主　题:	Re: Do you think these developers will help you reduce your workload?
发件人:	[redacted] 2019-1-4 0:10:22
收件人:	"Xin Tan" <tanxin16@pku.edu.cn>

Hi,

On 1/3/2019 3:10 PM, Xin Tan wrote:
> [redacted]
>
> I am Xin Tan, a PhD. student from Peking University. Our team is doing a
> research about "how to reduce the burden of the Linux kernel maintainers".
> We have already obtained some results and hope to get your opinions. We
> sincerely hope that our results would help you.
>
> We notice that you are the maintainer of "ACPI" subsystem and you have
> signed off a large number of patches in the past two years. Your workload is
> ranked in the top 50 driver subsystems list, and you might feel busy about
> this work.

I don't really feel so busy in that particular respect.

>   We would like to introduce you the multiple-committer model,
> which could probably reduce your workload.

I am familiar with this model and had I thought it would have helped me,
I would have tried to use it already.

I don't think so, however.

> As you know, in traditional Linux workflow, only the maintainers have the
> right to commit the patches to the repositories from developers. The
> multiple-committer model, first adopted by "i915" subsystem in 2015, gives
> the commit right to some regular contributors (aka committers, usually they
> are driver engineers who do core changes). They can review and commit
> patches directly to the same repository as the maintainer. We evaluated the
> new model and found it could significantly reduce the maintainers' workload,
> latency, and overwork.
>
> The multiple-committer model runs well on i915 subsystem so far. However,
> not all subsystems are suitable to this model. There should be a relatively
> stable core team in the subsystem. The developers from this team are not
> only competent but also enthusiastic about community activities, e.g.,
> actively participating in patch review. They are trustworthy and may be
> elected as candidate committers. We considered the developers' ability and
> their reviewing relationship and used a graph theory method to select
> candidate committers for your subsystem.
> Here is the list of candidate committers.
>
> Rafael J. Wysocki, Len Brown, Peter Zijlstra, Borislav Petkov, Josh
> Poimboeuf, Juergen Gross, Andy Lutomirski
>
> What do you think of the multiple-committer model?

I don't think that it is suitable for the code I maintain.

> Do you want to apply it in the future?

I don't think so.

> How do you like these candidate committers?

I respect all of these people and I collaborate with all of them. I
don't think any of them will be interested in the role you are proposing
for them, however, and frankly I don't think I need help with applying
patches.

Code review is a different matter, but you don't need to be a committer
to do that.

> Wish to receive your precious feedback to us.

To me, the whole multi-committer model is misguided as it focuses of one aspect of code maintenance only.

If somebody likes it and wants to use it, good for them, but I don't.

Rafael