[Ksummit–discuss] [MAINTAINER TOPIC] Succession Planning: Is It time to Throw Linus Under a Bus?

Laurent Pinchart [laurent.pinchart at ideasonboard.com](#)
Sun Sep 9 13:56:35 UTC 2018

- Previous message: [[Ksummit–discuss] [MAINTAINER TOPIC] Succession Planning: Is It time to Throw Linus Under a Bus?](#)
- Next message: [[Ksummit–discuss] [MAINTAINER TOPIC] Succession Planning: Is It time to Throw Linus Under a Bus?](#)
- Messages sorted by: [date] [thread] [subject] [author]

```
Hi Mauro,

On Saturday, 8 September 2018 13:47:08 EEST Mauro Carvalho Chehab wrote:
> Em Thu, 06 Sep 2018 12:51:15 -0700 James Bottomley escreveu:
> > On Thu, 2018-09-06 at 21:47 +0200, Daniel Vetter wrote:
> >> On Thu, Sep 6, 2018 at 9:44 PM, James Bottomley wrote:
> >>> Since our fearless leader apparently can't even remember the dates
> >>> of the only conference he goes to, perhaps now might be a good time to
> >>> talk about how we'd run an orderly succession process (purely
> >>> theoretically, of course ...)
> >>>
> >>> I think a vote amongst the Maintainer Summit attendees might be the
> >>> way to elect a new leader, but others probably have different ideas.
> >>
> >> Why I single leader?
> >>
> >> This group maintainer ship thing ... it  works.
> >
> > I also note that group maintainership seems only to work for you in
> > DRM; most of the other subsystems seem to have single leader
> > hierarchical maintainership.
>
> If we're seriously thinking on have a plan for maintainership
> replacement with a group, we need first to answer:
>
>        Q: How many Kernel developers does it take to replace Linus?
>
> That's said, I doubt that the DRM model would work outside DRM
> subsystem: too many people with commit rights can be hard, specially
> if they all can touch the core.

Sorry, I can't agree here. I see no reason why the DRM model couldn't be used
by other subsystems. It's not the panacea, it will certainly not apply as-is
to any and every project without considerations of local peculiarities, but
there's also no reason why it would work well (or at least well enough, as I
have my doubts about a few aspects of the current DRM multi-committers model)
for DRM only and nothing else.

Ruling a multi-committers model out without seriously considering it first
seems to me the sign of a power struggle (and yes, advocating such a model is
also the sign of a power struggle to some extent). That's unfortunately
nothing new in the kernel and open-source in general.

When it comes to top-level maintenance I certainly would consider a proposal
for a 20+ committers model with a high level of worry, but a group
```

maintainership model should be at least considered in my opinion.

> *Btw, we tried a similar model in the past on media (by the time we were*
> *using Mercurial, about 10 years ago): all core developers had commit rights.*

That's not what DRM does. Not all developers have commit rights, trust
relationships have to be built first, and commit rights come with an
expectation of responsible behaviour. Mistakes happen, especially with new
committers who are not familiar enough with the system, but tooling and
official maintainers taking the blame and cleaning things help keeping the
problems under control. Commit rights work quite well as an incentive for
committers to not screw up, and they can be taken back in case of serious
problems.

> *It worked smoothly for years, until one of the developers decided to commit*
> *a very intrusive patch touching all drivers at the subsystem and stepping on*
> *everyone else feet.*

That seem to me like a sign of issues in the community that were not addressed
and left to rot, until something bad enough happened.

> *Handling it was really painful. If our official model would be a group*
> *maintainership, it would probably take months for us to put the tree on a*
> *sane state.*

Why so ? All it takes is one person to commit a revert (or even rebase if
needed), most likely after discussing the problem with co-maintainers.

> *I ended by using my maintainership status to revert the patch, returning the*
> *tree to a sane state and allowing the others to keep sending patches. Yet,*
> *it took months, even years for us to recover from the disaster. Also, due to*
> *the heated discussions, we end by losing several developers due to that.*

That would happen the same way without a multi-committers model. Disagreements
can lead to people leaving the project when no acceptable solution can be
found.

> *A triumvirate like x86 and arm might work, but I suspect that it would be*
> *easier to start with a more hierarchical model, like for example having one*
> *(sub-)maintainer for arch, another one for core and a third one for drivers.*
>
> *Also, if we want to consider the bus scenario and other disaster threats, it*
> *could be worth to consider having them geographically distributed (if*
> *possible even on different Countries).*

--
Regards,

Laurent Pinchart

- Previous message: [Ksummit-discuss] [MAINTAINER TOPIC] Succession Planning: Is It time to Throw Linus Under a Bus?
- Next message: [Ksummit-discuss] [MAINTAINER TOPIC] Succession Planning: Is It time to Throw Linus Under a Bus?
- Messages sorted by: [date] [thread] [subject] [author]

More information about the Ksummit-discuss mailing list