LONG TERM SUPPORT INITIATIVE

Developing Industry's Kernel Together – LTSI; how it works and its value to the business

Noriaki Fukuyasu

2013.11.13 @ Korea Linux Forum 2013

AGENDA

- About LTSI
- Value of LTSI to your business

About LTSI

There are 2 Simple Questions Often asked …

- What is LTSI?
- Why it is there?

LTSI is ...

LTS + I

LTSI is … (1)

A project which the effort of the kernel community and industry bring together under the Linux Foundation.

LTS + I

Community effort, maintained by Greg Kroah-Hartman

Community + Industry effort, maintained by Greg Kroah-Hartman, as a Linux Foundation project

LTSI is … (2)

A project which provides a long term stable kernel that include industry’s requirements

LTS + I

Maintained for 2 years.
Bug and security fixes can be backported

Maintained for 2 years.
New feature, function, drives required by industry are backported in addition to bug and security fixes.

LTSI is there because …

- **“Fragmentation”** in embedded Linux industry has been a big issues
- **“In-House Patch”** has been an issue among consumer electronics companies.
- **“Speed of Innovation”** ironically has become a pain for the industry.

Single industry-managed kernel could help solving these issues.

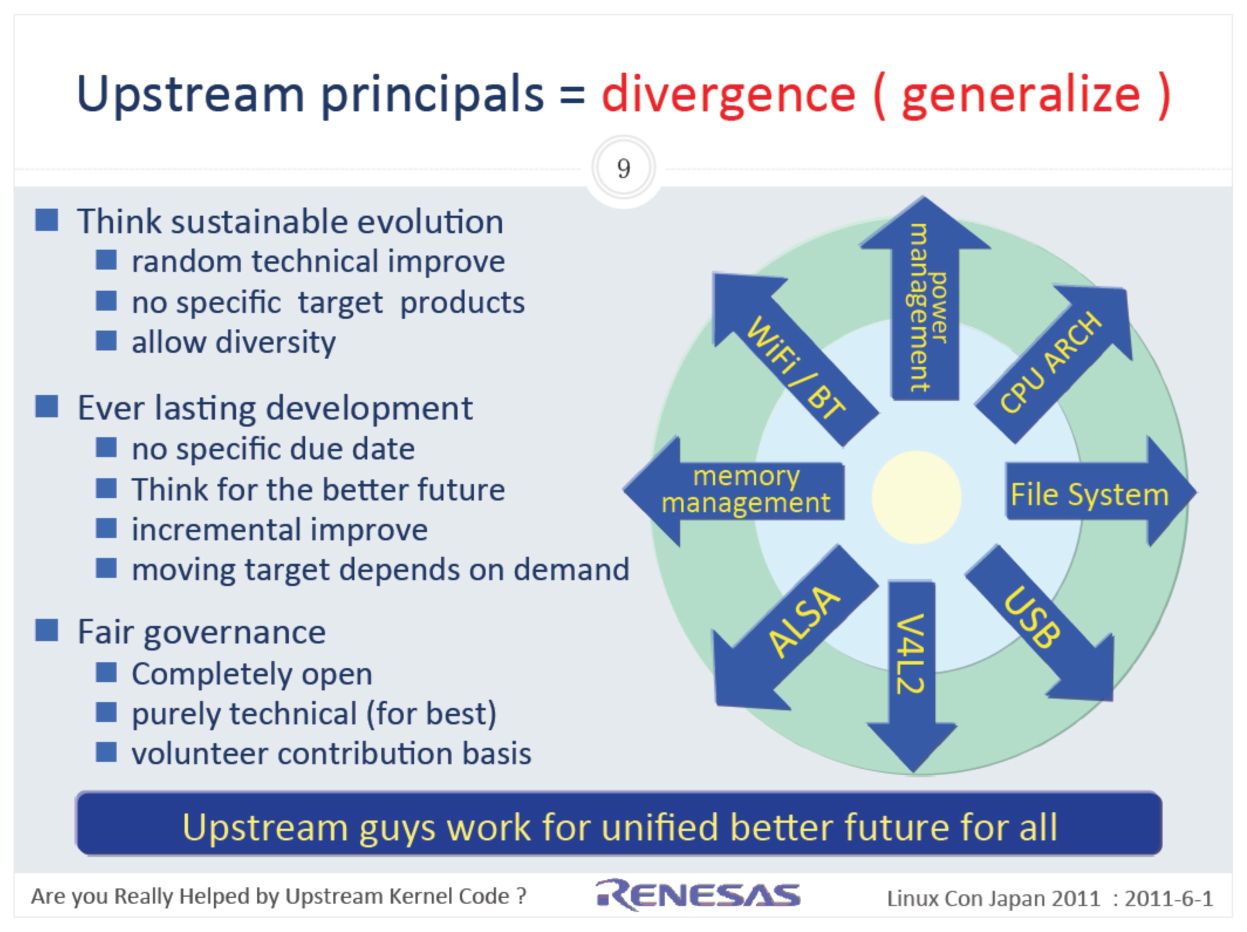
Fragmentation

- Unlike in Enterprise industry, there is no de facto distributions which industry can rely on. (Yes, Android is big in mobile, but "embedded industry" is not limited to mobile)
- Take a example android devise development, you have to deal with 1) Google's patch, 2) Soc's patch specific to their chip set, 3) vendors's patch, and 4) bug and security fix from upstream kernel.
- Companies tend to maintain many difference kernels within their company on their own.

In-House Patch Issue

- Embedded Linux developers pays huge attention to their device driver code quality, and they do internal improvements to optimize to their systems.
- Thus, embedded developers (companies) owns very good/important patches within their companies but these patches are mostly not share with the community (and therefore rest of the industry)
- EVEN WORSE, since these patches are not upstreamed, they will most likely have to work on the same enhancement next time they update the kernel

Reason behind the in-house patch issue (1)



Upstream principals = divergence (generalize)

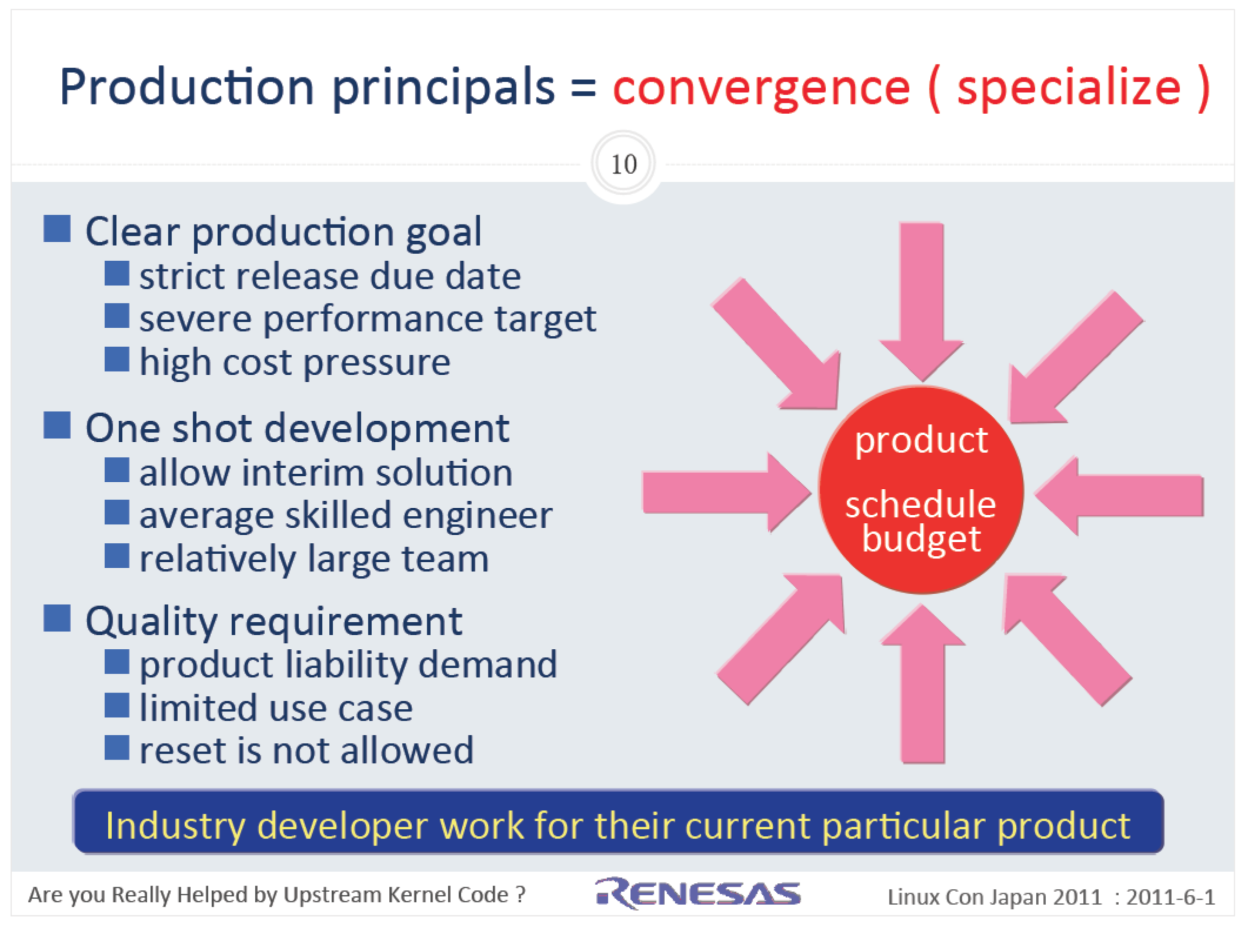
9

- Think sustainable evolution
  - random technical improve
  - no specific target products
  - allow diversity
- Ever lasting development
  - no specific due date
  - Think for the better future
  - incremental improve
  - moving target depends on demand
- Fair governance
  - Completely open
  - purely technical (for best)
  - volunteer contribution basis

Upstream guys work for unified better future for all

Are you Really Helped by Upstream Kernel Code ?   RENESAS   Linux Con Japan 2011 : 2011-6-1

Reason behind the in-house patch issue (2)



Production principals = convergence (specialize)

10

- Clear production goal
  - strict release due date
  - severe performance target
  - high cost pressure
- One shot development
  - allow interim solution
  - average skilled engineer
  - relatively large team
- Quality requirement
  - product liability demand
  - limited use case
  - reset is not allowed

product
schedule
budget

Industry developer work for their current particular product

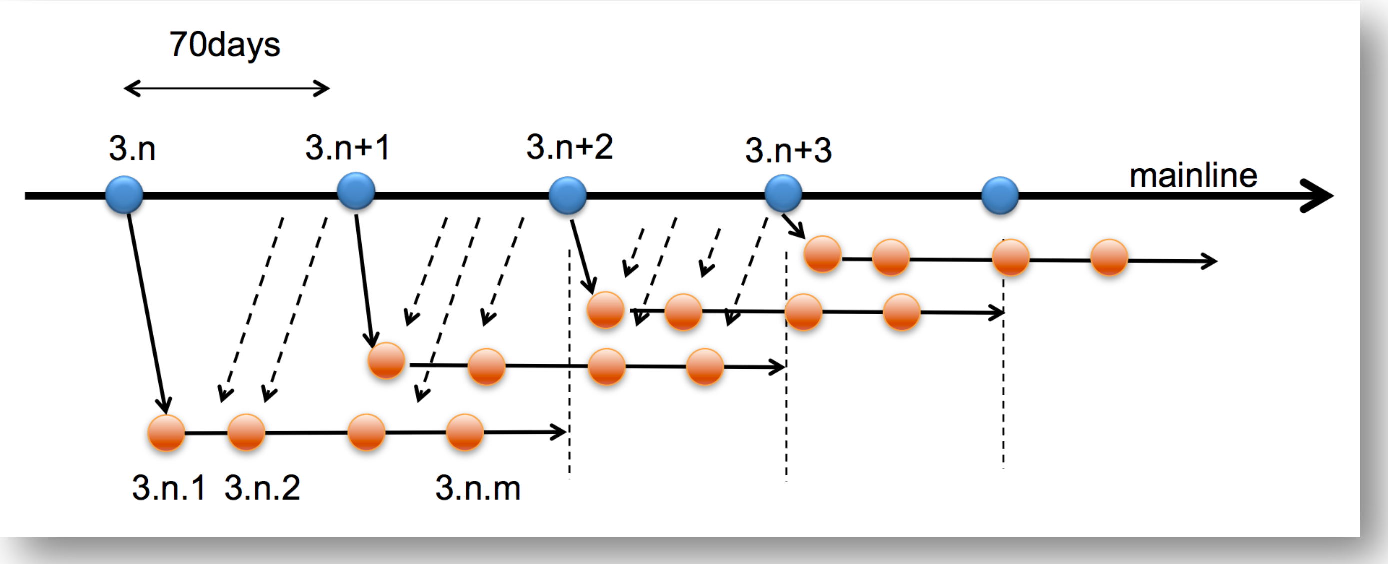
Are you Really Helped by Upstream Kernel Code ? RENESAS Linux Con Japan 2011 : 2011-6-1

Speed of Innovation Issue

- Unlike enterprise business, product life cycle of embedded business is around 1-3 years (shorter than Enterprise business)
- Meanwhile, embedded products are getting more and more complex.
- Can you keep up with the speed of innovation on your own?

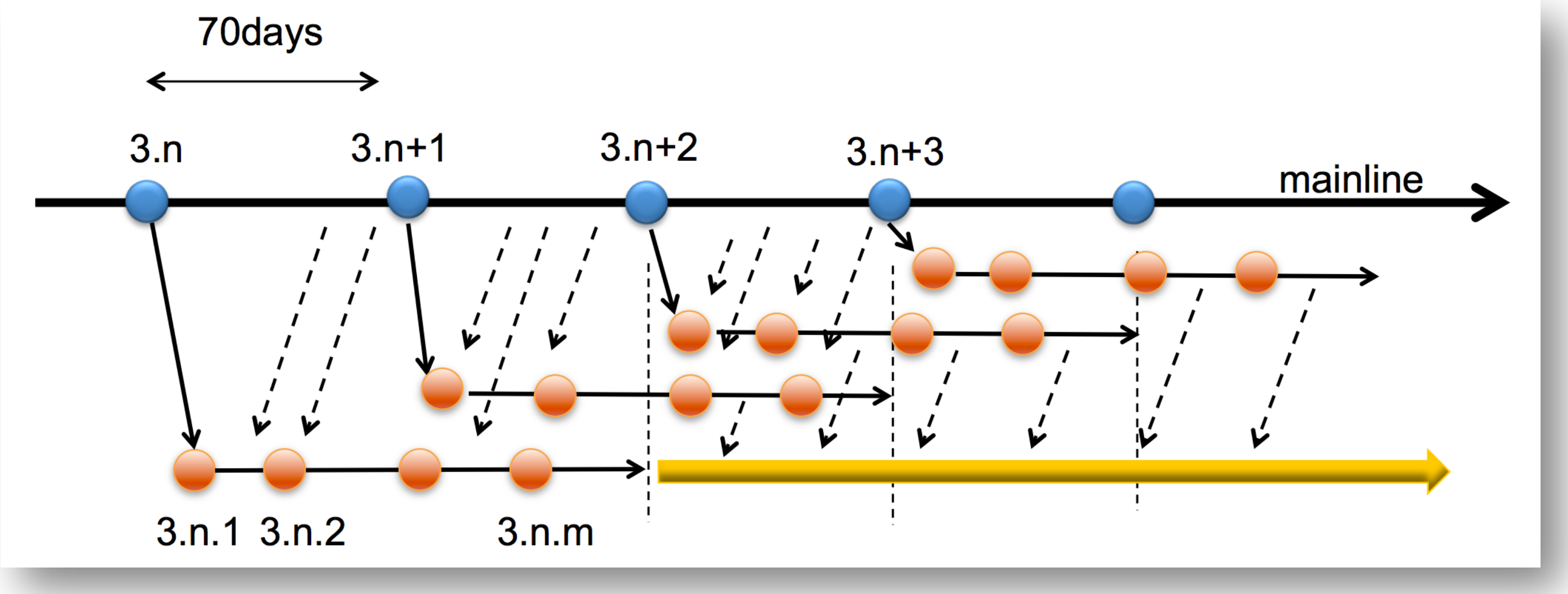
How LTSI Works?

Stable Kernel Development



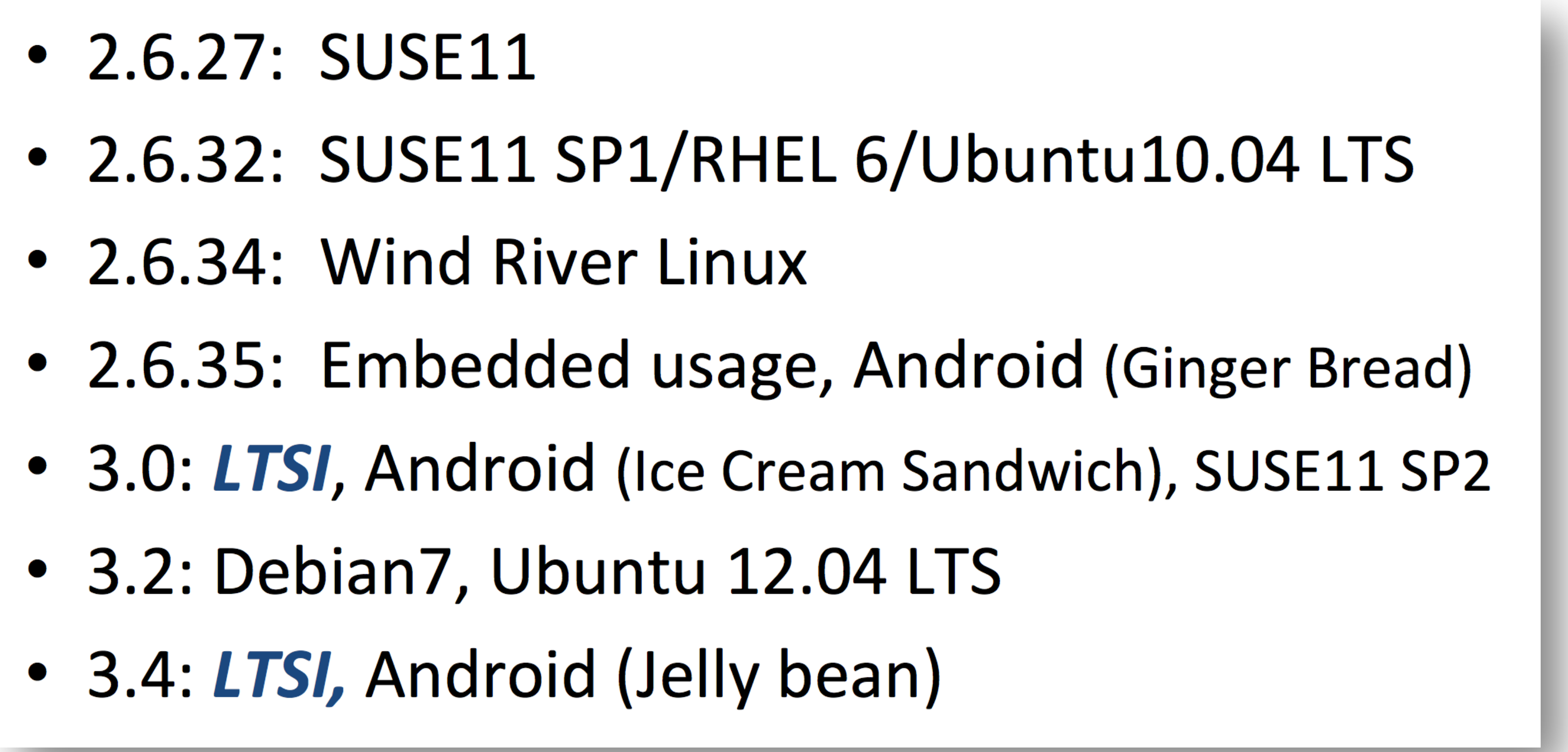
- Kernels are maintained for n+2 version after their release.
- “Maintain” means the adoption of “Bug fix” and “security fix” that are found in the later versions.

LTS Development



- After n+2 version of maintenance period, some kernel version are selected as "Long Term Stable" (= LTS) and continued to maintain for 2 years.
- LTS will be selected once a year.
- LTSI is based on this but add new features (based on the demand) etc on top of it.

List of LTS and its Consumers



- 2.6.27: SUSE11
- 2.6.32: SUSE11 SP1/RHEL 6/Ubuntu10.04 LTS
- 2.6.34: Wind River Linux
- 2.6.35: Embedded usage, Android (Ginger Bread)
- 3.0: ***LTSI***, Android (Ice Cream Sandwich), SUSE11 SP2
- 3.2: Debian7, Ubuntu 12.04 LTS
- 3.4: ***LTSI,*** Android (Jelly bean)

The next LTS and LTSI is v. 3.10

LTSI v3.10 Development Schedule

- Merge period: December 31st
- Testing Period: January, 2014
- Expected release date: early February

We are now accepting your patch!

LINUX FOUNDATION

Economic Value of LTSI to your business?

How to calculate the value of Open Source Projects

There are number of approaches to evaluate the value of open source project such as:

- Lines of code
- Comparison with proprietary products in the same market
- etc…

The value of LTSI can be evaluated based on # of patches backported.

What are the costs of your products?

- Software Development cost
- Maintenance cost
- HW/Product cost
- Sales/Marketing cost
- …

What are the costs of your products?

- Software Development cost
  - Specific application or middleware
  - Tuning for overall system
  - Specific driver for kernel
  - **Patch** porting to newer kernel
- Maintenance cost
  - **Back porting** bugs and security fixes
  - Fixes for own application and middle ware
- HW/Product cost
- Sales/Marketing cost

Why patch back-porting costs so much ?

```
While every single in-hose patches
  if a patch cannot apply to the target kernel
      Investigate the reason
    rewrite the patch for target kernel
    test the patch on the target kernel
```

Why patch back-porting costs so much ?

Reason may depends on for both in-house patch and kernel itself

```
While every single in-house patches
   if a patch cannot apply to the target kernel
         Investigate the reason
      rewrite the patch for target kernel
      test the patch on the target kernel
```

Why patch back-porting costs so much ?

Reason may de
in-house patch ar

Engineer who wrote the patch
may not in the team

```
While every single i     se pa    hes
  if a patch cannot ap  y to the target kernel
      Investigate the reason
    rewrite the patch for target kernel
    test the patch on the target kernel
```

Why patch back-porting costs so much ?

```
While every single i      se pa    hes
  if a patch cannot ap   y to the target kernel
      Investigate the reason
    rewrite the patch for target kernel
    test the patch on the tar   t kernel
```

Reason may de
in-house patch ar

Engineer who wrote the patch **may not in the team**

Need **review the patch**

Why patch back-porting costs so much ?

Reason may de
in-house patch ar

Engineer who wrote the patch **may not in the team**

```
While every single i     se pa   hes
  if a patch cannot ap  y to the target kernel
      Investigate the reason
    rewrite the patch for target kernel
    test the patch on the tar   t kernel
```

Need to **create testing environment again**

review the patch

LINUX FOUNDATION

LONG TERM SUPPORT INITIATIVE

Why patch back-porting costs so much ?

Reason may de

in-house patch a

Engineer who wrote the patch **may not in the team**

These must continue every time new release come out and you adopt it, and this effort will last thru/out the product life!

While every single ... se pa ... hes

if a patch cannot ad ... to the target kernel

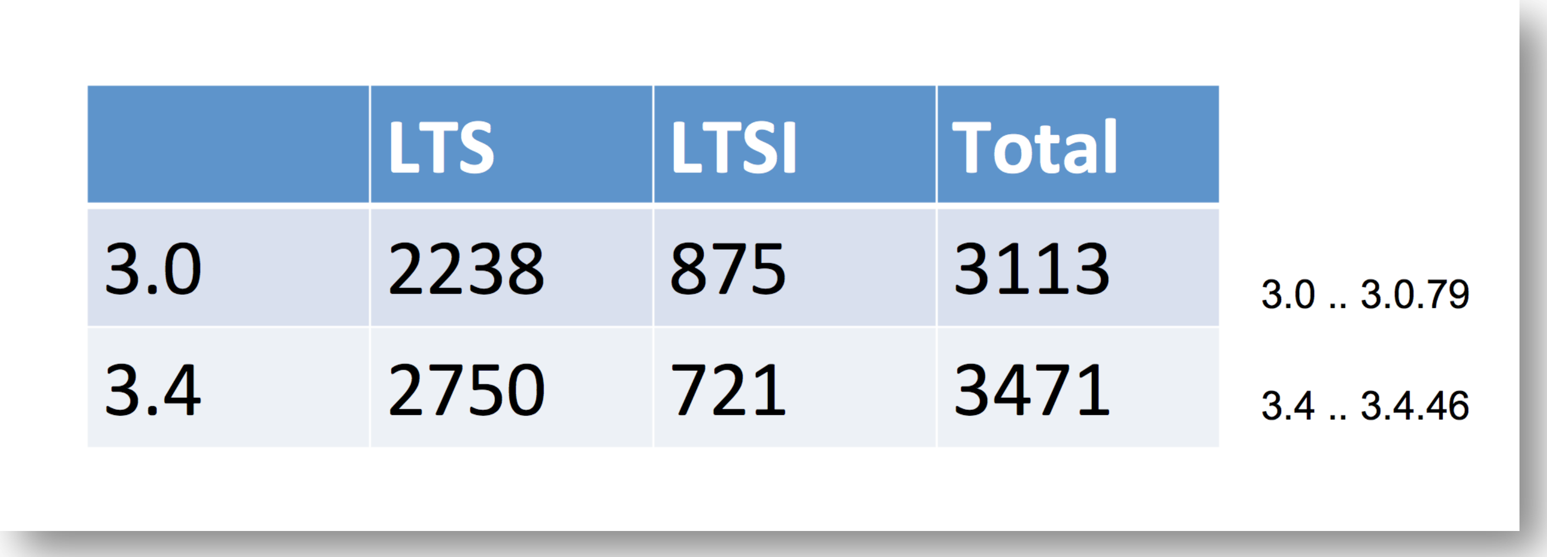
investigate the reason

test the patch on the tar ... t kernel

Need to **create testing environment again**

review the patch

Number of patches back-ported to LTSI (as of today)



	LTS	LTSI	Total	
3.0	2238	875	3113	3.0 .. 3.0.79
3.4	2750	721	3471	3.4 .. 3.4.46

Economic value for a PASSIVE USERS

ASSUMPTIONS

- It takes an average of one day (8 hours) to port each patch
- The cost of a kernel developer is USD $18,000 per month
- Developers work 20 days per month.

LTSI 3.0

3,113 patches = 3,113 (man Day) = 155.7 (man Month)
155.7 (man Month) x US $18,000 = **US $2,802,600**

LTSI 3.4

3,452 patches = 3,452 (man Day) = 172.6 (man Month)
172.6 (man Month) x US $18,000 = **US $3,106,800**

Economic value for an ACTIVE PARTICIPANTS

- By upstream your patches, developers in the Community will develop based on the assumptions that your codes exists already. By keeping your patches “in-house” there is a risk of conflict between your in-house patch and upstream kernel, and it happens over and over every time new kernel are released.
- In case of v3.4 it happened 48 times (nearly every week!)

1) Time consumed for porting in-house patches to the latest kernel

= 48 times (48 man days) x Pn (number of patches)

= 384 hours x Pn

2) Time consumed for resolving patch conflicts

= 2,712 patches x 0.1 (conflict rate) x One day (porting time) x Pn

= 2,170 hours x Pn

Total (1 +2) = 2,254 Man hours x Pn

= 14 Man Month x Pn (1 Man Month = 160 Man Hours)

# of Patches	Man Month	Cost (USD 1,000)
1	14	252
2	16	504
3	24	756
:	:	:
10	80	2,520

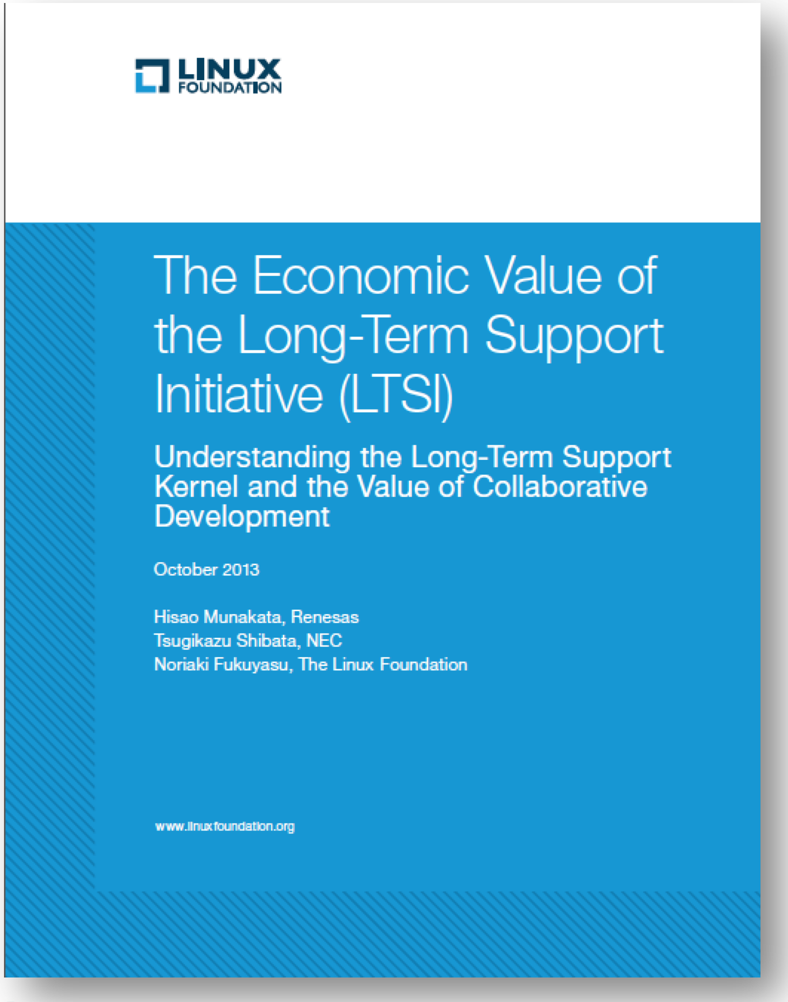
Active participants can get this economic value is on top of just downloading the kernel and consume it.

Once again....

- Merge period: December 31st
- Testing Period: January, 2014
- Expected release date: early February

Get these economic value by participating to LTSI today!

http://www.linuxfoundation.org/publications/workgroup/value-of-ltsi



LINUX FOUNDATION

The Economic Value of the Long-Term Support Initiative (LTSI)

Understanding the Long-Term Support Kernel and the Value of Collaborative Development

October 2013

Hisao Munakata, Renesas
Tsugikazu Shibata, NEC
Noriaki Fukuyasu, The Linux Foundation

www.linuxfoundation.org

PLEASE JOIN US!

- Follow on Twitter account:
  @LinuxLTSI
- Web site:
  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