popcon is not installed in enough machines, the sample we have of the software installed is too developeloper-biased, we need to acquire better knowledge of what programs the users are really using.

If we knew what the users actually want, it would be easier to work on fixing those programs, and making them *release ready* along the whole release cycle.

4.2 Internationalization

It's important that we all make sure all the menus, docs, manpages, help messages, etcetera are internationalized in as many languages as possible, in as many packages as possible. You might think that this is upstream's responsibility, but since we are providing the programs to the users, we **need** to take the responsibility of making more complete translations as well.

Also, we need to make sure that all the basic programs correctly support the many encodings that are out there. This is to say, it's not sufficient to support French, Portuguese, German or any other latin1 language. We have to make sure the whole system works in Japanese, Arabic or Vietnamese.

This is a task that involves a larger group of people than just Debian Developers. There are language teams, and they usually work very hard to have as much software translated as possible. But when a maintainer ignores an l10n patch that has been sitting in the BTS for months, it's not good for the translators.

It's very important to acknowledge the works of the language teams, and treat the l10n bugs as really important bugs, not as wishlists.

It's not difficult to change our approach about this matter, starting from today. If you maintain a package which is not translated, or not completely translated, you should approach the language-team mailing-list and ask for help, today.

This could, of course, be automated. And the good thing is that we have the skills needed to automating this process and make it easier for everyone. The only thing left is to do it.

4.3 Documentation

A very important thing that is missing from our distribution is to make the information that is located in /usr/share/doc more accesible to the public. Users tend to be clueless, not only because they tend not to read the documentation, but also because they have no idea where it is.

Tools like **yelp** could help us. Or anything that makes the information be easy enough to find, no matter how newbie the user is. We need to integrate them and make them part of the whole system.

4.4   Package Management

Since users need to install software into their boxes, we need to provide a very simple way of doing it.

Simplifying Synaptic might be a way to do it. Expanding tasksel (to include a bit more granular tasks than today's) might be another.

The important point is to allow users to have a good, updated system, even if they don't want to spend hours in front of their boxes tuning them up.

4.5   Boot time

Work in making the system boot and shutdown more quickly is desperately needed. This is specially important when trying to reach the double-booting public.

A freshly installed Debian system, with the common tasks selected, takes a painful time to boot, and almost as much to shutdown. This should be fixed so that no processes are run at boot time when they are not really needed.

For example, exim is installed as a daemon even if the machine has no Internet access, therefore it takes around one minute to timeout; hardware is discovered again and again on every single boot; etc. These are not so easy tasks to fix, yet we should at least have a goal of decreasing boot time.

4.6   Speed

We have to work on making the programs run faster, on older hardware.

Since we don't develop the software, but only package it, this is one of the trickiest things. But we should keep in mind that not everyone in the world has the money to buy a new machine with lots of RAM and a fast processor. And do as much as we can to increase the speed and decrease the memory requirements of our software.

There are some things that we can do as a distribution. For example, we could include a different libc6 and a diferent libstdc++, and some other libraries, for each model of processor (not just one per arch).

4.7   Releases

The fact that Debian should release more often is much more than well known, and therefore will not be covered here.

However, it's important to keep in mind that we really need to think about it and make it happen.

5   How we could achieve these goals

When presented with this many things to improve, it's difficult to find a solution to all of them. However, it's not impossible to think up some ways of making progress for these areas.

5.1   Set global goals

Since working in a distribution is so different from working on a simple application project, there's a general feeling that we can't set a global goal.

We should leave that impression behind and set a global goal for us. For example: improve boot-time. Many different packages need to be worked on and changed to do this, so we need the collaboration of everyone involved. It's a global goal that needs to be addressed as a team. We **can** do it, we just need to have the initiative to go do it.

5.2   Involve more people

Many would agree that incorporating more than a thousand Debian Developers is too many people to coordinate and to make sure that everyone is working with the same quality of work in mind.

However, there are many ways of involving more people that do not mean increasing the number of Debian Developers, just the number of contributors. Some of them are described in the next sections.

Accepting and organizing ways for more people to help us out, might be the key of turning Debian into the universal distribution, or just staying at one of the many distributions.

5.3   Unify i18n-teams work

Becoming a translator is one of the first things people do when starting to participate in Debian. Ee need to take this into account, and make these people's time worth the effort.

One of the problems related to i18n is that this is not a unified task. There are people who work on debconf screens, people who work with the debian-installer, people who fix upstream translations, people who work on the web-page, etc.

We really need to unify the teams, so that no strength is wasted.

Resurrecting DDTP might be important to achieve this, but there could be other ways too. The most relevant point is to think and then implement the best way of unifying all the Debian translators.

5.4   Improve the BTS interface

We've been using the same Bug Tracking System for many many years now, we might be a bit attached to it, because of the amount of time it's been with us. But we need to ackowledge that the BTS needs our attention in order for us to do a better job at bug tending.

We need a better BTS interface, with many more tags and labels, so that it's easier to find bugs that a person might be able to contribute to. For example, it would be really nice to have a "language" (C, Python, Java, etc) tag for each package, or an "easy to fix" tag, or a "hardware related" tag, or a "tedious to fix", etc.

5.5 Advertise bugs that need help

If it were easier to tag bugs that need help (including the language skills required and the like), it would be possible to make a daily or weekly short list of the "Current bugs that need your help".

As people start sending patches to these bugs, advertise the names of those that helped solving most bugs lately. This could work like ad-honorem bounties.

In this way, we could incorporate more people into the bug fixing process give them some special acknowledgement for their work, and finish with better programs for our users.

5.6 Encourage team work and NMUs

With the birth of Alioth, many teams have been started, and a nice number of packages are co-maintained and therefore better maintained.

This should be the rule rather than the exception. All packages should be in Alioth, and they should all be co-maintained.

This is also a way of involving more people, that are not really Debian Developers: they can just have access to contributing in a certain package.

On the other hand, we all know that NMUs tend to be seen as an aggression, when they should be seen as a way of helping each other out. The only way that an attitude change towards NMUs can be achieved is realizing that we are all in the same team, working towards the same goal.

NMUs are teamwork. We have to keep that in mind, both when we get an NMU and when we do an NMU. We are all in this together.

6 Conclusion

Many more things can and should be done. The important part to keep in mind is that we want to make a better distribution for everyone. We want it to be good and usable, and the only way of doing it is by working together and helping each other achieve the best possible quality.