Towards a Linux Kernel Maintainer Handbook

Dan Williams

Linux Plumbers 2018

Intel OpenSource TECHNOLOGY CENTER

First a quick survey

- XMAS Tree
- Reverse XMAS Tree

- XMAS Tree

- Reverse XMAS Tree

- What?

This talk is not...

- ...the end state of a completed work, it’s the start of a conversation
- ...for contributors, but you’re free to join the conversation

Kernel Maintenance is Easy

- The Merge Window is as simple as “git request pull”
- Maintaining one subsystem is the same as any other
- If you go vacation, or are hit by a bus, there’s continuity

Kernel Maintenance is Easy?

- The Merge Window is as simple as “git request pull”

  It takes topic organization, cross-tree coordination, and soak time

- Maintaining one subsystem is the same as any other

- If you go vacation, or are hit by a bus, there’s a plan

Kernel Maintenance is Easy?

- The Merge Window is as simple as “git request pull”

- Maintaining one subsystem is the same as any other

  Maintenance processes are as diverse as the number of subsystems

- If you go vacation, or are hit by a bus, there’s a plan

Kernel Maintenance is Easy?

- The Merge Window is as simple as “git request pull”

- Maintaining one subsystem is the same as any other

- If you go vacation, or are hit by a bus, there’s a plan

  What plan?

Agenda

Why a “Handbook”?

Why me?

What does the community want / need in a Handbook?

Why a “Handbook”?

- More reference less rules
- A la carte
- Repository for knowledge sharing
- Capture maintainer feedback from upstream

Why a “Handbook”?

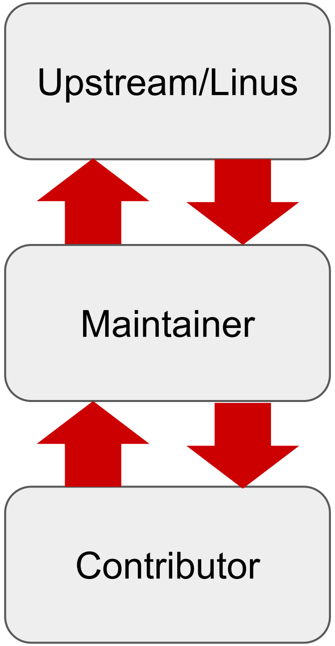
- More reference less rules
- A la carte
- Repository for knowledge sharing
- Capture maintainer feedback from upstream

...the most pressing reason?

Pain.

Pain... in all directions

- Maintainer silence
- Unaligned contributor expectations
- Preventable maintenance mistakes



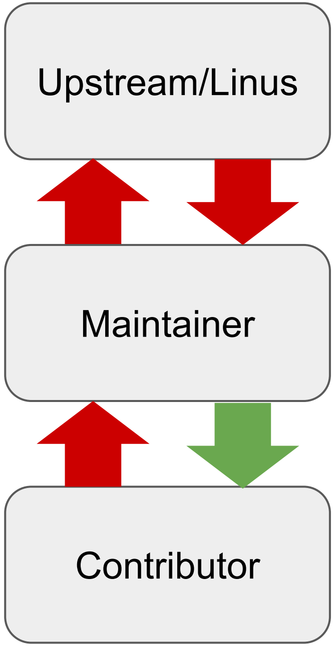
Upstream/Linus

Maintainer

Contributor

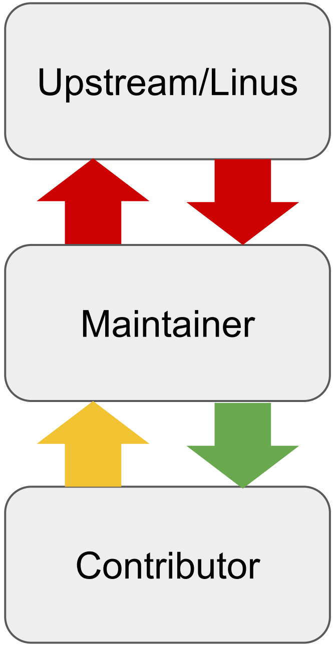
Pain... in all directions

- Maintainer silence
  - Advertise an SLA
  - Identify trusted reviewers
- Unaligned contributor expectations
  -
  -
- Preventable maintenance mistakes
  -
  -



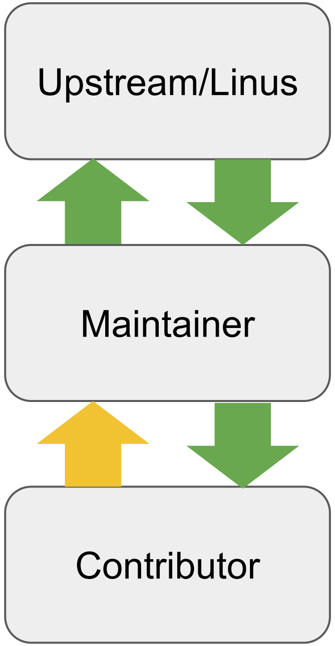
Pain... in all directions

- Maintainer silence
  - Advertise an SLA
  - Identify trusted reviewers
- Unaligned contributor expectations
  - Pass the test suite
  - Consult the resend cadence
- Preventable maintenance mistakes
  -
  -



Pain... in all directions

- Maintainer silence
  - Advertise an SLA
  - Identify trusted reviewers
- Unaligned contributor expectations
  - Pass the test suite
  - Consult the resend cadence
- Preventable maintenance mistakes
  - Share lessons of successes / tooling
  - Share lessons of your screw ups



Why Me?

Oh the wonderful places you’ll go...

- ARM
- MD/RAID
- DMAENGINE*
- SCSI / LIBSAS
- AHCI / LIBATA
- XHCI / USB
- NVDIMM*
- MM
- X86
- DAX

Oh the wonderful places you’ll go...

- ARM
- MD/RAID
- DMAENGINE*
- SCSI / LIBSAS
- AHCI / LIBATA
- XHCI / USB
- NVDIMM*
- MM
- X86
- DAX

Personal Lessons

- A contributor’s experience can be widely divergent from subsystem to subsystem.
- Subsystems with a team are healthier.
- I’m guilty of some of the same behaviors as a maintainer that caused me stress as a contributor.

What does the maintainer community want in a Handbook?

Handbook Goals

- Collect a gallery of “after the blow up” mails
- Provide a documentation framework for maintainers to drop-off contributions
- Establish Subsystem Maintenance Profiles

About those follow up mails

That time I learned how to merge

On Sat, Jun 9, 2018 at 8:17 AM Dan Williams <dan.j.williams@intel.com> wrote:
>
> Well, crap. I've been doing it the wrong way for a while.

... you're not the only one ;*(

I only really catch it when it's very obvious, like it was now when
the last merge was just before so it stood out like a sore thumb when
I looked at the resulting "git log".

... or then I catch it in the (happily rare) case when a merge causes
issues, and then I curse.

A framework for contributions

In fact the Maintainer Handbook already exists, thanks Tobin and Jon!

A framework for contributions

In fact the Maintainer Handbook already exists, thanks Tobin and Jon!

...although no other contributions since, let’s change that.

Subsystem Profile

- Describe maintenance of your sub-system as a “bus-factor” document
- Document the policies that each subsystem typical decides differently

Patches or Pull requests
Last day for new feature submissions
Last day to merge features
Non-author Ack / Review Tags Required
Test Suite

Trusted Reviewers
Resubmit Cadence
Time Zone / Office Hours
Checkpatch / Style cleanups
Off-list review

Others?

Call To Action

- Watch linux-kernel and linux-doc for a collection of Linus gems going into the Handbook

- Submit your own examples of upstream feedback to the gallery

- Submit a Subsystem Profile

Thank you