DPL platform

Stefano Zacchiroli
zack@debian.org
http://upsilon.cc/~zack

January 9, 2010

version 1.1

[Other formats available: .html, .pdf, .txt.]

Executive summary

Hi, I'm Stefano Zacchiroli—mostly known as *Zack*—and I'm running for DPL. The **main points** of my platform are as follows:

- *I intend to be a* present *DPL, both in discussions and as the responsible of the project agenda.*
- *I will provide a constant stream of DPL activity news, to be frozen and posted* monthly *to* `d-d-a`*.*
- *To resolve the impasse of vocal minorities, I will apply mechanisms that make rough* consensus *emerge when it exists.*
- *I will push for more* gradual *and* rewarding *access paths to Debian.*
- *I will push to diminish* strong package ownership *when it conflicts with package quality.*
- *I will do my best to support, with money and other resources, contributors get-togethers.*

The reminder of the platform provides background information about me (Section 1), describes the above points in more detail (Section 2), and highlights more specific plans for my term (Section 2.3). Rebuttals can be found in Section A.

1 Introduction

1.1 Who am I

I became a Debian Developer (DD) in March 2001, shortly after the current NM process was introduced. My Debian involvement has been through two distinct phases. In the first one I only cared about my packages, ignoring the community: no IRC, no `-devel` subscription, etc. Then, at LinuxTag 2004, I discovered Debian as a *community* and got fascinated by it, gradually increasing my involvement in the project.

Technically, my main activities during all these years have been:

PTS I've been co-maintaining the Package Tracking System (PTS) for the past 3-4 years, contributing day to day maintenance as well as more significant changes. Details are available on my blog.

Vcs-* fields I've made popular the notion of VCS (Version Control System) fields in `debian/control`, and I wrote `debcheckout`.

Quality Assurance ... more generally, I've joined the QA team because I enjoy thinking of the project as a whole and looking for new solutions (e.g. UDD) to our persisting problems.

OCaml Better support for the OCaml programming language was my motivation to join the project. I've contributed forming the Debian OCaml Task Force in which I've ranged from the newbie, to leading activities. The team nowadays maintains more than 120 source packages with insanely intricate inter-dependencies, and complex transition needs.

Packaging Obviously, I've maintained (and continue to maintain) several packages. Beside OCaml-related packages, I'm proud of contributions to `devscripts`, `vim` (in spite of my recent betrayal `:-)`, and `python-debian`.

Believing in the community as the real strength of Debian, I'm a regular attendee of DebConf and, time permitting, of any other face to face meetings, like the Extremadura work ones.

In real life, I'm a computer science researcher, currently working as a post-doc in the PPS laboratory, University Paris Diderot. This laboratory is somewhat of a Debian contributors nest of the French Debian community; our coffee breaks frequently turn into exciting Debian discussions. I mainly work on the Mancoosi project, where we apply techniques such as formal methods to the study of FOSS distributions, including but not limited to Debian, and to the improvement of upgrade planning algorithms. Mancoosi is the successor of EDOS, which was the source of various tools used for Debian QA, like `edos-debcheck`, running periodically at `edos.debian.net`.

1.2 Why I am running for DPL

Being the DPL is about two distinct aspects: the institutional role and a set of extra goals the prospective DPL wants to pursue during the term. This distinction matters.

The institutional role is described in our constitution and is about three tasks: ordinary and extraordinary decision making, public relations with the outside world, and leading discussions inside the project.

My personal view of why we need a DPL builds on the observation that we operate as a do-ocracy: anyone willing to do things can decide what and how they are done, and no one can be forced to do anything. Given our size, we have the irresistible tendency of turning into an *imperfect* do-ocracy:

a. access restrictions get added to inhibit people doing potentially dangerous actions;

b. non-exciting tasks can rot because nobody is motivated enough to take care of them;

c. those in acquired positions may neglect their duties and lower the quality of the service they offer, because they do not admit they are no longer fit for the task.

The DPL has the duty, for a *limited amount of time*, to mitigate the imperfections of our do-ocracy:

a. spot overly constraining access restrictions which inhibit capable people to work, inducing inefficiencies and frustration;

b. find people willing to do non-exciting tasks for the project's global good (making sure they get credit for it);

c. assign people to key positions to improve service quality inside and outside the project.

The reward is the occasion to push for project-wide improvements by the only virtue of occupying the DPL position.

I want to challenge myself to be a transparent and present DPL, and to improve the mechanisms of our project. That is why I'm running for DPL. (Actually, I've also been *asked* by some of you to run, so the blame for bothering you in reading all this does not fall exclusively on me `:-)`

2 My goals

DPL institutional tasks deal with decision-making in situations that are, in general, unknown *a priori*. Hence, I present my goals as follows.

- First I give my *vision* encompassing key themes of Debian "politics". This, I believe, is the only way to give a rough idea of how I can react to unforeseeable scenarios.
- Then I present the *approach* I intend to apply in carrying on DPL institutional tasks.
- Finally I list some specific projects I intend to pursue as DPL.

2.1 Vision

Non-DD contributors The introduction of DMs (Debian Maintainers) has been a fortunate event for Debian. Some people argue that it has opened up our archive to packages of sub-standard quality. That might be true, but we also have (plenty of) sub-standard packages maintained by full-fledged maintainers who have lost their interest in Debian. The solution to both is *more QA*.

Through the DM process, many enthusiastic people have found their way into Debian, increasing our work force. In addition, the DM process induces a *more controlled process* to become a full-fledged DD, with greater insurance of serious commitment to Debian, and experience levels. All in all, the DM process is a more fine-grained access path to Debian than what we had before it, which enables us to give back to our contributors *gradually* through recognition. Contributing to Debian is no longer all-or-nothing, it is now way less frustrating for packagers, who previously might have turned their efforts to other projects.

We need to generalize the lessons learned from the DM process. We have a lot of potential valuable contributors out there. They just need better documentation about *how* to join. They simply demand something in exchange, to be proud of, that acknowledges their efforts. I do not have preconceptions on the different ways of achieving this (e.g., ACLs vs linearly increasing privileges), but we need to go in that direction. In doing so, we should also relax our implicit assumptions that *only* technical abilities matter in Debian. The truly "Universal Operating System" is *mainly*, not only, made of software; it is also made of translations, graphics, musics, etc.

I will push for more gradual *and* rewarding *access paths to Debian.*

Collaborative maintenance The introduction of the `Uploaders:` field is another example of a fortunate development in Debian. While it can, in principle, form teams of people with no clear responsibility for specific tasks, on average it works very well. It creates efficient technical spaces for collaboration on specific topics, and it also scales through (re)creating organizational structures with specific position and task-assignments.

Collaborative maintenance should not be mandatory (we do have several very efficient one-man-band developers), but should be our default. Packaging newbies should start in collaborative maintenance teams and gain experience there. Team member feedback can then be useful to take some of the weight off the shoulders of the AMs (Application Managers). Similarly, we should not accept one-man-band maintenance when it comes with de facto unmaintained packages (to be identified with suitable QA activities). In those cases, we should suggest—or even force if needed—collaborative maintenance. It can provide a more acceptable exit strategy than public, yet useless, shaming.

I will push to diminish strong package ownership *when it conflicts with package quality.*

Vocal minorities Our mailing lists have substantially improved over the last 5 years. Every now and then though, they get polluted by (apparently) very vocal minorities capable of polarizing discussions, which is neither productive, nor fun. Given how we are attached to our community, we sometime take part in such discussions, inevitably burning out (how frequent are temporary VAC messages due to this? Too much). My take:

> *Freedom of speech: good.*
> *Vocal minorities holding hostage the silent majorities: bad.*

While we should consider—and have actually applied in the past—moderation measures to counter repeated community disrupting behaviors, we cannot take the risk of applying them only on the *assumption* that the posters actually are a vocal minority. Even though there are no optimal solutions for this kind of problems, technical devices can help. One such device I want to put into use are polls, as proposed by Jeroen van Wolffelaar and other years ago. The intention is to enable every DD to start mail-based, authenticated polls *à la* `devotee`.

Polls can give a reasonable idea where the community stands, without having to engage the whole GR (General Resolution) machinery. A poll can either make it clear to participants in discussions (or flame fests) that they are in disagreement with the rest of the project and better stop beating the dying horse, or indicate that they are on the right path.

To resolve the impasse of vocal minorities, I will apply mechanisms that make rough consensus *emerge when it exists.*

Face-to-face meetings Meetings are essential to improve the quality of collaboration within the project, no matter how good we are in communicating digitally. Get together face to face, hack for hours, do stuff together, and get back home. The day after, your remote collaboration will be better.

Helping the organization of meetings is a wonderful way of spending Debian money or other investing resources, such as specifically appointed people. Luckily we do have DebConf, which is organized by a very efficient specific team. Nevertheless it should not be the only get-together, and we should push more for local meetings, employing our resources for that. I see as perfectly reasonable supporting economically the trips of active contributors when that will make possible joining others to work on specific topics.

A simple metric for deciding when to do that and when not, beside the required amount of money, can be the number x of other developers asking for that. If and *when* money will become an issue, I do not see any showstopper in organizing specific sponsoring campaigns.

I will do my best to support, with money and other resources, contributors get-togethers.

Derivatives We are part of the FOSS ecosystem, in which patches flow both downstream and upstream. We have promised to give back to the free software community, yet sometimes we fail to do so. Initiatives like the recent patch tracker by Sean Finney are a way to ensure that all our changes are visible both to downstream and upstream.

Our derived distributions (AKA *derivatives*) have us as theirs upstream, and are in a similar situation. We cannot *force* them to give back to us, because our promises are not necessarily theirs, still we should:

1. be exemplar in our giving back practices, for example by tracking publicly our efforts in sending patches upstream;

2. make as easy as possible to give back to us, for example by participating in cross-distribution initiatives which make maintainers know each other and share distributed VCSs.

Doing both will strengthen our give back *requests* to derivatives that we should not stop posing.

2.2 DPL ↔ project interaction

2.2.1 Being present

As a DD, I have often suffered from a perceived DPL "absence". Maybe it was just me not being pro-active enough to ping the DPL on IRC or email and ask, but I consider it a DPL duty to communicate his or her presence. That comes in two forms: one is to **lead discussions** among developers, as prescribed in the constitution, something I have rarely seen. While we do not want that by default (no need for a "post-in-every-thread" DPL), I will try to be present in "hot" discussions (vague on purpose) which concern the organization and the big picture of the project. I will also encourage seeking the DPL opinion on specific topics, by pinging me explicitly.

The DPL opinion can also be a reasonable first attempt to solve a conflict; if it fails, we do have other mechanisms to settle it. In this respect, I believe my personal qualities can be useful for the project: I am thoughtful, listen to others, and open to be convinced by good arguments.

More generally, managing the **project agenda** is something that should be expected from the DPL. Managing the agenda means having a road map of topics the project should *consider* with in some time frame. The DiscussionsAfterLenny page is an example of such need. The DPL should ensure the project has similar agendas to avoid forgetting about important topics to remember them, say, just before a release.

Management also means keeping track of what happened. The DEP proposal—started by myself, Adeodato Simó, and Lars Wirzenius—was an attempt to deliver a device to achieve that: no excessive extra burden induced, but a workflow to remember what is the status of "large" project-changing proposals.

We should expect the DPL to take care of "orphaned" DEPs by reassigning or driving them in first person. DEPs have not taken off due to the lack of some technical bits and of representative examples. I will ensure that we give a try to DEPs, or similar devices, to check whether we can finally have a sane choice between hyper-formal decisions by the mean of GRs and folklore decisions which too often resemble no decisions at all.

I intend to be a present *DPL, both in discussions and as the responsible of the project agenda.*

2.2.2 Transparency

Another way for a DPL to be present is to disclose *periodically* what she or he is doing; let's call it "transparency". Recently we have got better, but only 4 or 5 "bits from . . . " mails during a DPL term is still too little for a role which is meant to represent such a big and diverse project.

I'm sure that a given number of DPL activities are not suitable for disclosure, whether they are personal to some, plain boring, or otherwise should not be archived forever on the Web. I'm also aware that writing from scratch a "bits from DPL" mail can be time consuming and possibly intimidating.

The solution I will implement is to have some constant **feed of DPL activity news**. "Feed" as a concept, the implementation can vary: an IRC channel, blogging or micro-blogging, a wiki page `BitsFromTheDPL` handled *à la* `DeveloperNews`, posts to `-private` for sensitive material, etc. No feed activity will mean no DPL activity and the right for DDs to complain and demand

explanation. I believe that activities which are not yet ready to be disclosed *at all*, not even by censoring or rewording details, are scarce enough *not* to imply empty feeds.

The resulting feed will then be frozen each month and posted to `d-d-a`. If I fail to post and freeze twice, I will admit my failure by explaining the reasons and seeking opinions on how to continue the term (e.g., using a poll, which can also contain an option "resign, you do not communicate enough").

I will provide a constant stream of DPL activity news, to be frozen and posted monthly *to* `d-d-a`*.*

2.2.3 (no) DPL board

According to past DPLs, carrying over the DPL burden all alone *is* daunting. To counter that, I will be constantly seeking advice from others, on the basis of their project expertise. I do not however believe in the utility of a **DPL board**: the DPL term is too short to spend time with extra coordination hops, so I will not have a DPL board. For more specific task assignment we do have delegates, which is more than enough.

However, as life goes, the DPL is in this sense a single point of failure, and I will be looking for a **2IC** (second in charge, vice-leader). The duties of the 2IC will be: DPL backup (in case of unexpected absences), and help in communicating outside and inside the project (e.g., for the news feed). The 2IC will be appointed with an ordinary, term-limited, delegation. As the DPL, I will take full responsibility for 2IC actions.

I will not have a DPL board; I will rather look for a 2IC for backup and communication.

2.3 Specific plans

2.3.1 Clarify delegates

Remember: TINC (there is no cabal). Good news. Then:

1. all current delegates should be clearly stated at our organization page with reference to the delegation mail;

2. all people in core teams which predate widespread use of delegation should be officially delegated. This will avoid disparities between "young" team members who are delegates and "old" team members who live in an unclear limbo.

2.3.2 The website issue

We all want a sexier website, i.e., a website where people can find what they look for, and which does not make us look like Debian is an operating system of the 1980s. While work on the issue has been going on in the WWW team, we have not delivered visible improvements yet.

I intend to help the WWW team in order to solve the main issues within the term. While it will be pointless to set the precise technical goals in a platform, my intended course of actions will be as follows. All steps are meant to be performed in agreement with the WWW team.

1. Establish the minimal requirements for an improved website in terms of: accessibility, site structure, team work flow. Make those requirements *public* as well as a detailed plan of the work that needs to be done to implement them: we will not hide problems.

2. Send out a call for help to our community, looking for people willing to take over the job and complete it in a given time frame. (Yes, I know we are all volunteers, but we do take responsibilities and aim for deadlines, this issue is relevant enough to ask for it.)[1]

[1] FWIW, even though it was way simpler, but my experience with the PTS face lift is that our community is responsive to our deficiencies in Web presence. I asked for a new PTS CSS layout, got several replies, and chose one of them. It required some back and forth round trips, but is nothing different from the usual patch work flow.

3. Make sure the takers, if any, have access to the needed resources and that they get credit for their attempt and, hopefully, success.

If all this fails, we will have an emergency plan. For instance, we can consider an external bounty (i.e. nobody involved in Debian can take it) to realize what is missing, under the supervision of the WWW team. We regularly pay for services we are unable to produce by ourselves, the website should be no difference.

2.3.3 Core teams

Our core teams have recently improved a lot, in terms of manpower and communication. Kudos for that go to Steve McIntyre and Sam Hocevar (the last two DPLs) and of course to the team themselves. Nevertheless some teams are still short, or so it seems from the outside. Adding members makes a team more tolerant to absences, and also helps to prepare the next generation of contributors. The distinction between assistants and regulars in several teams seems to work very well in that respect.

My naive intention would be to bring all core teams to at least three members and to push for campaigning for assistants when there are no well-established procedures for team joining. Also, by looking at our organization page it looks as if several teams are either not active anymore, or are very bad in communicating what they are doing. For their own good, involved people should be encouraged to replacement and something to work on that is more fun for them.

All these changes however should be attempted first with team agreement, to not bother potentially understaffed yet very efficient teams. I will start from the precious team reviews gathered by Steve McIntyre last year[2] to drive re-staffing and procedural changes in order not to bother properly working teams.

Additional info

Time commitment

Being DPL is challenging ; for it to succeed the job must be taken seriously. For the duration of the mandate I will therefore put on hold my other Debian tasks; it is an obligation towards habitual co-workers and a fair deal to avoid burning out. I'm not the only responsible person in most of my current Debian duties and I will leave behind efficient teams, so I'm confident the tasks will not remain unattended. The remaining parts are a handful of packages which will need new maintainers.

On the same topic and for the sake of clarity: some DPL candidates have in the past declared their ability to act as DPL full-time. I cannot grant that. What I offer is my Debian time, to be diverted to DPL tasks. Luckily though, I work at a university where I have a very FOSS-sensitive boss. I have the freedom to reorganize and possibly shrink my schedule both for emergencies, and for longer activities such as traveling to deliver Debian talks. Like most of us, I'm generally available on IRC, phone, etc.

Live platform

Beside rebuttals, the DPL platform is fixed once and for all at the end of the nomination period. While that makes perfect sense in the context of a vote, it makes it hard for candidates to communicate their positions on important topics, because of the volume of information the vote process generates. Hence, if noteworthy topics will come up during the campaigning, I will summarize my positions about them at `http://www.upsilon.cc/~zack/hacking/debian/dpl-2009/`.

[2] with specific exceptions, if interviewed people do not feel like sharing the actual content with me

A Rebuttals

Steve McIntyre (93sam) & Luk Claes (luk) I have known Steve and Luk for quite a while now and I deeply respect what they both did for Debian in all these years. I acknowledge all the problems they (yup, "they", because they are de facto running in tandem) mention in their platform: communication, humbleness and asking for help, ... In fact most of them are general problems of all free software projects.

That notwithstanding, their platform lacks hints and strategies that explain *how* they are going to attack those problems: it is not enough to say, e.g.,

> I want to see improvements made where they are clearly needed

to actually deploy the improvements. Also, I'm missing the vision of the two candidates on relevant topics such as Debian membership, mailing list discussion quality, derivatives, ...

All in all I found that the main contribution to the decision of voting for Steve and Luk is the example set by Steve in the current DPL term. The platform, which is what a rebuttal is meant to comment upon, does not have much more to offer.

I acknowledge that Steve did some great work during the current DPL term, but most of his practical tasks are approaching completion. It would have been great to know what, if elected, he plans to work on *additionally*. Presenting such a "diff" is, in my opinion, the duty of the DPL currently in charge, when standing for re-election.