主　题:	Re: Do you think these developers will help you reduce your workload?
发件人:	[redacted] 2019-1-3 22:17:04
收件人:	"Xin Tan" <tanxin16@pku.edu.cn>

Hi! We already use a multiple committer model for amdgpu and radeon. I just coordinate sending the pull requests upstream via my tree to deal with the way some of our internal processes work inside AMD.

[redacted]

From: Xin Tan <tanxin16@pku.edu.cn>
Sent: Thursday, January 3, 2019 9:10:38 AM
To: [redacted]
Subject: Do you think these developers will help you reduce your workload?

[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 "RADEON and AMDGPU DRM DRIVERS" 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. We would like to introduce you the multiple-committer model, which could probably reduce your workload.

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.

Harry Wentland, Alex Deucher, Christian König, David (ChunMing) Zhou, Tony Cheng

What do you think of the multiple-committer model?
Do you want to apply it in the future?
How do you like these candidate committers?

Wish to receive your precious feedback to us.
Thank you!

Xin Tan