

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 何故、営利企業が近年オープンソース開発に積極的になっているのか

宗像尚郎

ルネサスエレクトロニクス株式会社
オートモーティブソリューション事業本部

2020-1-21

自己紹介

ルネサスエレクトロニクス という半導体メーカーで働いています

- ルネサスの概要と私の担当業務
 - ルネサスエレクトロニクス = (日立+三菱) + NEC の半導体事業部門
 - 車載向け SOC (System on Chip) のグローバルリーディングサプライヤーです
 - R-CarH3 = 64bit ARM Octa-core + GPU + LPDDR4 8G、1,384pin BGA
 - 私は SOC を動かすために必要な **Linux BSP** の開発や提供を担当しています
- 産業コンソーシアム活動、コミュニティ活動でも指導的な役割をもっています
 - **Linux Foundation** (後述) の全世界で 22 名の理事メンバー の一人です
 - Automotive Grade Linux、yocto project の理事メンバーです
 - 社内オープンソース開発チームを運営、コミュニティ開発でも顕著な実績

ルネサスはオープンソース開発活動への貢献実績において日本を代表する企業です

Linux kernel 開発実績 (kernel 5.3, 2019-9-16 release の実績)

Most active 5.3 employers

By changesets			By lines changed		
Intel	1545	10.7%	AMD	432537	37.0%
AMD	966	6.7%	Linaro	115926	9.9%
(Unknown)	963	6.7%	Intel	76950	6.6%
Red Hat	925	6.4%	Mellanox	46732	4.0%
(None)	865	6.0%	Samsung	37869	3.2%
Google	638	4.4%	Google	31666	2.7%
Renesas Electronics	638	4.4%	(Unknown)	31310	2.7%
Linaro	548	3.8%	Red Hat	31089	2.7%
IBM	460	3.2%	IBM	29449	2.5%
Mellanox	440	3.0%	(None)	27803	2.4%
(Consultant)	410	2.8%	SUSE	18928	1.6%
Huawei Technologies	372	2.6%	Cirrus Logic	16509	1.4%
SUSE	337	2.3%	(Consultant)	14192	1.2%
Samsung	311	2.2%	Facebook	13723	1.2%
NXP Semiconductors	309	2.1%	Linux Foundation	13405	1.1%
Linux Foundation	291	2.0%	ARM	12755	1.1%
ARM	280	1.9%	Pengutronix	12398	1.1%
Facebook	188	1.3%	NXP Semiconductors	12202	1.0%
Oracle	170	1.2%	Renesas Electronics	11345	1.0%
BayLibre	170	1.2%	Huawei Technologies	10165	0.9%

Most active 5.3 developers

By changesets		By changed lines			
Kuninori Morimoto	271	1.9%	Hawking Zhang	364371	31.2%
Christoph Hellwig	262	1.8%	Arnd Bergmann	84993	7.3%
Mauro Carvalho Chehab	205	1.4%	Mauro Carvalho Chehab	34041	2.9%
Nishka Dasgupta	162	1.1%	Harry Wendland	33568	2.9%
Chris Wilson	160	1.1%	Jason Gunthorpe	22421	1.9%
Greg Kroah-Hartman	145	1.0%	Chris Wilson	17382	1.5%
Yue Haibing	137	0.9%	Richard Fitzgerald	15791	1.4%
Masahiro Yamada	130	0.9%	Linus Walleij	13392	1.1%
Gustavo A. R. Silva	128	0.9%	Greg Kroah-Hartman	11713	1.0%
Geert Uytterhoeven	120	0.8%	Bernard Metzler	10963	0.9%
Takashi Sakamoto	113	0.8%	Christoph Hellwig	10844	0.9%
Wolfram Sang	101	0.7%	Pawel Laszczak	7702	0.7%
Arnaldo Carvalho de Melo	99	0.7%	Hannes Reinecke	7255	0.6%
Colin Ian King	98	0.7%	Andrii Nakryiko	7143	0.6%
Arnd Bergmann	96	0.7%	Maxime Ripard	6709	0.6%
Adrian Hunter	96	0.7%	David Ahern	6557	0.6%
Russell King	94	0.7%	Daniele Ceraolo Spurio	5781	0.5%
David Howells	92	0.6%	Darrick J. Wong	5776	0.5%
Hawking Zhang	90	0.6%	Kuninori Morimoto	4717	0.4%
Maxime Ripard	84	0.6%	Thomas Zimmermann	4706	0.4%

<https://lwn.net/Articles/798505/>

Linux kernel 開発実績 (2005 年 4 月からの 14 年間強の累積)

Total patch sets of this kernel release: 822303
888 companies contribute their works to this kernel release.
Averagely, every companies committed 926 patch sets.

ⓧ No.1	Unknown	115707(14.07%)	ⓧ No.21	Linutronix	6499(0.79%)
ⓧ No.2	Intel	84981(10.33%)	ⓧ No.22	Linux Foundation	6092(0.74%)
ⓧ No.3	Red Hat	72879(8.86%)	ⓧ No.23	VISION Engraving and Routing Systems	6045(0.74%)
ⓧ No.4	Hobbyists	71581(8.70%)	ⓧ No.24	Canonical	5976(0.73%)
ⓧ No.5	Novell	35687(4.34%)	ⓧ No.25	Pengutronix	5679(0.69%)
ⓧ No.6	IBM	32948(4.01%)	ⓧ No.26	Analog Devices	5536(0.67%)
ⓧ No.7	Linaro	24061(2.93%)	ⓧ No.27	NXP	5466(0.66%)
ⓧ No.8	AMD	17729(2.16%)	ⓧ No.28	NVIDIA	5420(0.66%)
ⓧ No.9	Google	16856(2.05%)	ⓧ No.29	Fujitsu	5076(0.62%)
ⓧ No.10	Renesas Electronics	16624(2.02%)	ⓧ No.30	QUALCOMM	4902(0.60%)
ⓧ No.11	Samsung	16534(2.01%)	ⓧ No.31	Code Aurora Forum	4785(0.58%)
ⓧ No.12	Oracle	16006(1.95%)	ⓧ No.32	Freescale	4694(0.57%)
ⓧ No.13	Texas Instruments	15182(1.85%)	ⓧ No.33	Wolfson Microelectronics	4180(0.51%)
ⓧ No.14	Mellanox Technologies	9718(1.18%)	ⓧ No.34	Nokia	4072(0.50%)
ⓧ No.15	Academics	8311(1.01%)	ⓧ No.35	Cisco	4021(0.49%)
ⓧ No.16	HuaWei	8125(0.99%)	ⓧ No.36	Parallels	3805(0.46%)
ⓧ No.17	Consultants	8020(0.98%)	ⓧ No.37	Imagination Technologies	3773(0.46%)
ⓧ No.18	ARM	7815(0.95%)	ⓧ No.38	Marvell	3635(0.44%)
ⓧ No.19	Broadcom	7578(0.92%)	ⓧ No.39	QLogic	3390(0.41%)
ⓧ No.20	Bootlin	7013(0.85%)	ⓧ No.40	NetApp	2827(0.34%)

http://www.remword.com/kps_result/all_whole.html

オープンソース開発コミュニティの実像

オープンソースとは、どのようなソフトウェアなのか

何故わざわざ「オープン」とか「フリー」なソフトと呼んでいるのか？

オープンソースソフトウェア、フリーソフトウェアの意味するもの

- 「オープン」 = 文字通り開かれているという意味
 - ソースコードが公開されている
 - 開発者にアクセスすることができる
- 「フリー」には「無料」と「自由」の意味があるが、ここでは「自由な」という意味
 - ソフトウェアにおける「自由」とは何だろうか
 - Richard Stallman は 3つの自由 が担保される必要があると主張した
 - Stallman の GNU Free Software 運動 がオープンソース活動の起点となった

ソフトウェアは本来「開かれていなくて」「不自由」なものなのであえて区別している

Richard Stallman が Free Software Foundation (=FSF) を創設

Stallman が主張する Free Software(= 自由ソフトウェア) が担保すべき 3 つの自由

- 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。

オープンソースソフトウェア (= OSS)、開発コミュニティとは

関わり方を変えれば オープンソースソフトウェアの色々な側面 も見えてくるはず

- ユーザー (=使うだけの人)
 - 無料で使えてうれしい、寄付を要請されたけどどうしようかな
 - サポートとか、セキュリティ対策とかは大丈夫かな
- デベロッパー (=開発に参加する人)
 - プログラムを改造してみたいけど、ソースコードはどこにあるのかな？
 - 機能追加やバグ対策のパッチを書いたけど、受け取ってもらえるのかな？
- プロジェクトメンバー (=プロジェクトの成長を考える人)
 - 今後のリリース計画をどうしよう？ ユーザーは何を望んでいるかな？
 - 中長期的にプロジェクトを安定運営させるためのリソース(人、金)をどうしよう？

「利用段階」からステップアップして、背後にどんな仕組みがあるのかを考えてみよう

コミュニティで Good Citizen, First Class Citizen になるために

オープンソースの世界を支えているのは ” お互い様 (give & take) の精神 ”

- オープンソース開発者は、ユーザーが SW を無償利用できるようにしている
- ユーザーは OSS から商用 SW を作らない等の制約を守っていれば OK なのだが...
- **Good Citizen (善良な市民)、First Class Citizen (一級市民) とは OSS の健全な発展のために OSS 開発者を支援する方法を考え行動するグループ** を指す
 - ブログなどを使って OSS の紹介記事を書いて SW を宣伝する
 - 開発者に問題点や改善のアイデアなどをフィードバックする
 - 問題対策のパッチを作成してコミュニティにフィードバックする
 - OSS 開発プロジェクトに参加して、コミュニティの運営を支援する
 - 自分でも別の OSS を開発してコードをコミュニティにフィードバックする

露骨なただ乗り (**Free Ride**) をするとバッシングされる懸念もあることに注意

オープンソースのフリーライドに対抗するライセンス変更が顕在化

Redis、MongoDB、Kafkaらが相次いで商用サービスを制限するライセンス変更。AWSなどクラウドベンダによる「オープンソースのいいとこ取り」に反発

2019年1月16日

オープンソースソフトウェアの開発元がクラウドベンダへの不満を表明し、商用サービス化を制限するライセンス変更を行う例が続いています。

高速なインメモリデータストアを実現するオープンソースソフトウェアとして知られる「Redis」の開発元「Redis Labs」は、2018年8月に同社が開発したRedis拡張モジュールに関するライセンスの変更を発表しました（Redis本体のライセンスはBSDライセンスのまま）。拡張モジュールには、全文検索を行うRediSearchなどが含まれていません。

変更後のライセンスでは、クラウドベンダによる商用サービスでの利用を制限するものとなっています。

<https://www.publickey1.jp/blog/19/redismongodbkafkaaws.html>

コンピュータプログラムは **著作物** なので複製や改造には制限がかかる

コンピュータプログラムは音楽や小説と同じように **著作権による保護対象** である

- 言語 (論文、小説、脚本、詩歌、俳句、講演など)
- 音楽 (楽曲及び楽曲を伴う歌詞)
- 美術 (絵画、版画、彫刻、漫画、書、舞台装置など)
- 建築 (芸術的な建造物 (設計図は図形の著作物))
- 地図や図形 (地図と学術的な図面、図表、模型など)
- 映画 (劇場用／テレビ、ビデオソフト、ゲームソフト)
- 写真、グラビアなど
- **コンピュータプログラム**

著作権（財産権）

複製権	著作物を印刷、写真、複写、録音、録画などの方法によって有形的に複製する権利
上演権・演奏権	著作物を公に上演したり、演奏したりする権利
上映権	著作物を公に上映する権利
公衆送信権・公の伝達権	著作物を自動公衆送信したり、放送したり、有線放送したり、また、それらの公衆送信された著作物を受信装置を使って公に伝達する権利 *自動公衆送信とは、サーバーなどに蓄積された情報を公衆からのアクセスにより自動的に送信することをいい、また、そのサーバーに蓄積された段階を送信可能化という。
口述権	言語の著作物を朗読などの方法により口頭で公に伝える権利
展示権	美術の著作物と未発行の写真著作物の原作品を公に展示する権利
頒布権	映画の著作物の複製物を頒布（販売・貸与など）する権利
譲渡権	映画以外の著作物の原作品又は複製物を公衆へ譲渡する権利
貸与権	映画以外の著作物の複製物を公衆へ貸与する権利
翻訳権・翻案権など	著作物を翻訳、編曲、変形、翻案等する権利（二次的著作物を創作することに及ぶ権利）
二次的著作物の利用権	自分の著作物を原作品とする二次的著作物を利用（上記の各権利に係る行為）することについて、二次的著作物の著作権者が持つものと同じ権利

著作者人格権 (from 公益社団法人著作権情報センター)

著作者人格権

公表権	自分の著作物で、まだ公表されていないものを公表するかしないか、するとすれば、いつ、どのような方法で公表するかを決めることができる権利
氏名表示権	自分の著作物を公表するときに、著作者名を表示するかしないか、するとすれば、実名か変名かを決めることができる権利
同一性保持権	自分の著作物の内容又は題号を自分の意に反して勝手に改変されない権利

参考条文…[著作権法第18条～第20条](#)

<http://www.cric.or.jp/qa/hajime/hajime2.html>

OSD (Open Source Definition) = Open Source と名乗るための条件

OSI (Open Source Initiative) が OSD を規定した

- 1. 再頒布の自由
- 2. ソースコード (を入手できること、再配布できること等)
- 3. 派生ソフトウェア (を作れて、同じライセンスを付与すること)
- 4. 作者のソースコードの完全性 (integrity) (が担保されること)
- 5. 個人やグループに対する差別の禁止
- 6. 利用する分野 (fields of endeavor) に対する差別の禁止
- 7. ライセンスの分配 (distribution) (に際し追加ライセンスが不要)
- 8. 特定製品でのみ有効なライセンスの禁止
- <http://www.opensource.jp/osd/osd-japanese.html>

OSI が承認したオープンソースライセンスのリストが公開されている

Licenses & Standards

About Open Source Licenses

Open source licenses are licenses that comply with the [Open Source Definition](#) — in brief, they allow software to be freely used, modified, and shared. To be approved by the Open Source Initiative (also known as the OSI), a license must go through the [Open Source Initiative's license review process](#).



Popular Licenses

The following OSI-approved licenses are popular, widely used, or have strong communities:

- Apache License 2.0
- BSD 3-Clause "New" or "Revised" license
- BSD 2-Clause "Simplified" or "FreeBSD" license
- GNU General Public License (GPL)
- GNU Library or "Lesser" General Public License (LGPL)
- MIT license
- Mozilla Public License 2.0
- Common Development and Distribution License
- Eclipse Public License version 2.0

2020年1月時点で、既に無効な物も含め、96種類のライセンスがオープンソースライセンスとして登録されている

<https://opensource.org/licenses/alphabetical>

オープンソース利用時に守るべき事項 がライセンスに規定されている

特に OSS を利用して開発した コードの公開義務 の考え方が異なる

■ Copy-left 型ライセンス

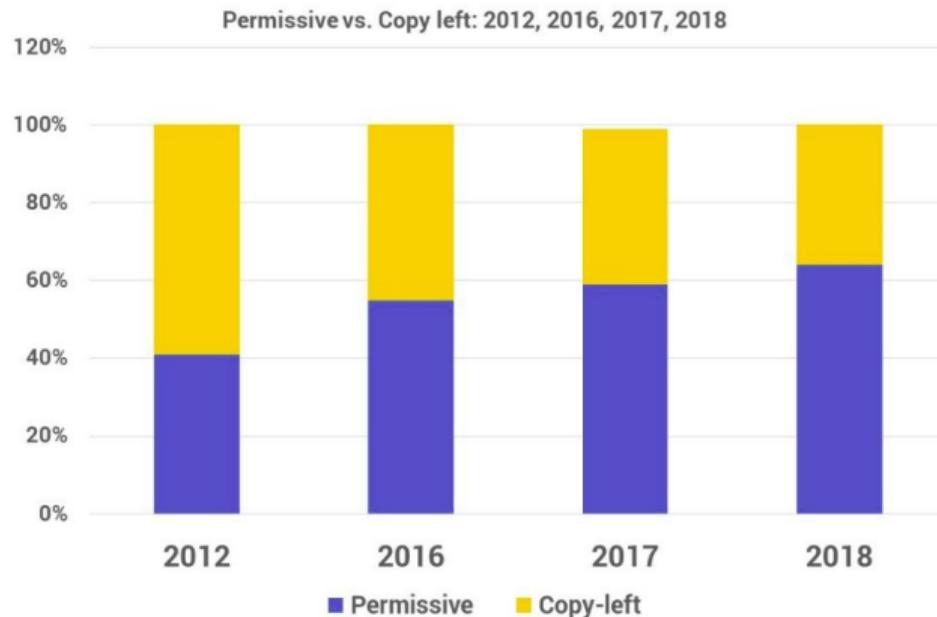
- Copy-left とは Copyright (= 著作権/著作者人格権) の否定
- OSS を利用して開発した 製品のソースコード開示要請に応える事が求められる
- 開発成果は確実に OSS コミュニティに還元され、コミュニティの成長に寄与する
- GNU GPL が代表例

■ Permissive 型ライセンス

- OSS を利用して開発した 製品のソースコード開示を義務を要求しない ライセンス
- OSS 開発コミュニティへのフィードバックが弱く「ただ乗り」を助長するリスクも
- Apache 2.0、BSD など が代表例

産業界視点では **Permissive** 型ライセンスが **OSS** の採用を後押ししている形である

オープンソースライセンスのトレンド：Permissive 型が増化中



<https://resources.whitesourcesoftware.com/blog-whitesource/top-open-source-licenses-trend>

OSS 先行成功事例としての Linux kernel 開発

kernel 開発総本山 = kernel.org には複数バージョンが置かれている

Protocol	Location
HTTP	https://www.kernel.org/pub/
GIT	https://git.kernel.org/
RSYNC	rsync://rsync.kernel.org/pub/

①



① : 現在の最新版
(差分更新反映)

② : 開発途上版

③ : 短期保守版
(最新 -2 版まで)

④ : 長期保守版
(2年+ 保守期間)

⑤ : 次期マージ候補

②	mainline: 5.5-rc5	2020-01-05	[tarball]	[patch]	[inc. patch]	[view diff]	[browse]		
③	stable: 5.4.8	2020-01-04	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
③	stable: 5.3.18 [EOL]	2019-12-18	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
④	longterm: 4.19.93	2020-01-04	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
④	longterm: 4.14.162	2020-01-04	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
④	longterm: 4.9.208	2020-01-04	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
④	longterm: 4.4.208	2020-01-04	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
⑤	linux-next: next-20200106	2020-01-06	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]

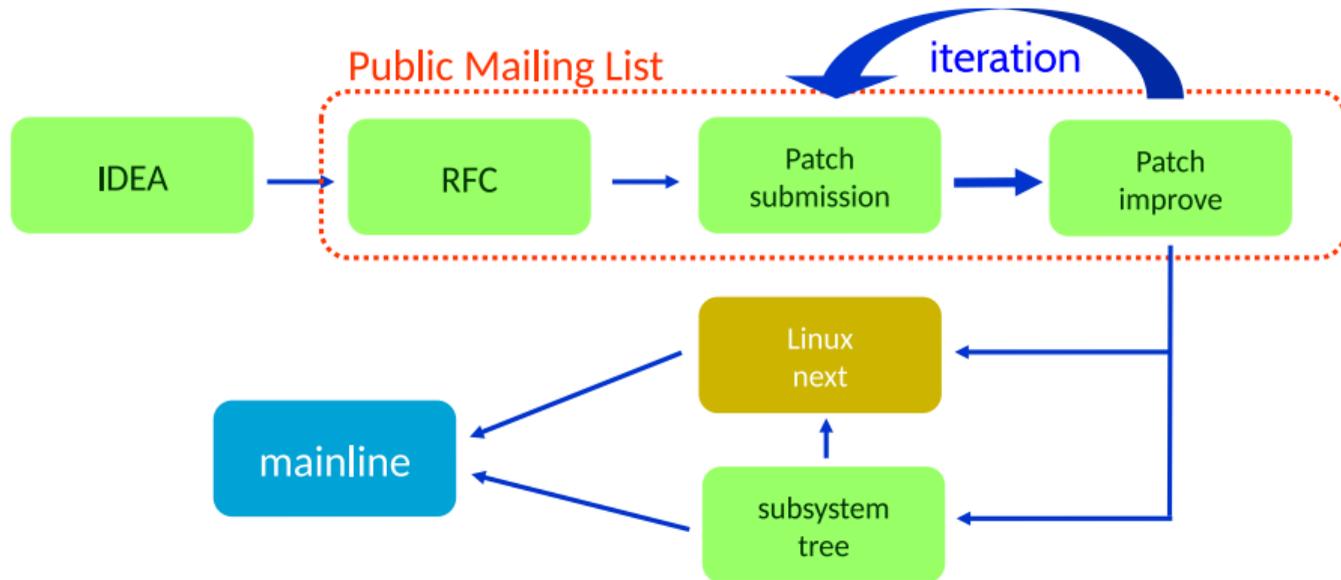
<https://www.kernel.org/>

Linux kernel 5.4 (2019-11-26 release) の開発実績

項目	集計結果
ソースコード行数総数 (コメント行を除く)	17,503,400 行
ソースコード行数差分 (5.3 → 5.4)	12行 (注、大規模な整理が行われた)
ファイル数 (ディレクトリーを除く)	63,320 個
sloccount によるソフトウェア価値の試算	\$ 770,761,765 (約 847億円)
ver. 5.4 の開発期間 (5.3 → 5.4)	70 日 (10 週間)
ver. 5.4 の開発に参加した開発者	1,802 人
はじめてパッチを投稿した開発者	266 人
加えられた変更数 (= チェンジセット数)	14,000 個
ver. 5.4 で追加された価値 (sloccount 差分)	555\$ (注、削除された行数で相殺された)

<https://lwn.net/Articles/804119/>

完全にオープンな コードレビュープロセス



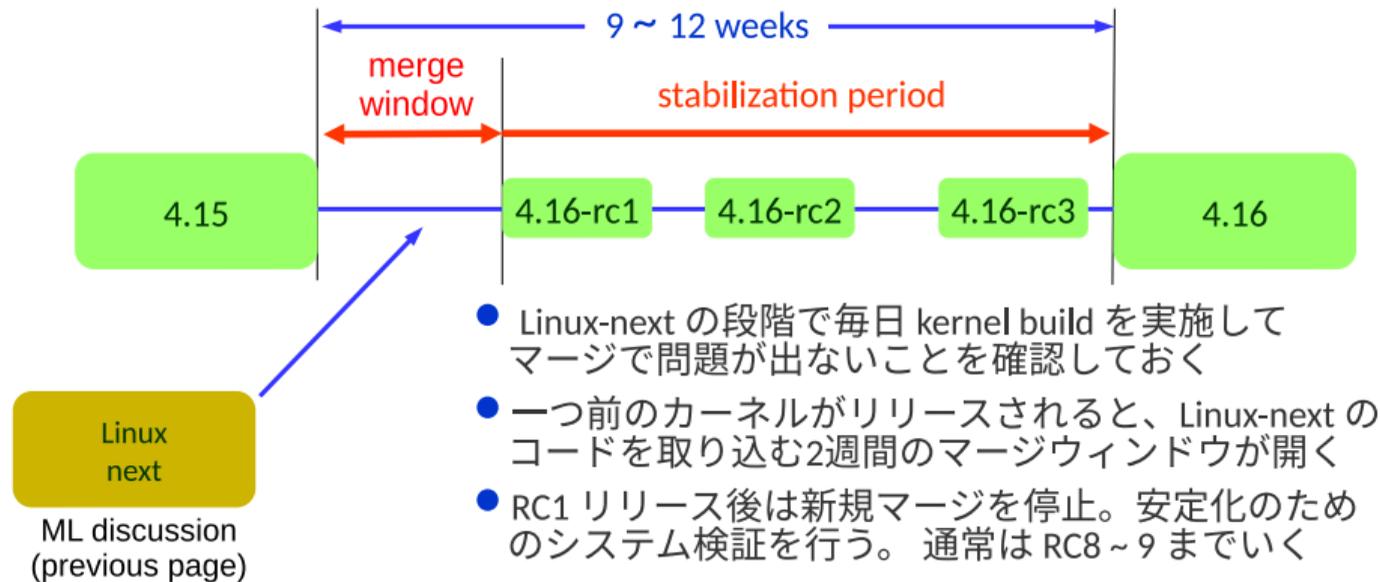
公開のML上で新提案をオープンに検討→改善を重ねる（イタレーション）、
どんな経験者でも数回の書き直しが要求される（顔パスは無い、皆で推敲して良いコードにする）

OSS 開発における コードレビュープロセス

徹底した第三者レビューの実施 が OSS のコードクオリティの裏付け

- オープンソースプロジェクトではコミュニティ開発者から投稿されたパッチを集約してプログラムを構築するので**低品質なコードが混入しないよう** 入り口での制御 (**コードレビュー**) が重要な砦となる。
 - 組織的なコードレビューの仕組み (Gerrit など) があるか
 - メンテナーはタイムリーにパッチをレビューしているか、
 - **レビュー内容に" 透明性" や" 中立性" があるか**
 - メンテナーの技術バックラウンド、スキルセットは適切か
 - 幅広く関連する別プロジェクトの開発者もレビューに参加しているか
- コミュニティ著名開発者からのコード指導もあるので **レビュー自体のレベルも高い**
- パッチレビューでは **ベテラン開発者も最低 3 回程度は書き直す** のが普通

パッチマージとコードリリースプロセス



パッチレビューに合格して Linux-next に登録されてからマージが完了するまでには**最大で22週**かかる

Upstream kernel と Linux ディストリビューション

Upstream kernel

- コミュニティ (kernel.org) で開発
- カーネル部のソースコードのみ
- 公開 ML でパッチを集約して開発
- 約 70 日周期でバージョンアップ
- 全ディストリビューションのベース
- コミュニティによる保守 (2~6年)
- セキュリティパッチを適用した LTS 版

Linux Distribution

- ディストリビュータがライブラリーやアプリもインテグレーションしたもの
- 概ね半年周期でバージョンアップ
- 商用版 (有償サポート有)
- 無償版 (コミュニティサポート)
- PC ターゲット (Ubuntu, Fedora 等)
- 組み込みターゲット (yocto 等)

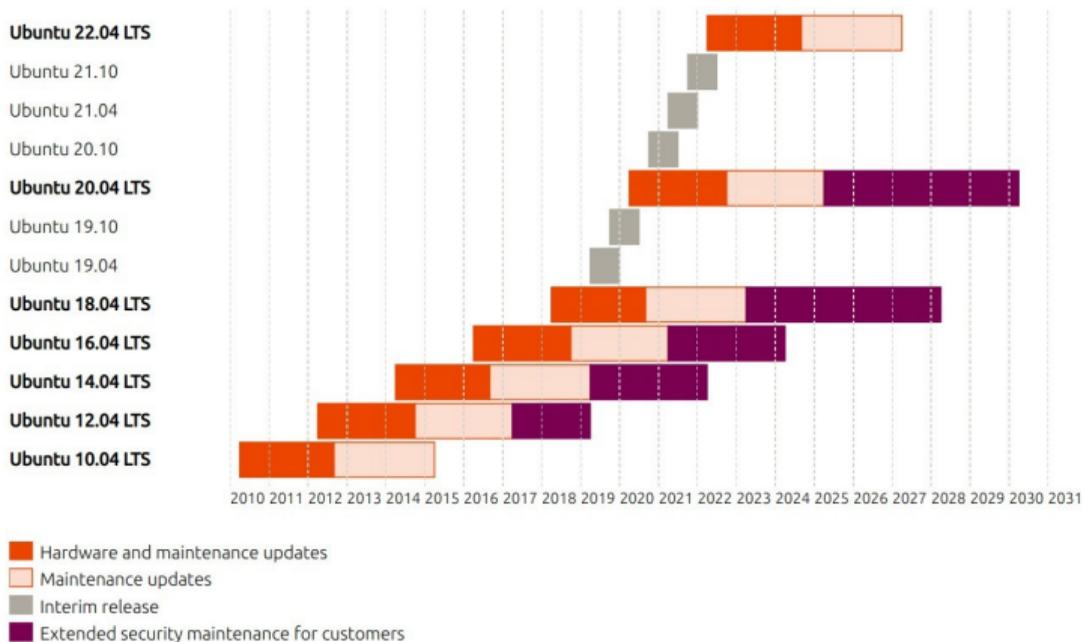
Upstream Linux kernel リリース、メンテナンスマイルストーン

Longterm release kernels

Version	Maintainer	Released	Projected EOL
5.4	Greg Kroah-Hartman & Sasha Levin	2019-11-24	Dec, 2021
4.19	Greg Kroah-Hartman & Sasha Levin	2018-10-22	Dec, 2020
4.14	Greg Kroah-Hartman & Sasha Levin	2017-11-12	Jan, 2024
4.9	Greg Kroah-Hartman & Sasha Levin	2016-12-11	Jan, 2023
4.4	Greg Kroah-Hartman & Sasha Levin	2016-01-10	Feb, 2022
3.16	Ben Hutchings	2014-08-03	Jun, 2020

<https://www.kernel.org/category/releases.html>

ディストリビューションのリリースマイルストーン (Ubuntu の例)



<https://ubuntu.com/about/release-cycle>

プロジェクト毎に異なる開発コミュニティの特性

オープンソース開発プロジェクトの活性度の評価

OSS プロジェクトのステージ（黎明期 → 成長期 → 安定期 → 衰退期）を確認する

- 多くの人々が利用している OSS の中には現在も活発に開発が継続されているものばかりではなく、**開発が終了していて保守も行われなくなっているもの**も多い。OSS を採用する時には **プロジェクトの活性度を確認** する必要がある。
- **メーリングリスト (=ML)** のトラフィック
- 何人くらいの人々が ML に参加しているか (特定の人に偏っていないか)
- メンテナーは ML への投稿に適切にレスポンスしているか
- **パッチ**の投稿数 (パッチはメッセージの一部として投稿される)
- **パッチレビュー**が適切に行われているか (レビュー期間やコメントの内容)

公開メーリングリストのトラフィック量は開発プロジェクト活性度のバロメータ

OSS プロジェクト活性度統計 (https://www.openhub.net/ による)

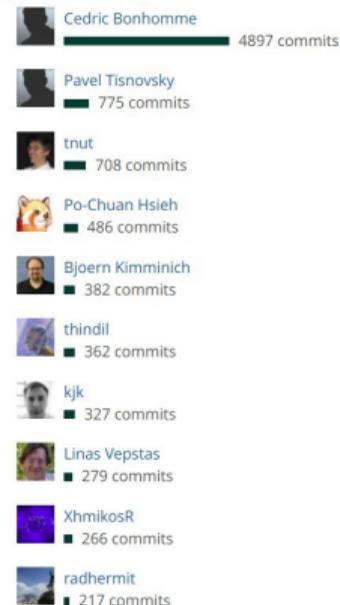
Most Popular Projects



Most Active Projects



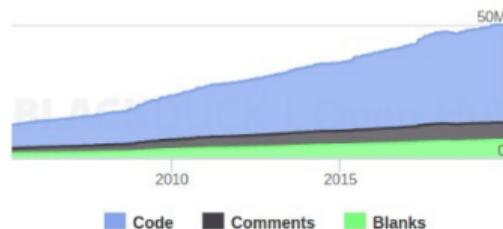
Most Active Contributors



Linux kernel = 健全安定成長型 (<https://www.openhub.net/p/linux>)

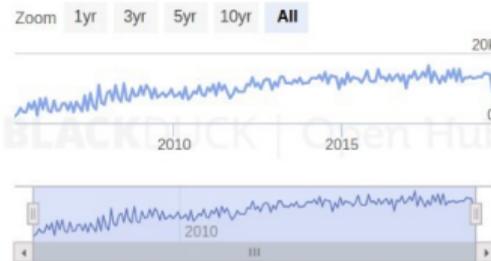
Code

Lines of Code



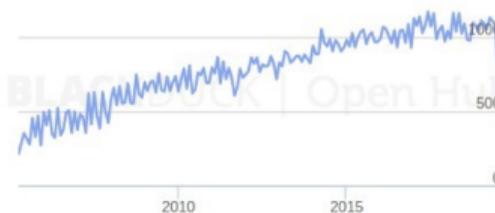
Activity

Commits per Month



Community

Contributors per Month



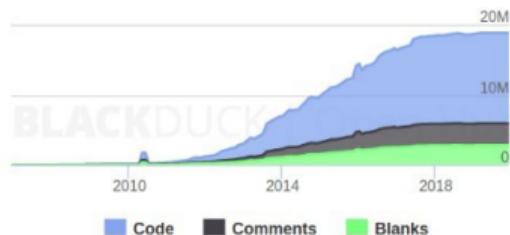
オープンソース開発コミュニティの実像
世界のトップ企業が今「共創開発」に注目
「事業戦略」と「オープンソース開発の理念」は整合できるのか

オープンソースとは、どのようなソフトウェアなのか
OSS 先行成功事例としての Linux kernel 開発
プロジェクト毎に異なる開発コミュニティの特性

OpenStack = 一過性爆発型 (<https://www.openhub.net/p/openstack>)

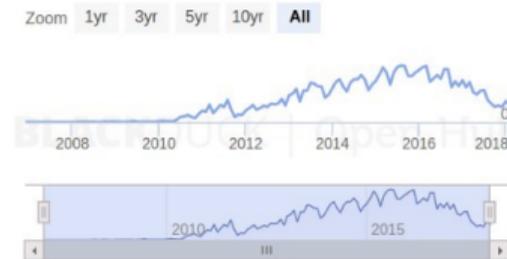
Code

Lines of Code



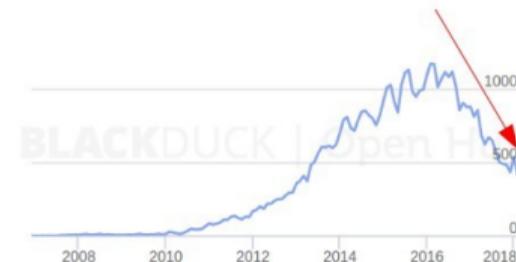
Activity

Commits per Month



Community

Contributors per Month



OSS プロジェクトの盛衰がエンジニアの生活を直撃してしまった事例

HP Enterprise, Mirantis sack couple of hundred OpenStack engineers

Still super committed, though, we're told

By Thomas Claburn in San Francisco 2 Nov 2016 at 00:06 9  SHARE ▼



Hewlett Packard Enterprise and Mirantis have laid off roughly 200 OpenStack developers in recent weeks, calling into question the appeal of OpenStack, the open source project for cloud computing infrastructure.

https://www.theregister.co.uk/2016/11/02/hpe_mirantis_sack_openstack_workers/

世界のトップ企業が今「共創開発」に注目

一例として、TOYOTA も異業種間連携（開発）を加速している

トヨタ「仲間作り」貪欲に 回顧19年 CASE対応深く

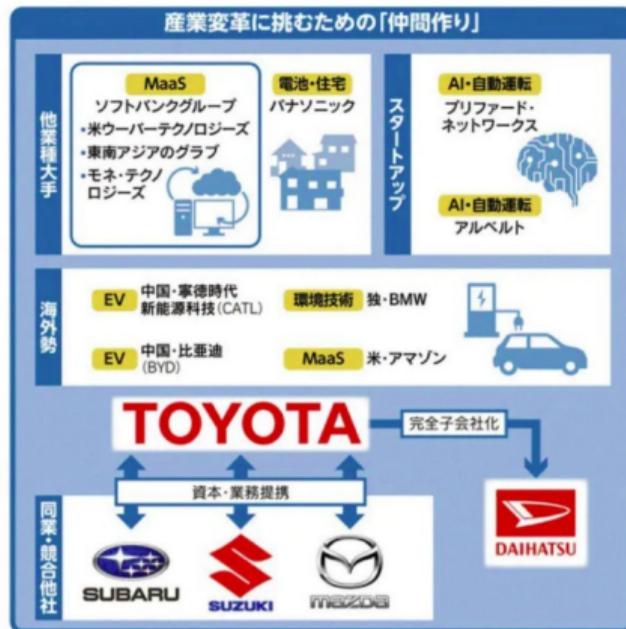
日経産業新聞 コラム (ビジネス) 自動車・機械 中部

2019/12/24 2:00 | 日本経済新聞 電子版

保存 共有 印刷 通知 電子メール その他

2019年の産業界は業界勢力図の様変わりを予感させるニュースが相次いだ。主な企業・業界の動きを振り返りつつ、残る難題を探り、20年を展望する。初回はトヨタ自動車。電動車や自動運転といった「CASE」の大波が迫るなか、「仲間作り」と称する提携戦略を19年も貪欲に進めた。

トヨタ自動車のアライアンスは様変わりする



<https://www.nikkei.com/article/DGXMZ053682220T21C19A2X11000/>

オープンソース開発プロジェクトの類型

オープンソース開発の原点 = 個人プログラマーによる草の根活動

個人ベースの小さな開発がコミュニティプロジェクトに発展していくのが自然な形

- オープンソースソフトウェアとしてプログラムを配布することが起点となる
- 開発に参加してくれる人が現れた時に、色々考える 必要ができてくる
 - コードを受け取ってレビューする仕組み（開発プロセス）
 - 次のバージョンを何時出すか、どんな機能を追加するか（ロードマップ）
 - メンバーの役割、議決の方法、収支管理（プロジェクトの体制）
- 脆弱性問題への対応が求められることがある（セキュリティパッチリリース）
- 開発者、資金、組織体の運営を考える必要がある（プロジェクトの継続性）
- openssl 脆弱性問題（=Heartbleed CVE-2014-0160）が顕在化

現在でも多くの **OSS** コンポーネントは小規模プロジェクト活動に依存している

近年 企業主導型の OSS プロジェクト が増えている

企業がコミュニティにリソース（資金、開発者等）を提供しているプロジェクト

- （当初は）**純粋な社会貢献型**の開発コミュニティ支援
 - **金は出すが口は出さないパトロン型**の支援活動
 - 開発者を急激に増やすことは考えず制御可能な規模を維持する
 - 短期での成果刈り取りよりも活動の継続性を重視
- （近年は）**企業戦略（＝収支目論見）に基づいた戦略的な取り組み**
 - トップタレントをもったエンジニアを効率的に集めることができる
 - 非競争領域の共通技術基盤をオープンソース化する事で開発投資を効率化
 - プロジェクト開発成果は適宜自社製品にフィードバックしていく（製品リンク）
 - **企業戦略との整合のためにコミュニティに影響力を行行使するケースも**

多くの企業がイノベーション実現には共創（コラボレーション）が必須と考え始めた

「赤い Chrome」と「青い Chromium」の違いをご存じですか？



「Chrome」(製品) は「Chromium」(OSS) がベースになっている

 Chromium

バージョン: 79.0.3945.79 (Official Build) Built on Ubuntu, running on Ubuntu 18.04 (64 ビット)

[Chromium のヘルプを見る](#) 

Chromium
Copyright 2020 The Chromium Authors. All rights reserved.

Chromium は [Chromium](#) オープンソース プロジェクトやその他の [オープンソース ソフトウェア](#) によって実現しました。

 Google Chrome

バージョン: 79.0.3945.117 (Official Build) (64 ビット)

[Chrome のヘルプを見る](#) 

[問題を報告](#) 

Google Chrome
Copyright 2020 Google LLC. All rights reserved.

Google Chrome はオープンソース プロジェクト [Chromium](#) やその他の [オープンソース ソフトウェア](#) によって実現しました。

[Google Chrome 利用規約](#)

現実には「Chromium」と「Chrome」は同時並行開発されている

Chromium (青いロゴマーク)

- Google ホストの OSS プロジェクト
- html5 の事実上のレファレンス実装
- Chromium 版 Microsoft Edge 公開
- オープンソース (Apache2)
- OSS 版で提供されるコア機能
 - Google V8 JavaScript engine
 - Media Codec (VP9 等)
 - サンドボックス (セキュリティ)
 - Service Worker (代理応答)

Chrome (赤いロゴマーク)

- Google ブランド
- Windows, Linux (Android), iOS
- OSS 版と同バージョン (遅延なし)
- クローズドソース (=ソース非公開)
- Chromium + 独自拡張
- 拡張される機能
 - Flash Player
 - 自動アップデート機構
 - ユーザートラッキング機構

Android オープンソース版 AOSP は Chrome とは考え方が違う

AOSP = Android Open Source Project、Google がホストする OSS 開発プロジェクト

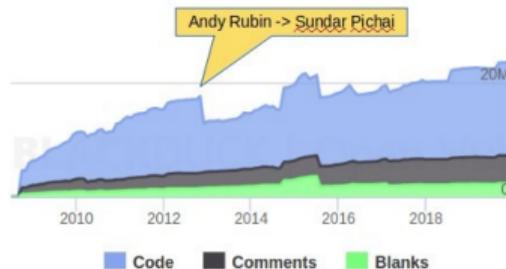
- Google 社内開発（非公開）と AOSP コミュニティ開発（公開）が別々に存在
- 社内開発版にも Apache2 ライセンスが適用される OSS コードがある
- Google 開発のコードに OSS でないコードが含まれるかは知るすべはない
- Google は社内開発版が完成した時点で OSS 部を AOSP にコードドロップ
- AOSP ではドロップされたコードに合わせるための後追い変更を実施

- ハンドセットベンダーは製品発売時に OSS 利用部分のソースコードを公開
- AOSP のコードを使ってスマートフォンを構築する別のプロジェクトもある
 - CyanogenMod (2016 年末に終了)
 - LineageOS (<https://lineageos.org/>)

AOSP (= Android Open Source Project) 活動実績

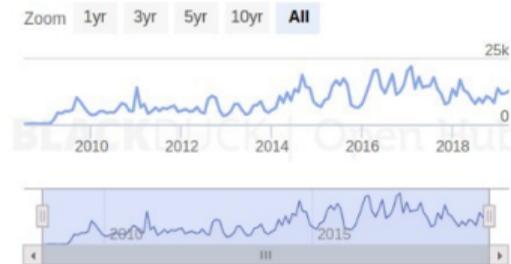
Code

Lines of Code



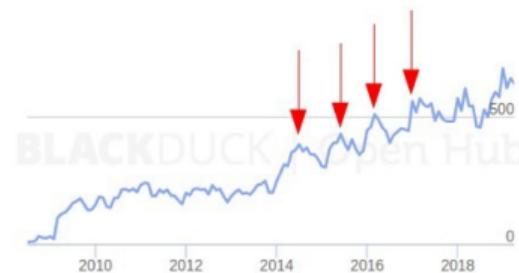
Activity

Commits per Month



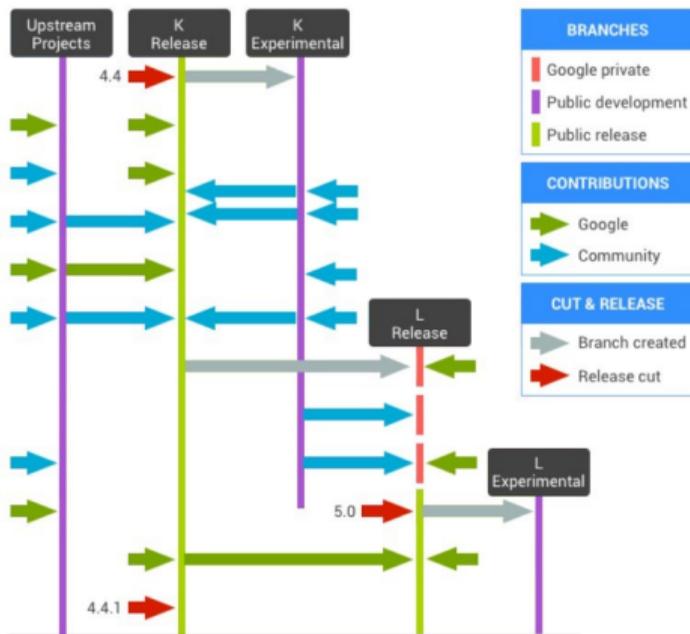
Community

Contributors per Month



<https://www.openhub.net/p/android>

Google は社内開発成果を周期的に AOSP にドロップ



Android プラットフォームには、どの時点でも常に最新のリリースが存在しており、通常は、ツリー内のブランチの形式をとっています。

デバイスのビルドやコントリビューションを行う場合、現在の最新リリースで作業を行います。バグの修正、新しいデバイスの立ち上げ、新機能の試験運用などもこのリリースで行います。

同時に Google では、製品のニーズや目標に合わせて Android プラットフォームとフレームワークの次期バージョンを開発しています。次期バージョンの開発にあたっては、デバイスパートナーと協力し、Android の今後の方向性をふまえたうえで、その仕様を選定するフラグシップ デバイスに取り組みます。

現行バージョンの次のバージョンの準備が完了すると、公開ソースツリーに公開されて、最新のリリースとなります。

<https://source.android.com/setup/start/codelines>

企業がリーディングする OSS 開発プロジェクトで懸念されるポイント

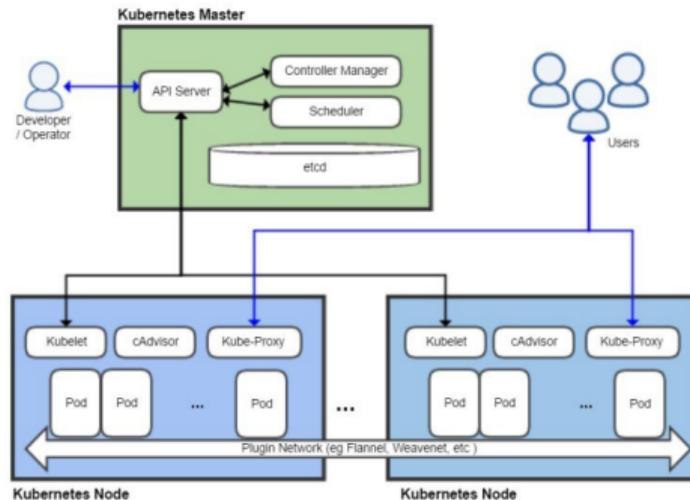
草の根型のコミュニティとは オペレーションモードが異なる ケースが散見される

- オープンソースの中には **特定企業に実効的に支配** されているものもある
- 企業主導プロジェクトで懸念されるコミュニティ運営リスクの例
 - パッチを投稿してもメンテナーにレビューしてもらえない
 - OSS 開発と並行して **企業内でクローズド開発が平行に行われているケース**
 - 障害発生時に原因究明、対策パッチの開発が円滑に行われぬ
 - 企業の事業方針の転換に伴い、**セキュリティパッチの提供が滞る**
 - **企業の事業方針転換により開発資金支援が突然ストップ** し開発が頓挫する
 - 企業の意向により **ライセンス条件が変更** されオープンに利用できなくなる
- プロジェクトのメンテナーが **技術的に中立、公平である事** が極めて重要

Google は **AOSP** 成果を刈り取っていない。何のために **OSS** 化しているのだろうか？

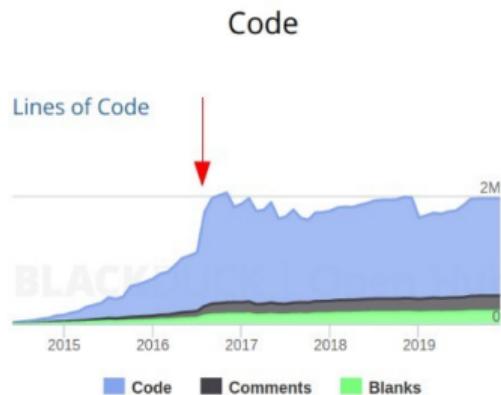
企業主導のオープンソース開発における中立性の確保

Kubernetes (=K8S) = container orchestration OSS project



<https://ja.wikipedia.org/wiki/Kubernetes>

Kubernetes 活動実績 (<https://www.openhub.net/p/kubernetes>)



kubernetes 開発コミュニティは誰がリーディングしている？

2016年3月にプロジェクトを Linux Foundation に移管し中立化

- 草の根型プロジェクトとは全く性質の異なる **企業主導型コミュニティ** が起点
 - 元々は Google がホストする OSS プロジェクトとしてスタート
- OSS であっても **特定企業に帰属するコードだと使いにくいと感じる企業** は多い
 - プロジェクトの **将来性への不安**（開発が続いていくか心配）
 - プロジェクトの **ガバナンスへの懸念**（オーナー企業の意向に強く左右される）
 - プロジェクトへの **貢献成果がすべてオーナー企業にもっていかれないか**
- 以降、開発者数、ユーザー数が爆発的に伸びてプロジェクトが急成長した
- **コンテナオーケストレーションのデファクト**として幅広く IT 業界に浸透した

企業に帰属する **OSS** プロジェクトを中立化させることで大きく飛躍できた成功例

K8S は今 CNCF (Could Native Computer Foundation) 傘下に

KubeCon + CloudNativeCon Amsterdam | March 30 – April 2 | Best Pricing Ends February 2 | [Learn more](#)

 **CLOUD NATIVE**
COMPUTING FOUNDATION

About ▾ Projects ▾ Certification ▾ People ▾ Community ▾ Newsroom ▾ [JOIN NOW](#) 

CNCF serves as the vendor-neutral home for many of the fastest-growing open source [projects](#), including Kubernetes, Prometheus, and Envoy.

			
76,899	99,485	94	144,258
# of contributors to CNCF projects	Registered for free Kubernetes EdX course	Certified Kubernetes Distributions and Platforms	CNCF Meetup members

Linux Foundation = プロジェクトの中立性を担保する仕組み

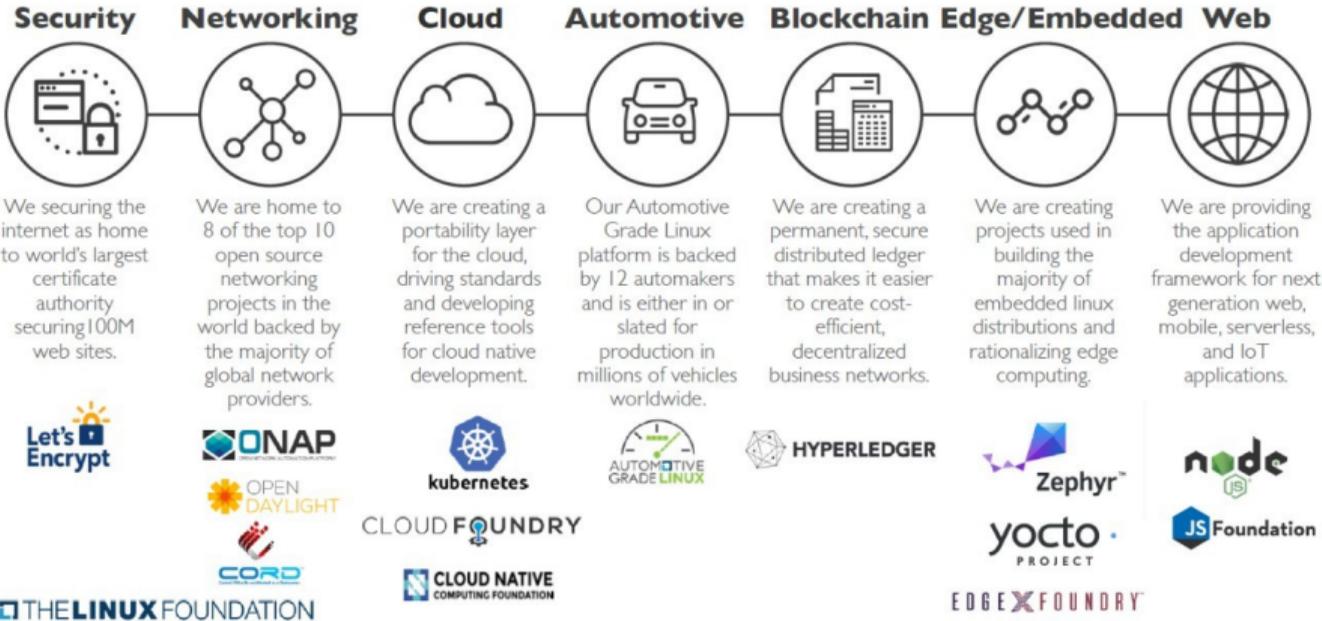
OSS 開発プロジェクトの多くは Linux Foundation (=LF) 傘下で活動

- Linux kernel 開発コミュニティを支援するための産業コンソーシアムとして発足
- **世界の大手 SW テクノロジー企業** が Foundation の運営資金を会費の形で負担
 - 発足当初は Linux kernel 創始者の Linus Torvalds 氏に給料を払うためのスキーム
 - 有料企業会員 = 17 (プラチナ) + 15 (ゴールド) + 891 (Silver) = **923 社**
 - 2019 年終了時点で LF 傘下で **228 のプロジェクト** が活動中
 - 多くの”草の根型の開発プロジェクト”も LF の傘下 (年会費なしのアソシエート)
 - LF 年会費は LF の運営費用なので、各プロジェクトは独自で費用を捻出して運営中
- 企業主導の OSS プロジェクトの多くが **中立化担保のために所有者を LF に変更**
- 近年は企業連合が **最初から LF 傘下で新プロジェクトを立ち上げるケース** が多い

FaaS (Foundation as a Service) を目指した産業コンソーシアム活動

Linux Foundation のカバー領域 は多岐に広がっている

Today the Linux Foundation is Much More than Linux...



LF アンブレラプロジェクト という建付け

類似の (場合によっては競争関係にある) プロジェクトをまとめて一つのグループに

- プロジェクトをグループに集める狙い = 共通のゴールを持った活動の緩い連携
 - 別々の活動 (=方式) を統合して **一つの開発に束ねるという意図ではない**
 - **重複する活動 (開発、運営) が見つかったら 共有することで効率化**を図る
 - 業界全体で共通化した方が良く **非競争領域の仕様 (API 等) の統一** を行っていく
- グループ形成によって期待される効果
 - 類似の複数のプロジェクトに参加しなくて済むので、**年会費等の費用抑制**
 - 業界関係者 (サプライヤー、クリエイター、サポーター等) が **情報交換機会の創出**
 - 異業種間 (IT ベンダー、通信キャリア、サービス等) の **イノベーション創成**

オープンなコラボレーションを活性化させる為のプロジェクト間連携を促進する仕組み

オープンソース開発コミュニティの実像

世界のトップ企業が今「共創開発」に注目

「事業戦略」と「オープンソース開発の理念」は整合できるのか

オープンソース開発プロジェクトの類型

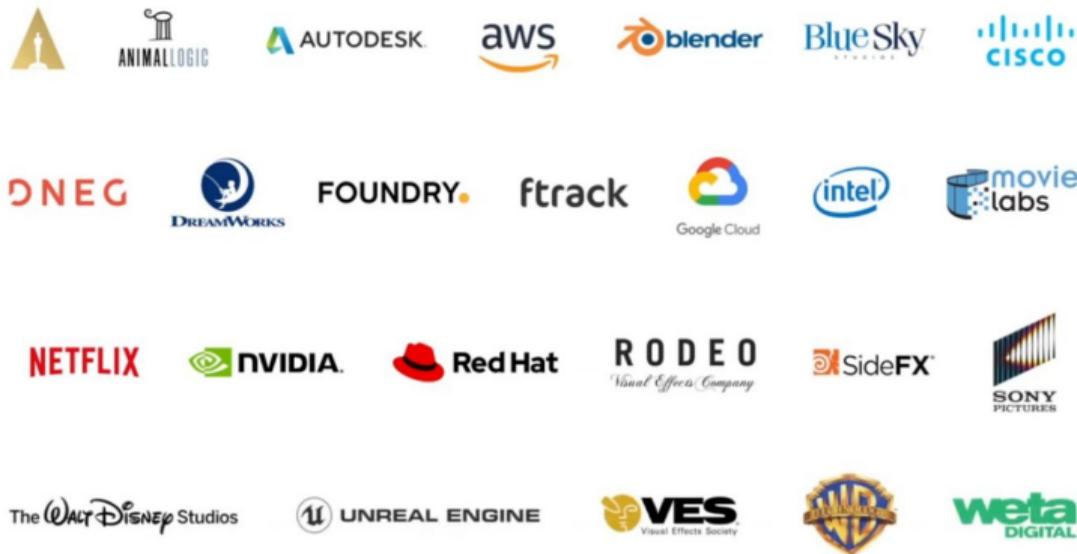
企業主導のオープンソース開発における中立性の確保

LF Academy SW Foundation (アンブレラの事例)



LF Academy SW Foundation (アンブレラの事例)

ASWF IS SUPPORTED BY



<https://www.aswf.io/>

「事業戦略」と「オープンソース開発の理念」 は整合できるのか

共創開発の基盤としてのオープンソースプラットフォーム

企業の研究開発は 自社の競争優位性を獲得を目指した活動 である

研究開発 (R&D = Research & Development) とは新しい競争優位性獲得 の取り組み

■ アカデミア (学术界)

- **基礎技術領域** では純粋な技術探求 (= 何に役立つかが明確でない活動) も重視する
- 研究成果は論文などを通じて公開され、成果は共有される
- **産学連携活動** として産業界の意向を反映させた開発もある (増えている)

■ インドustry (産業界 = 企業)

- 企業の **競争優位性の源泉** としての **差異化技術** の獲得が動機
- 明確な開発ロードマップと比較的短期間での開発成果の取り組みが期待される
- 研究開発成果は **特許化して公開** するケースを除き **原則として非公開**

各企業は自社 **R&D** は競争差異化の源泉だと考えるので成果は「非公開」が常識

では何故企業が 共創開発 (=オープンな R&D 活動) に取り組むのか

多くの先端企業が「make or buy」から「make or buy or collaborate」に転換

- make 戦略 = 差異化技術については自社内でクローズに技術開発
- buy 戦略 = 非差異化領域の技術、時間が求められる技術は社外から技術導入
- collaborate 戦略 = 他社と連携して差異化技術のベースを共同開発
 - 開発規模が大きく自社だけでは必要なリソース（資金、開発者等）を集められない
 - オープンな活動とすることで、技術の中立性担保や開発スピードアップを図る
 - 企業間の連携による マジョリティ確保でデファクトを獲得する
 - 中長期的にメンテナンス可能な サステナブルなソリューションの構築
 - 企業同士で 似て非なる技術を重複開発するリスクを回避する

技術進化の速度、規模の拡大、グローバル化等のトレンドを反映した技術の新潮流

技術開発における 標準化競争のトレンド

「デジュールスタンダード」から「デファクトスタンダード」の流れが加速

■ デジュール (dejure = 原則上の) スタンダード

- 公的な **技術標準化団体** であらかじめ決められた手順を踏んで方式を策定
- 各企業の思惑などにより合意形成にはとても時間がかかる
- ISO/IEC、IEEE、JIS などの標準化団体がある

■ デファクト (de facto = 事実上の) スタンダード

- ユーザーの選択によって結果的に主流となったもの
- 公式な技術標準化よりも早くトレンドが決まることも多い (流行り廃りは激しい)
- 企業はマーケティング活動、プロモーション活動を通じてデファクトを取りに行く

ハードウェアではデジュール標準が主流、ソフトウェアではデファクト標準が拡大中

html5 spec : W3C (デジュール) と WHATWG (デファクト)

👤 HTML Living Standard

トップ > HTMLリファレンス > HTML概要 > HTML Living Standard

HTML Living Standardの概要

HTML5 や HTML 5.1 は W3C が標準化を進めていますが、W3C とは別に、Apple, Chrome, Opera が設立した **WHATWG** という団体が独自に策定を進めている HTML 仕様です。HTML Living Standard と呼ばれ、バージョン番号や、何年何月何日版という概念もなく、毎日改版、修正、強化が進められています。

- HTML Living Standard: <https://html.spec.whatwg.org/> (2011-10)

現時点では、W3C が勧告する HTML5/HTML 5.1 との差分は少ないですが、今後はどのような方向になるかわからない状況です。

Microsoft はまだ、W3C を標準としていますが、Chrome, Firefox, Opera などのブラウザでは、W3C よりも、HTML Living Standard の方を標準として実装していくようです。

HTML Living Standard の歴史

W3C は、1996年から 2001年にかけて、HTML, CSS, XML, XHTML に関する様々な仕様を勧告してきました。

- CSS 1.0: <https://www.w3.org/TR/REC-CSS1-961217/> (1996-12-17)
- HTML 3.2: <https://www.w3.org/TR/REC-html32.html> (1997-01-14)
- HTML 4.0: <https://www.w3.org/TR/REC-html40-971218/> (1997-12-18)
- XML 1.0: <https://www.w3.org/TR/1998/REC-xml-19980210/> (1998-02-10)
- HTML 4.01: <https://www.w3.org/TR/1999/REC-html401-19991224/> (1999-12-24)
- XHTML 1.0: <https://www.w3.org/TR/2000/REC-xhtml1-20000126/> (2000-01-26)
- XHTML 1.1: <https://www.w3.org/TR/2001/REC-xhtml11-20010531/> (2001-05-31)

しかし、無理に XML/XHTML への移行を進めようとしたりなど、Web 開発者のニーズを軽視する W3C の方針に不満もありました。2004年2月の W3C ワークショップで、Opera と Mozilla が Web 開発者のニーズを取り込んだ HTML の開発再開を提案しましたが否決され、その後、Apple, Opera, Mozilla の 3 社が、2004年6月に WHATWG(Web Hypertext Application Technology Working Group) を設立しました。

<http://www.tohoho-web.com/html/memo/html1s.htm>

CLA (=Contributor's License Agreement)

TensorFlow Project にコードを投稿するには Google との契約 が必要

Announcing the TensorFlow Dev Summit 2020 [Learn more](#)

TensorFlow > コミュニティ > 貢献 ☆☆☆☆☆

Contribute to the TensorFlow code

Whether you are adding a loss function, improving test coverage, or writing an RFC for a major design change, this portion of the contributor guide will help you get started. Thank you for work and interest in improving TensorFlow.

Before you get started

Before you contribute source code to a TensorFlow project, please review the `CONTRIBUTING.md` file in the GitHub repo of the project. (For example, see the [CONTRIBUTING.md file for the core TensorFlow repo](#).) All code contributors are required to sign a [Contributor License Agreement \(CLA\)](#).

To avoid duplicating work, please review [current RFCs](#) and contact the developers on the TensorFlow forums before you start work on a non-trivial feature. We are somewhat selective when deciding to add new functionality, and the best way to contribute and help the project is to work on known issues.

<https://www.tensorflow.org/community/contribute/code>

CLA (Contributor License Agreements) は法的に有効な契約書である

How to become a contributor and submit your own code

Contributor License Agreements

We'd love to accept your patches! Before we can take them, we have to jump a couple of legal hurdles.

Please fill out either the individual or corporate Contributor License Agreement (CLA).

- If you are an individual writing original source code and you're sure you own the intellectual property, then you'll need to sign an [individual CLA](#).
- If you work for a company that wants to allow you to contribute your work, then you'll need to sign a [corporate CLA](#).

Follow either of the two links above to access the appropriate CLA and instructions for how to sign and return it. Once we receive it, we'll be able to accept your pull requests.

NOTE: Only original source code from you and other people that have signed the CLA can be accepted into the main repository.

<https://github.com/tensorflow/tensorflow/blob/master/CONTRIBUTING.md>

TensorFlow CLA の Grant (= 許諾条項) に書かれていること

CLA の中の Grant of Copyright License, Grant of Patent License 条項

- Grant of Copyright License. Subject to the terms and conditions of this Agreement, **You hereby grant to Google** and to recipients of software distributed by Google a **perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license** to reproduce, prepare derivative works of, publicly display, publicly perform, sublicense, and distribute Your Contributions and such derivative works.
- Grant of Patent License. Subject to the terms and conditions of this Agreement, **You hereby grant to Google** and to recipients of software distributed by Google a **perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license** to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by You that are necessarily infringed by Your Contribution(s) alone or by combination of Your Contribution(s) with the Work to which such Contribution(s) was submitted. If any entity institutes patent litigation against You or any other entity (including a cross-claim or counterclaim in a lawsuit) alleging that your Contribution, or the Work to which you have contributed, constitutes direct or contributory patent infringement, then any patent licenses granted to that entity under this Agreement for that Contribution or Work shall terminate as of the date such litigation is filed.

TensorFlow CLA の Grant (= 許諾条項) に書かれていること

”みんなの自動翻訳 (by NICT)” を使って簡易的に日本語化（出力結果のまま）

■ 著作権ライセンスの付与

本契約の条件に従うことを条件として、お客様は、Google に対して、恒久的、世界的、非排他的、無報酬、ロイヤリティフリー、取消不可能な著作権ライセンスを複製、準備、公表すること、サブライセンスおよび配布物を頒布すること、および拠出金などの派生物を頒布することを Google に許諾するとともに、ソフトウェアの受領者に対して、お客様に対して譲渡することを許諾します。

■ 特許ライセンスの付与

本契約の条件に従うことを条件として、本書は Google に対して、Google が提供する特許ライセンスに該当することを条件として、お客様が本契約の対象とする特許クレームのみを対象とすることを条件として、Google が配布するソフトウェアの受領者に対して、恒久的、世界的、非排他的、無報酬、ロイヤリティフリー、取消不能、取消不能 (本条に記載する場合を除く)、特許ライセンスを付与することを許諾するものとします。

お客様がお客様のお客様または他の団体 (訴訟におけるクロスクレームまたはカウンタークレームを含む) に対して、お客様の貢献または貢献した作業が直接的または寄与的な特許侵害を構成することを主張する特許訴訟を提起した場合は、この契約の下で当該団体に与えられた特許ライセンスは、当該訴訟が提起された日から終了するものとします。

OSS プロジェクト開発への参加判断には総合的な判断力が求められる

Open Source Program Office (=OSPO) という専門職制が注目されている

- (技術面) ソフトウェアのテクノロジートレンドの理解と開発マネジメント
 - ソフトウェア技術の「流行り廃り」を理解し、適切な技術を選定する力
 - 開発リソースのマネジメント (外の世界で戦えるエンジニアの確保)
 - オープンなソリューションから 自社競争優位性を導き出せる技術的な洞察力
- (法務面) オープンソースに関わる法務、知財に対する理解と当事者能力
 - 法律の専門家としてオープンソースに関わる OSS ライセンス理解と適切な運用
 - 企業の 知財戦略と整合した技術コントリビューションのバランス
 - 知財権オーナー (ライセンサー) との 契約交渉、契約実務の遂行能力

エンジニアリング部門だけでは判断が難しい種類の問題に遭遇する場面が増えている

本日のまとめ

色々な場面で オープンソース開発プロジェクト との関わりができる

- オープンソースソフトウェア (=OSS) を利用するとき
 - ソフトウェアライセンス を確認し何が許諾、要請されているかを確認する
 - ソースコードの取り扱い、ライセンス文章のバンドル 等には十分は注意が必要である
- 自分で開発したコードをオープンソースとして公開しようと考えたとき
 - OSI が規定する OSS ライセンスのどれかを付与する
 - ユーザーの サポートや将来的な保守（脆弱性対策）も想定しておくべき
- 企業が主導する OSS 開発プロジェクトに参加するとき
 - 事前に プロジェクトの運営スタイルや活動の活性度 を把握する
 - CLA 締結を求められた場合、法務的視点 で契約内容を確認する

企業の事業戦略としての「共創開発プロジェクト」に対する的確な審美眼が必要です