

# デジタル化社会における オープンソースプラットフォーム

1

(株)ルネサスソリューションズ  
第三応用技術部 エグゼクティブ  
宗像尚郎 (hisao.munakata.vt(at)renesas(dot)com)

# 本日の話題

2

- オープンソース とは何か
- 何故オープンソースが広まっているのか
- リナックス開発の実態（統計情報）
- リナックスの品質、メンテナンス体制
- 開発プロセス革新ツール “git”
- オープンソース開発への参加方法
- 製品開発の現場とオープンソース開発
- オープンソース活用課題と皆さんへの期待

# オープンソース とは何か

# 代表的なオープンソース

4

	Name	Category	developer
1	Linux kernel	OS	kernel community
2	GCC (GNU Compiler Collection)	toolchain	GNU (FSF)
3	Android	smartphone platform	AOSP (Google)
4	Ruby on Rails	web app. platform	Rails Core Team
5	Apache	web server	Apache SW Foundation
6	MySQL	database	ORACLE
7	Free Type	font library	Free Type Project
8	OpenOffice.org	office suite	Apache Software Foundation

# オープンソース の定義

5

[Open Source Initiative](#) (OSI) は、オープンソース・ライセンスの要件として、以下のような定義 “The Open Source Definition(OSD)” を掲げている。これはもともとは [Debianフリーソフトウェアガイドライン](#) をベースとしたものである。

- 自由な再頒布ができること
- ソースコードを入手できること
- 派生物が存在でき、派生物に同じライセンスを適用できること
- 差分情報の配布を認める場合には、同一性の保持を要求してもかまわない
- 個人やグループを差別しないこと
- 適用領域に基づいた差別をしないこと
- 再配布において追加ライセンスを必要としないこと
- 特定製品に依存しないこと
- 同じ媒体で配布される他のソフトウェアを制限しないこと
- 技術的な中立を保っていること

Wikipedia : オープンソース

# オープンソースの定義

6

<http://opensource.org/docs/osd>



## Open Source Initiative

Home

Search this site:

Search

Navigation

- ▷ About the OSI
- ▽ The Open Source Definition
  - Annotated
- ▷ Open Source Licenses
- ▷ Open Standards
- Open Source Education
- ▷ Mailing lists
- ▷ Getting Help

## The Open Source Definition

### Introduction

Open source doesn't just mean access to the source code. The distribution terms of open-source software must comply with the following criteria:

#### 1. Free Redistribution

The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.

#### 2. Source Code

The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost preferably, downloading via the Internet without charge. The source code must be the



Netscape ブラウザーのソースコード公開 [1998年] が発端

# OSI 承認オープンソースライセンス ①

7

<http://sourceforge.jp/projects/opensource/wiki/licenses>

Academic Free License	Common Development and Distribution License	GNU General Public License version 3.0 (GPLv3)
Affero GNU Public License	Common Public License	GNU Library or "Lesser" General Public License (LGPL)
Adaptive Public License	CUA Office Public License Version 1.0	GNU Lesser General Public License version 3.0 (LGPLv3)
Apache Software License	EU DataGrid Software License	Lucent Public License (Plan9)
Apache License, 2.0	Eclipse Public License	Lucent Public License Version 1.02
Apple Public Source License	Educational Community License	IBM Public License
Artistic license	Eiffel Forum License	Intel Open Source License
Artistic license 2.0	Eiffel Forum License V2.0	Historical Permission Notice and Disclaimer
Attribution Assurance Licenses	Entessa Public License	Jabber Open Source License
new BSD license	Fair License	ISC License
Boost Software License (BSL1.0)	Frameworkx License	Microsoft Public License (Ms-PL)
	GNU General Public License (GPL)	Microsoft Reciprocal License (Ms-RL)

# OSI 承認オープンソースライセンス ②

8

<http://sourceforge.jp/projects/opensource/wiki/licenses>

MIT license	OCLC Research Public License 2.0	Sleepycat License
MITRE Collaborative Virtual Workspace License (CVW License)	Open Group Test Suite License	Sun Industry Standards Source License (SISSL)
Motosoto License	Open Software License	Sun Public License
Mozilla Public License 1.0 (MPL)	PHP License	Sybase Open Watcom Public License 1.0
Mozilla Public License 1.1 (MPL)	Python license (CNRI Python License)	University of Illinois/NCSA Open Source License
Multics License	Python Software Foundation License	Vovida Software License v. 1.0
NASA Open Source Agreement 1.3	Qt Public License (QPL)	W3C License
NTP License	RealNetworks Public Source License V1.0	wxWindows Library License
Naumen Public License	Reciprocal Public License	X.Net License
Nethack General Public License	Ricoh Source Code Public License	Zope Public License
Nokia Open Source License	Simple Public License 2.0	zlib/libpng license

# オープンソースライセンス

From IPA 『OSS ライセンスの比較および利用動向ならびに係争に関する調査』

[http://www.ipa.go.jp/software/open/oss/download/OSS\\_License\\_Survey.pdf](http://www.ipa.go.jp/software/open/oss/download/OSS_License_Survey.pdf)

9

OSSライセンスの比較および利用動向ならびに係争に関する調査

調査報告書

2010年5月

独立行政法人 情報処理推進機構

表 2: 世界のオープンソースプロジェクトにおけるライセンス採用状況(トップ 20)<sup>7</sup>

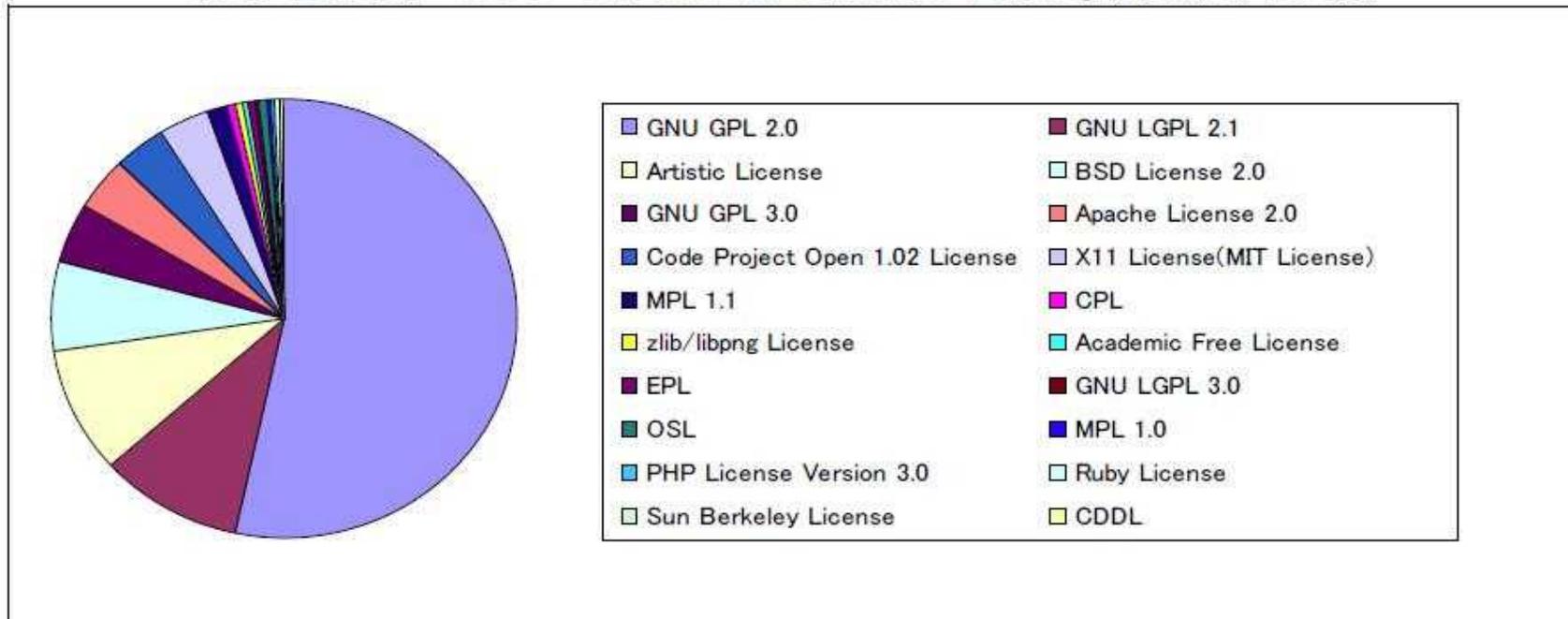


表 3: OSIにより主要とされている OSS ライセンス

- |   |   |
|---|---|
| <ul style="list-style-type: none"> <li>• GNU GPL</li> <li>• Apache License, 2.0</li> <li>• MIT License (MIT License)</li> <li>• CDDL</li> <li>• Eclipse Public License</li> </ul> | <ul style="list-style-type: none"> <li>• LGPL</li> <li>• MPL 1.1</li> <li>• New and Simplified BSD licenses</li> <li>• CPL 1.0</li> </ul> |
|---|---|

# フリーソフトウェア

10



The Free Software Definition (フリーソフトウェアの定義) は、[リチャード・ストールマン](#)により起草され[フリーソフトウェア財団](#)により公開された、[フリーソフトウェア](#) (フリーとは「言論の自由」の意味における[自由](#)) を定義する文書である。

- 目的を問わず、プログラムを実行 する自由 (第 0 の自由)。
- プログラムがどのように動作しているか研究し、そのプログラムにあなたの必要に応じて修正を加え、採り入れる自由 (第 1 の自由)。ソースコードが入手可能であることはこの前提条件となります。
- 身近な人を助けられるよう、コピーを再頒布 する自由 (第 2 の自由)。
- プログラムを改良し、コミュニティ全体がその恩恵を受けられるようあなたの改良点を 公衆に発表 する自由 (第 3 の自由)。ソースコードが入手可能であることはここでも前提条件となります。

フリーソフトウェア と オープンソースは同一概念ではない  
GNU GPL/LGPL はフリーソフトウェア財団が定義したもの

# GPL / LGPL

11

- GPL/LGPL はオープンソースソフトウェアの代表的なライセンス
- GPL と LGPL の違い
  - GPL ライセンスソフト + Proprietary\*1 ライセンスソフト  
⇒ 全てのソフトウェアに GPL ライセンス要件の適用を要求
  - LGPL ライセンスソフト + Proprietary\*1 ライセンスソフト  
⇒ Proprietary ライセンス部分は LGPL の影響を受けない

1: Proprietary = 商用ライセンスなどオープンソースライセンス以外のライセンスを指す

- LGPL version 2 vs. version 3
  - 全てのソフトが v2 → v3 に移行したわけではない。
    - Linux kernel は明示的に v2 を利用することを宣言
    - 組み込み機器の多くは v3 への移行が困難である

# 何故オープンソースが広まっているのか

# オープンソースの採用事例

13

分野	採用事例
基幹システム	東京証券取引所、シカゴ証券取引所 三菱東京UFJ銀行、航空会社発券システム 楽天、Amazon、Google など
科学計算	スーパーコンピュータの9割以上
組み込み機器	Android 携帯、MOAP-L (DoCoMo)、MeeGo デジタルテレビ、HDDレコーダー、HDD Audio
デスクトップ	海外では Windows から Linux への切替え例

# オープンソースの採用動機

14

## 機能

ファイルシステム対応（データ保全、PC連携）  
標準ネットワークプロトコル（機器間の接続性）  
アプリケーション実行環境（データベース等）

## 経済性

独自 OS からの転換（開発負担過大のため）  
商用 OS からの転換（Black Box 回避）  
開発、サポートの国際化対応

## 標準化

プラットフォームのベース（Android、Rails 等）  
チップがサポートするから（組込み系 TDV, BD 等）  
各種ネットサービスのベース

# リナックス開発の実態（統計情報）

出典について：この章で紹介する各種の統計データは、NEC OSS推進センター長の柴田次一氏が2011年11月28日に横浜で開催されたAutomotive Linux Summitで講演された資料を本人の了解のもと引用しています。本資料は近日公開される予定です。

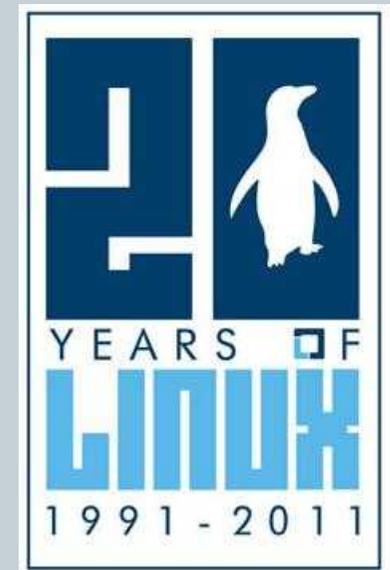
# リナックス開発コミュニティの実像

16



2011年、リナックスは誕生20周年を祝いました

- リナックス開発トレンドの紹介
  - コードサイズ変遷、開発者数変遷、リリース実績
- リナックス開発プロセスの紹介
  - レビュープロセス、コード管理
- リナックスのソフトウェア構造の紹介
  - コンフィギュレーション、抽象化



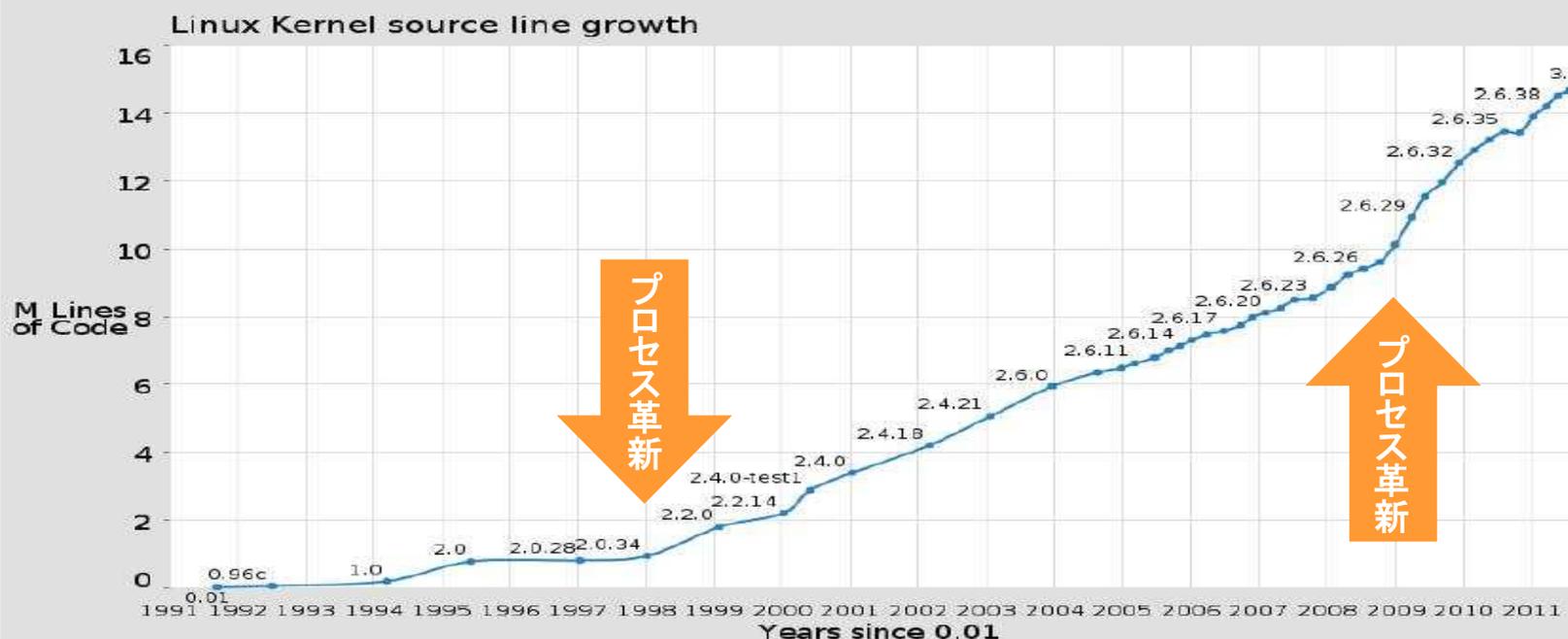
# リナックスカーネルサイズの変遷 [長期]

17

## This year is 20<sup>th</sup> anniversary of Linux

- Linux is growing with rapid pace and further acceleration is on going from 2009

柴田さん  
講演資料



20年の歴史の中で 開発プロセス革新 により生産性を高めながら成長を続けてきた

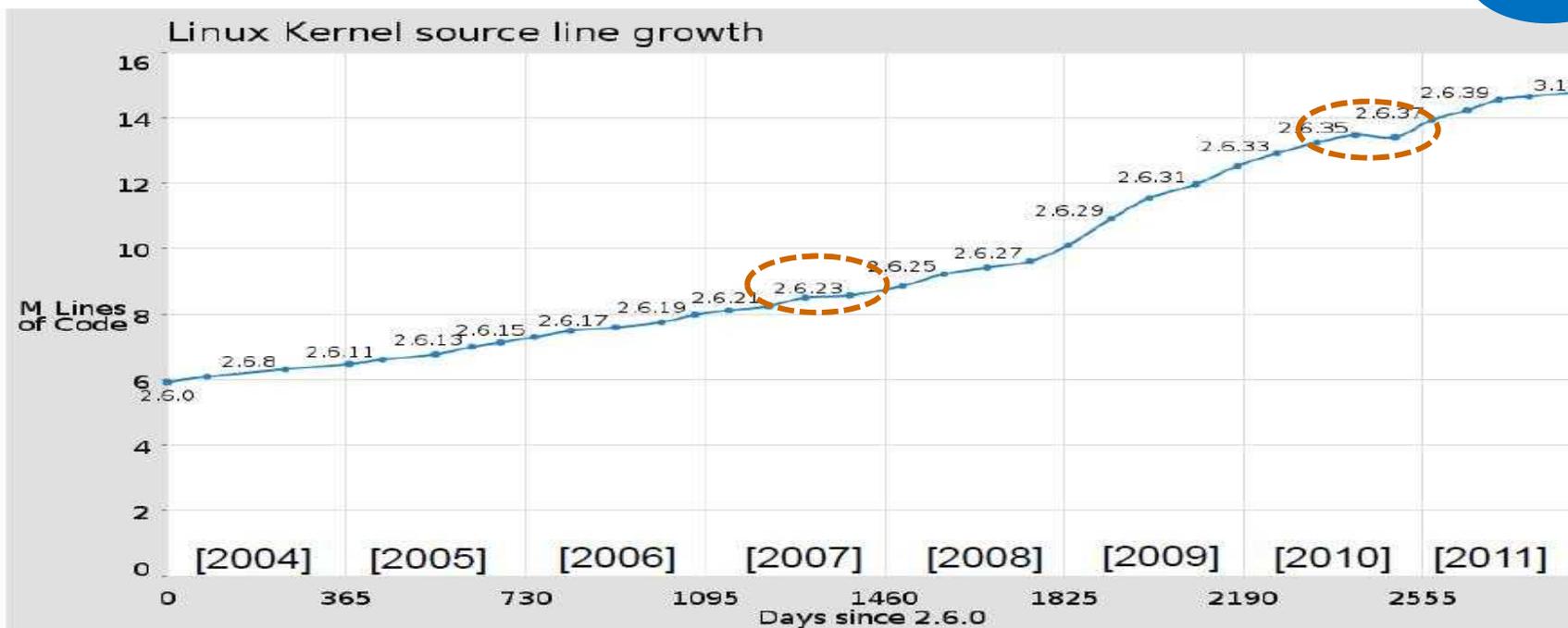
# リナックスカーネルサイズの変遷 [直近]

18

## Source code growth

- 0.8ML/year from 2004-2008, 1.8ML/year since 2009,
- Large code removal happened 2.6.23 and 2.6.36

柴田さん  
講演資料



最新の リナックスカーネル 3.1 は **37,085 ファイル**、**14,770,555 行** の規模

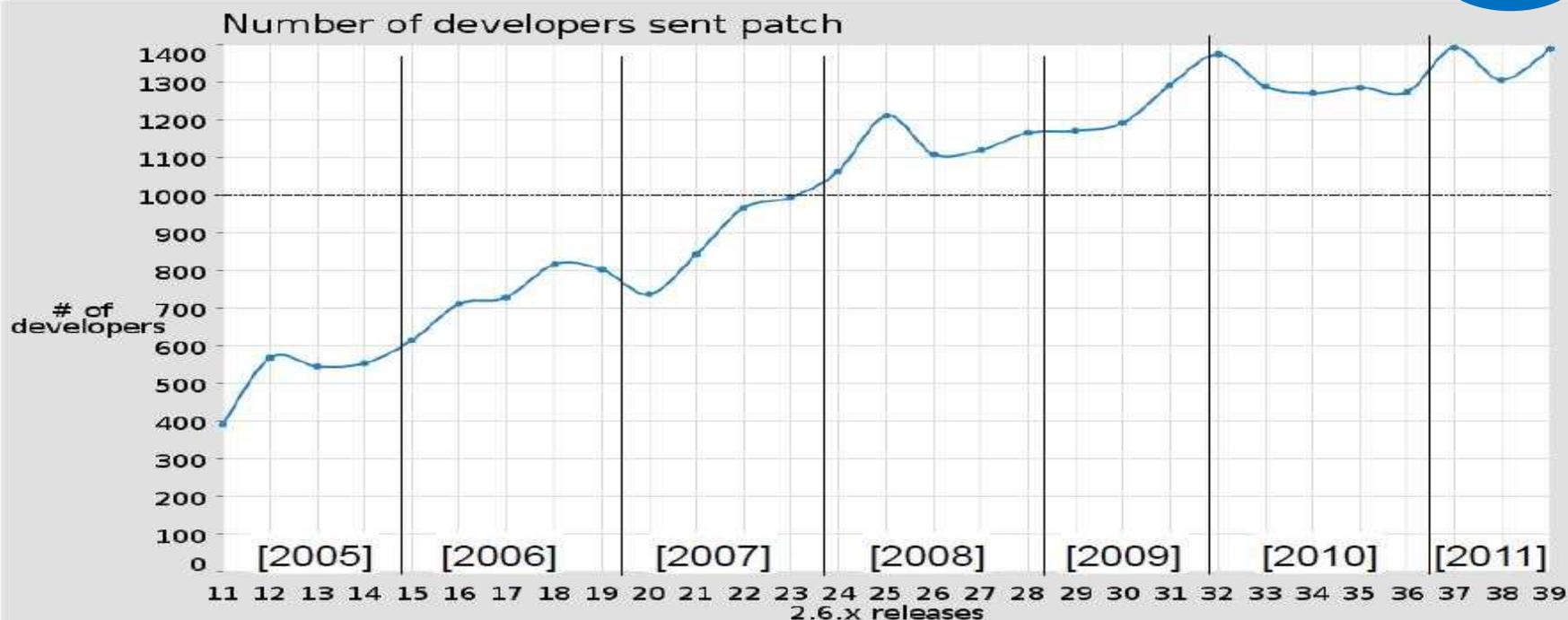
# パッチ投稿数（＝開発者数）の変遷

19

## Number of developers sent patches

- More than 1000 of developers since 2.6.23
- Become average 1,300 developers this year

柴田さん  
講演資料



開発者数も確実に増えてきており、最近は毎回 **1,300人** 程度が開発に寄与する

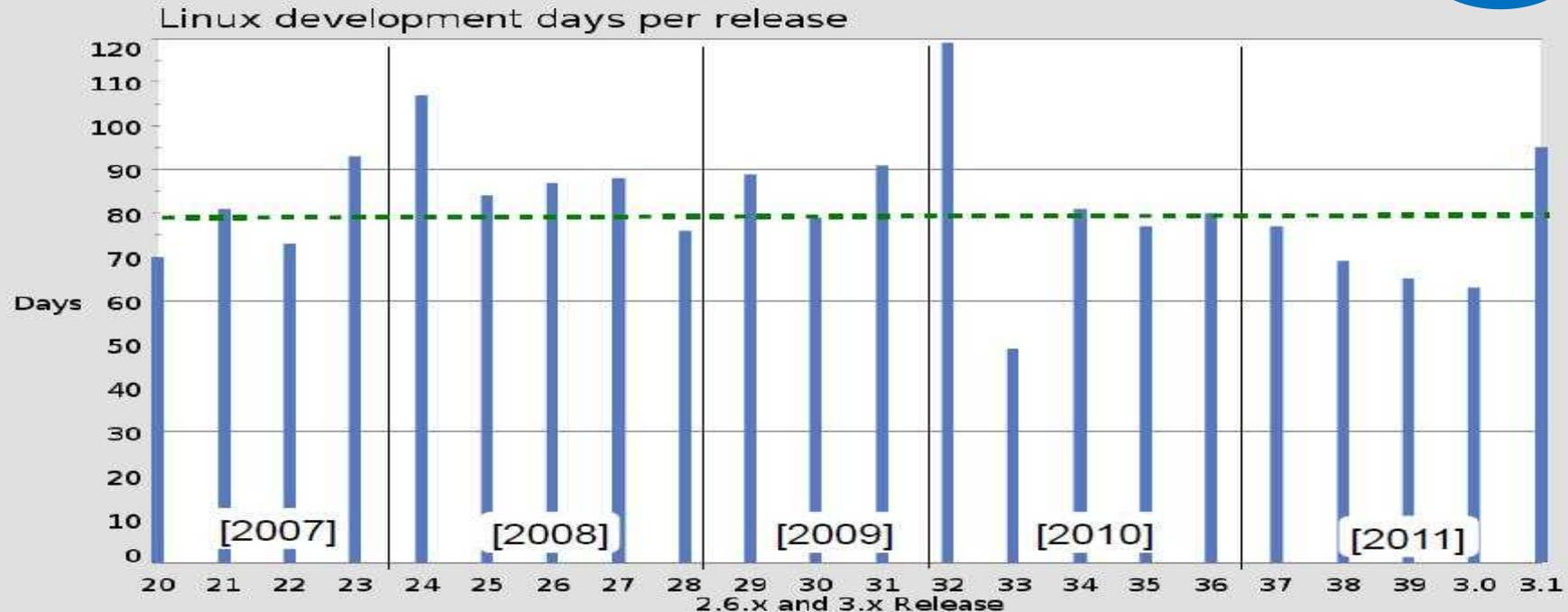
# カーネルリリースインターバル

20

## Days to develop each releases

- Average 82 days in last 4 releases
- Average 84 days in last 10 releases

柴田さん  
講演資料



これまでコンスタントに 約 80日周期 で カーネルメジャーバージョンアップ

## The Linux Kernel Archives

Welcome to the Linux Kernel Archives. This is the primary site for the Linux kernel source, but it has much more than just Linux kernels.  
[Frequently Asked Questions](#)

Protocol	Location
HTTP	<a href="http://www.kernel.org/pub/">http://www.kernel.org/pub/</a>
FTP	<a href="ftp://ftp.kernel.org/pub/">ftp://ftp.kernel.org/pub/</a>
RSYNC	<a href="rsync://rsync.kernel.org/pub/">rsync://rsync.kernel.org/pub/</a>

Latest Stable Kernel:



[3.2.1](#)



stable:	<b>3.2.1</b>	2012-01-12	<a href="#">[Full Source]</a>	<a href="#">[Patch]</a>	<a href="#">[View Patch]</a>	<a href="#">[Gitweb]</a>	<a href="#">[Changelog]</a>
mainline:	<b>3.2</b>	2012-01-04	<a href="#">[Full Source]</a>	<a href="#">[Patch]</a>	<a href="#">[View Patch]</a>	<a href="#">[Gitweb]</a>	
stable:	<b>3.1.9</b>	2012-01-12	<a href="#">[Full Source]</a>	<a href="#">[Patch]</a>	<a href="#">[View Patch]</a>	<a href="#">[View Inc.]</a>	<a href="#">[Gitweb]</a> <a href="#">[Changelog]</a>
stable:	<b>3.0.17</b>	2012-01-12	<a href="#">[Full Source]</a>	<a href="#">[Patch]</a>	<a href="#">[View Patch]</a>	<a href="#">[View Inc.]</a>	<a href="#">[Gitweb]</a> <a href="#">[Changelog]</a>
stable:	<b>2.6.39.4</b>	2011-08-03	<a href="#">[Full Source]</a>				<a href="#">[Gitweb]</a>
stable:	<b>2.6.35.14</b>	2011-08-01	<a href="#">[Full Source]</a>	<a href="#">[Patch]</a>	<a href="#">[View Patch]</a>	<a href="#">[View Inc.]</a>	<a href="#">[Gitweb]</a> <a href="#">[Changelog]</a>
stable:	<b>2.6.33.20</b>	2011-11-07	<a href="#">[Full Source]</a>	<a href="#">[Patch]</a>	<a href="#">[View Patch]</a>	<a href="#">[View Inc.]</a>	<a href="#">[Gitweb]</a> <a href="#">[Changelog]</a>
stable:	<b>2.6.32.54</b>	2012-01-12	<a href="#">[Full Source]</a>	<a href="#">[Patch]</a>	<a href="#">[View Patch]</a>	<a href="#">[View Inc.]</a>	<a href="#">[Gitweb]</a> <a href="#">[Changelog]</a>
linux-next:	<b>next-20120113</b>	2012-01-13					<a href="#">[Gitweb]</a>

Changelogs are provided by the kernel authors directly. Please don't write the webmaster about them.  
[Customize the patch viewer](#)

① 開発中コード、② 最新リリース、③ 長期メンテナンスコードが公開されている。

# 実際にコードを書いているのは誰か

22

<http://lwn.net/Articles/460597/>

## Most active 3.1 employers

### By changesets

(None)	1111	13.1%
Red Hat	882	10.4%
(Unknown)	749	8.8%
Intel	616	7.3%
Broadcom	428	5.1%
Novell	380	4.5%
IBM	301	3.6%
Texas Instruments	276	3.3%
(Consultant)	223	2.6%
Freescale	182	2.2%
Linaro	170	2.0%
Samsung	162	1.9%
Google	150	1.8%
Wolfson Microelectronics	142	1.7%
Fujitsu	131	1.5%
Renesas Electronics	100	1.2%
Oracle	82	1.0%
MiTAC	80	0.9%
Nokia	79	0.9%
(Academia)	73	0.9%

### By lines changed

Novell	162583	19.8%
(None)	90119	11.0%
Broadcom	76810	9.4%
Red Hat	58262	7.1%
Intel	43505	5.3%
(Unknown)	27109	3.3%
Metzler Brothers Systementwicklung GbR	23681	2.9%
Samsung	23238	2.8%
Rising Tide Systems	23090	2.8%
IBM	22231	2.7%
Texas Instruments	21130	2.6%
Freescale	17270	2.1%
Brocade	16587	2.0%
Realsil Microelectronics	15868	1.9%
Wolfson Microelectronics	14004	1.7%
(Consultant)	13710	1.7%
South Pole AB	12087	1.5%
Linaro	11129	1.4%
Oracle	9390	1.1%
Nokia	7450	0.9%

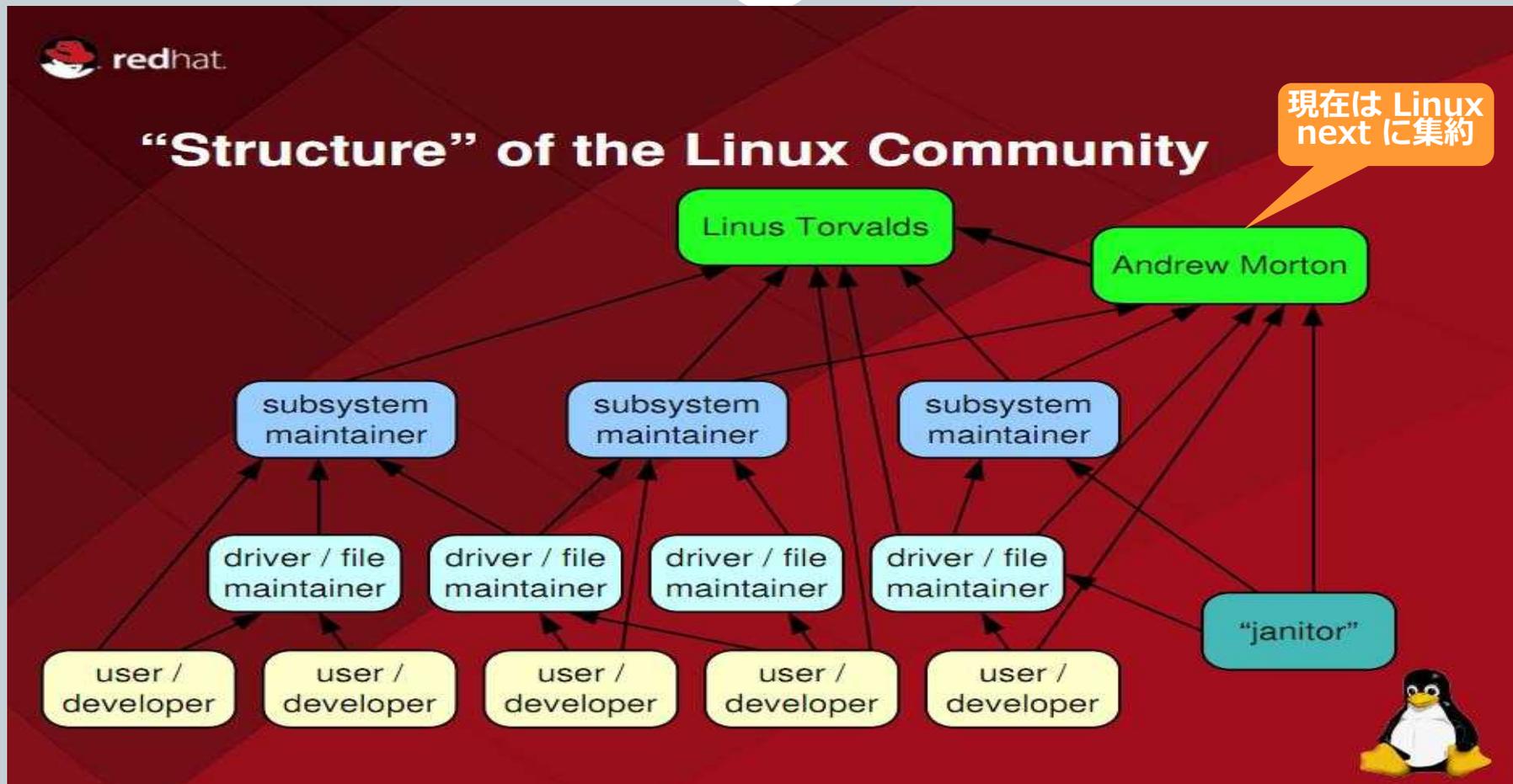
リナックスの大部分のコードは各企業のプロフェッショナルが作成

# リナックスカーネル開発コミュニティ

23

From the world of Community David Woodhouse (Red hat)

<http://tree.celinuxforum.org/CelfPubWiki/JapanTechnicalJamboree19>



サブシステム毎に分割されたメンテナーによる階層的国際分散開発体制で開発

# カーネルコンフィギュレーション

24

## Choosing feature of Linux

- Possible by using “make manconfig” (or xconfig)
- There are 768 of Kconfig files in 3.1
- 11,662 of configuration items

柴田さん  
講演資料

```
munakata@muna-E420: ~/linux-2.6
munakata@muna-E420: ~/linux-2.6
.config - Linux/i386 3.2.0-rc6 Kernel Configuration

Linux/i386 3.2.0-rc6 Kernel Configuration
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys.
Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?>
for Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module < > module capable

General setup --->
[*] Enable loadable module support --->
-* Enable the block layer --->
Processor type and features --->
Power management and ACPI options --->
Bus options (PCI etc.) --->
Executable file formats / Emulations --->
-* Networking support --->
Device Drivers --->
Firmware Drivers --->
File systems --->
Kernel hacking --->
Security options --->
-* Cryptographic API --->
v(+)
```

全てのアーキテクチャ、プラットフォームで共通のソースコードを利用している

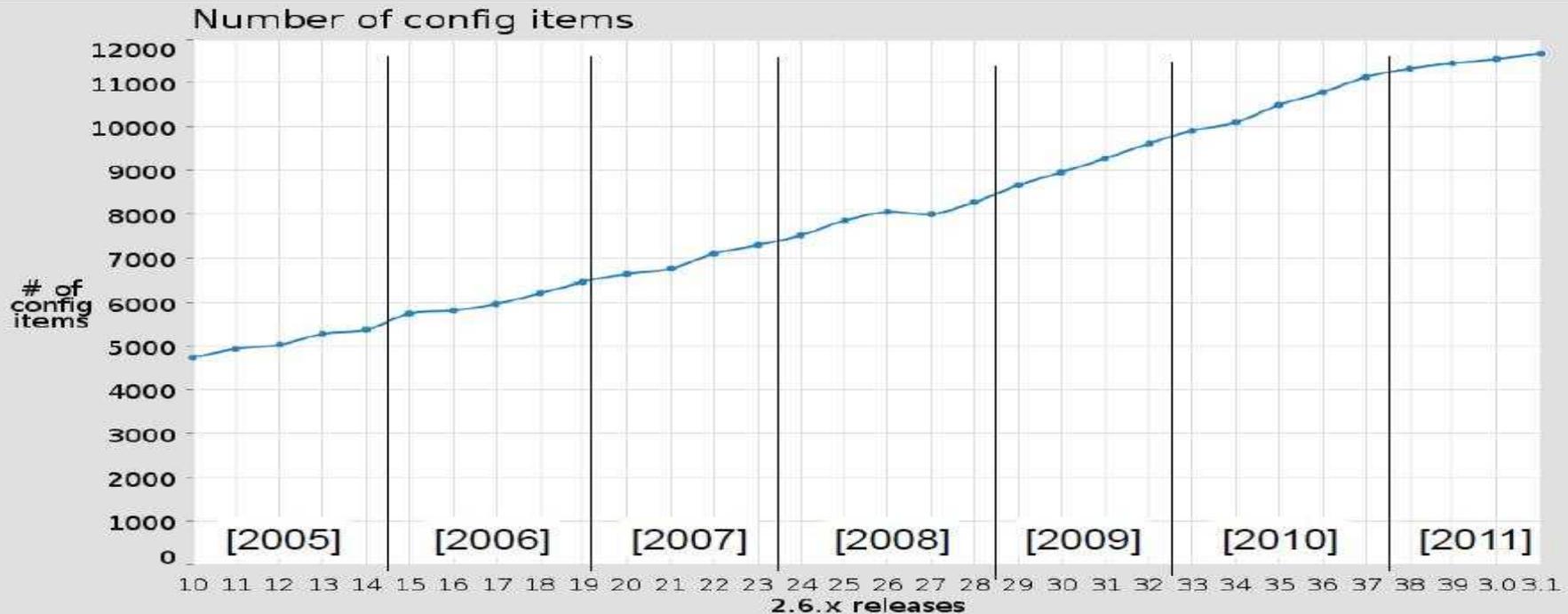
# 設定項目拡大の歴史は多様性拡大の指標

25

## # of configuration items

- 11,662 of configuration items in 3.1

柴田さん  
講演資料



カーネルコア機能、CPUアーキテクチャ、デバイスドライバーの拡張

# リナックスの実像 = 進化の継続

26

- 1,500万行以上、37,000ファイル の膨大なコード
- 1,500人近い開発者が世界中で連続的に開発中
- 約80日周期でメジャーバージョンアップ  
(新機能の追加、新しいデバイス対応の追加 など)
- カーネル設定パラメータ 12,000 項目 の最適設定で  
超大型マシンから組み込み機器まで共通ソースコード



Water Flow Model 設計 (解析→仕様→設計→実装→検証)  
が適用できない。常に進化する超大規模ソフトを効果的に追従  
できる新しい開発の仕組みを構築しないと対応できない。

# リナックスの品質、メンテナンス体制

# リナックスの品質（対立する意見）

28

-  *Community people says “Linux is really great !”*
  - 他の商用ソフトウェアなどと比較して信頼性が高い
  - 既に基幹コンピュータに多く採用されている
-  *Some industry people says “Linux sucks !”*
  - リナックスは機能実装も不完全でテストもされていない
  - 商用ソフトのようなサポート（バグ修正の仕組み）がない
  - コミュニティがリリースしたコードでは製品は作れない

実際には両方の主張ともに 一定の妥当性があるものだと思う

# コミュニティの品質に対する見方 ①

Jonathan Corbet

29



## Kernel Quality

“I believe the 2.6 kernel is slowly getting buggier. It seems we're adding bugs at a higher rate than we're fixing them.”

-- Andrew Morton, May, 2006

Some fear that kernel quality is declining

Bugs not getting fixed

Too many features added too quickly

Too little stabilization time

Kernel developers tend not to agree

But everybody agrees fewer bugs would be better

# コミュニティの品質に対する見方 ②

Andrew Morton

30



## Reporting bugs

- It's your fault that the bug got into a vendor kernel anyway
  - You should have tested the kernel during its development
- Option a: report to vendor
  - Slow, can cost
  - Costs the vendor too
  - Others are being affected
  - We'd prefer that the bug never was there in the first place
  - It's still your fault
  - Vendor then needs to remember to forward-port the fix into mainline
    - Can take ages, and might even get lost
- Option b:
  - Verify that the bug is still present in mainline kernel. If so:
    - Tell the vendor!
      - They will realize that they just need to find the right patch to backport
      - They'll probably act more promptly
  - If the bug is in mainline, report it to kernel.org team instead.
    - Chances are, it'll get fixed quickly
    - Then the vendor can backport it promptly

使う人が十分テストしなきゃ駄目だよ

# コミュニティの LTS メンテナンスルール

31

コミュニティによるメンテナンス期間中に発見されたバグは、コミュニティ BTS (Bug Tracking System) に報告

バグ発見者、当該プログラム開発者、又は第三者が対策用修正コード (パッチ) をコミュニティ ML に投稿する

コミュニティのレビュープロセス (ML上の議論) を通し対策内容の妥当性を確認する

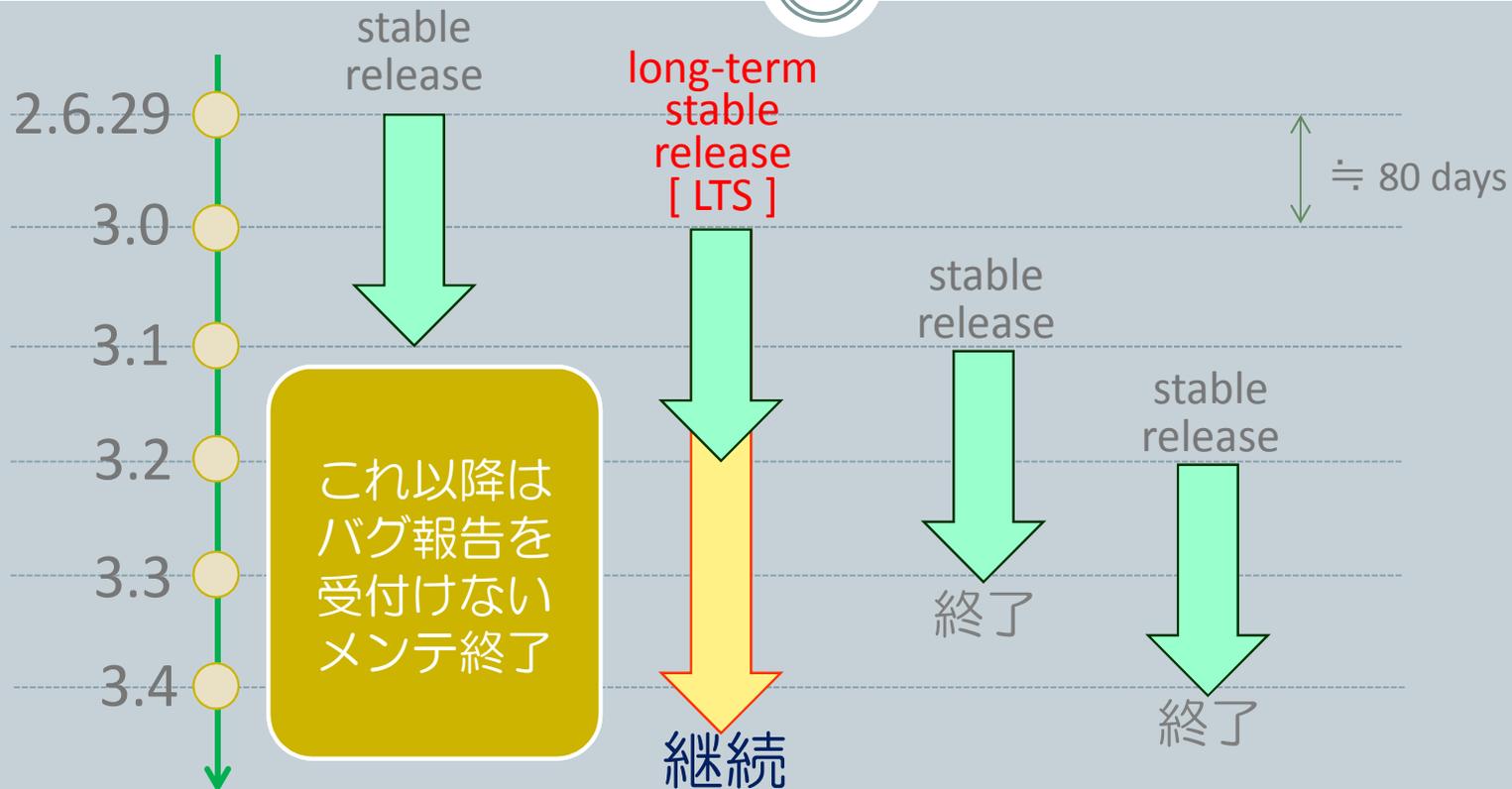
妥当性が確認されたコードは、その時点で開発中のコードにまず登録される (feed forward 思想)

新しいカーネルで採用された確実なバグ修正だけを選んで、ステータブルリリースに登録

目先の問題を修正するより前に将来の再発を防止する手順を踏む

# リリース後のバグ修正取り込みの仕組み

32



community は期間限定でメンテナンス（バグ修正）に対応する

- 通常は 2つ先のバージョンのカーネルがリリースされるまで（リリース後 約5か月間）
- 特定バージョン（LTSバージョン）は一定期間メンテナンスが継続される

# コミュニティLTS 管理方針改訂

<http://lwn.net/Articles/454886/>

33



## Possible changes to longterm kernel maintenance

[Posted August 13, 2011 by corbet]

Greg Kroah-Hartman has posted [a proposal](#) for some changes to how the stable and (especially) longterm kernels are maintained. The changes are being driven by users other than the enterprise distributors. "Now that 2.6.32 is over a year and a half, and the enterprise distros are off doing their thing with their multi-year upgrade cycles, there's no real need from the distros for a new longterm kernel release. But it turns out that the distros are not the only user of the kernel, other groups and companies have been approaching me over the past year, asking how they could pick the next longterm kernel, or what the process is in determining this." The core idea is to pick a new longterm kernel once a year; that kernel would then be maintained for two years thereafter. There is [some discussion](#) on Google+; it should move to the mailing list around August 15.

毎年一つ LTS kernel を選び、それを 2年間継続メンテナンスする

# Long Term Stable kernel Initiative

34



## The Linux Foundation Announces Consumer Electronics Long Term Stable Kernel Initiative

By Linux\_Foundation - October 25, 2011 - 9:48pm

New kernel tree addresses electronics makers' need for regular updates and long-term support

PRAGUE {LinuxCon Europe} October 26, 2011 – The Linux Foundation, the nonprofit organization dedicated to accelerating the growth of Linux, today announced it is hosting a new project created by its Consumer Electronics (CE) workgroup.

<http://www.linuxfoundation.org/news-media/announcements/2011/10/linux-foundation-announces-consumer-electronics-long-term-stable-ke>

コミュニティによる LTS メンテナンスに対して、より新しい必要なものを取り込んだ派生バージョンを作成して組込み製品開発に適用させる取り組み

**コミュニティの成果を拡張メンテナンスしていく業界独自の取り組み**

# 組み込み機器関連産業界が主導となって推進

35

**LONG TERM SUPPORT INITIATIVE**

Home What is LTSI? Developers Users Downloads Participate

The LTSI is an ecosystem-wide collaborative project hosted at the Linux Foundation to create and maintain a common Linux base for the use in a variety of CE products and to enable faster contributions upstream and better alignment with the mainline kernel.

### LTSI Blog

The latest from LTSI.

#### Introducing the LTSI Project

Submitted by brian on December 26, 2011 - 00:42

On October 25, 2011, [The Linux Foundation](#), the nonprofit organization leading the growth of Linux, announced it is hosting a new project created by its Consumer Embedded Workgroup.

The new project, the Long Term Support Initiative (LTSI), provides an annual release of a Linux kernel suitable for supporting the lifespan of consumer electronics and regular updates of those releases for two years.

more blogs

### Participating Organizations

See who is involved in LTSI.

HITACHI hp  
IBM intel  
NEC Panasonic  
Qualcomm ATHEROS RENESAS  
SAMSUNG SONY

Tentative web site image

# LTSI で拡張対応される内容

36

## LTSI : long term support kernel

- Back-port newly added device / platform support from new kernel.
- Back-port new kernel feature to improve code functionality like PM.  
We can apply newly added (to 3.2 and later) Android code to 3.0.
- SoC vender specific new code that are still in Linux-next can be added.
- Some embedded target kernel enhancement also can be added.

## Community LTS

- Safe bug fix already verified and merged new kernel.
- Serious security fix already verified and merged new kernel.

## Upstream vanilla kernel

# 開発プロセス革新ツール “git”

# ソースコードバージョン管理システム [VCS]

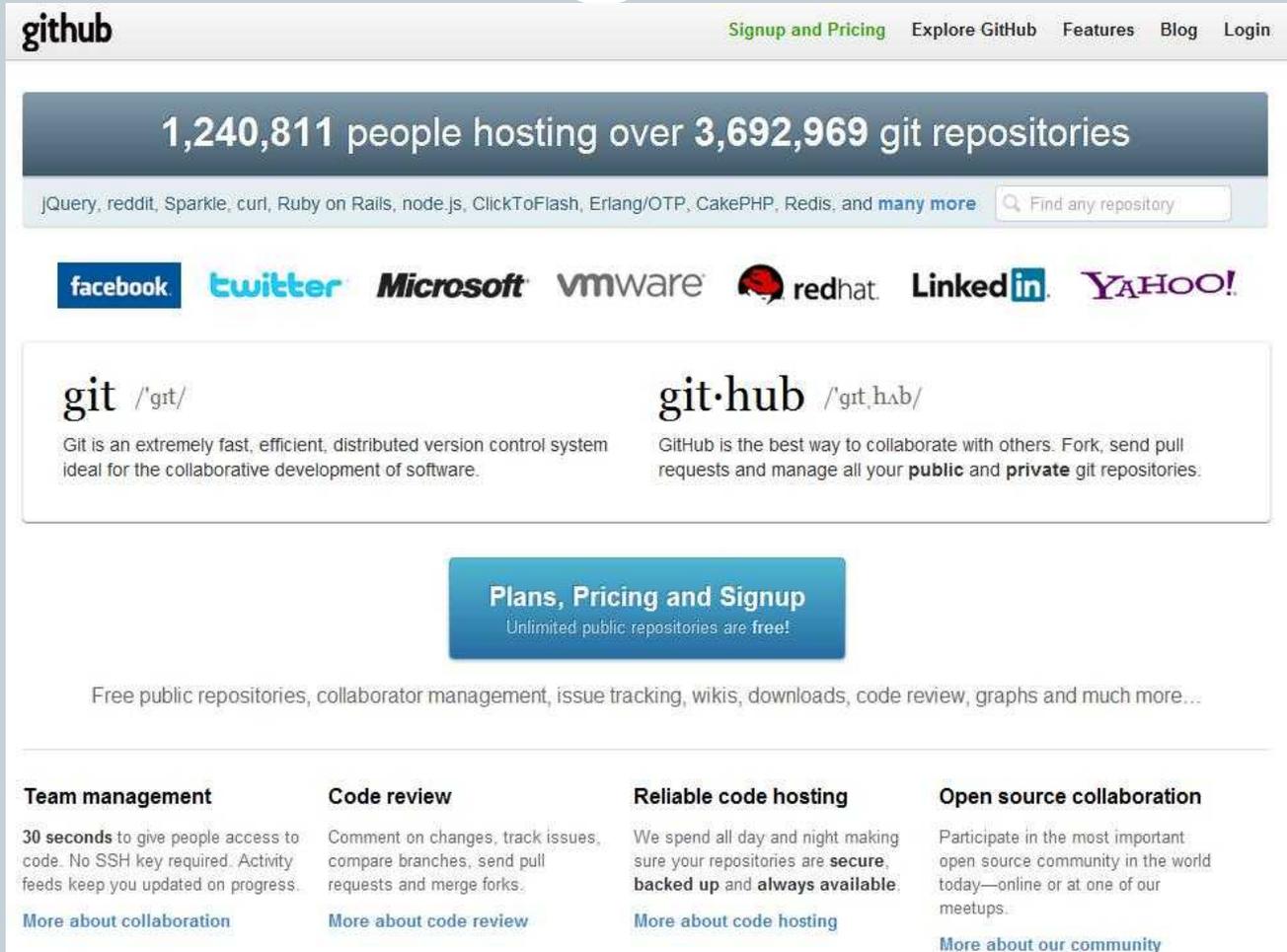
38

- VCS の基本的な機能
  - ファイル履歴管理 …… 複数開発者による共同開発では必須
  - 差分抽出
  - 古いバージョンの取り出し
  - 派生バージョンへの分岐
- ファイル管理方法
  - 個別 VCS …… SCCS、RCS
  - クライアント/サーバー型 集中VCS …… CVS、Subversion
  - 分散バージョン管理 分散 VCS …… git、BitKeeper、Mercurial

git は多くのオープンソース開発で利用される分散管理 VCS の代表

# github : free opensource “git” repository

39



The screenshot shows the GitHub homepage. At the top left is the 'github' logo. To the right are navigation links: 'Signup and Pricing', 'Explore GitHub', 'Features', 'Blog', and 'Login'. Below the navigation is a dark blue banner with the text '1,240,811 people hosting over 3,692,969 git repositories'. Underneath the banner is a search bar containing 'jQuery, reddit, Sparkle, curl, Ruby on Rails, node.js, ClickToFlash, Erlang/OTP, CakePHP, Redis, and many more' and a search icon with the text 'Find any repository'. Below the search bar are logos for 'facebook', 'twitter', 'Microsoft', 'vmware', 'redhat', 'LinkedIn', and 'YAHOO!'. The main content area is divided into two columns. The left column is for 'git /git/' and describes it as 'an extremely fast, efficient, distributed version control system ideal for the collaborative development of software'. The right column is for 'git·hub /git\_hΛb/' and describes it as 'the best way to collaborate with others. Fork, send pull requests and manage all your public and private git repositories.' Below these columns is a blue button that says 'Plans, Pricing and Signup' with the subtext 'Unlimited public repositories are free!'. Underneath the button is the text 'Free public repositories, collaborator management, issue tracking, wikis, downloads, code review, graphs and much more...'. At the bottom of the page are four columns of features: 'Team management' (30 seconds to give people access to code), 'Code review' (comment on changes, track issues), 'Reliable code hosting' (secure, backed up, always available), and 'Open source collaboration' (participate in the most important open source community).

github [Signup and Pricing](#) [Explore GitHub](#) [Features](#) [Blog](#) [Login](#)

1,240,811 people hosting over 3,692,969 git repositories

jQuery, reddit, Sparkle, curl, Ruby on Rails, node.js, ClickToFlash, Erlang/OTP, CakePHP, Redis, and [many more](#)

facebook twitter Microsoft vmware redhat LinkedIn YAHOO!

**git** /git/  
Git is an extremely fast, efficient, distributed version control system ideal for the collaborative development of software.

**git·hub** /git\_hΛb/  
GitHub is the best way to collaborate with others. Fork, send pull requests and manage all your **public** and **private** git repositories.

**Plans, Pricing and Signup**  
Unlimited public repositories are free!

Free public repositories, collaborator management, issue tracking, wikis, downloads, code review, graphs and much more...

**Team management**  
30 seconds to give people access to code. No SSH key required. Activity feeds keep you updated on progress.  
[More about collaboration](#)

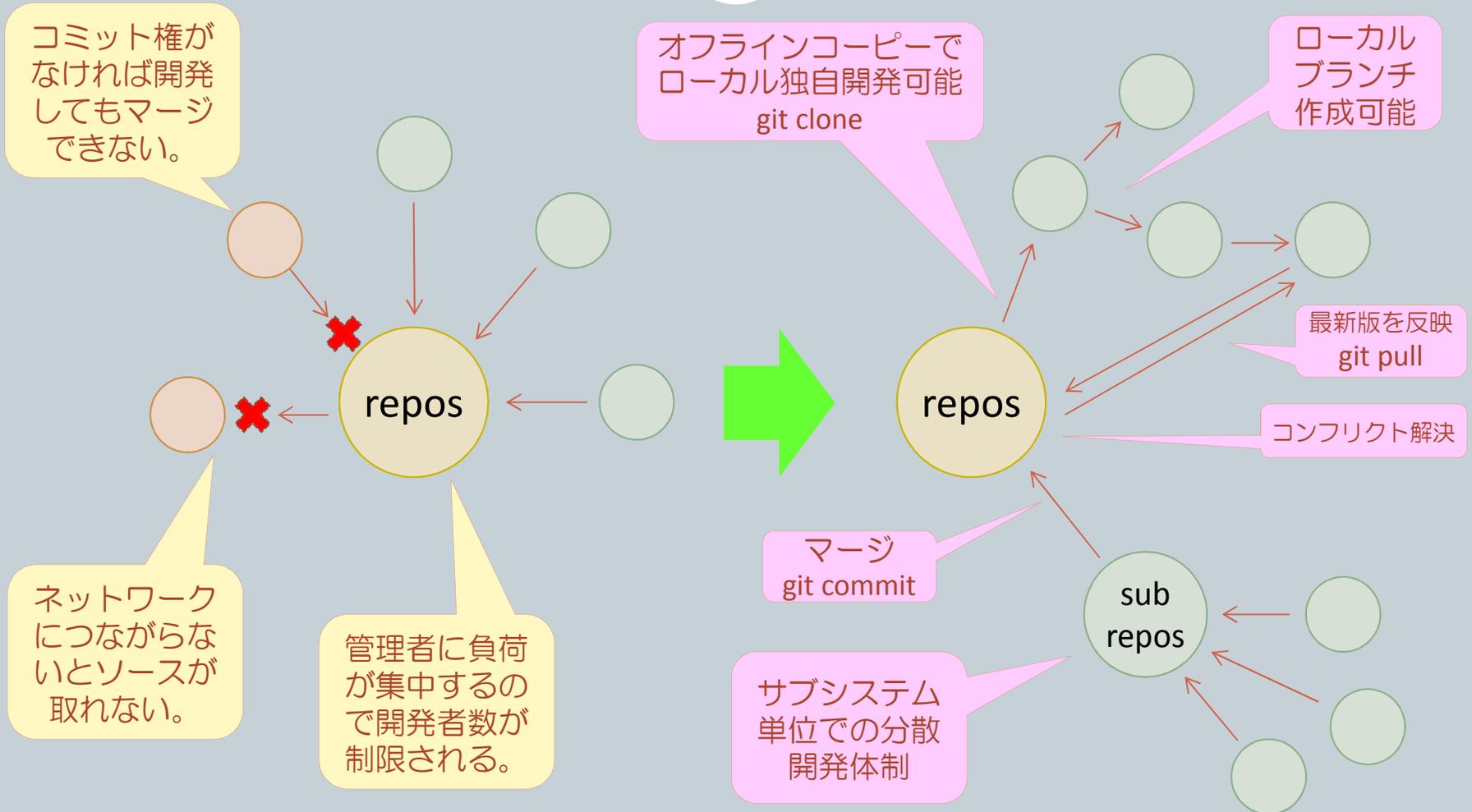
**Code review**  
Comment on changes, track issues, compare branches, send pull requests and merge forks.  
[More about code review](#)

**Reliable code hosting**  
We spend all day and night making sure your repositories are **secure**, **backed up** and **always available**.  
[More about code hosting](#)

**Open source collaboration**  
Participate in the most important open source community in the world today—online or at one of our meetups.  
[More about our community](#)

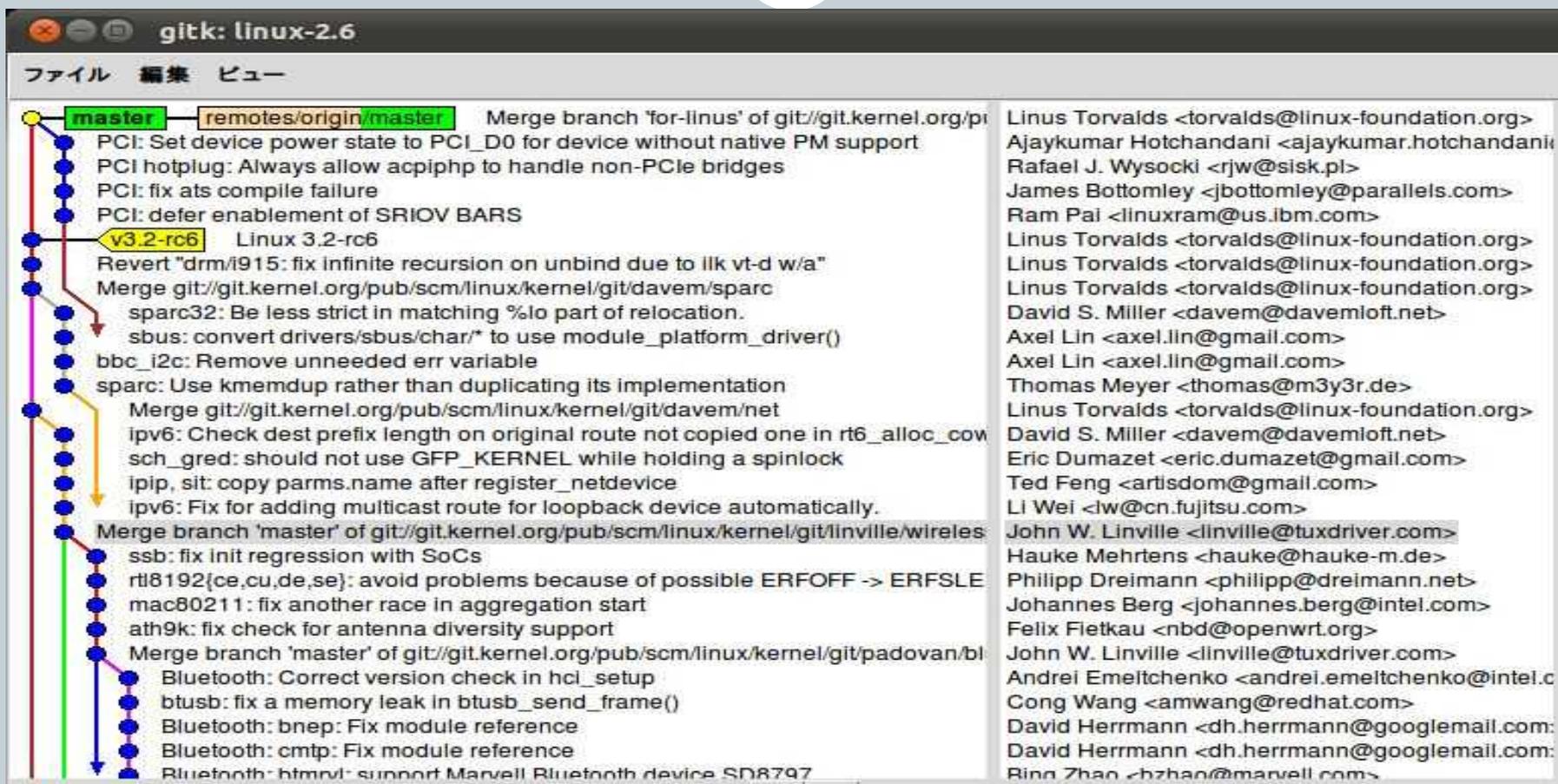
# 集中管理 vs. 分散管理

40



# マージプロセスの可視化ツール gitk

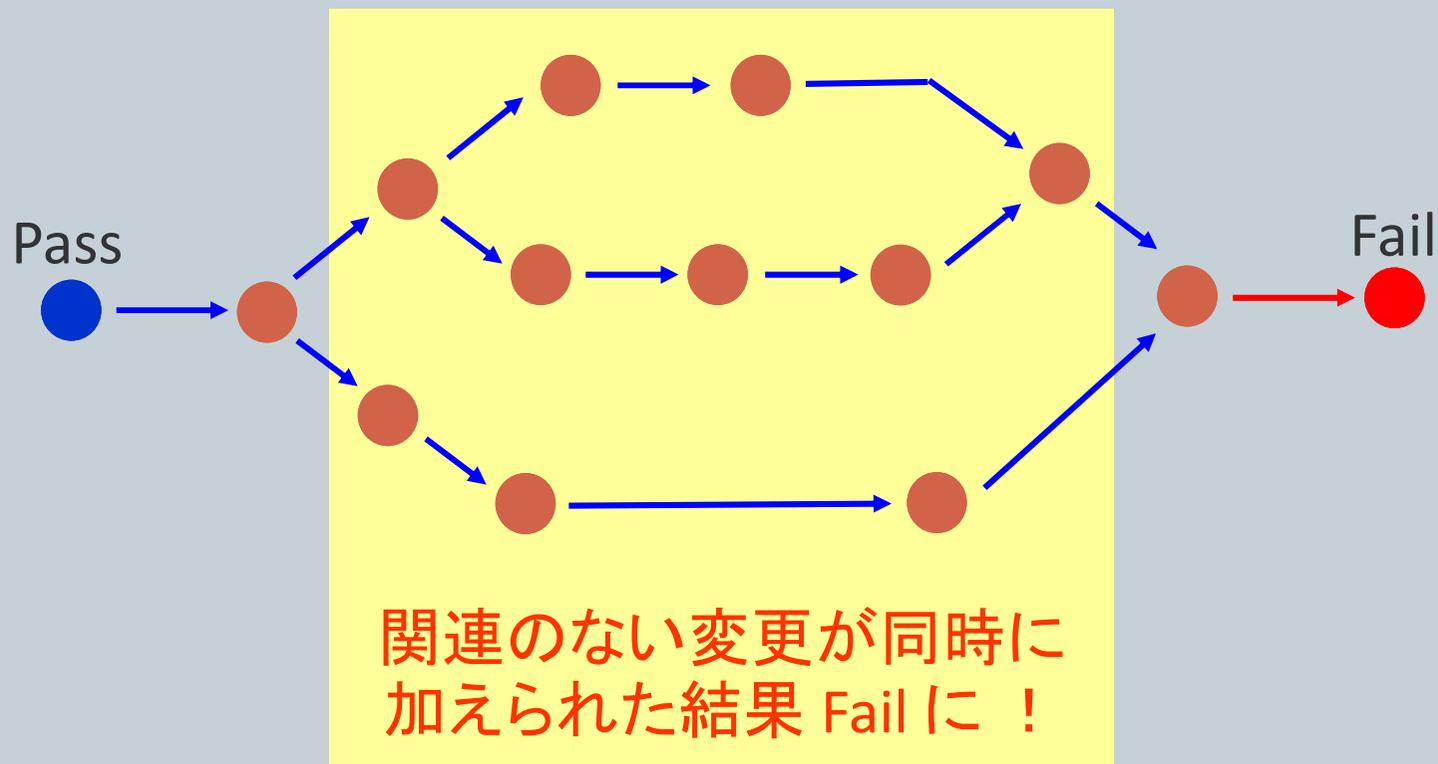
41



↑ 多点同時変更されたソースコードを集約管理する極めて優れたコード管理機構

# git bisection (二項探索)

42

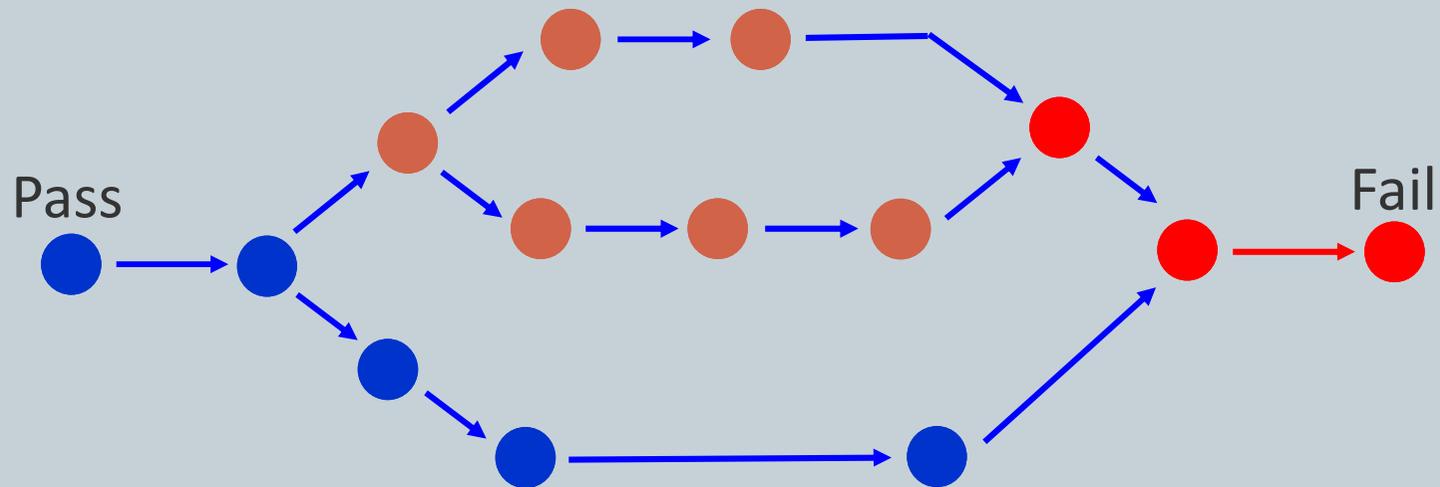


どのマージラインでバグが混入したかを切り分ける作業の自動化



# git bisection (二項探索)

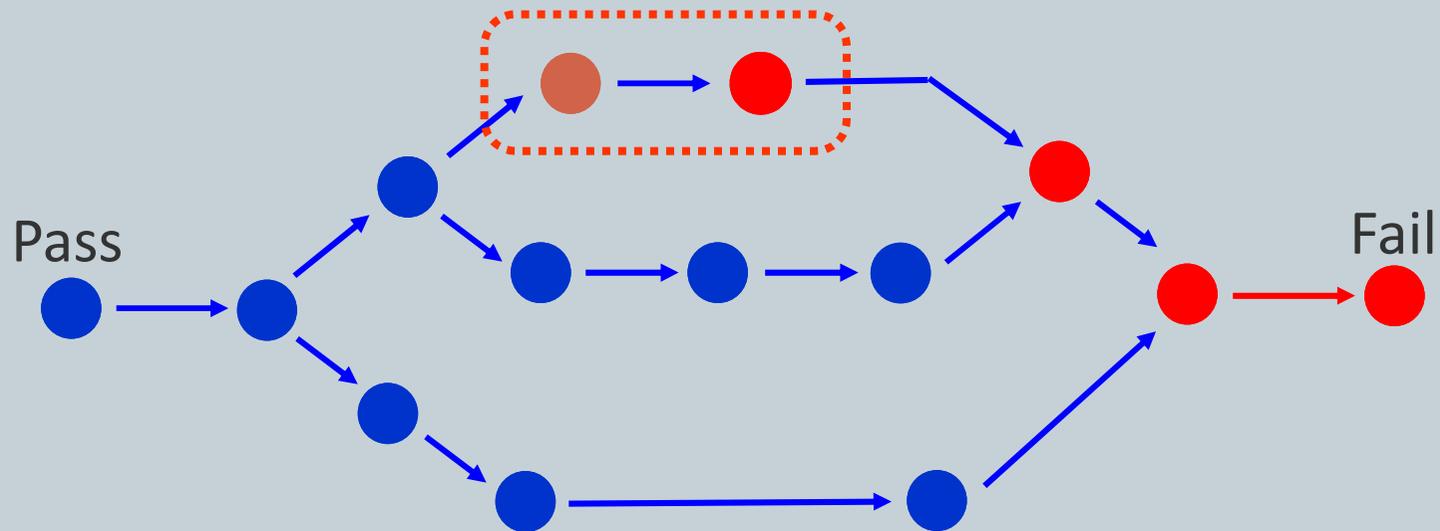
44



どのマージラインでバグが混入したかを切り分ける作業の自動化

# git bisection (二項探索)

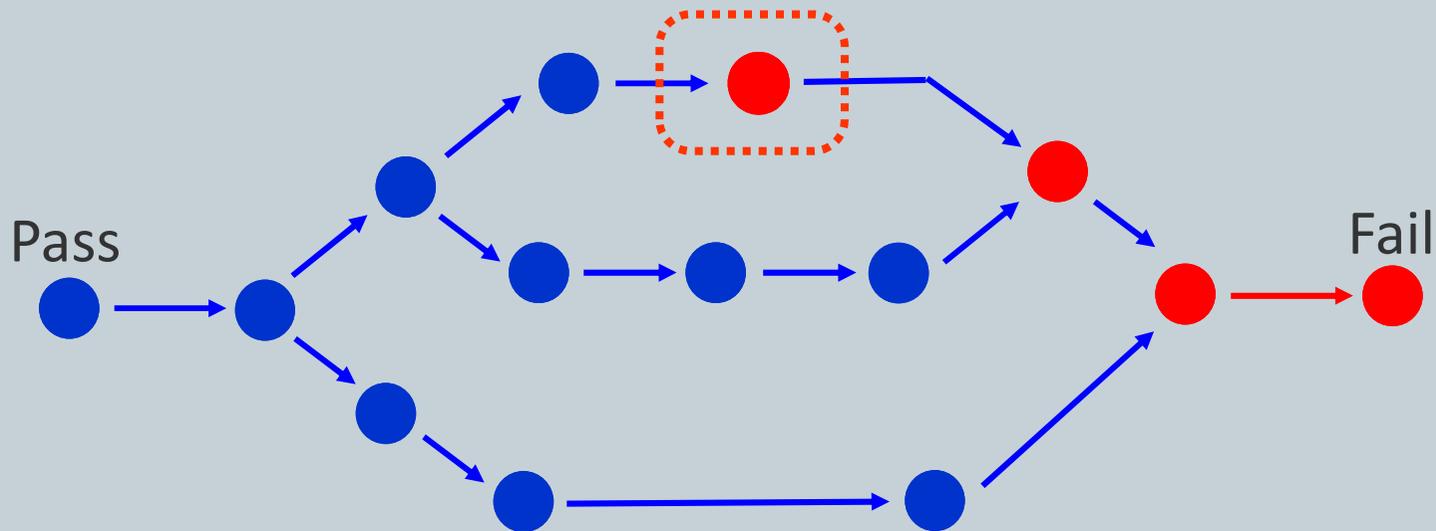
45



どのマージラインでバグが混入したかを切り分ける作業の自動化

# git bisection (二項探索)

46



バグを混入させる原因となったパッチが特定できた！

どのマージラインでバグが混入したかを切り分ける作業の自動化

# オープンソース開発への参加方法

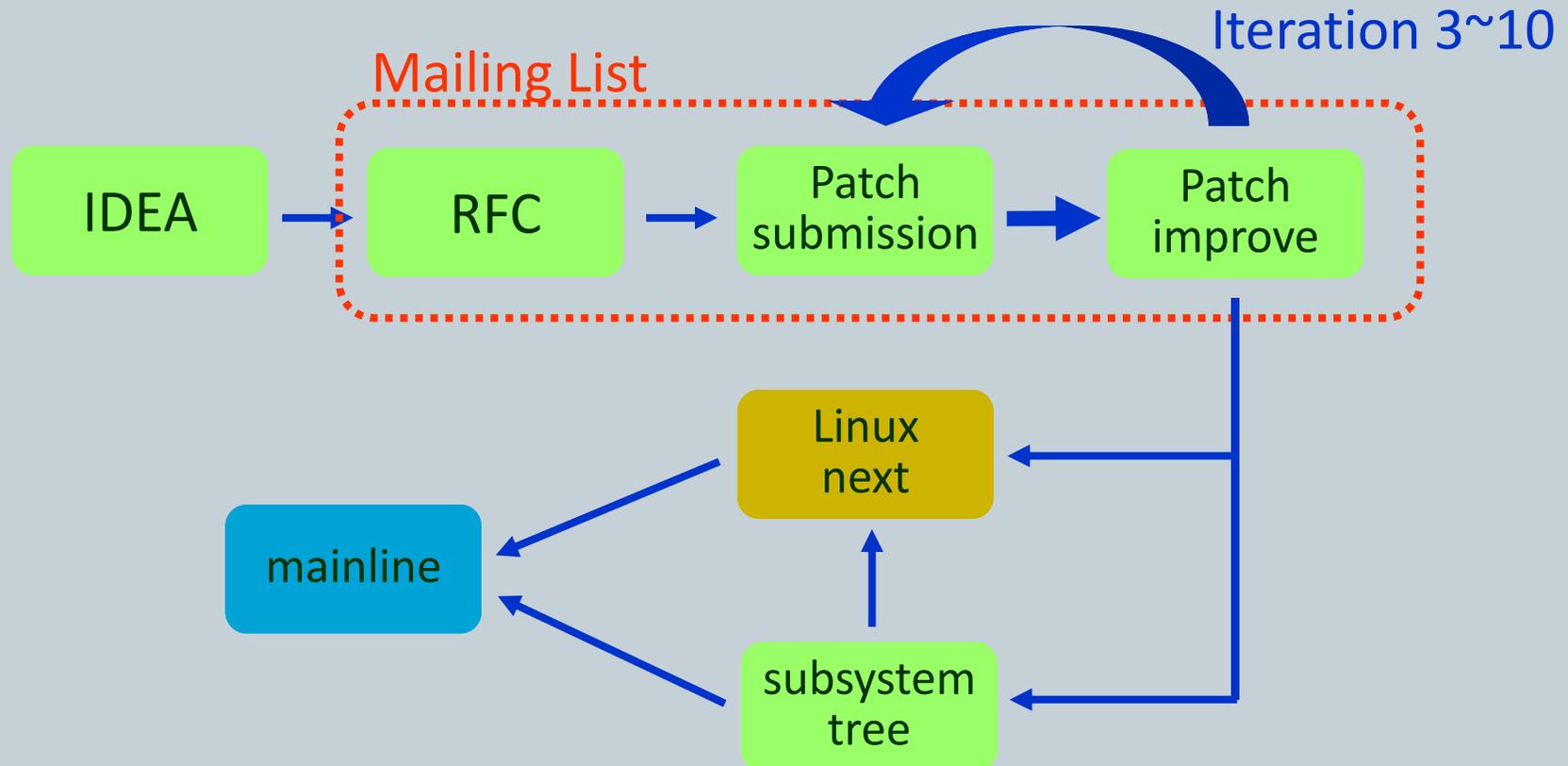
# オープンソース開発の仕組みを理解する

48

- 基本的には ML (Mailing List) での議論である
  - LKML (Linux kernel mailing list)
  - 各サブシステム毎の個別 ML
  - ML 購読は自由だが、膨大な数のメールがやり取りされる
  - 提案する場合 ML のスピード感に追従できないと取り残される
- 実際のコードもパッチとして ML 本文に書く
  - 添付ファイルは禁止
  - 1つのメールには1つのパッチ (提案) のみ
  - 決められたパッチの書き方 (フォーマット) に従う
    - ✦ git-format-ptach コマンドで生成される文章をメールで送信する
    - ✦ [scripts/checkpatch.pl](http://sourceforge.jp/projects/linuxjf/wiki/SubmittingPatches) というチェックスクリプトで送信前に確認する
  - 参考ドキュメント = <http://sourceforge.jp/projects/linuxjf/wiki/SubmittingPatches>
- アイディアは構想段階から出す (小さな段階で出す)
  - 完成した機能を提案しても受け入れられない。
  - 二項探索でデバッグ可能な小さな単位でパッチを区分けする。

# パッチ投稿プロセスを理解する

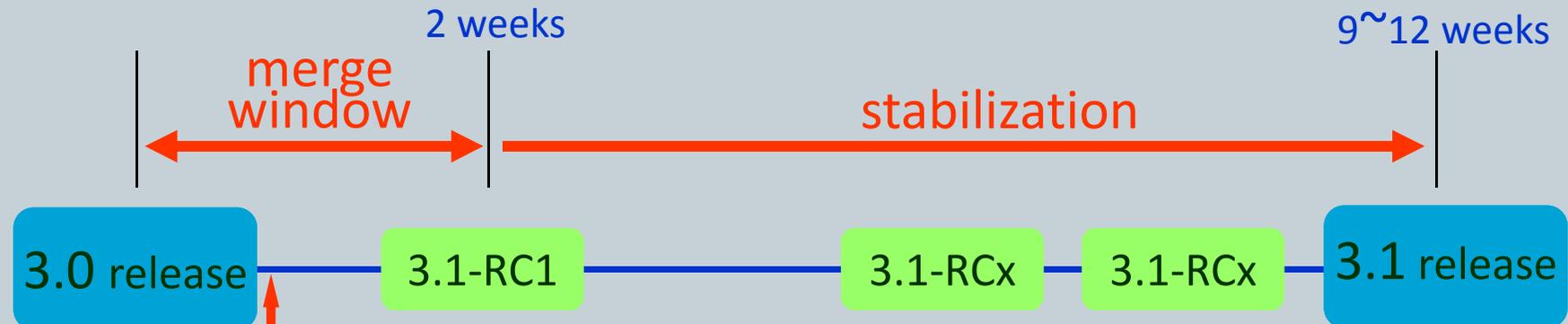
49



ML上で提案を議論、どんな経験者でも数回の書き直しを行って採用

# コードレビュープロセスを理解する

50



- Linux-next の段階で毎日 kernel build を実施しマージで問題が出ないことを確認しておく
- カーネルがリリースされると、Linux-next のコードを取り込む2週間のマージウィンドウが開く
- RC1 リリース後は、新規マージを停止して安定化のためのシステム検証期間。通常は RC8~9 までいく

Linux-next に登録されてから最大22週間後にマージが完成する

# オープンソース開発参加の意義

51

- メインラインのコードに採用されて適切なメンテを実施していれば、次期バージョン移行にもそのコードが継承される。
  - ローカルに拡張したコードは、バージョンが新しくなった時にはあらためて独自で開発して実装しなければならない。
- パッチレビューでは各技術領域の世界のトップ技術者からのアドバイスを得ることができる。（学内、社内より有効）
- メインラインにパッチが採用された実績、それまでの ML 上での議論の経緯などが記録として残る。 個人の実績になる。
- 類似のアイデアを持っている人との連携により、より良い実装を実現できる可能性がある。レビューの過程で紛糾するケースもあるが、双方誠意をもって最善な方法を探る柔軟性も必要になる。

# 製品開発の現場とオープンソース開発

# 私の問題意識

ソフトウェア  
規模・複雑度

53

リナックスの製品適用が本格化



NetFront



Google TV



DTV では

2005

2010

最近のデジタル家電製品の多くはリナックスがなければ成立しなくなっている。  
組込み業界でも **ソフトウェア規模複雑度が非常に高くなっているのに対して開発  
(支援) 体制の整備が追い付いていない** のではないか？ どう解決したらよいのか？

# カーネルメンテナンスに必要なリソース規模

54

## How many of developers does the team need to have?

柴田さん  
講演資料

- Latest Linux have 14.7MLines
  - If you were a manager of the proprietary development team, you may count the number of developers as follows..

A developer can handle 100KL and completely understand after 2 years

Development team need to be = 147 person !  
 $14.7\text{ML} / 100\text{KL}/\text{person} = 147$

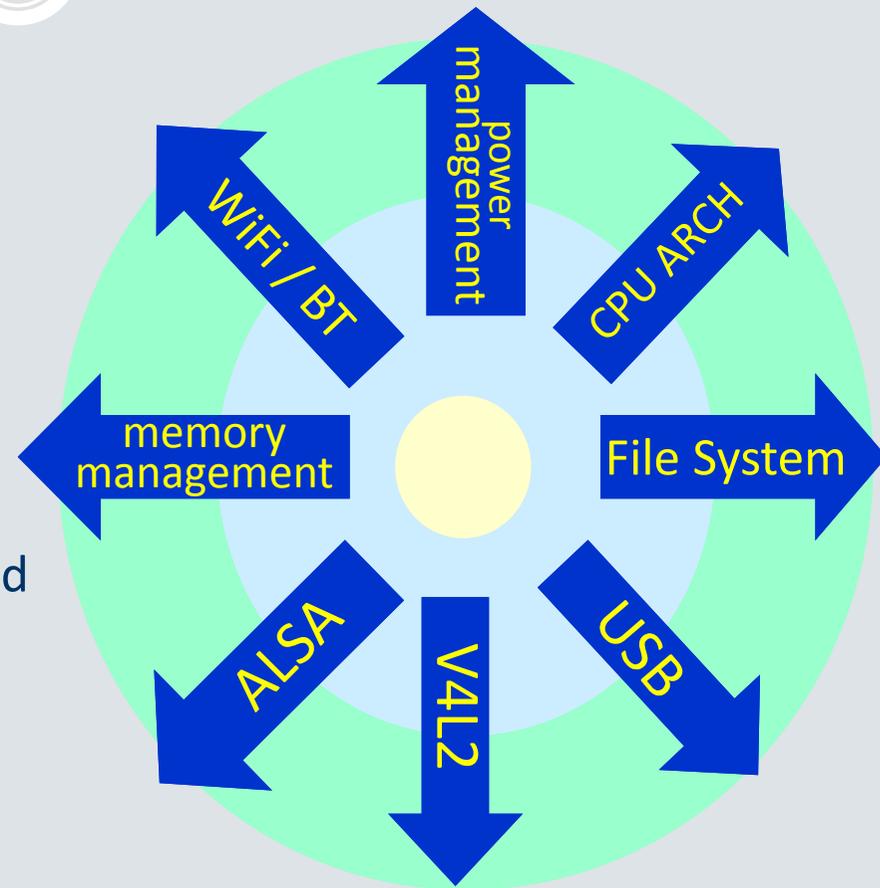
When hiring a developer takes 150K\$/year  
 $147 \times 150 = 2.2\text{M}\$/\text{year} \dots 176\text{MYen}/\text{year}$

リナックスを巨大なブラックボックスにしておくわけにいかなくなるが  
各社が社内で独自に完全なメンテナンス体制を構築するのは事実上不可能

# Upstream principals = divergence

55

- Think sustainable evolution
  - random technical improve
  - no specific shred narrow target
  - allow diversity
- Ever lasting development
  - no specific due date
  - Think for 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**

# Production principals = convergence

56

- Clear production goal
  - strict release due date
  - sever 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



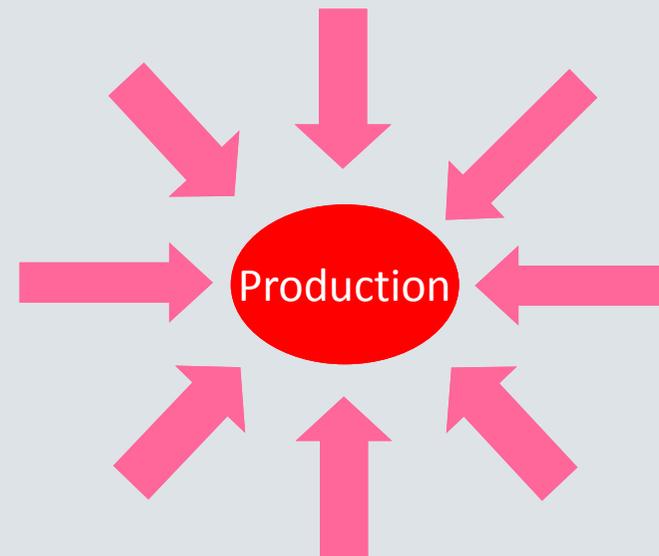
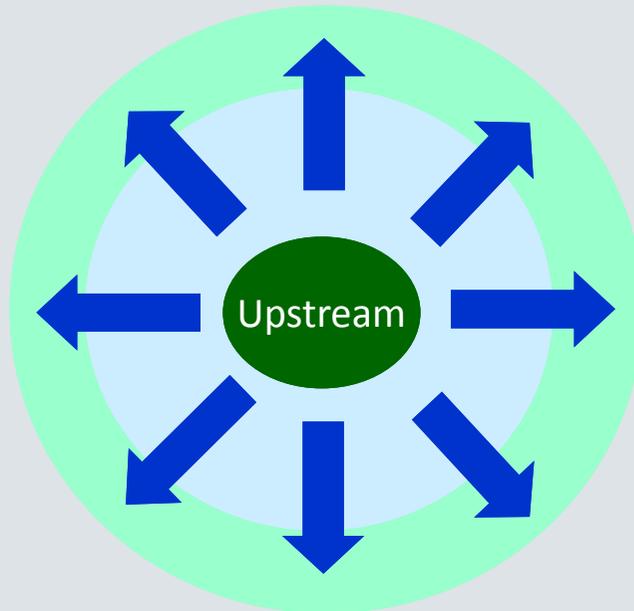
**Industry developer work for their current particular product**

# completely controversial mentality

“Upstream work can not be a part of production development”

57

- It is hard to share upstream development and production development together, as their motivation is completely opposite.

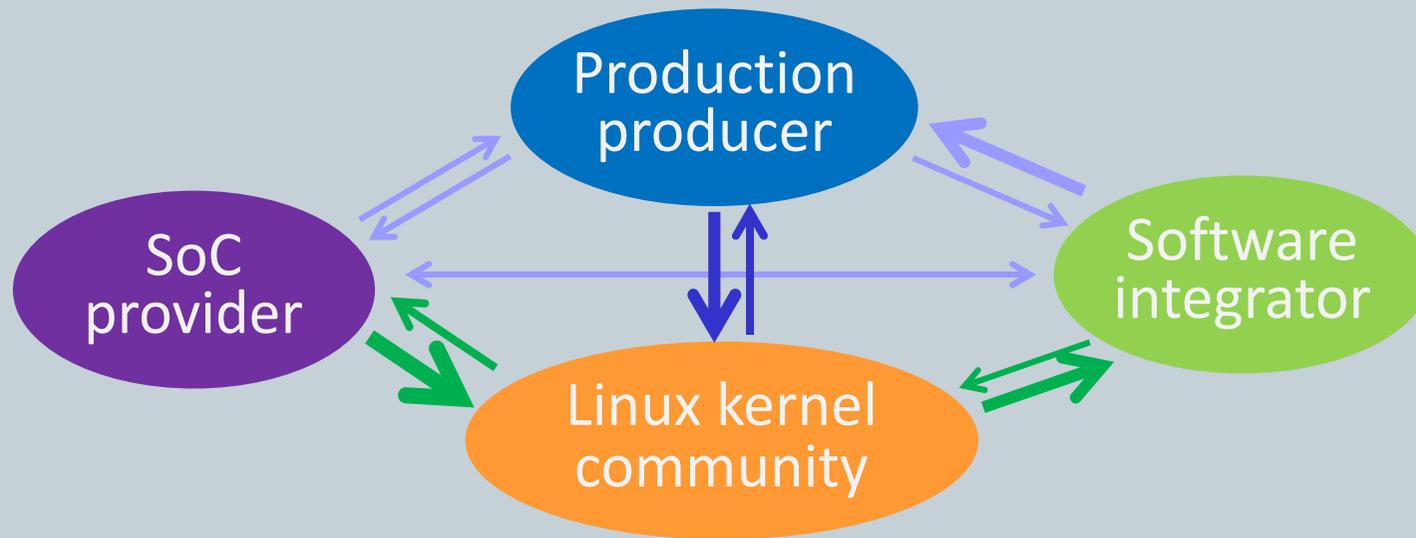


This circumstances should not be specific for embedded forks

# 組み込みリナックス エコシステム

58

各社が非差異化技術であるオープンソース開発、保守について、別々に同じ様な事をやってしまうのは冗長。



カーネル開発コミュニティを中心とした組み込み機器向けのオープンソース開発エコシステムの再構築が課題である

# オープンソース活用課題と皆さんへの期待

# オープンソース活用レベル 3段階

60

レベル	活用内容
Consumer	evaluating using deploying
Contributor	contributing community relations
Leader	building communities starting new opensource project opensourcing proprietary code

# Community = Trusted network

61

- オープンな世界とは言っても、全く実績の無い開発者の書いたコードが、どんどんメインラインに入る訳ではない
- 企業のバックアップ体制は重要だが、最終的には個人対個人のこれまでの実績、貢献に裏付けられた信頼があつての世界
- 正直、これまで日本（及びアジア圏）の開発者でこの信頼のネットワークの中に入れている人は少数派。今後日本が世界の中で地位を確保していくためには、皆さんが世界レベルの信頼された開発者になっていくことが非常に重要である

trusted connection の中に確固たる地位を築いてグローバルに活躍してほしい

## エンジニアがコミュニティと接する機会は実は沢山ある

62

- リナックス開発は文字通りオープンな開発プロセス。
  - 世界中の開発者（英語ノンネイティブも珍しくない）
  - 特定企業や団体の意向が開発に反映されることはない
  - ライセンス、ルールを守れば誰にでも門戸は開かれている
  - 近年韓国、中国などアジア圏からの参加が急増中
  - 一方で日本から国際会議に参加するエンジニアは減少中
  - 通常メールベースの議論だが、顔を知ってもらっただけで開発がスムーズに進むようになるのがコミュニティ
- 実際の技術会議には、極めて充実したコンテンツと実際の大物開発者に会って直接勉強するチャンスが沢山ある！

皆さんもこのような世界規模の開発に積極的に参画して頂きたい

# 実例 : Embedded Linux Conference Europe 2011

<https://events.linuxfoundation.org/archive/2011/embedded-linux-conference-europe/schedule>

63

8:00	Registration & Continental Breakfast		
9:00	Imagine a World Without Linux Jim Zemlin, Executive Director of The Linux Foundation		
9:30	Kernel Developer Panel Linus Torvalds, Alan Cox, Thomas Gleixner and Paul McKenney Moderated by Lennart Poettering		
10:30	Break		
10:45	Linaro's Android Platform Zach Pfeffer, Linaro	Another Mile Down the RT Road Thomas Gleixner, Linutronix	The Yocto Project Eclipse plug-in: An effective IDE environment for both Embedded Application and System developers Jessica Zhang, Intel
11:45	Contributing to the Community? Does your Manager Support You? Satoru Ueda, Sony Corporation / Japan OSS Promotion Forum	Embedded Linux Optimization Techniques: How Not To Be Slow Benjamin Zores, Alcatel-Lucent	Remote Processor Messaging Ohad Ben-Cohen, Texas Instruments
12:30	Lunch		
14:00	Collaborative Initiatives in Embedded Linux Jeff Osier-Mixon, Intel	Leveraging Android's Linux Heritage Karim Yaghmour, Opersys, Inc.	Using pytimechart For Real World Analysis Pierre Tardy, Intel
15:00	Optimizations for Cheap Flash Media Arnd Bergmann, Linaro	Saving Power with Wi-Fi: How to Prolong Your Battery Life and Still Stay Connected Vitaly Wool, Sony Ericsson	Developing Embedded Linux Devices Using the Yocto Project and What's new in 1.1 David Stewart, Intel
15:45	Break		
16:15	Android is NOT Just "Java on Linux" Tetsuyuki Kobayashi, Kyoto Micro Computer	Using Buildroot For a Real Project Thomas Petazzoni, Free Electrons	
17:15	Status of Embedded Linux BoFs Tim Bird, Sony Network Entertainment	The Linux NFC Subsystem Lauro Ramos Venancio, Instituto Nokia de Tecnologia & Samuel Ortiz, Intel	Board Bringup: LCD and Display Interfaces David Anders, Texas Instruments
18:30	Attendee Reception U-Fleku Brewery		

	Nadir	Kepler	Tycho	Stella
8:00	Registration & Continental Breakfast			
9:00	Re-Defining the Cloud Phone Antti Aumo, President of Global Solutions at Ixonos			
9:30	Reflection on 20 Years of Linux Dirk Hohndel, Chief Linux and Open Source Technologist at Intel			
10:00	Break			
10:15	Device Tree Status Report Grant Likely, Secret Lab	Success Story of the Open-Source Camera Stack: The Nokia N9 Case Laurent Pinchart, Ideas on Board	SOC Power Management - Debugging and Optimization Techniques Avinash Mahadeva, Texas Instruments & Vishwanth Sripathy, Texas Instruments	Android Development Platform with the Snapdragon Processor Tia Cassett, Qualcomm & Mike Chalupa, BSquare
11:15	Power Management Using PM Domains on SH7372 Rafael J. Wysocki, Faculty of Physics, U. Warsaw / SUSE Labs	A Generic Clock Framework in the Kernel: Why We Need It & Why We Still Don't Have It Sascha Hauer, Pengutronix e.K.	Android Platform Optimizations Ruud Derwig, Synopsys	Kernel Development using the DragonBoard with the Snapdragon Processor Tia Cassett, Qualcomm & David Brown, Qualcomm
12:00	Lunch & Technology Showcase Be sure to check out the technology showcase, displays will be open from 12:00 - 14:30.			
13:30	Linux for In-Car Infotainment Matt Jones, Vice President, GENIVI Alliance & Technical Lead - Next Generation Infotainment, Jaguar Land Rover (This Keynote Presentation Will Be Held in the Nadir Room)			
14:30	Break			
15:00	DRM Driver Development For Embedded Systems Inki Dae, Samsung Electronics	Consolidating Linux Power Management on ARM Multiprocessor Systems Lorenzo Pieralisi, ARM Ltd.	Using Qt For Non-Graphical Applications Thomas Petazzoni, Free Electrons	PandaBoard Workshop: Booting the PandaBoard David Anders, Texas Instruments
16:00	ARM DMA-Mapping Framework Redesign and IOMMU Integration Marek Szymanski, Samsung Electronics Polska Sp. z o.o. & Kyungmin Park, Samsung Electronics Corporation	Thermal Framework for ARM based SOCs Keerthy Jagadeesh, Texas Instruments & Vishwanath Sripathy, Texas Instruments	Efficient JTAG-Based Linux Kernel Debugging Marc Tittiger, ST Microelectronics	PandaBoard Workshop: PandaBoard Expansion I/O David Anders, Texas Instruments
17:00	Toward the Long Term Stable Kernel tree for The Embedded Industry Tsugikazu Shibata, NEC & Linux Foundation Board Member	Secure Virtualization in Automotive Lisko Lappalainen, MontaVista Software	Yocto Project Community BoFs Jeff Osier-Mixon, Intel	PandaBoard Workshop: WLAN kernel hacking with PandaBoard Luca Coelho, Texas Instruments
18:30	VIP Dinner Reception A reception at Duplex for our speakers and sponsors.			

組み込みリナックス関連だけで約50セッション、基調講演8本の充実した内容で、最先端のコミュニティ開発状況を吸収可能

# linux.conf.au

## Ballarat Australia 2012-1/16~20

ICT

英語

海外

64



linux.conf.au  
ballarat 2012  
16th - 20th january

Home About Sponsors Programme Register Wiki Blog Contact

Sign in  
Create an account

News

[Humanitarian FOSS Miniconf cancelled](#)  
(2011-12-13)

[Jacob Appelbaum watches the watchers - fourth keynote for linux.conf.au 2012 announced](#)  
(2011-11-29)

[With its stellar line up, linux.conf.au 2012 should be on your horizon!](#)  
(2011-11-19)

[Miniconfs are reaching for a star - you!](#)  
(2011-11-15)

[More news...](#)

Press

Welcome to linux.conf.au 2012!

Welcome!

Welcome to the home of linux.conf.au 2012, Australia's premier free and open source conference!

Registrations now open!

**Early bird registrations are now sold out!** Hopefully you managed to grab yours, but if not, don't fret! Regular registrations will remain open until the conference starts or sold out.

Does internet censorship interest you? Jacob Appelbaum to give 4th keynote address at linux.conf.au 2012

Ever worried about the state of surveillance on the internet - and just who has access to your private data?

So does Jacob Appelbaum!

Jacob Appelbaum ([@ioerror](#), [http://en.wikipedia.org/wiki/Jacob\\_Appelbaum](http://en.wikipedia.org/wiki/Jacob_Appelbaum)) is a renowned independent internet security professional, accomplished photographer, software hacker and world traveller.

# Embedded Linux Conference Redwood City, CA 2012-2/15~17

組み込み

英語

海外

<https://events.linuxfoundation.org/events/linuxcon-japan>

65

[Home](#) [About](#) [Events](#) [Sponsorship](#) [Newsroom](#) [Multimedia](#) [Archive](#) [Contact Us](#)



February 15 - 17, 2012 ■ Hotel Sofitel San Francisco Bay ■ Redwood Shores, CA

[Home](#) | [CFP](#) | [Sponsor](#) | [Hotel & Travel](#) | [Register](#)

Register  
Now

Submit  
Proposal

## 2012 Embedded Linux Conference

February 15 - 17, 2012 · Hotel Sofitel SF Bay, Redwood Shores, CA

The Embedded Linux Conference (ELC) is the premier vendor-neutral technical conference for companies and developers using Linux in embedded products. This conference, now in its 6th year, has the largest collection of sessions dedicated exclusively to embedded Linux and embedded Linux developers. ELC is embedded Linux experts talking about solutions to your embedded Linux problems. ELC consists of 3 days of presentations, tutorials and Bird-of-a-Feather sessions. There are over 50 sessions to choose from, on a wide variety of topics.

### About the CE Workgroup

The Linux Foundation and the CE Linux Forum announced a merger on October 27, 2010. CELF is now a workgroup of the Linux Foundation. The use of Linux in embedded products has skyrocketed in recent years, with Linux now being

## Keynote Speakers

Keynote speakers will be announced shortly.

## Latest News

3 November 2011 2012 Embedded Linux Conference  
Call For Participation Announced. [Submit a proposal now.](#)

## Newsletter Sign-Up

Platinum Sponsor



Media Sponsor



For information on sponsorship opportunities at the 2012 Embedded Linux Conference, please contact [Angela Brown, Events Director.](#)

# LinuxCon Japan 2012 Pacifico Yokohama, 2012-6/6~8

組み込み

英語

国内

<https://events.linuxfoundation.org/events/linuxcon-japan>

66



LINUXFOUNDATION.ORG | LINUX.COM | TRAINING | VIDEO

LOGIN SIGN UP

THE LINUX FOUNDATION

Home About **Events** Sponsorship Newsroom Multimedia Archive Contact Us

**LINUXCON JAPAN 2012**

June 6 - 8, 2012 • Pacifico Yokohama • Yokohama, Japan

Home | CFP | Sponsor | Hotel & Travel | Register

**Submit Proposal**

**LinuxCon Japan 2012**

**June 6 - 8, 2012 • Pacifico Yokohama • Yokohama, Japan**

LinuxCon Japan is the premiere Linux conference in Asia that brings together a unique blend of core developers, administrators, users, community managers and industry experts. It is designed not only to encourage collaboration but to support future interaction between Japan and other Asia Pacific countries and the rest of the global Linux community. The conference includes presentations, tutorials, birds of a feather sessions, keynotes, sponsored mini-summits. LinuxCon Japan will be the showcase Japan and Asian conference for The Linux Foundation and we greatly appreciate your support as we bring together top talent from Japan, Asia and around the world.

**Who Should Attend:**  
Software Developers, Linux IT Professionals, Corporate End Users, Senior Business Executives, IT Operations Experts, System Administrators, Students, Media and anyone else with an interest in the Linux ecosystem.

**Keynote Speakers**

Keynote speakers will be announced shortly.

**Latest News**

5 December 2011 **LinuxCon Japan Call For Participation Announced. [Submit a proposal now.](#)**

**Platinum Sponsor**

**intel**

For information on sponsorship opportunities at LinuxCon Japan 2012, please contact [Angela Brown](#), Events Director.

# CEWG Jamboree (年4回程度)

組み込み

日本語

国内

67

## Japan Technical Jamboree 39

### *CE Linux Forum Japan Technical Jamboree*



Date: December 9th / 日付: 12月9日 (金)

At Nakano Sunplaza / 於、中野サンプラザ



We are calling for session proposal now! / セッション提案募集中!

Session proposal how-to. / 提案の方法

#### Contents [hide]

- 1 Introduction / はじめに
  - 1.1 Special remarks for non Japanese speakers
  - 1.2 Previous Jamboree
- 2 Date and venue... / 日付・場所...
  - 2.1 Registration / 参加登録
- 3 Main Topics
- 4 Agenda / 進行
  - 4.1 Agenda
  - 4.2 Special Remarks
- 5 Ask for your help / お願ひ
  - 5.1 Presentation Materials
  - 5.2 English Translation Volunteer

現場のエンジニア同士がオープンな技術を語り合うホットなサロン

# まとめ

68

- オープンソースの適用範囲は今後更に拡大していきます。この機会にオープンソースとは何か、誰が何のために開発しているのかその実像を理解してください。
- オープンソースの特性として未来志向、進化の追及など商用ソフトウェアとは異なった特徴がある事を理解して、適切な活用方法を考えてください。
- 皆さんの中で将来のオープンソース開発を主導するようなワールドクラスの開発者が誕生することを期待しています。今日説明した開発プロセス、コミュニティへの参加方法を手掛かりに是非トライしてください。世界は開かれています。