Trust and the Linux kernel

Greg Kroah-Hartman

gregkh@linuxfoundation.org
git.sr.ht/~gregkh/presentation-linux-trust

Disclaimer

Nothing in here reflects the opinion of the Linux Foundation or any other Linux kernel developer. It's all my personal opinion.

Open source software is more trustworthy than closed source software.

Open source software is more trustworthy than closed source software.

Because it can be audited by anyone at anytime.

Open source software is more trustworthy than closed source software.

Because it can be audited by anyone at anytime and fixed by anyone.

University of Minnesota “episode”

or

How to NOT do research on an open source community

University of Minnesota “episode”

Kernel Recipes 2022 talk

umn.edu “episode”

Proof that you can go back in time and audit code based on new information.

Trust

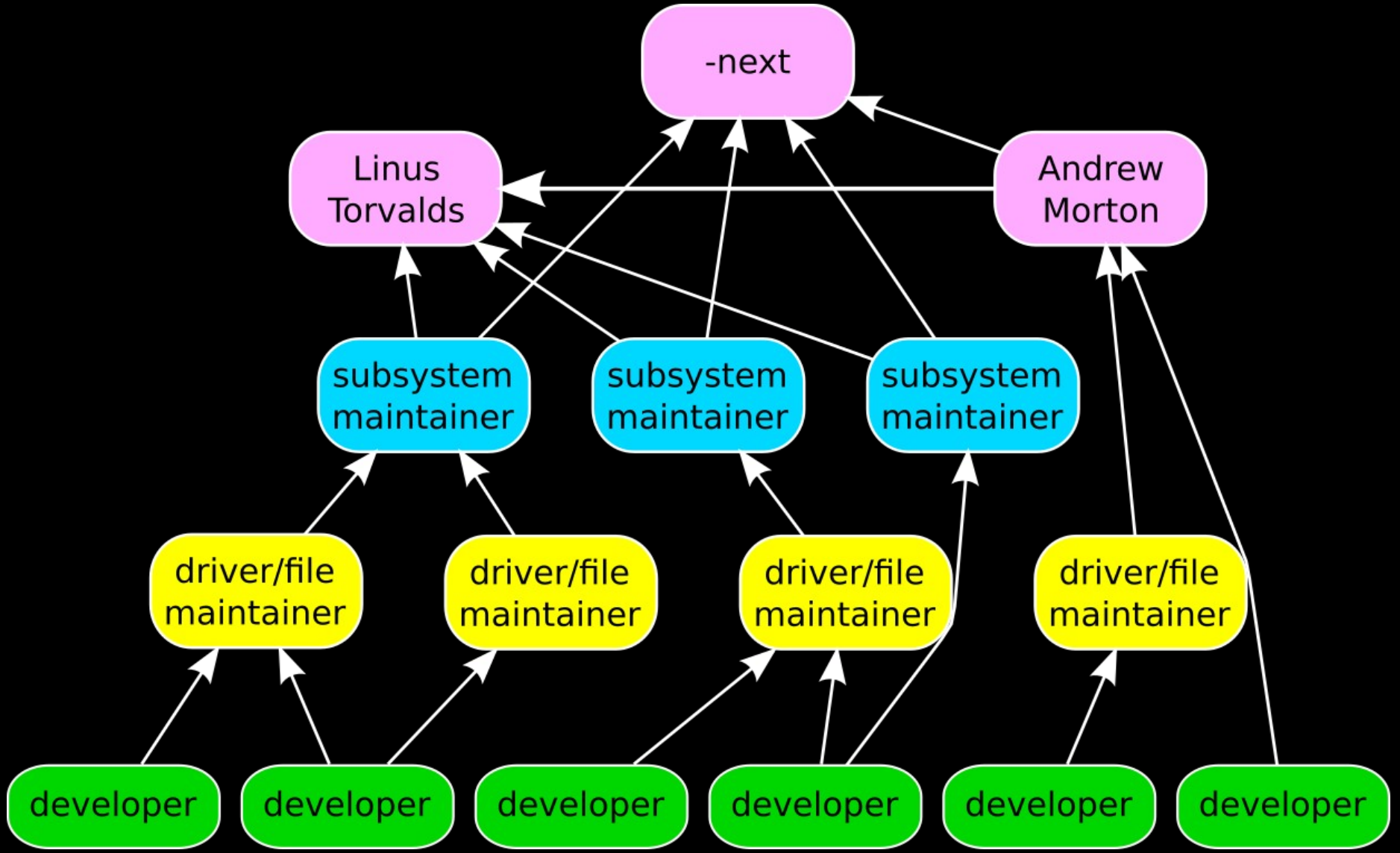
Trust

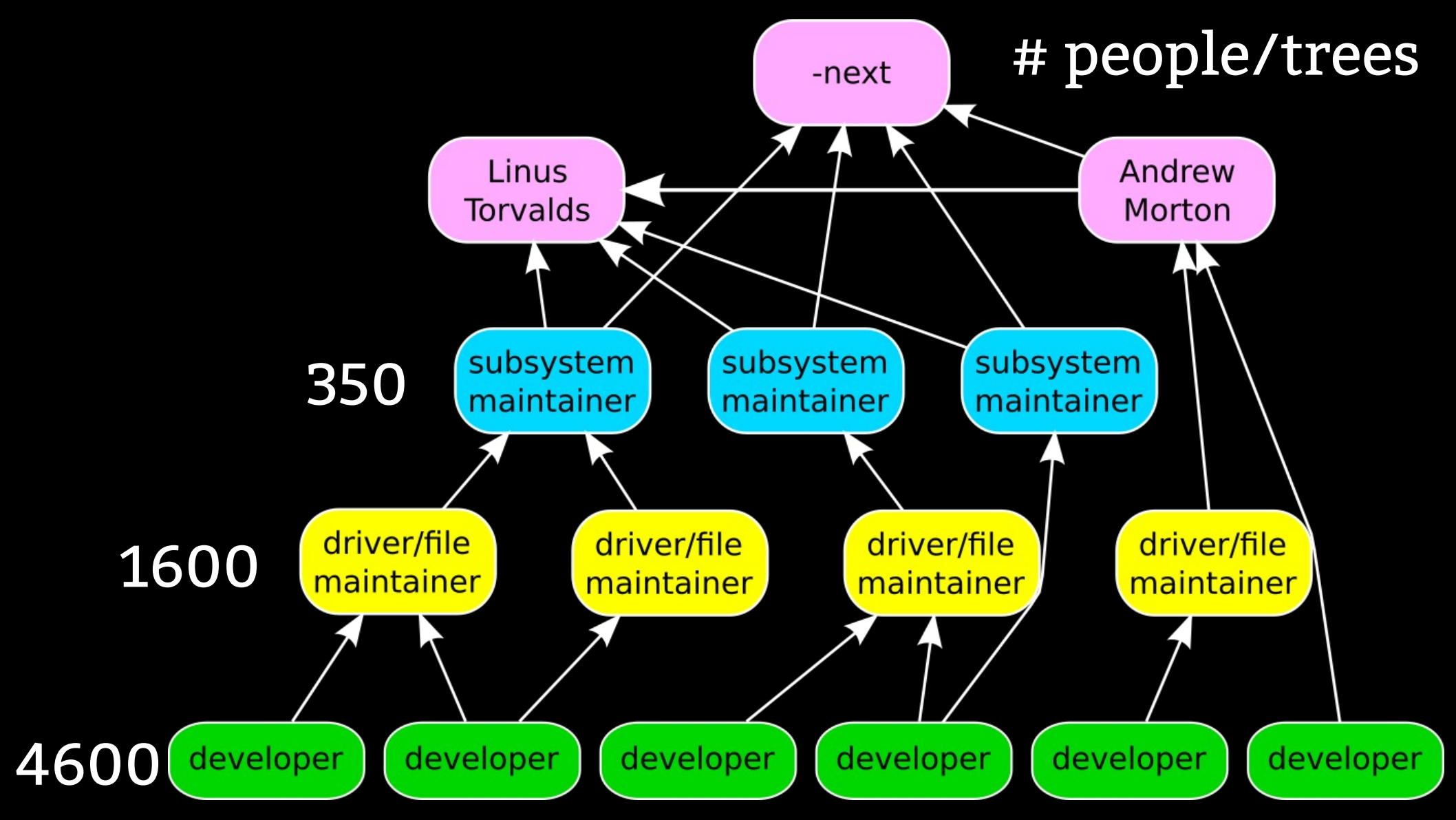
NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

Trust

“You need to verify all developers to ensure you know who they are.”


-next
Linus Torvalds
Andrew Morton
subsystem maintainer
subsystem maintainer
subsystem maintainer
driver/file maintainer
driver/file maintainer
driver/file maintainer
driver/file maintainer
developer
developer
developer
developer
developer
developer


people/trees
-next
Linus Torvalds
Andrew Morton
350
subsystem maintainer
subsystem maintainer
subsystem maintainer
1600
driver/file maintainer
driver/file maintainer
driver/file maintainer
driver/file maintainer
4600
developer
developer
developer
developer
developer
developer

Development stats for 2021

79.662 total commits

Fixes for 2021

79.662 total commits
13.587 commits marked with Fixes: tag

17% are fixes

Fixes for 2021

79.662 total commits
13.587 commits marked with Fixes: tag

17% are fixes

Found after commits hit subsystem trees

Fixes for 2021

79.662 total commits
13.587 commits marked with Fixes: tag

17% are fixes

Found after commits hit subsystem trees

26% of the fixes were for issues before -final

2021 changes

- ~ 12% of all commits were fixes for problems in older releases.

2021 - Top developers

Christoph Hellwig	960 (1.2%)
Lee Jones	737 (0.9%)
Andy Shevchenko	704 (0.9%)
Mauro Carvalho Chehab	642 (0.8%)
Pavel Begunkov	624 (0.8%)
Vladimir Oltean	600 (0.8%)
Sean Christopherson	597 (0.7%)
Colin Ian King	573 (0.7%)
Arnd Bergmann	535 (0.7%)
Geert Uytterhoeven	487 (0.6%)

2021 - Top fixers

Dan Carpenter	340 (2.5%)
Arnd Bergmann	227 (1.7%)
Colin Ian King	165 (1.2%)
Sean Christopherson	160 (1.2%)
Vladimir Oltean	143 (1.1%)
Christophe JAILLET	142 (1.0%)
Randy Dunlap	140 (1.0%)
Geert Uytterhoeven	132 (1.0%)
Johan Hovold	125 (0.9%)
Eric Dumazet	119 (0.9%)

2021 - Authors of commits fixed

[masqué]	207 (1.5%)
[masqué]	161 (1.1%)
[masqué]	121 (0.8%)
[masqué]	109 (0.8%)
[masqué]	93 (0.7%)
[masqué]	89 (0.6%)
[masqué]	77 (0.5%)
[masqué]	75 (0.5%)
[masqué]	69 (0.5%)
[masqué]	67 (0.5%)

2021 - Authors of commits fixed

[redacted]	207 (1.5%)
[redacted]	161 (1.1%)
[redacted]	121 (0.8%)
[redacted]	109 (0.8%)
[redacted]	93 (0.7%)
[redacted]	89 (0.6%)
[redacted]	77 (0.5%)
[redacted]	75 (0.5%)
[redacted]	69 (0.5%)
[redacted]	67 (0.5%)

Over time, the most prolific developers will write the most bugs.

Over time, the most prolific developers will write the most bugs.

So make it easy to find and fix those bugs.

Lifecycle of a kernel change

Lifecycle of a kernel change

- Submit through email

Lifecycle of a kernel change

- Submit through email
  - Development is done in public
  - No central authority (i.e. login)
  - Can be verified
  - Lowest possible barrier to entry

Lifecycle of a kernel change

- Submit through email
- Change is reviewed, and rejected

Lifecycle of a kernel change

- Submit through email
- Change is reviewed, and rejected
  - All through email
  - Patchwork instances to check status

Lifecycle of a kernel change

- Submit through email
- Change is reviewed, and rejected
- Resubmit through email

Lifecycle of a kernel change

- Submit through email
- Change is reviewed, and rejected
- Resubmit through email
  - Properly document your changes

Lifecycle of a kernel change

- Submit through email
- Change is reviewed, and rejected
- Resubmit through email
- Average change takes 3 attempts

Lifecycle of a kernel change

- Submit through email
- Change is reviewed, and rejected
- Resubmit through email
- Average change takes 3 attempts
  - Some less (1), some more (25+)

Lifecycle of a kernel change

- Submit through email
- Change is reviewed, and rejected
- Resubmit through email
- Average change takes 3 attempts
- Every email submission is tested by bots

Lifecycle of a kernel change

- Submit through email
- Change is reviewed, and rejected
- Resubmit through email
- Average change takes 3 attempts
- Every email submission is tested by bots
  - and often automatically rejected

Date: Fri, 5 Aug 2022 05:38:30 +0800
From: kernel test robot <lkp@intel.com>
To: Johan Hovold <johan+linaro@kernel.org>, Greg Kroah-Hartman <gregkh@linuxfoundation.org>, Felipe Balbi <balbi@kernel.org>
Cc: kbuild-all@lists.01.org, linux-arm-msm@vger.kernel.org, linux-usb@vger.kernel.org, devicetree@vger.kernel.org, linux-kernel@vger.kernel.org
Subject: Re: [PATCH v2 6/9] usb: dwc3: qcom: fix peripheral and OTG suspend
Link: https://lore.kernel.org/r/202208050544.ijUhoUyB-lkp@intel.com

Hi Johan,

I love your patch! Perhaps something to improve:

[auto build test WARNING on usb/usb-testing]
[also build test WARNING on linus/master next-20220804]
[cannot apply to robh/for-next v5.19]
[If your patch is applied to the wrong git tree, kindly drop us a note.
And when submitting patch, we suggest to use '--base' as documented in
https://git-scm.com/docs/git-format-patch#_base_tree_information]

url: https://github.com/intel-lab-lkp/linux/commits/Johan-Hovold/usb-dwc3-qcom-fix-wakeup-implementation/20220804-231122
base: https://git.kernel.org/pub/scm/linux/kernel/git/gregkh/usb.git usb-testing
config: arc-randconfig-r002-20220804 (https://download.01.org/0day-ci/archive/20220805/202208050544.ijUhoUyB-lkp@intel.com/config)
compiler: arc-elf-gcc (GCC) 12.1.0
reproduce (this is a W=1 build):

If you fix the issue, kindly add following tag where applicable
Reported-by: kernel test robot <lkp@intel.com>

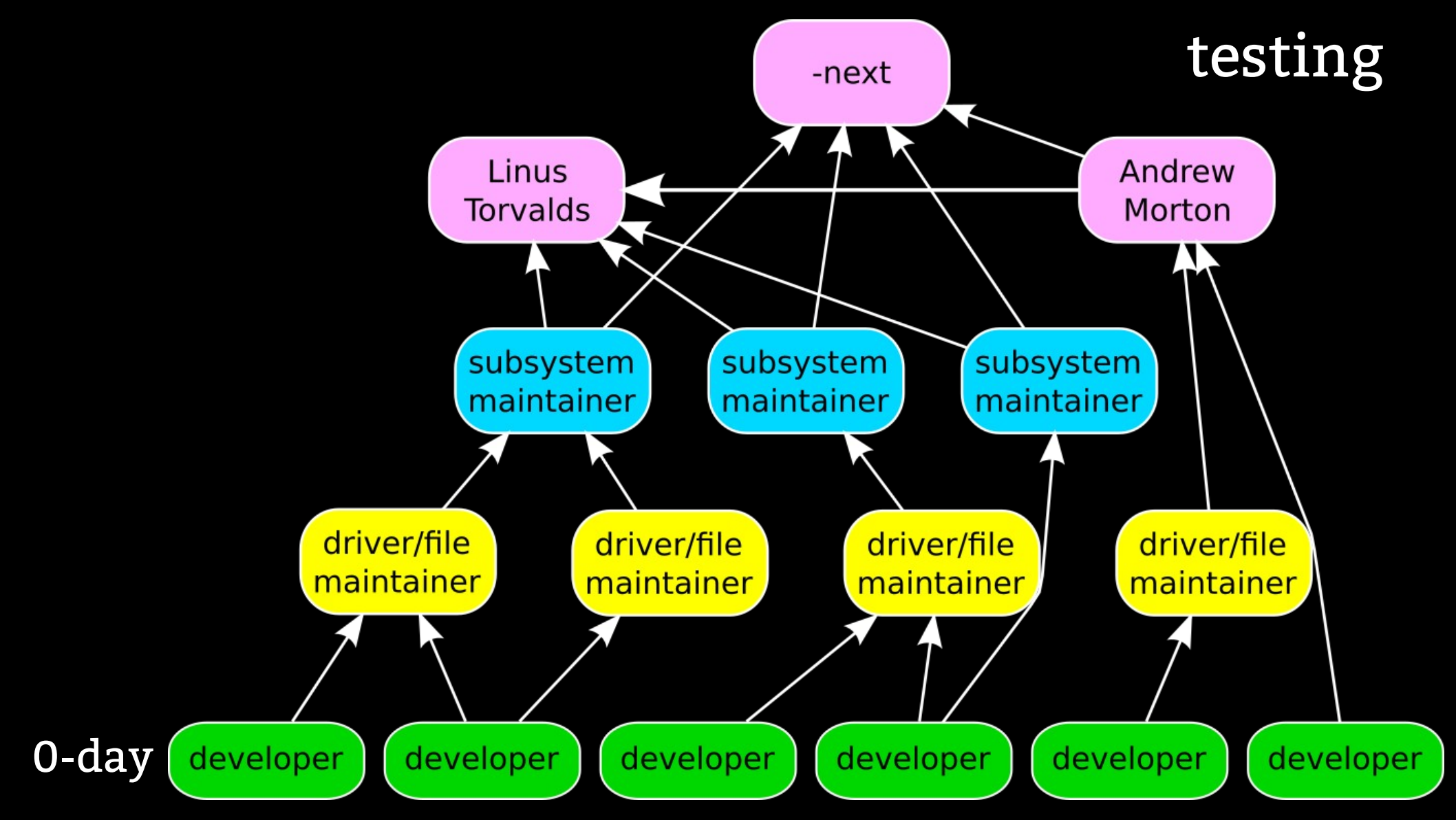
All warnings (new ones prefixed by >>):

0-day CI Kernel Test Service https://01.org/lkp

- Provides a one-hour response time
- Performs patch-by-patch tests
- Covers all branches of a developer tree
- Performs kernel build and static semantics-level testing
- Performs boot tests, functional, and performance tests
- Bisects code automatically when tests fail

0-day CI Kernel Test Service – benchmarks

- Virtual memory management
- I/O subsystem
- Process scheduler
- File system
- Network
- Device drivers

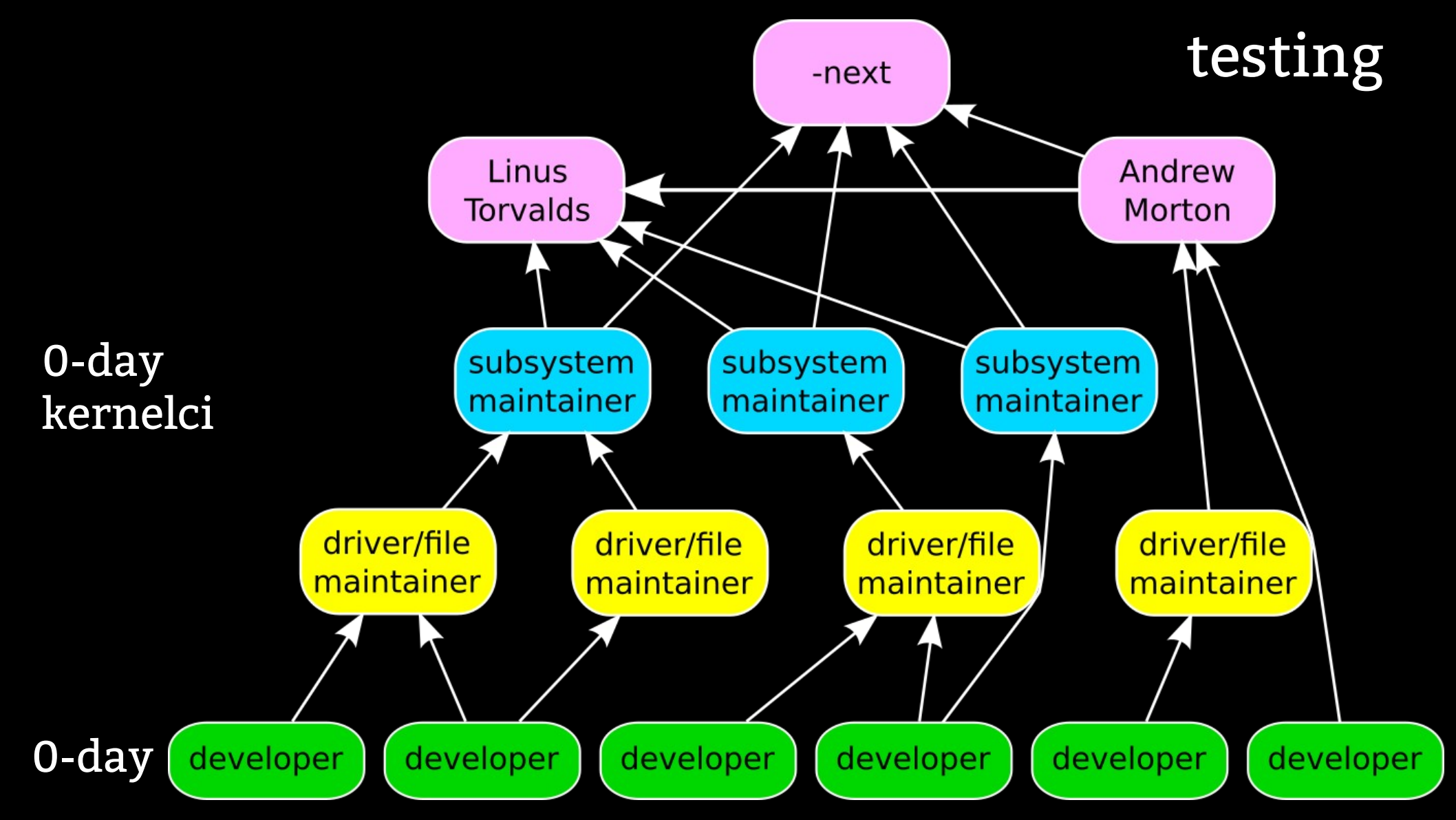

testing
-next
Linus Torvalds
Andrew Morton
subsystem maintainer
subsystem maintainer
subsystem maintainer
driver/file maintainer
driver/file maintainer
driver/file maintainer
driver/file maintainer
0-day
developer
developer
developer
developer
developer
developer

Lifecycle of a kernel change (cont.)

- Change accepted by maintainer

Lifecycle of a kernel change (cont.)

- Change accepted by maintainer
- Now the real testing starts


testing
-next
Linus Torvalds
Andrew Morton
0-day kernelci
subsystem maintainer
subsystem maintainer
subsystem maintainer
driver/file maintainer
driver/file maintainer
driver/file maintainer
driver/file maintainer
0-day
developer
developer
developer
developer
developer
developer

KernelCI https://kernelci.org/

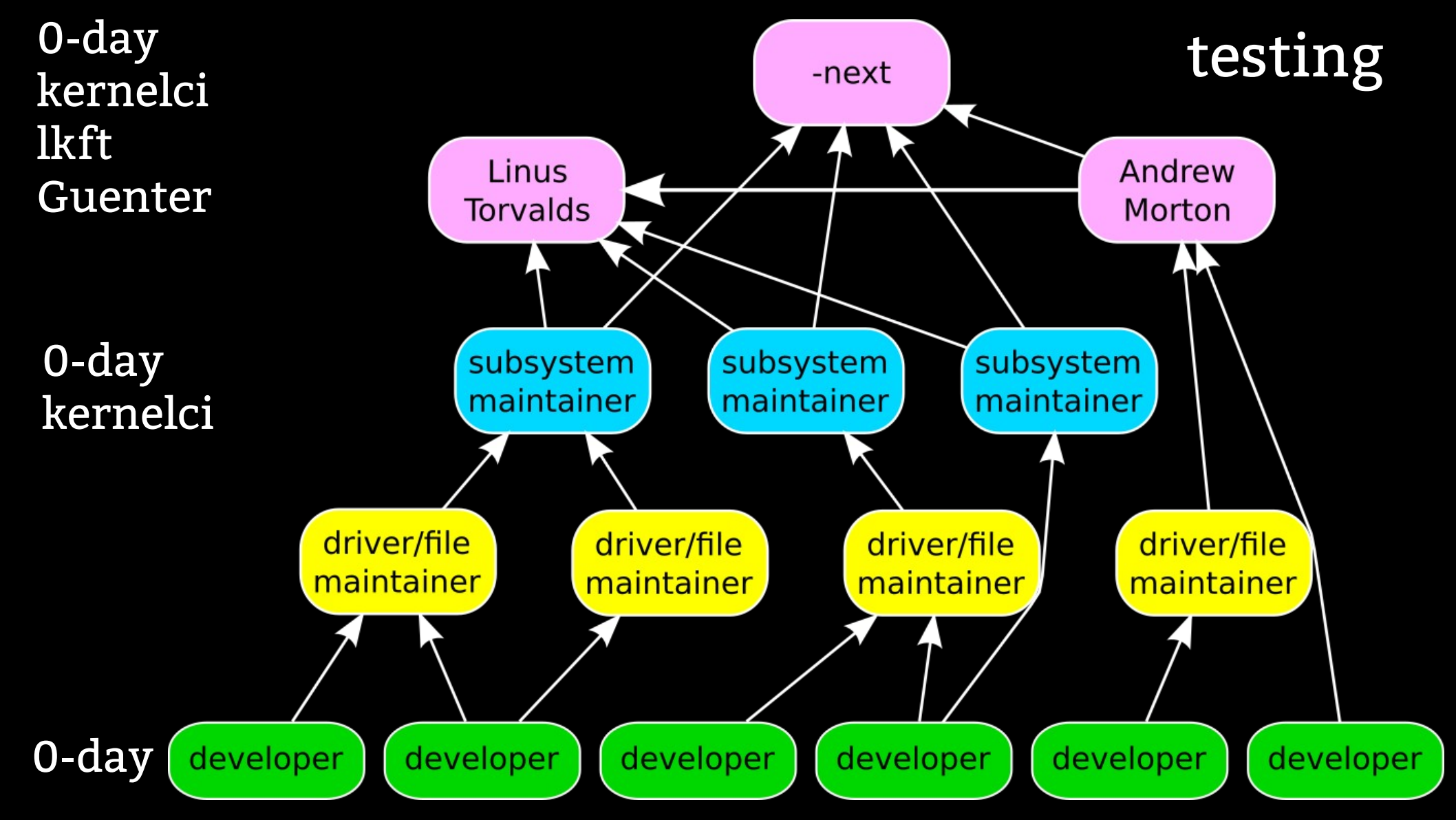
- Community-led test system focused on the upstream Linux kernel.
- Follows the open testing philosophy to enable the same collaboration to happen with testing as open source does to the code itself.

KernelCI Dashboard https://linux.kernelci.org/

- Tracks 62 different branches
- 13000+ different build/boot tests
- Thousands of different devices/architectures
- Different labs contribute from around the world

KernelCI Dashboard https://linux.kernelci.org/

- Common reporting framework (KCIDB)
- Allows any kernel testing system to submit results
- Already used by RedHat and Google test labs
- One unified location to see testing results


0-day
kernelci
lkft
Guenter
testing
-next
Linus
Torvalds
Andrew
Morton
0-day
kernelci
subsystem
maintainer
subsystem
maintainer
subsystem
maintainer
driver/file
maintainer
driver/file
maintainer
driver/file
maintainer
driver/file
maintainer
0-day
developer
developer
developer
developer
developer
developer

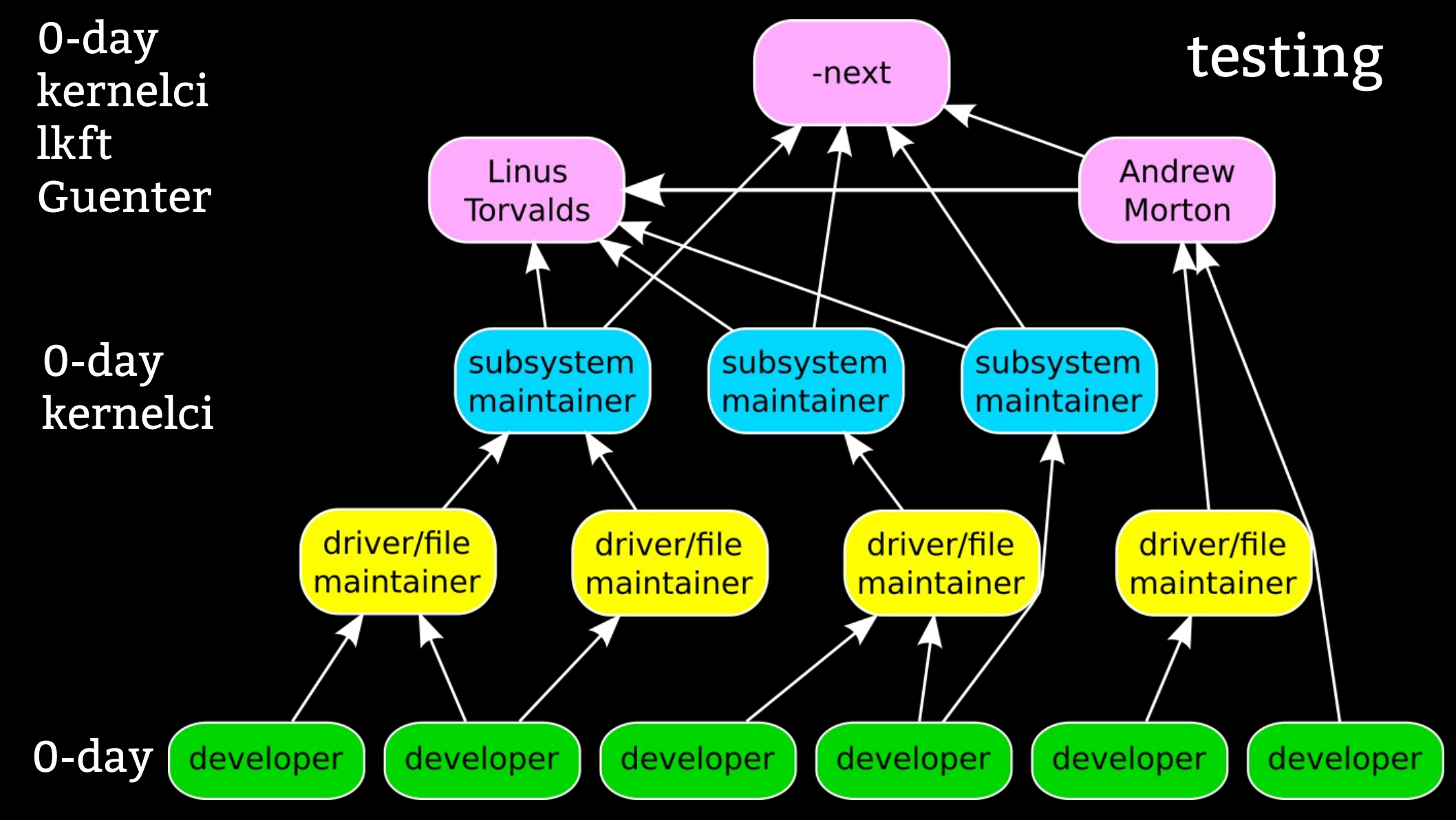
lkft (Linux Kernel Functional Testing)

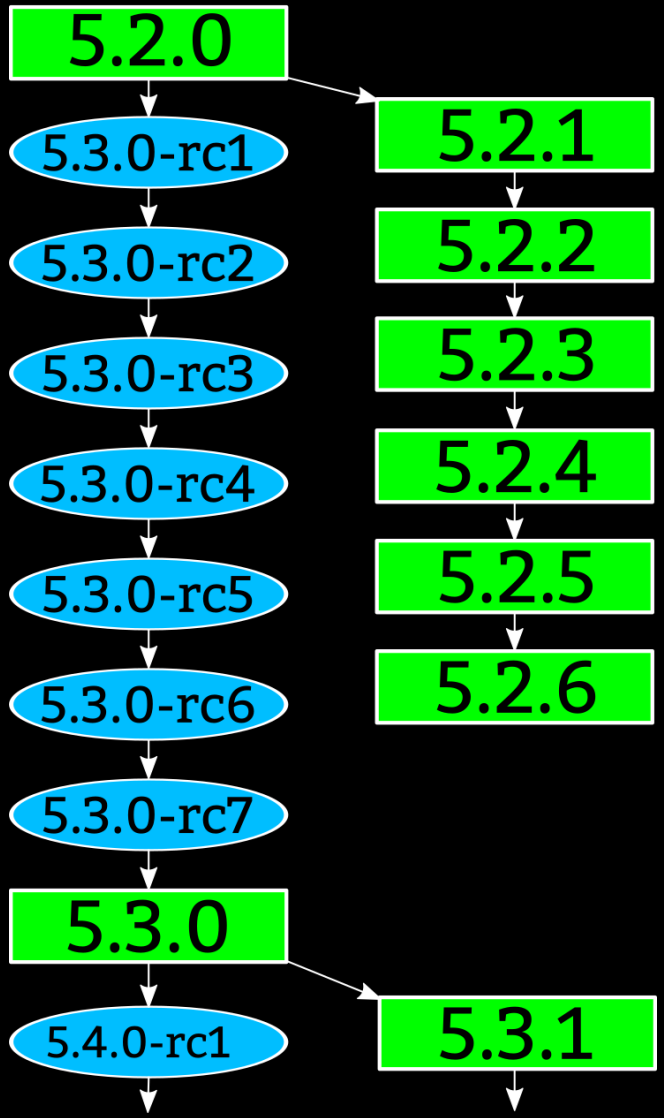
https://qa-reports.linaro.org/lkft/

- Sponsored by Linaro member companies
- Testing for stable and Linus's -rc releases
- Testing for linux-next
- Run by tuxsuite
- 125000+ tests for modern stable releases
- Subset of arches and configs and compilers

Guenter Roeck's test system

- Kernel maintainer doing it on their own time
- All supported kernel arches test-built (150+)
- Many build/boot tests for qemu targets (480+)
- Invaluable for stable kernel -rc testing
- Also tests linux-next and Linus's tree


0-day
kernelci
lkft
Guenter
testing
-next
Linus
Torvalds
Andrew
Morton
0-day
kernelci
subsystem
maintainer
subsystem
maintainer
subsystem
maintainer
driver/file
maintainer
driver/file
maintainer
driver/file
maintainer
driver/file
maintainer
0-day
developer
developer
developer
developer
developer
developer



Testing every release

kernelci
lkft
Guenter
Shuah
Android
Huawei
Nvidia
Debian
Fedora
Many others

Trust in Linux kernel development

Trust but verify.

Trust in Linux kernel development

Trust but test.

Trust in Linux kernel development

We trust not that you will always get it right, but that you will be there to fix it when you get it wrong.