LTSI Project update
Long Term Support Initiative

Tsugikazu SHIBATA, NEC

Hisao Munakata, Renessas

14, October 2014

Embedded Linux Conference Europe@Dusseldorf

Who am I

Tsugikazu Shibata, NEC

Project lead of LTSI

Member of CE Working group

Member of the board of the Linux Foundation

Key activities of LTSI
Long Term Support Initiative

- LTSI had been started 2011 in Prague
- After long discussion of industry demand and how to solved

- ☐ Provide a industry managed kernel and maintain Long Term

- ☐ Provide a common place for embedded industry to share information: Mailing list, Workshop …

- ☐ Provide place to support upstream activity : Help industry developer

Contents

- Linux Development process and status
- LTSI Development process
- LTSI New challenges : TEST Project

# Status of Latest Linux Kernel

- Latest Kernel : 3.17
  - Released October 5th
  - Lines of code : 18,864,486
  - Files : 47,490
- Current Stable Kernels: 3.16.5
- Current development kernel: 3.18-rc1 is not yet released

Linux Development process

- Just after the release of 3.n, two weeks of merge window will be opened for proposal of new features

- After 2 weeks of merge window, -rc1 will be released and the stabilization will be started

- 3.n+1 will be released when it becomes reasonably stable by some of −rcX released
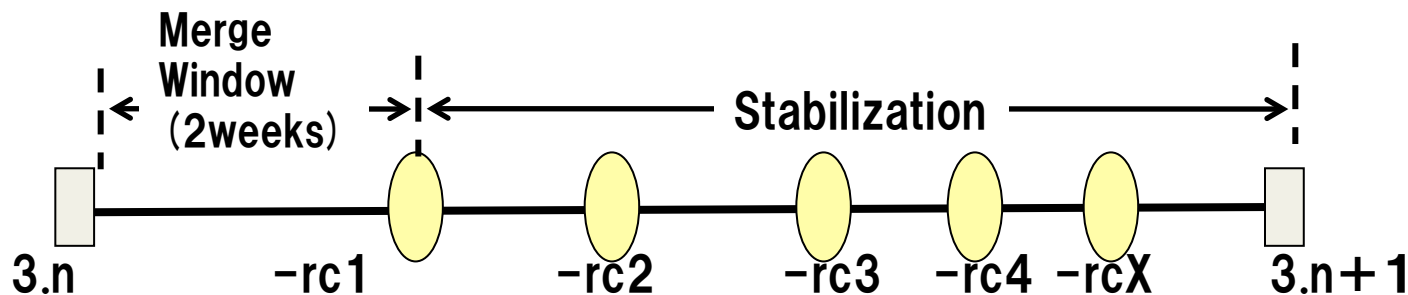
Kernel release cycle

- Release cycle of Linux kernel is about 60-70 days

Version	Release	Duration
3.0	2011-7-21	
3.1	2011-10-24	95
3.2	2012-1-4	72
3.3	2012-3-18	74
3.4	2012-5-20	63
3.5	2012-7-21	62
3.6	2012-9-30	71
3.5	2012-12-10	71
3.8	2013-2-18	70
3.9	2013-4-28	69

Version	Release	Duration
3.10	2013-6-30	63
3.11	2013-9-2	64
3.12	2013-11-15	74
3.13	2014-1-21	67
3.14	2014-3-30	68
3.15	2014-6-8	70
3.16	2014-8-3	56
3.17	2014-10-5	63

Stable kernel release
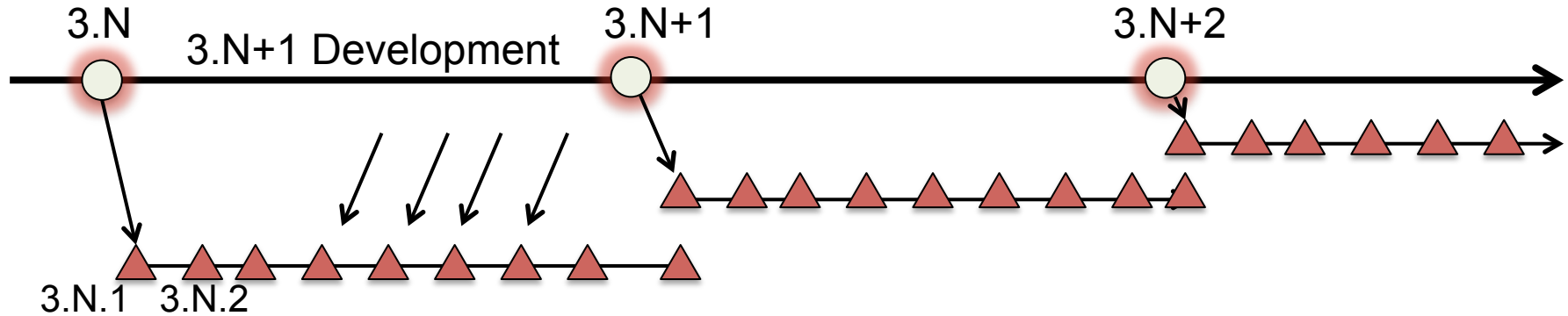
- 3 part version like 3.x.y

- Recommended branch for users who want the most recent stable kernel

- Contain small and critical fixes for security problems or significant regressions discovered in a latest development version

- Bug fixes/Security fixes are back ported from latest version

- Drop when next stable kernel were released

Stable_kernel_rules.txt

- Published in 2006. Strict rule to back port from latest version

See Documentation/stable_kernel_rules.txt

- It must be obviously correct and tested.
- It cannot be bigger than 100 lines, with context.
- It must fix only one thing.
- It must fix a real bug that bothers people.
- It must fix a problem that causes a build error, an oops, a hang, data corruption, a real security issue, In short, something critical.
- Serious issues as reported by a user may also be considered.
- New device IDs and quirks are also accepted.
- No "theoretical race condition" issues, unless an explanation of how the race can be exploited is also provided.
- It cannot contain any "trivial" fixes in it .
- It or an equivalent fix must already exist in Linus' tree (upstream). ……

LongTerm Stable Kernel

- Extended maintenance period for stable kernel
- Pick one version per year and maintain 2 years
- Bug and security fixes found in latest version are continued to back ported
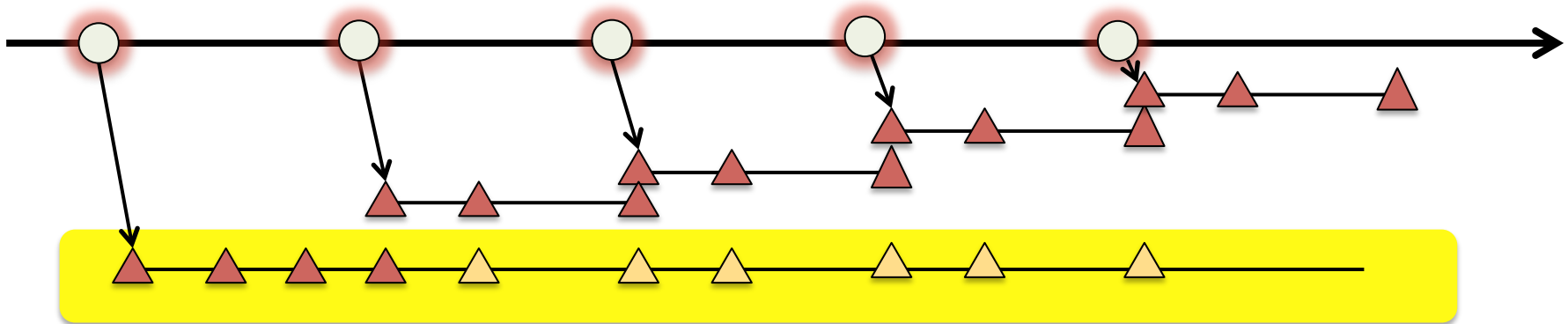
LTS includes large number of fixes

- Stable release only back port important bug and security fixes
- Thousands of fixes are there   2014/10/11

Version		# of commits
From	**To**	
3.0	**3.0.101**	**3953 (EOL)**
3.1	3.1.10	695 (EOL)
3.2	**3.2.62**	**5353**
3.3	3.3.8	698 (EOL)
3.4	**3.4.103**	**4759**
3.5	3.5.7	816 (EOL)
3.6	3.6.11	757 (EOL)
3.7	3.7.10	718 (EOL)
3.8	3.8.13	996 (EOL)
3.9	3.9.11	746 (EOL)

Version		# of commits
From	**To**	
3.10	**3.10.53**	**3610**
3.11	3.11.10	677 (EOL)
3.12	**3.12.26**	**3162**
3.13	3.13.11	903 (EOL)
3.14	**3.14.17**	**1850**
3.15	3.15.10	703 (EOL)
3.16	3.16.1	687

Current LTS versions

Version	Maintainer	Released	Projected EOL
3.14	Greg Kroah-Hartman	2014-3-30	Aug, 2016
3.12	Jiri Slaby	2013-11-3	2016
3.10	Greg Kroah-Hartman	2013-6-30	Sep, 2015
3.4	Li Zefan	2012-5-20	Sep, 2016
3.2	Ben Hutchings	2012-1-4	2016
2.6.32	Willy Tarreau	2009-12-3	Mid-2015

https://kernel.googlesource.com/pub/scm/docs/kernel/website/

LTSI DEVELOPMENT PROCESS

LTSI development process

| Preparation | Merge Window | Validation | Maintenance |

Release
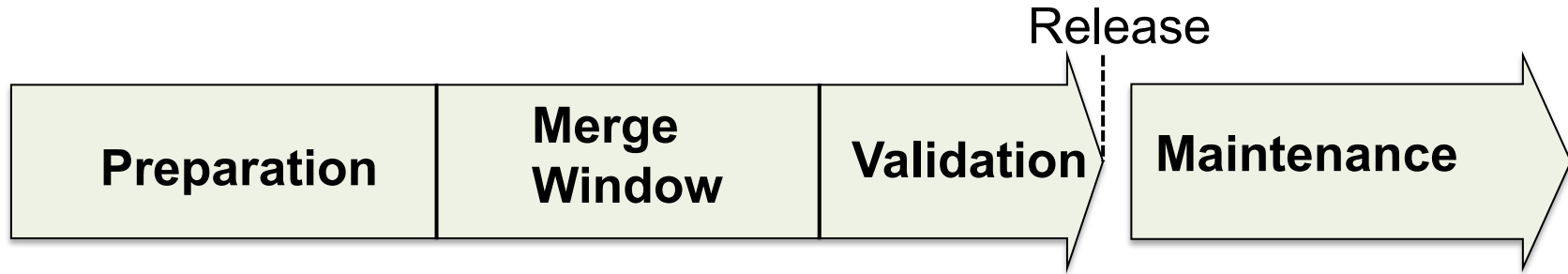
Discussing about process as;

- Preparation

- Merge window

- Validation

- Maintenance

LTSI Process: Preparation

Release

Preparation	Merge Window	Validation	Maintenance
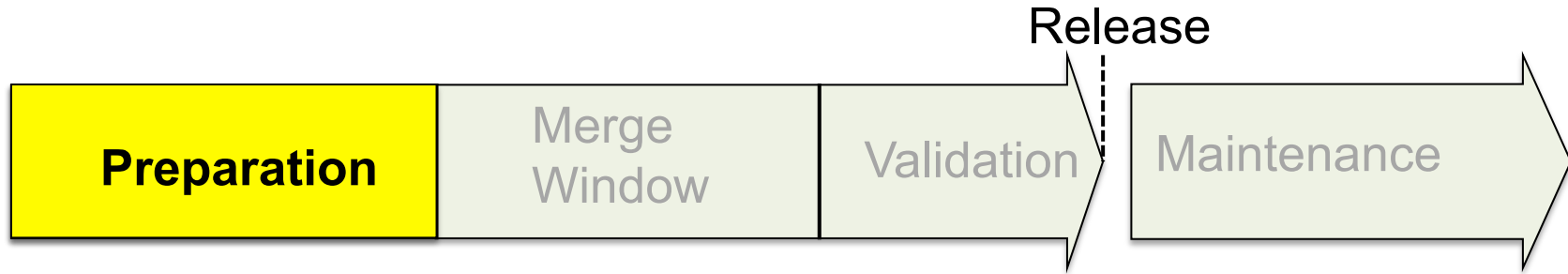
- Decide LTS version

- Decide LTSI development schedule

- Before the merge window, developer should;

  – Merge own patches into upstream to prepare merge into LTSI. That will be easier to maintain production kernel (ie. Driver)

  – Prepare to back port important features in the upstream to LTSI

LTSI Process: Merge window

Release

| Preparation | **Merge Window** | Validation | Maintenance |
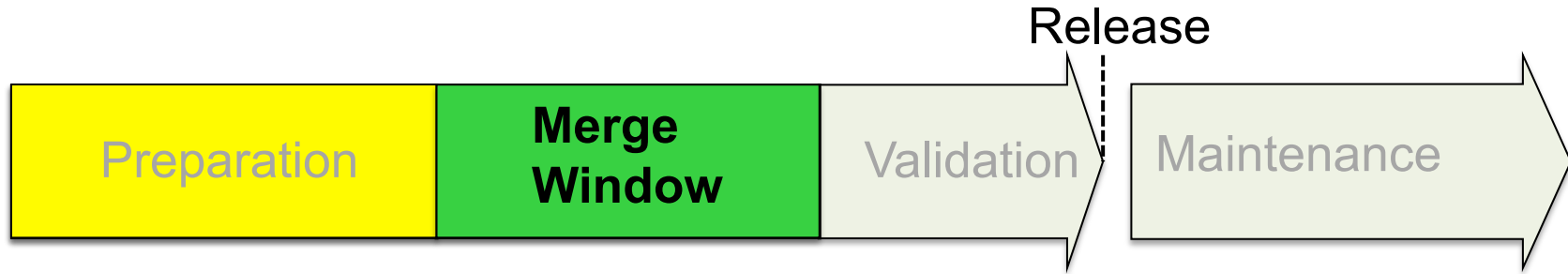
- Merge window will open after 4-5 month of LTS release
- Anyone can propose patches in the merge window period
- Patches should be in the upstream or at least in the –next
- Exceptional case: beneficial for the wide range of users, or projected to be in the upstream
- Merge window period is about two month and –rc1 will be released right after the window was closed
- No additional patches are accepted after the window closed

LTSI process: Validation

Release

| Preparation | Merge Window | **Validation** | Maintenance |
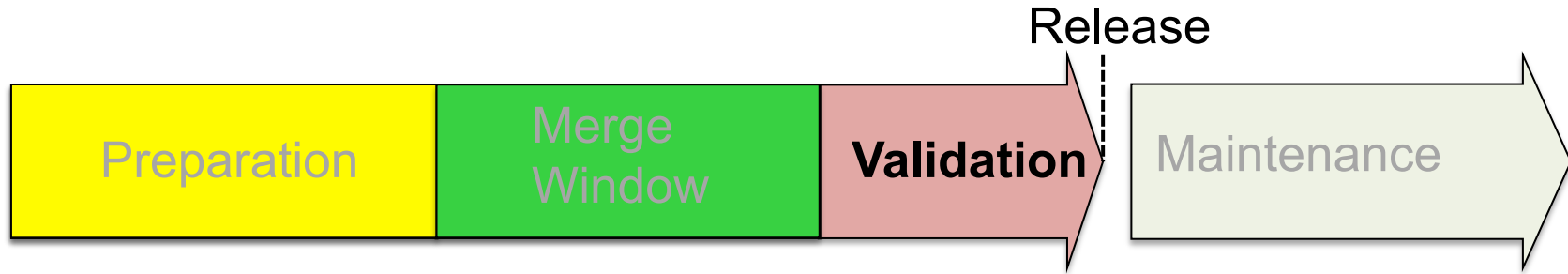
- After release of –rc1, developer should validate whether its features were correctly merged and worked fine

- Patches to fix problem should be sent

- Validation period is about a month or more

- LTSI development will be finished when maintainer recognized reasonably stable and then new LTSI kernel will be released

LTSI Process: Maintenance

Release

Preparation	Merge Window	Validation	Maintenance
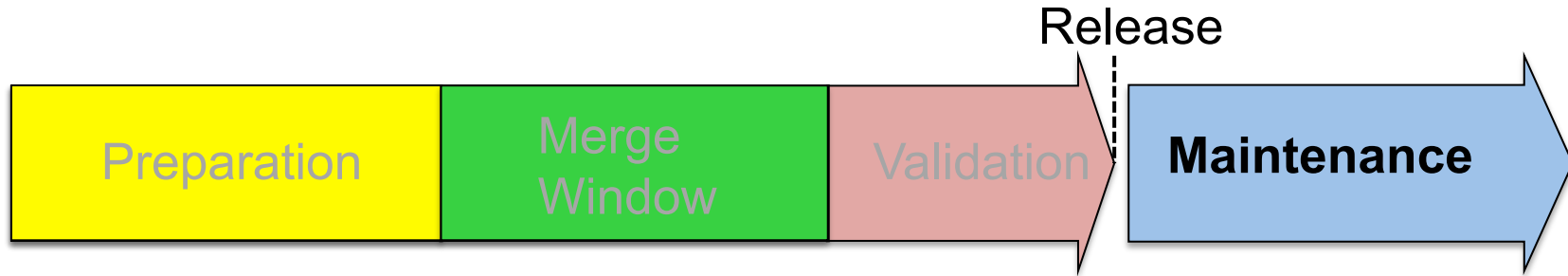
- LTS is continue to maintain in two years since that version was released

- LTSI will be rebased at the same time each new LTS was released
  - If you want to create own kernel based on LTS such as 3.10.x, you will be able to find same version in LTSI git tree
  - There are tool to create LTSI full kernel source in "scripts" directory

- LTSI full source tarball will be released every 3month after QA team have done its test as a volunteer work

LTSI: maintenance stream
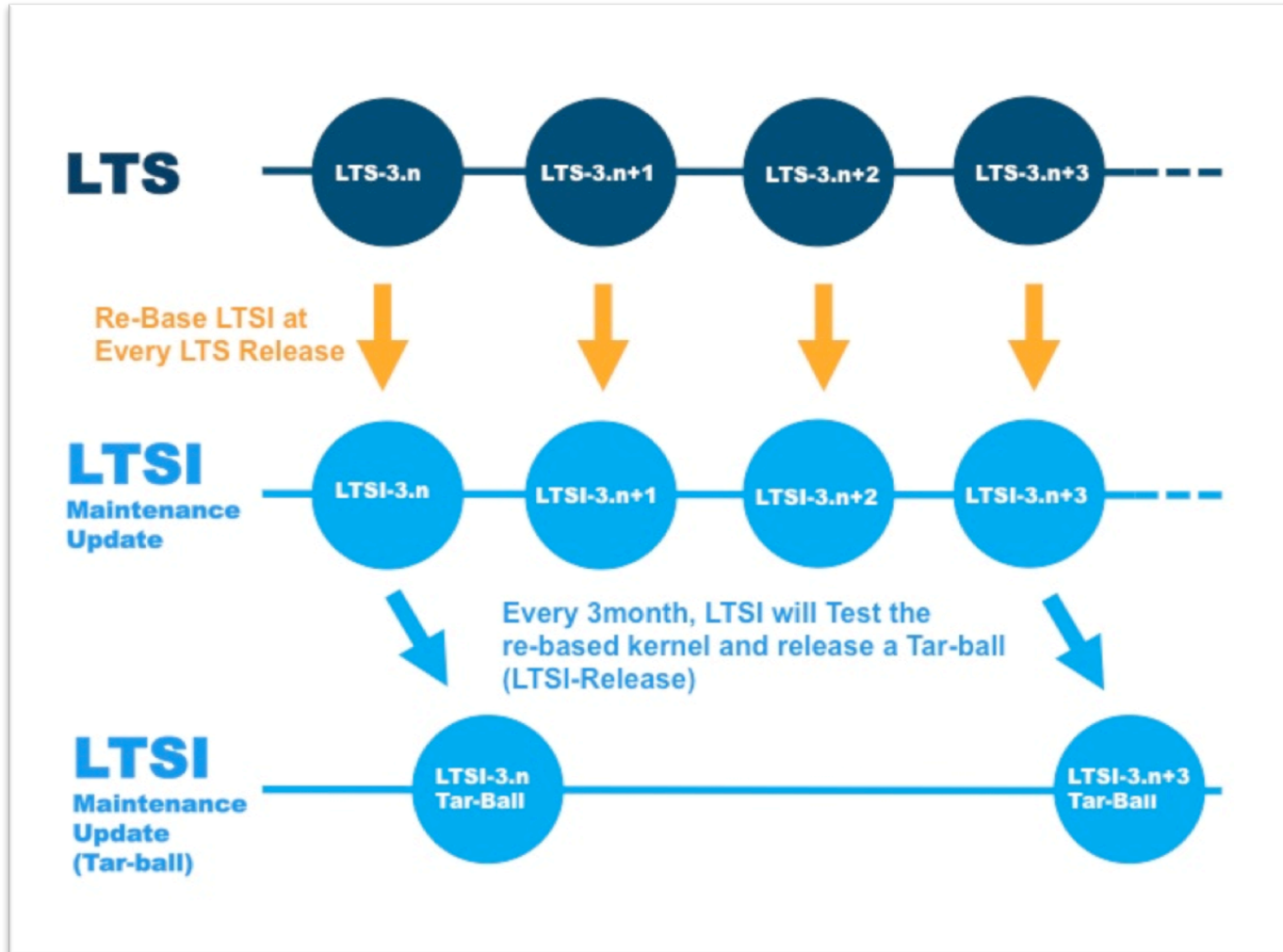
Patchwork is ready

- Patchwork is tool to cherry pick patches posted on ML

- LTSI providing patchwork

- In case you still have patches after closed the merge window, you can send patch to LTSI ML anytime.

- Using Patchwork, you can share patches even after the release

Patchwork LTSI Project development

login
register
mail settings

Project: ltsi-dev : patches : project info : other projects   about

Incoming patches

« Previous   1   2   3   4   5   Next »

Filters: State = Action Required ⊖ | Archived = No ⊖

Patch	⚠ Date	Submitter	Delegate	State
[12/12] ARM: shmobile: armadillo800eva: set clock rates before timer init	2013-05-22	Do Quang Thang		New
[11/12] ARM: shmobile: use FSI driver's armadillo800eva				

https://patchwork.kernel.org/project/ltsi-dev/list/

How to create own BSP
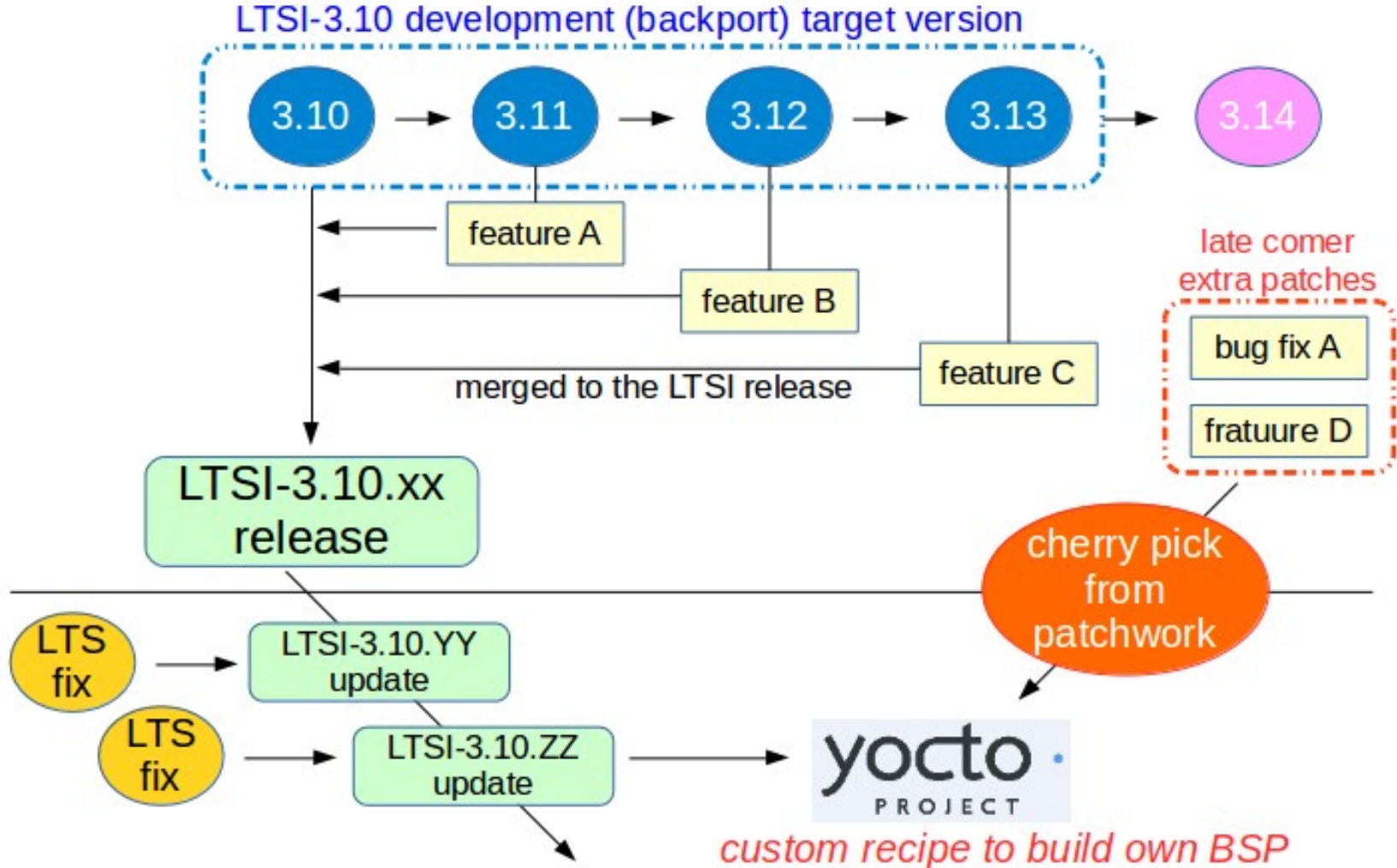
LTSI-3.10 history

Event	Date
kernel 3.10 merge window open	2013.4.28
kernel 3.10 merge window close	2013.5.12
kernel 3.10 release	2013.6.30
Announce of 3.10 becomes LTS	2013.8.4
LTSI-3.10 merge window open	2013.11.13
Merge Window period	75 days
LTSI-3.10-rc1 (=merge window close)	2014.1.29
Validation period	26 days
LTSI-3.10 release	2014.2.24

Major contributors for LTSI-3.10

Contributor	Patch count
Darren Hart (Intel)	1,197
Simon Horman (for Renesas)	1,122
Daniel Sangorrin (Toshiba)	123
Patrik Jakobsson (for Intel)	46
Mark Brown (linaro)	11
Greg Kroah-Hartman (Linux Foundation)	11
Total	2,510

Major achievement of LTSI-3.10

- LTTng
- Power efficient work queues
- Intel BayTrail support
- Intel Minnow board support
- Renesas R-Car H2/M2 series support
- Xilinx Zinq board support

LTSI-3.14 Release plan

NEW

Event	Date
kernel 3.14 merge window open	2014.1.19
kernel 3.14 merge window close	2014.2.2
kernel 3.14 release	2014.3.30
3.14 becomes LTS	2014.7.3
LTSI-3.14 merge window open	**2014.8.21**
Merge window period	75 days
LTSI-3.14-rc1 (=merge window close)	**2014.10.30?**
Validation period	50+ days
LTSI-3.14 release	**2014.12.25?**

Please send patch to LTSI mailing list now!

LTSI NEW CHALLENGES TEST PROJECT

LTSI Test : Background

- LTSI is patch set added on top of LTS

- LTS is quite stable because of picking patches with strict rule (stable_kernel_rules.txt)

- However, LTSI will include
  - Back port features in latest upstream
  - Industry demanded non mainlined code

We want to validate LTSI kernel that does not include any bug or regression against LTS

LTSI Test : Share test case!

- Now LTS and LTSI is used by lots of companies

- Each companies may spend a lot of time for its validation

- Some of fundamental "feature test" might be duplicated and such portion might be shared

We can consider sharing some part of kernel test across the industry
- **Test environment**
- **Test case**

LTSI Test environment

- We have developed LTSI test environment
  - Fully automated execution
  - Easy to manage
  - Monitoring what have happened
  - GUI
  - Deliverable (Be able to install each companies)
  - 28 benchmarks and 33 test programs are integrated in initial test environment

LTSI Test environment

Current stage is just started but You can down load from:
https://bitbucket.org/cogentembedded/jta-public/

https://bitbucket.org/cogentembedded/jta-public/

LTSI Test environment

- Initial documentation:

  https://bytebucket.org/cogentembedded/jta-public/raw/7cefe53a09b5028bf2c99663d81ecde39b486713/docs/jta-guide.pdf

  Your feed back is welcome!

NEXT STEP for LTSI Test environment

- Daily/Nightly test of LTSI for specific hardware
  - Public server 24h/7d up/running for LTSI testing with Intel Minnow, Renesas Henninger
- More I/O and platform-specific tests

- Polished docs, deployment/installation scripts

You will be able to refer its result

http://145.255.234.170/

Test case sharing

- Held discussion at LTSI workshop in Chicago

- What we have discussed :
  - License to deliver; strong or weak?
  - How the test cases are maintained; maintainer or just gather
  - How we manage test cases;
    - Collect test case and Referencing test cases

Further discussion will be happen in our workshop, please join us!

THANK YOU

You can participate LTSI

- Follow on Twitter account:

  @LinuxLTSI

- Web:

  http://ltsi.linuxfoundation.org

- Mailing list:

  https://lists.linuxfoundation.org/mailman/listinfo/ltsi-dev

- Git tree :

  http://git.linuxfoundation.org/?p=ltsi-ernel.git;a=summary