

ソニーでのLinux商品導入

Introduction of Linux to the Sony products

ソニーでのOSS適用とOSSコミュニティとの関わり合い

OSS Application Sony and the relationship with the OSS community

R&D Center 主幹技師 佐藤和美

Kazumi SATO, Distinguished Engineer

Agenda

- 自己紹介 Self introduction
- OSSとのかかわりあい Relationship with the OSS
- ソニーでのLinux導入の歴史 History of introduction of Linux in Sony
- OSSの選び方、OSSを使った開発、フィードバック How to choose OSS, development using OSS, Feedback
- まとめ Summary

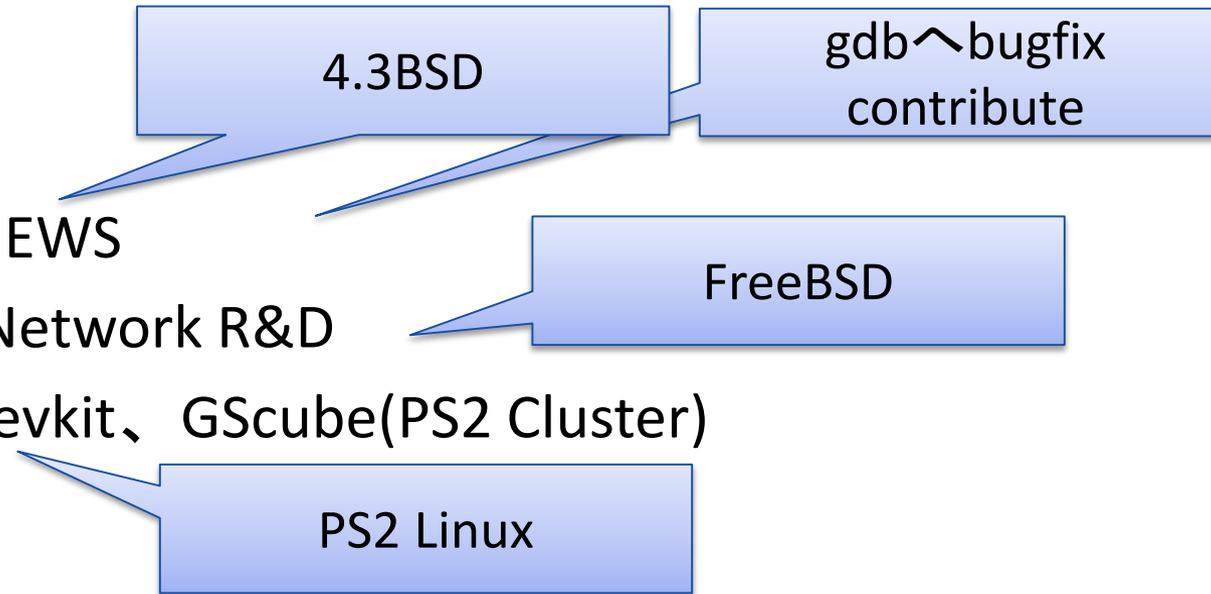
自己紹介 self introduction

- 1989年入社
- ~1996年 WorkStation NEWS
- ~1999年 PC Server、Network R&D
- ~2002年 PlayStation2 devkit、GScube (PS2 Cluster)

- 2002年~現在 Sony機器向けのLinux開発
- 2009年~現在 Sony OSS License委員会 副委員長 co-chair
- 2015年~現在 Sony OSS Board 技術担当

OSS とのかかわり合い

- 1989年入社
- ~1996年 WorkStation NEWS
- ~1999年 (PC Server、)Network R&D
- ~2002年 Playstation2 devkit、GScube(PS2 Cluster)
- 2002年~現在 Sony 機器向けのLinux開発
- 2009年~現在 Sony OSS License委員会 副委員長 co-chair
- 2015年~現在 Sony OSS Board Tech 担当

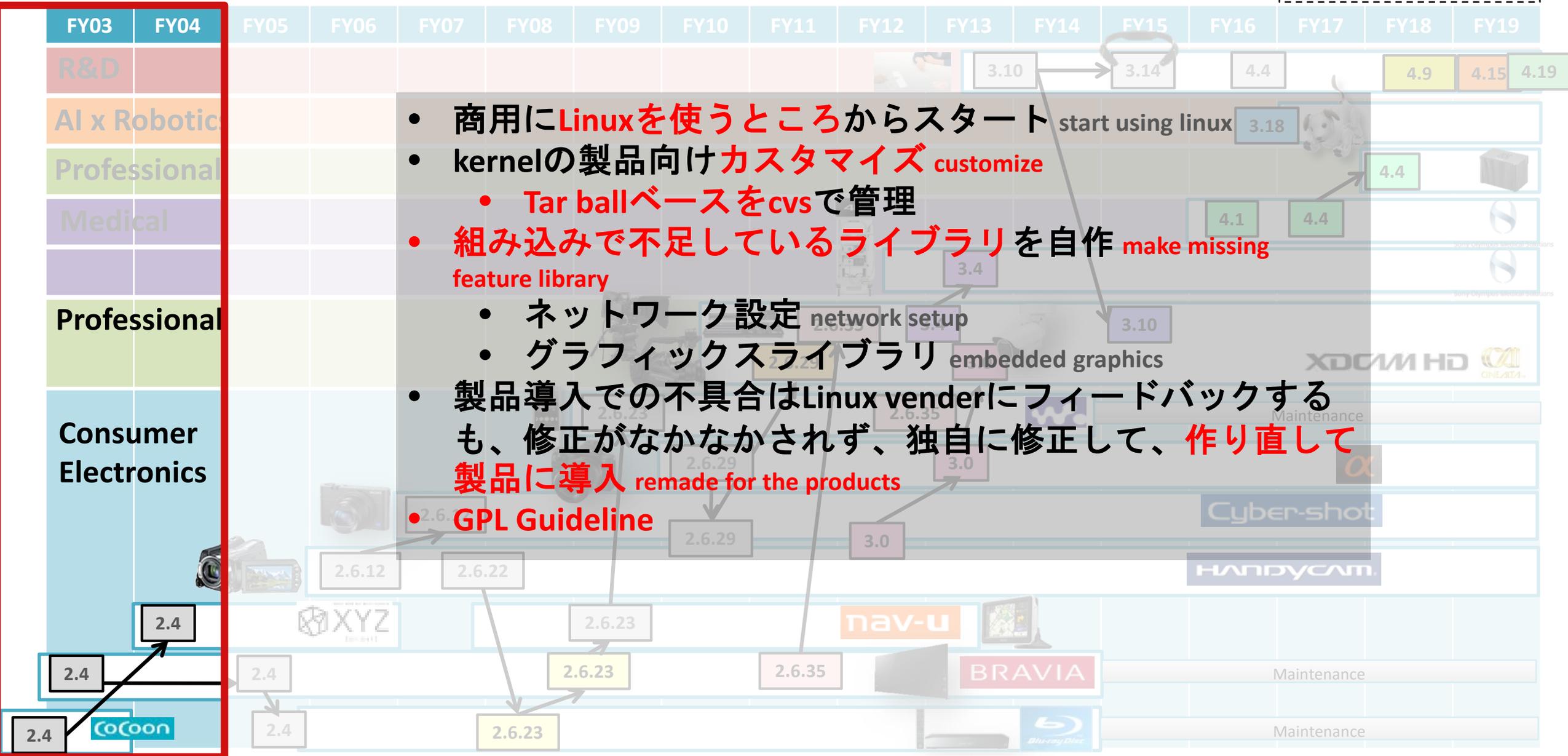
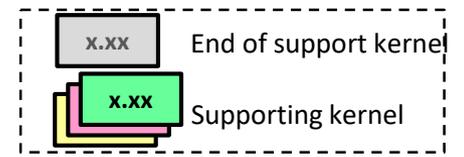


ソニー機器用のLinux関連技術を開発・導入する傍ら、その知見を全社向けのOSS活用・OSSコンプライアンスに展開

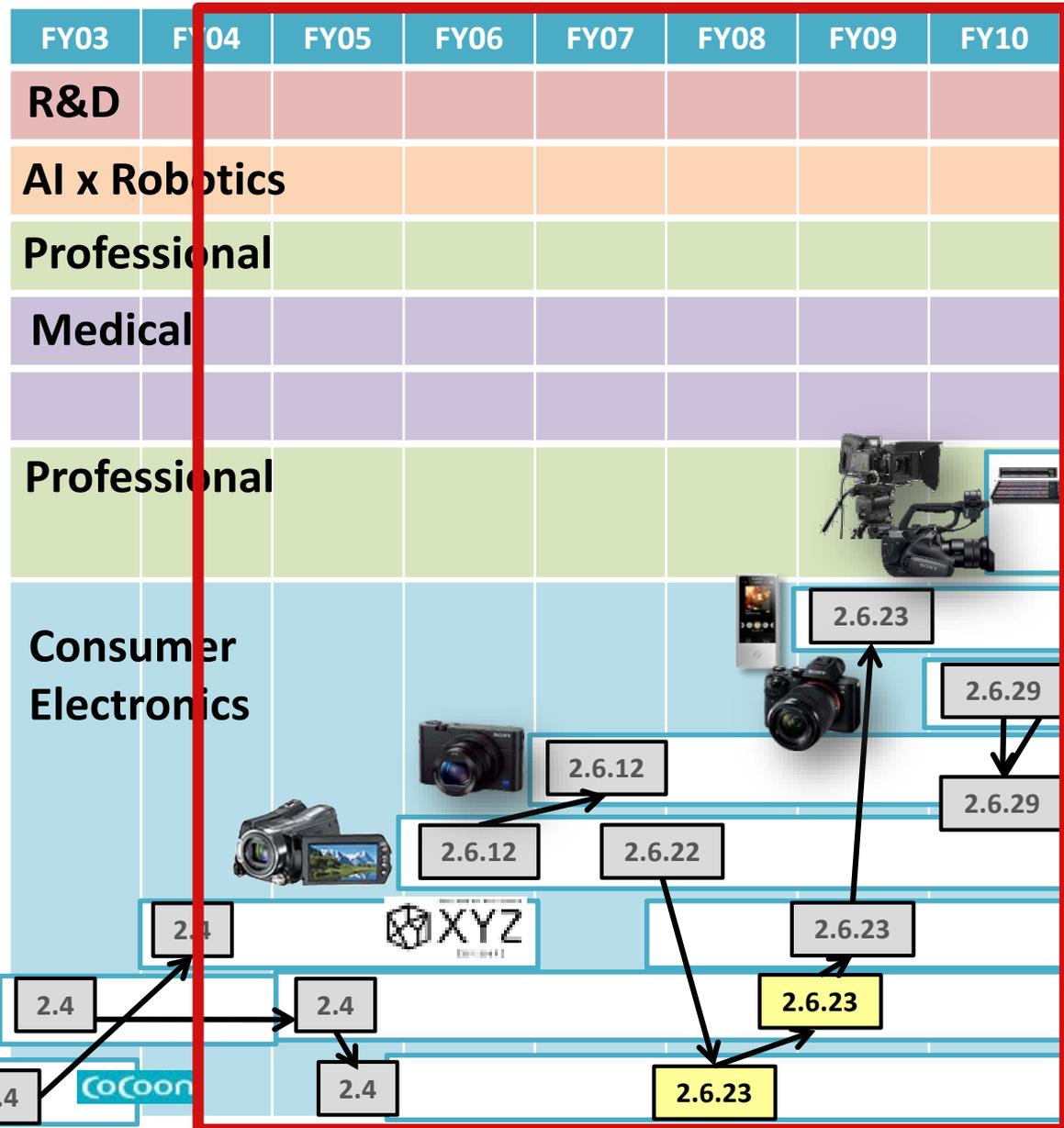
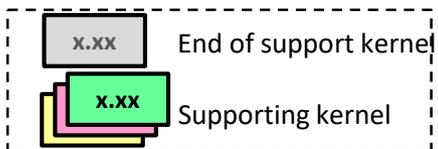
Agenda

- 自己紹介 Self introduction
- OSSとのかかわりあい Relationship with the OSS
- **ソニーでのLinux導入の歴史** History of introduction of Linux in Sony
- OSSの選び方、OSSを使った開発、フィードバック How to choose OSS, development using OSS, Feedback
- まとめ Summary

Stage 1: Commercial Linux Distribution



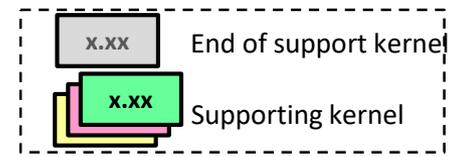
Stage 2: Original Small Distribution + Kernel



FY11	FY12	FY13	FY14	FY15	FY16	FY17	FY18	FY19
			3.10	3.14	4.4		4.9	4.15 4.19

- kernel開発を **stable+patchset** で開発
 - CVS → 2.6.12から **git** へ
- 他社SoC BSPの自社kernelへの取り込み
- rpmベースの **必要最小限のユーザーランド** **minimal required userland**
- OSSコンプライアンス、脆弱性対応、長期サポート
- 組み込み向け機能の開発
 - 省メモリ (Linux-Tiny), リアルタイム(Preempt-RT)**
 - CE Linux Forum** や **Tim, Frank** と連携し必要機能の開発
 - 耐電源断や組込向け改良** と **adhoc fix**
 - mainlineの試みと挫折と独自メンテの道
 ^ try to mainline but maintenance self

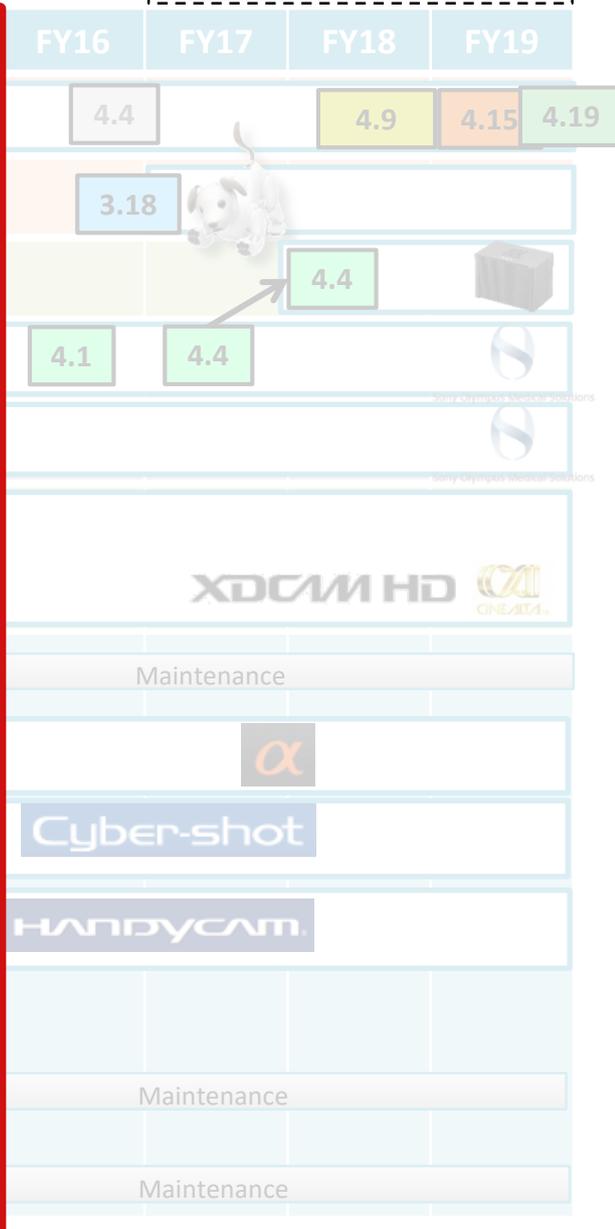
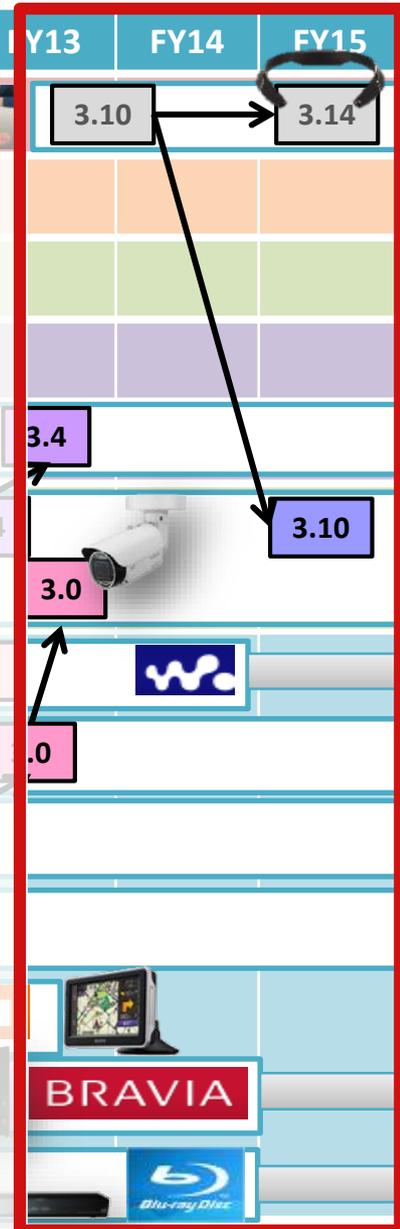
Stage 4: Ubuntu base small Distribution + kernel



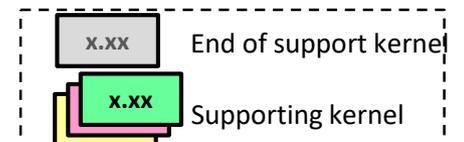
	FY03	FY04	FY05	FY06	FY07	FY08	FY09	FY10	FY11	FY12	FY13	FY14	FY15	FY16	FY17	FY18	FY19	
R&D											3.10	3.14		4.4		4.9	4.15	4.19

AI x Robotics

- AndroidやSoC BSPの充実の時代
- 従来の積み上げベースのsmall distributionも継続して開発提供
- R&D向けにはdeveloperが使いやすいubuntuベース14でユーザーランドを構築
 - 拡大するユースケース対応 *expanding usecases*
 - Ubuntuの商品搭載に課題となる依存関係の刈込やGPLv3関連のコンプライアンス対応 *dependency pruning and GPLv3 compliance*
- Kernel 開発をLTS(long term stable)ベースで開発
- 他社SoC BSPの修正の自社kernelへの取り込み *BSP change to my kernel*
 - LTSベースでgood。でもtarballで構成管理困る

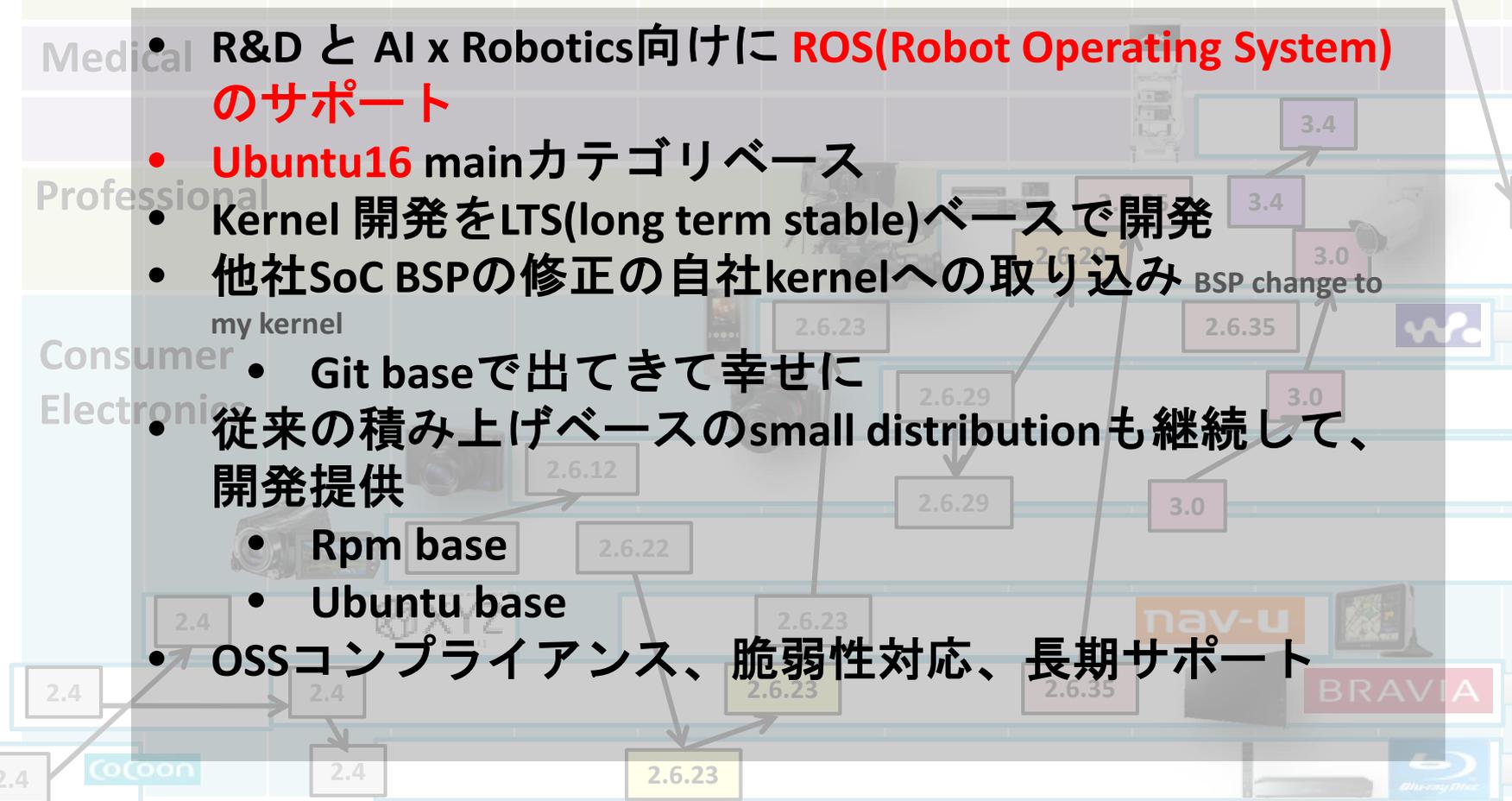
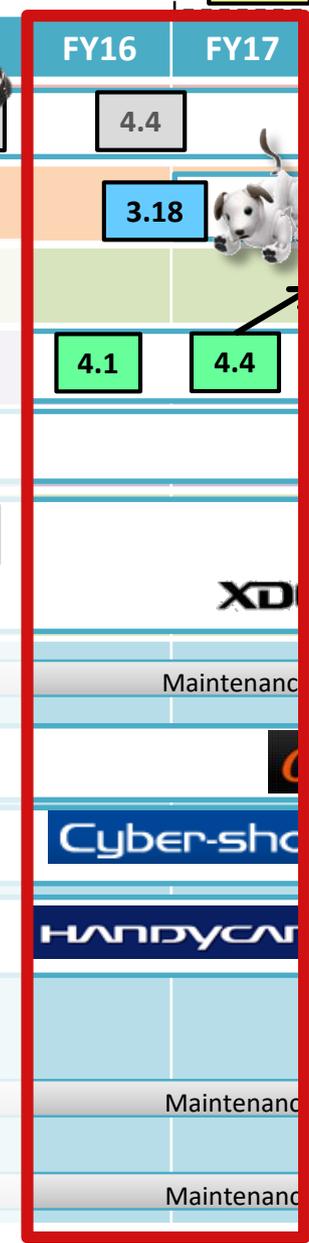


Stage 5: Ubuntu base+ROS+ kernel

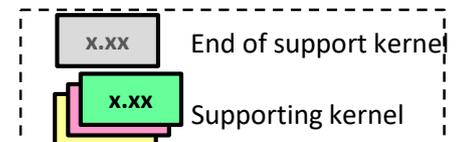


	FY03	FY04	FY05	FY06	FY07	FY08	FY09	FY10	FY11	FY12	FY13	FY14	FY15	FY16	FY17	FY18	FY19	
R&D											3.10	3.14		4.4		4.9	4.15	4.19
AI x Robotics														3.18				
Professional																		
Medical														4.1	4.4			
Professional																		
Consumer Electronics																		

- R&D と AI x Robotics向けに **ROS(Robot Operating System) のサポート**
- **Ubuntu16** mainカテゴリベース
- Kernel 開発をLTS(long term stable)ベースで開発
- 他社SoC BSPの修正の自社kernelへの取り込み
 - Git baseで出てきて幸せに
 - 従来の積み上げベースのsmall distributionも継続して、開発提供
 - Rpm base
 - Ubuntu base
- OSSコンプライアンス、脆弱性対応、長期サポート

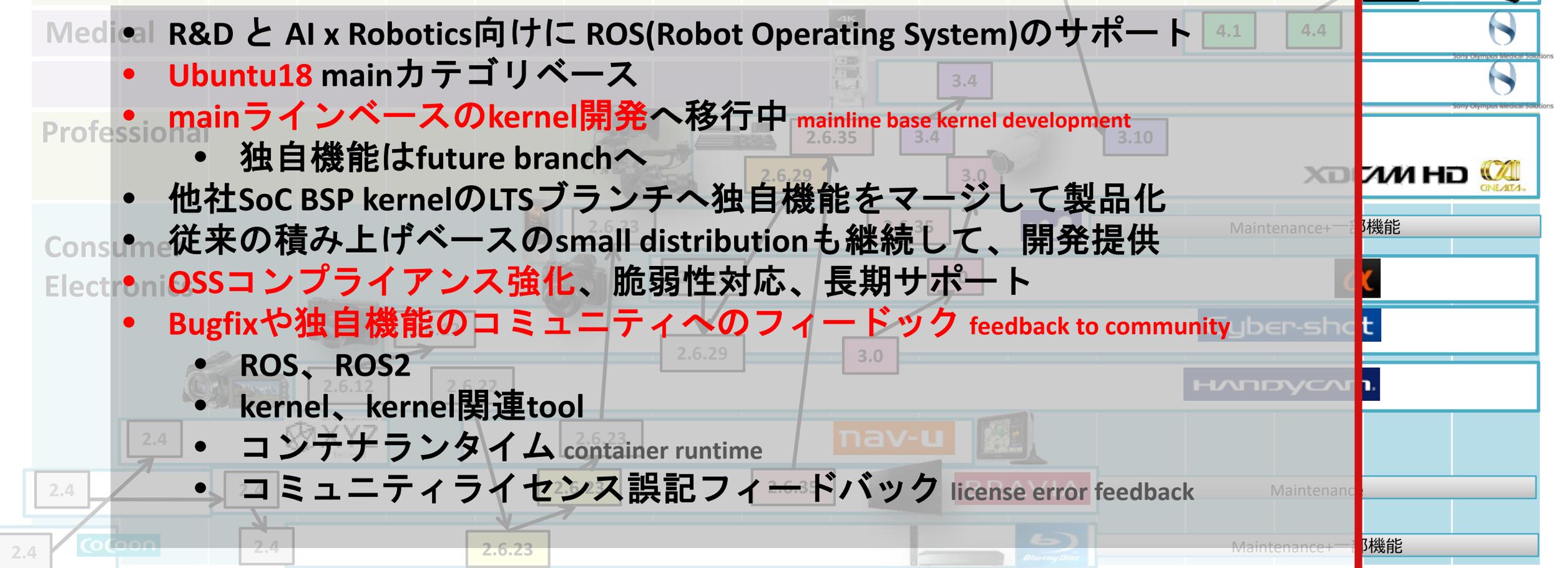


NOW: mainline baseの開発へ



	FY03	FY04	FY05	FY06	FY07	FY08	FY09	FY10	FY11	FY12	FY13	FY14	FY15	FY16	FY17	FY18	FY19	
R&D											3.10	3.14	4.4			4.9	4.15	4.19
AI x Robotics														3.18				
Professional																4.4		
Medical														4.1	4.4			
Professional																		
Consumer Electronics																		

- R&D と AI x Robotics向けに ROS(Robot Operating System)のサポート
- **Ubuntu18** mainカテゴリベース
- **mainラインベースのkernel開発へ移行中** *mainline base kernel development*
- 独自機能はfuture branchへ
- 他社SoC BSP kernelのLTSブランチへ独自機能をマージして製品化
- 従来の積み上げベースのsmall distributionも継続して、開発提供
- **OSSコンプライアンス強化、脆弱性対応、長期サポート**
- **Bugfixや独自機能のコミュニティへのフィードバック** *feedback to community*
- ROS、ROS2
- kernel、kernel関連tool
- コンテナランタイム *container runtime*
- **コミュニティライセンス誤記フィードバック** *license error feedback*



Agenda

- 自己紹介 Self introduction
- OSSとのかかわりあい Relationship with the OSS
- ソニーでのLinux導入の歴史 History of introduction of Linux in Sony
- **OSSの選び方、OSSを使った開発、フィードバック** How to choose OSS, development using OSS, Feedback
- まとめ Summary

製品開発における要求 requirements for the product development

- 機能 functionality
 - 最新のデバイス対応、最新技術
- 品質 quality
- 長期ソフトウェアサポート long SW support
 - Platformライフサイクル、脆弱性対策、バックポート... lifecycle, fix vulnerability, backport
- ライセンスクリアランス license clearance
 - 特許、OSSライセンス patent, OSS License
- 適切な開発コスト・製品開発プロセスとの整合性 cost & production cycle



- OSSの採用やバージョン選定でさまざまな**トレードオフ** tradeoff

品質から見たOSSの選択 quality

- 活発に活動しているか? Active?
 - <https://openhub.net>
- Distributionの主機能か? distribution main feature?
- OSSの主機能か? OSS main feature?
 - 基本はmain、baseに分類されている物を使う。 Use main/base
 - contrib, -bad, -uglyなどに分類されている部分は特許的に問題があるか、テスト不十分な部分なのでサポートは困難。 Some parts are hard to support

	Support	Hard to support, patent
ubuntu	「main」 「restricted」	「universe」 「multiverse」
Gstreamer	gstreamer, gst-plugin-base, gst-plugin-good	gst-plugin-bad, gst-plugin-ugly
OpenCV	opencv	opencv_contrib

ライセンスから見たOSSの選択 license

- 商品に搭載可能なOSSライセンスのみで構成されているか？ Productable?
 - 商品搭載できない hard to apply the product: AGPLv3, GPLv3, LGPLv3
 - 自社権利の保護に課題 risk to protect rights: GPLv2 Library
 - 依存しているOSSもクリアランスをするのは非常に大変 hard to clear dependent OSS license.
 - 手繰った先のGPLv2ライブラリやLGPLv3ライブラリが潜んでいるので依存を切る
- 特許ライセンス的にリスクがないか？ patent
 - OSSは第三者特許をライセンスしているわけではない OSS do not license 3rd party license.
 - 別途、特許ライセンスを取得する or 特許回避をする必要がある need to void other license
 - 他社特許明記OSSや特許懸念の記述があるOSS(OpenCV, gstreamer, ffmpeg...) patent risk OSS
 - OSSを使用することによる自社特許権行使の制限(GPL系, Apache2.0) restriction to our patents
 - 問題のあるモジュールを除外 delete inappropriate module
- 個々のOSSやOSSライセンスの判断は案件毎に法務・知財の判断が必要

OSS選択、OSSバージョン選択 choice of OSS and OSS version

- 製品開発プロセスでのイベントにあわせて選択 select by production event
- 各種OSS
 - プロジェクトスタート時の最新のubuntu LTSの「main」やYocto Stableを参考
 - ライセンスを精査しGPLv3/LGPLv3/AGPLv3でないバージョンに変更 downgrade SW to GPLv2
- Kernel version
 - X86の場合はubuntu LTSに採用されたkernel LTSを採用
 - ARMの場合はBSPが採用しているkernel LTSを採用
- 採用したバージョンが製品要求より古い場合 more newer version required
 - 次のdistribution LTSがでていれば、そのバージョン next LTS version
 - それぞれのOSSのstableをライフタイムを見ながら採用 stable version
 - stableがない場合は、サポートライフサイクルをみながら覚悟して採用 select by lifecycle

機能要求から起こる課題と対策 issue from functionality and measures

- 最新版のOSSを利用したい(機能 functionality、EOL end of life、脆弱性 vulnerability)
- ある最新デバイスを利用したい want to latest device
- ◆ 可能であればバックポート backport if possible
- ◆ API差分を提供して製品側で個別OSSバージョンアップ version up by product side
- あきらめる例 hard to support →大規模 OS乗り換えが必要 OS version up
 - ある一定以上のバージョンのgccが必要 newer gcc
 - gccはC++のABIがバージョンによって変更がある C++ ABI change → rebuild all
 - ある一定以上のバージョンのglibcが必要 newer glibc
 - 古いバージョンのkernelはサポートしていないのでkernelバージョンアップ必要 newer kernel
 - 古いSoCは新しいkernelではサポートされていない newer kernel do not support old SoC
 - ある一定以上のkernelのドライバフレームワークが必要 need to newer kernel framework
- ◆ 製品PJやSoCの切り替え時にkernel/distributionのversion up

製品向け技術開発とOSSフィードバック方針 development and OSS feedback

- **PI開始時にできるだけ最新の安定板を選択** latest LTS when project start
 - 不足機能があればmainlineからバックポートする missing function backport
- **足りないものは短期には自分で開発する** develop by self in short term
 - adhocなものは独自で保守する maintenance adhoc fix
 - Bugfixはコミュニティにフィードバックする feedback bugfix
- **次に同様の機能を開発する際には、最新の機能をベースに作り直す** rebase
 - メインラインで類似のことができるのであれば、独自機能は潔く捨てる throw adhoc
- **OSSで足りないところは、コミュニティに提案してコントリビューションする** contribute to community
- **新しいOSS技術はコミュニティとの連携も視野にいれる** make early technology with community

まとめ Summary

- OSSは製品開発では必須です OSS is essential for product development
- 製品ではライセンス、品質などを考慮にいったOSS選定が必要 OSS selection
 - ここらへんがノウハウでエンジニアの腕の見せ所 engineer knowhow
 - Sony internal distribution 頑張っています
- 新機能や新デバイス対応にはkernel/OSSバージョンアップの対策が必要 newer function/device require to version up OSS/kernel
 - Sony R&Dでは継続的なkernel/distributionの開発 continues development in R&D
 - 安心して製品が移行できる検証の充実、互換性情報提供など validation, compatibility info
- OSS活用をするためにはコントリビューションが重要 contribution is important
 - 効率的な開発、コミュニティとの相互信頼の確立、プレゼンスの向上 efficiencies, trust with community, presence

SONY

SONYはソニー株式会社の登録商標または商標です。

各ソニー製品の商品名・サービス名はソニー株式会社またはグループ各社の登録商標または商標です。その他の製品および会社名は、各社の商号、登録商標または商標です。