

CASE戦国時代に必要な ソフトウェアエンジニアのスキルとは

掲載および発言内容は各個人の見解であり、必ずしも所属する
企業や組織の立場、戦略、意見を代表するものではありません。

Yuichi Kusakabe

Agenda

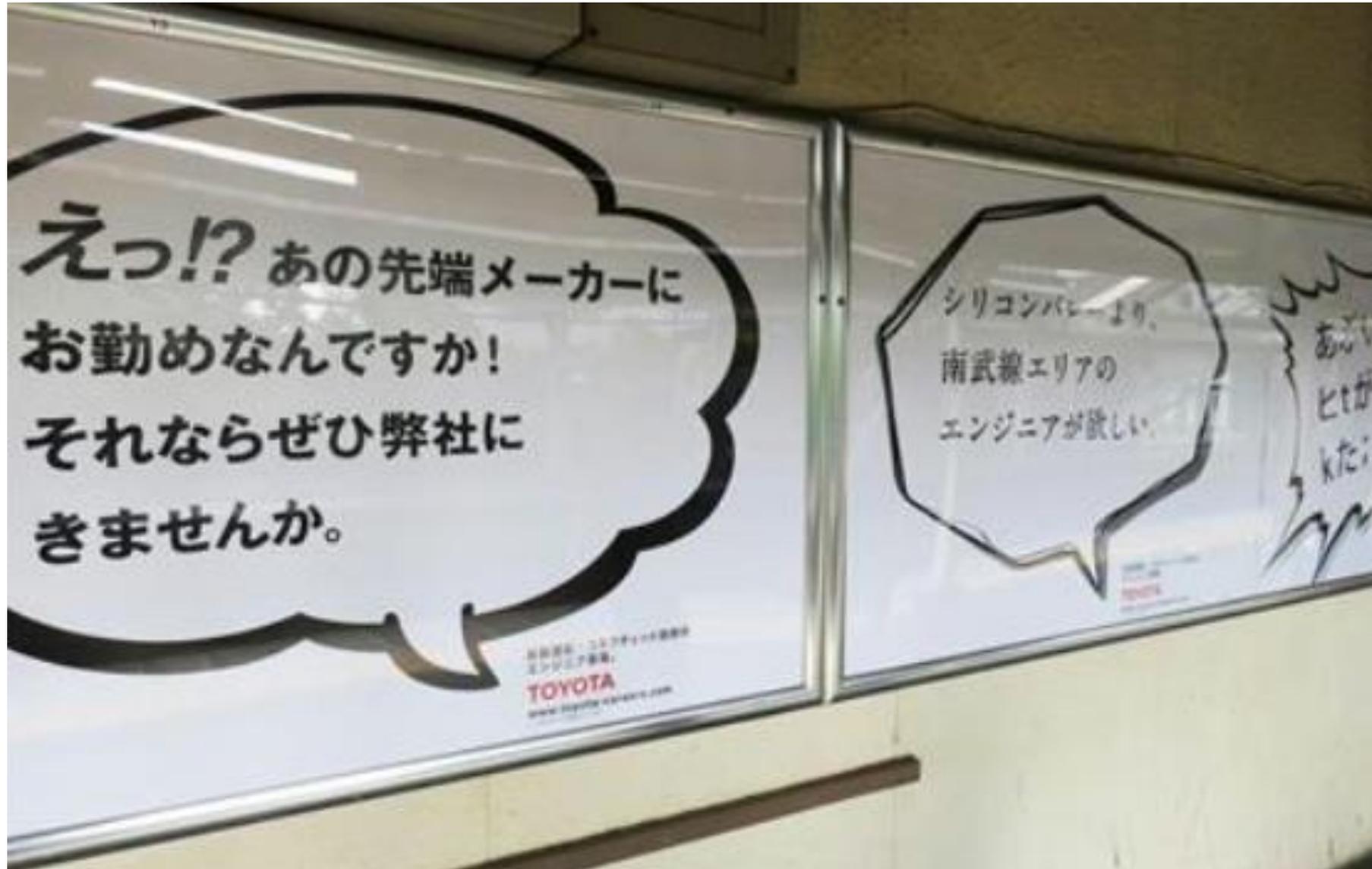
- 自動車メーカーのソフト内製化の動き
- 車両開発日程/各種法規対応
- ソフト内製化の課題
- エンジニアに必要なスキル

自己紹介

- Yuichi Kusakabe
- Software Engineer of IVI about 13 years
(for 16-bit and 64-bit architecture)
- Linux Software Engineer(2011-2013)
- Linux Software Lead Engineer(2013-Now)
- BSP Porting/Customizing
- Supporting for in-house software developers



自動車メーカーによるエンジニア取り合い(2017年)



自動車メーカーによるエンジニア取り合い

TOYOTA

[\[一覧に戻る\]](#)

【東京】デジタルコックピットソフトウェア開発：組み込みソフトウェアエンジニア（テクニカルリード）

職種

[コネクティッド]

勤務地

東京都（千代田区）

IT・システム | ソフトウェア開発 | IoT

特色

UXファースト、ソフトウェアファーストでトヨタのモビリティサービスを実現するため、お客様が実際に触れるデジタルコックピットシステム向けUI開発に携わっていただける組み込みソフトウェアエンジニアでテクニカルリードができる人材を求めています。

<MUST>

- リアルタイムOSあるいはLinuxでの組み込みC/C++開発経験
 - アーキテクチャ設計、システムデザイン、ハードウェアへのインテグレーション経験
 - 大規模システムの商品化経験
 - ユニットテスト・結合テストの経験（google testやmockなど）
 - 音声アーキテクチャのソフトウェア開発や運用性を考慮したソフトウェアデザインに関する基礎的な知識
 - オーディオ、グラフィック、Over-The-Airでのソフトウェア更新システムなどの量産開発経験、
- ※上記の経験ないしはそれに相応する能力があること

- コンピュータサイエンス、コンピュータ工学の学士以上、あるいは相当以上の経験
- 優れたリーダーシップ能力と卓越したコミュニケーション能力
- 現職での設計・コーディングでの優れた技術的な貢献

<WANT>

- LinuxあるいはリアルタイムOSの開発経験
- CI/CDの設定経験があること
- dockerの経験があること
- スクラム、アジャイルといったモダンな開発手法の経験

自動車メーカーによるエンジニア取り合い



JOB DESCRIPTION

募集職種詳細

【技術系】 コネクティッドカー&サービス

[C2016](コネクテッドカー)コネクテッドサービス&モバイルサービス バックエンド/Webフロントエンジニア

職務内容

<職務内容>

①所属組織の担当開発領域、業務概要とR&D内における役割、ポジション

ソフトウェア&ユーザーエクスペリエンス開発部では、クルマとつながるスマートフォンアプリやスマートスピーカー向けスキルなどのプロダクト開発や、MAASサービスに関する開発を担当しています。新規サービスの開発と、発売後にビジネスオーナーからの要求やマーケットの状況を元に継続的にプロダクトを継続的に改善するライフサイクルマネジメントや運用を両立しています。

②具体的な担当業務内容と、自部署内外で期待される役割、ポジション

コネクテッドアプリ開発・運用やMAASサービス開発・運用において、増加する車種やリージョン対応や、新規サービス立ち上げに向け、バックエンドやWebフロントエンドの開発・運用をご担当いただきます。ステークホルダーやチームメンバーと共に、要件/仕様策定・設計・プログラミング・テストなどを実施頂きます。実際にご担当頂く役割に関しては、開発者やリードリーダーなど多岐に渡り、ご本人様のスキルセットやプロジェクトの状況次第で調整させていただきます。

自動車メーカーによるエンジニア取り合い



募集職種詳細

本田技研工業株式会社

コネクテッドカー・サービス研究・開発（IVI及びデジタルコックピットの研究・開発におけるシステムアーキテクト）

職務内容

【具体的には】

車載情報システム（デジタルコックピット等）およびインフォテインメントシステムの

- IVIシステムのソフトウェア開発におけるアーキテクチャ設計
- 開発プロセスに則った品質監査およびプロセス改善
- ソフトウェアの全体設計検討、設計レビュー、品質分析
- 新技術調査、方針作成、リファクタリング検討
- 関係部署との仕様整合と協力会社との共同開発

※他部門や国内外のベンダー、研究機関など様々な関係者とコミュニケーションをとりながら開発業務を進めていきます。

海外出張など発生する可能性も大いにございます。

【開発ツール】

- ・開発言語: Java, C, C++, JavaScript
- ・開発環境: Windows, Linux
- ・開発管理: JIRA, Confluence
- ・構成管理ツール: Git, GitLab, Gerrit
- ・その他: Enterprise Architect (UML) 等

勤務地

- 本田技研工業株式会社 四輪事業本部 ものづくりセンター（六本木）

※将来的に国内・海外事業所への異動の可能性があります。

【求める経験・スキル】

- Linux / Android OSベースでの組込みソフトウェア開発経験

【上記に加え、あれば望ましい経験・スキル】

- IVIおよびメーター等の車載システム開発におけるアーキテクチャー/システム設計のご経験
- 複数の大規模ソフトウェアの同時開発経験

【求める人物像】以下の思い・適性をお持ちの方

- 自らの専門性にとらわれず、積極的にチームビルディングに参加できること
- チームに溶け込み、積極的なコミュニケーションを図ることができる方
- 組織、チームの中でリーダーシップを発揮できる方
- 新たな発想で技術提案、クリエイティブな顧客価値提案ができる方
- 柔軟な考え方で技術課題解決に取り組める方
- 自分の考えを積極的に発信し、周囲を巻き込んで課題解決の最良手段を見出せる力
- 世界初の技術を創り出し、世の中に貢献したいという思いのある方
- クルマを通し、お客様に新たな価値を提供したいという思いのある方
- 新しいことにチャレンジしたいという思いのある方
- 高い目標を掲げてやりきるエネルギーのある方

す（WEB面接で

収束後も積極

りに、これから

部品メーカーによるエンジニア取り合い

DENSO TEN

職種一覧(リスト形式)

職種検索

職種検索(都道府県)

職種カテゴリー一覧

マイページへ

【5】組込ソフトウェア開発※CI事業本部/東京

求人カテゴリー 東京その他

職種概要 車載マルチメディア機能（BT/wifi/スマートフォン連携/音声認識）のソフトウェア開発におけるミドルウェア層の開発業務をご担当頂きます。

業務内容

■業務詳細：

- ・顧客の要求を基にしたソフト仕様書作成
- ・ソフト仕様書を理解し、設計書を作成
- ・ソフト設計書に従い、プログラミング（ソフト作成/詳細設計）
- ・プログラミングしたソフトのデバッグ（検証）

スマホ連動ミドルウェア開発となるため、ソフト開発のみならず、apple社やGoogle社、Baidu社向けに認証取得の業務も担当を頂きます。

Apple：Apple CarPlay（全世界）、Google：Android Auto（中国除く全世界）、Baidu：Baidu CarLife(中国向け) となります。

■製品/ソフトウェアについて：

スマートフォンと車載機をUSBもしくはWireless(Bluetooth,WiFi)で接続し映像や音を転送します(車載機はディスプレイ、マイク、タッチパネルの役割を務めます)。ユーザーが運転中にスマートフォンを通してほしいことを車載ディスプレイで直接操作できるようにするための製品/ソフトウェアです。

■組織構成：

東京事業所には50名程の社員が在籍。そのうち、同じ部署(ソフト基盤技術部)は20名弱となっており、そこから機能ごとに3チームに分かれております

要件

■必須条件：

組込ソフトウェア設計の経験をお持ちの方
※開発フェーズは開発/設計などを問いません。
エンジニアとしてご自身の手で製品の開発～設計まで携わりたい方はぜひご応募ください

勤務地

東京事業所

何故ソフトウェアを内製化するのか？



2021年04月23日



社長就任会見 代表取締役社長 三部 敏宏スピーチ概要

本田技研工業株式会社 代表取締役社長 三部 敏宏（みべ としひろ）が、本日15時より社長就任会見を行いました。以下、会見でのスピーチの概要をお知らせします。

今後、クルマの競争軸はAI（人工知能）などの「ソフト」にシフトするといわれている。車両の販売後もOTA（Over The Air）によってソフトを追加・更新する「ソフトウェア・ディファインド・ビークル（SDV：Software Defined Vehicle）」が主戦場になる（図1）。SDVはすでに米Tesla（テスラ）が電気自動車（EV）「モデル3」などで実現しているが、トヨタ自動車やドイツVolkswagen（フォルクスワーゲン）なども追随する方向である。

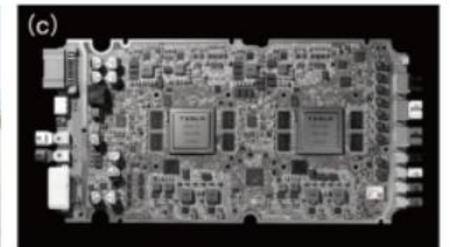
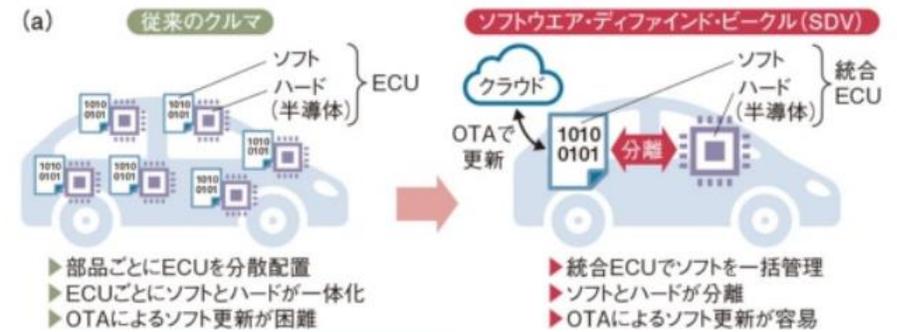


図1 ソフト定義のクルマへ

SDV：Software Defined Vehicle

自動車メーカーのソフト内製化の動き

ADOPTING LINUX ON BMW

THE LONG ROAD TO INTEGRATE LINUX AS A MAINLINE PLATFORM

Helio Chissin de Castro | October 2019

BMW
GROUP



BEHIND THE PROJECT ALMOST FROM THE GROUND

OS FROM SCRATCH – NON SAFETY RELEVANT

Full Linux based on Yocto

BMW GUI PORTED TO LINUX

Code adapted to be used on Linux, keeping legacy codebase

CONTINUOUS INTEGRATION IN HOUSE

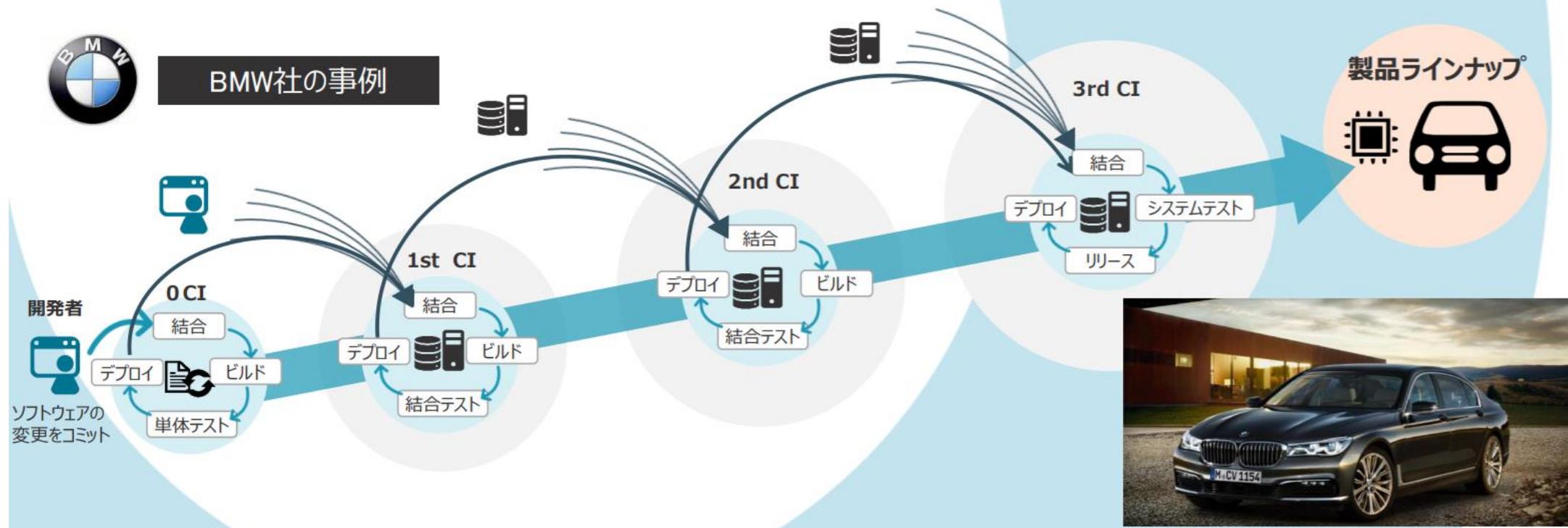
Not only integrating binary blobs anymore. All compilation done in-house



自動車メーカーのソフト内製化の動き

BMW社事例：段階的な継続的インテグレーションを実現

BMW社は、大規模のソフトウェア開発における段階的なCIシステムを導入している事例になります。BMW社の事例を実現するためには、コードがコミットされたら自動的にリリースまで行われるビルドパイプラインが整備されていることが必要になり、開発者の働き方の変更も必要になります。ですが、CI導入の基本としては、ビルドパイプラインは最初は小さな単位でビルドと検証を行いながら、フェーズが進みにつれて大きな単位での検証が行われるように設計します。そうすることで、エラーフィードバックが早くなり、影響範囲の小さなうちに修正されるようなプロセスとなります。

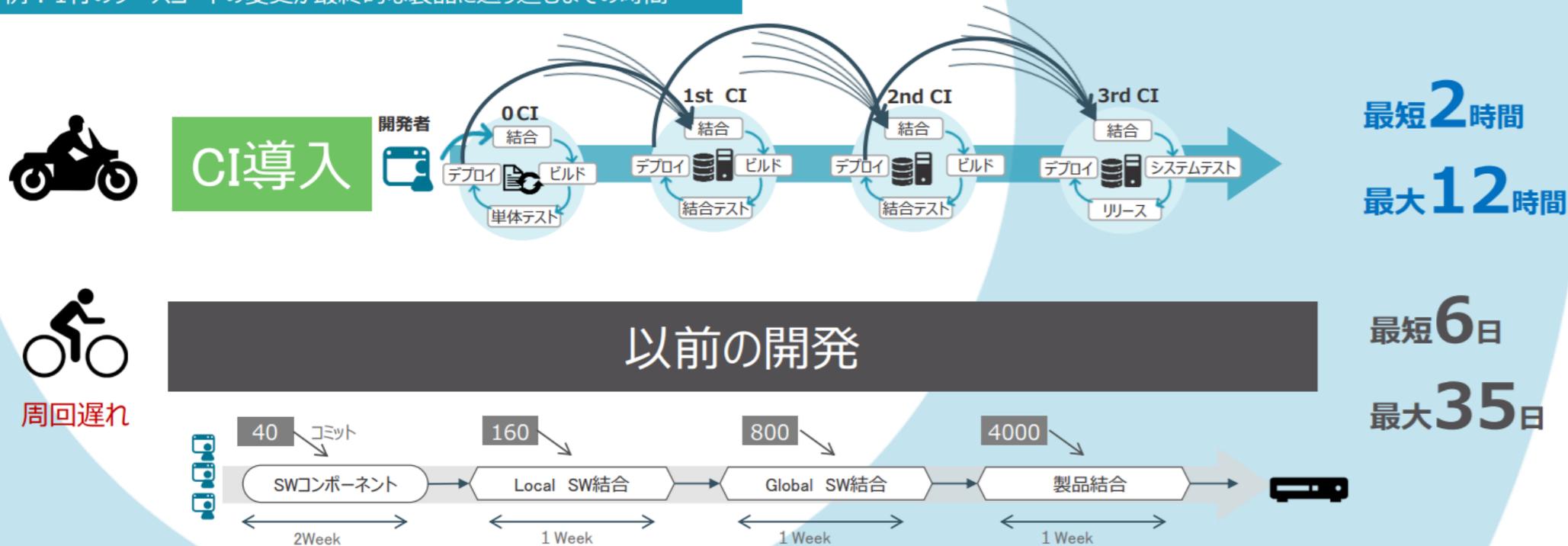


自動車メーカーのソフト内製化の動き

理想なソフトウェア開発では、製品のリリースを70倍速く

理想的なソフトウェア開発で、多段階CIシステムを導入した場合には、変更した内容が反映されるまでの時間に70倍の違いが生まれてきます。70倍の違いがある場合、目的地まで『6日』かかる乗り物と、目的地まで『2時間』で着く乗り物があるとしたら、どれだけの差が生まれるでしょうか？また、どれだけの費用対効果が得られるのでしょうか？

例：1行のソースコードの変更が最終的な製品に送り込むまでの時間



自動車メーカーのソフト内製化の動き

The banner features a dark background with abstract, glowing blue and white geometric patterns. The Toyota logo is in the top right corner. The main title 'TOYOTA Developers Night' is centered in large white font. Below it, the subtitle '内製開発チームで実践される“シリコンバレー流Flutter training Program”の全容' is highlighted with a blue underline. A smaller line of text below the subtitle reads '—元 Google エンジニアであるメンターの講義も生公開！—'. The date and time '2021.07.10 (Sat) 12:00 START' are in the bottom left corner.

TOYOTA

TOYOTA Developers Night

内製開発チームで実践される
“シリコンバレー流Flutter training Program”の全容

—元 Google エンジニアであるメンターの講義も生公開！—

2021.07.10 (Sat)
12:00 START

TOYOTA Developers Night 内製開発チームで実践される“シリ
コンバレー流Flutter training Program”の全容 ~元Googleエン
ジニアであるメンターの講義も生公開！~

Meetup

Flutter

UX

IoT

フロントエンド

チームビルディング

車両開発日程

試作 イベント	24ヶ月前 (試作)	18ヶ月前 (試作)	12ヶ月前 (正規工程)	6ヶ月前 (本型本工程)	2ヶ月前	1ヶ月前	量産
トヨタ	AS (試作)	FS (試作)	CV	1A	量確	品確	SOP
日産	Dロット	Cロット	VC	PT1	PT2	—	SOP
ホンダ	D1	段確	品確	量確	量確	—	LO

車両開発日程

AS (Advanced Stage) : 量産開始の24カ月前

試作検討段階で、試作品を使用します。場合によってはサプライヤーが確定しておらず、試作専門メーカーの部品が使用されることもあります。

FS (Final Stage) : 量産開始の18カ月前

最終試作評価のことで、ここでも試作品を使用します。自動車メーカーはこの段階で問題を潰しこみ、量産品の図面を作成します。ただし、近年では企画段階の部品共通化やCAEによる実証拡大が進んでおり、試作費削減という観点からこの段階での試作は行われない場合も増えています。

CV (Confirmation Vehicle) : 量産開始の12カ月前

性能確認車を製造します。設計の狙い通りの性能が達成できているか、性能確認車を使って各種試験が行われます。また、工場での組み立てが可能なのか、生産性も確認します。この段階では本型品が要求されます。

1A (1次号試) : 量産開始の6カ月前

量産試作です。本型、本工程品を使用します。自動車メーカーの実際の生産ラインで組み付けが行われます。この際に量産時と同様の設備や金型が求められるので、サプライヤーはこのタイミングまでに生産準備を完了させておく必要があります。生産される数量は少なく、LVPT (Low Volume Production Trial) としての位置付けです。

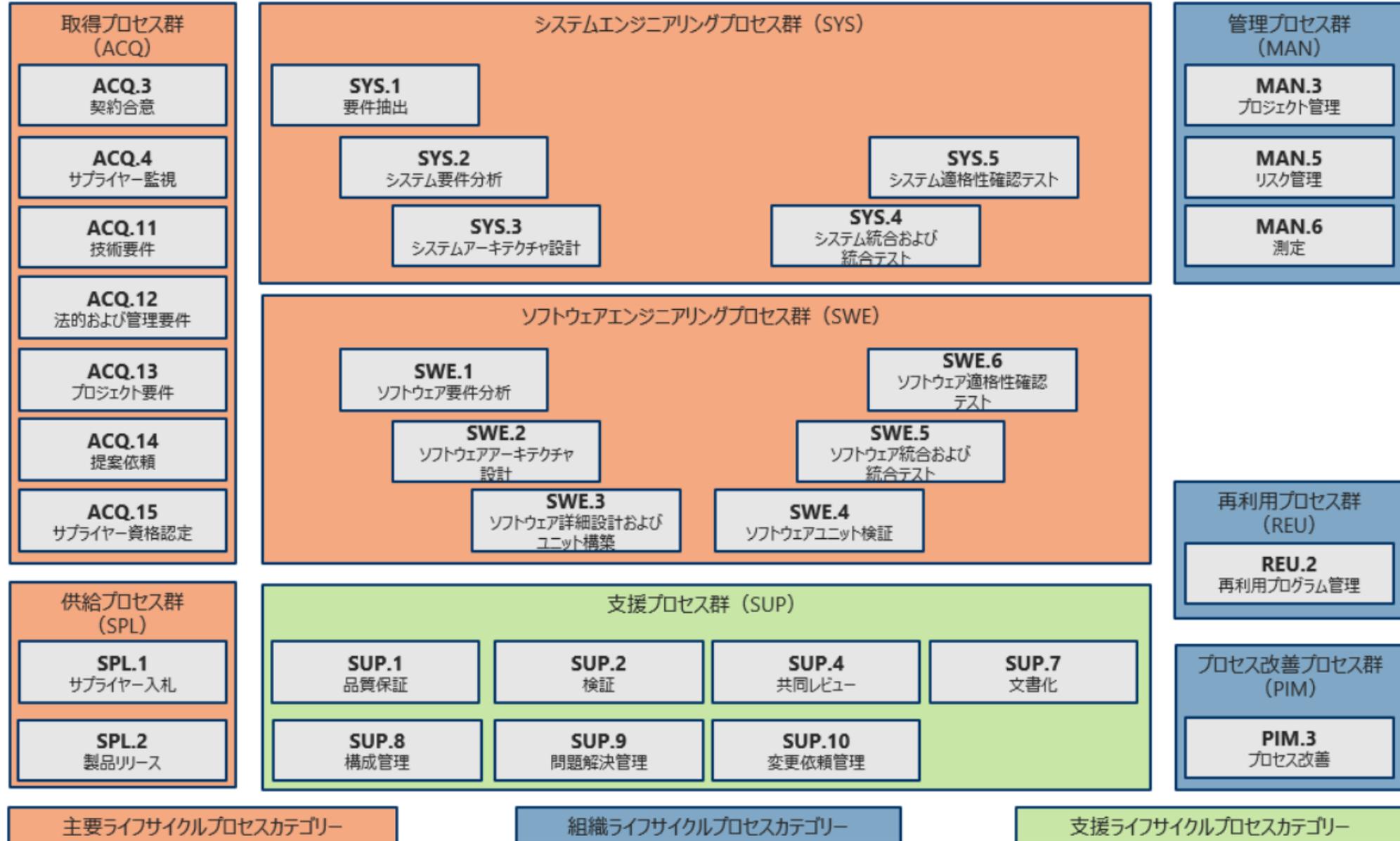
各種レギュレーション

- KT法(Cameron Gulbransen Kid's Transportation Safety Act of 2007)
- GDPR(General Data Protection Regulation)
- 自工会走行規制
- UN-R155/ISO 21434(サイバーセキュリティ対応)

車両開発における部品設計の難しさ

- 1台の車両には数多くのECUが内蔵されています
- IVIもいくつかのECUと接続しており、各ECUとの仕様整合や開発日程の整合など、車両開発日程がベースとなります
- つまりIVIは一つの製品ではなく、車両の部品として捉える必要があります

複雑な開発プロセス



車載特有の環境

- 電源が急に落ちる事があります
 - 工場での組立時のハーネス挿抜、ディーラでの作業、クランキング等
- いつどのタイミングで電源がOFFされても異常にならないソフトウェアの実装が必要となります
- 例えば下記コードは大丈夫でしょうか？
 - `fopen()` -> `fwrite()` -> `fclose()`

車載特有の環境

- 電源が急に落ちる事があります
 - 工場での組立時のハーネス挿抜、ディーラでの作業、クランキング等
- いつどのタイミングで電源がOFFされても異常にならないソフトウェアの実装が必要となります
- 例えば下記コードは大丈夫でしょうか？
 - `fopen() -> fwrite() -> fclose()`
 - `fwrite()`が完了してもデバイスにデータが書き込まれていない可能性があるため、**`fflush()/fsync()`**が必要です
 - `fopen() -> fwrite() -> fflush() -> fsync() -> fclose()`
- 但しOSS、調達ソフトだと対応が困難な場合があります

HWを考慮した設計

8GB eMMCリコールに関するよくある質問

Teslaは、ソフトウェアを更新し、メディアコントロールユニット(MCU)に搭載された8GB埋め込み型MultiMediaCard (eMMC)を交換するために、一部のModel S、およびModel X車両を自主的かつ積極的にリコールすることを決定しました。このリコールは、2018年3月より前に製造されたModel S、およびModel X車両にのみ適用されます。MCUに8GB埋め込み型MultiMediaCard (eMMC)が搭載されており、**書き換え摩耗**により当該症状が発生する可能性があります。2018年3月以降に製造されたModel S、およびModel X車両には、8GB eMMCは搭載されておらず、このリコールの対象ではありません。

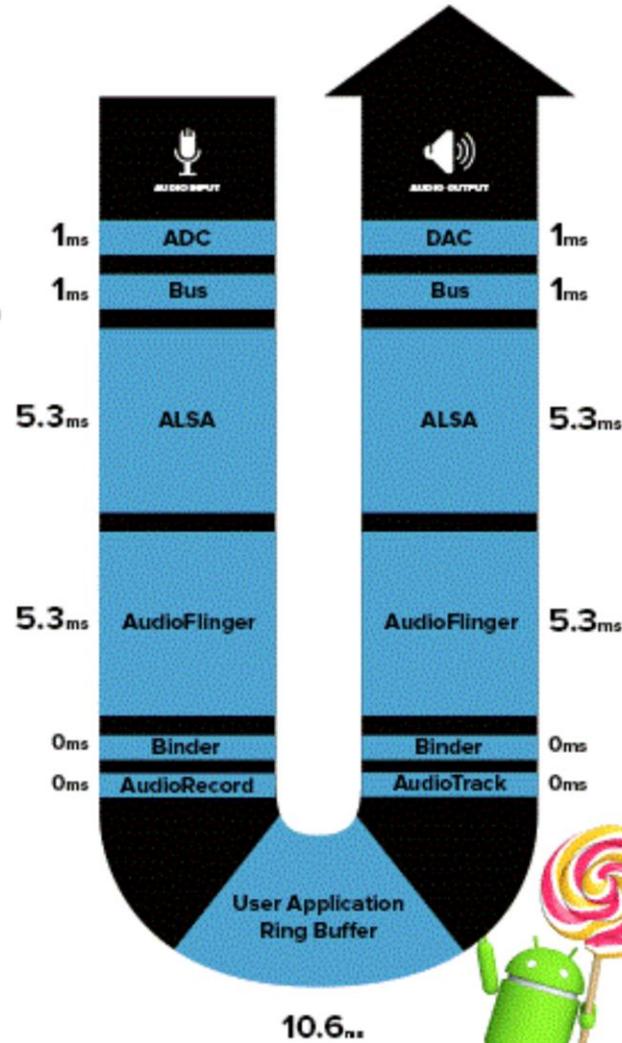
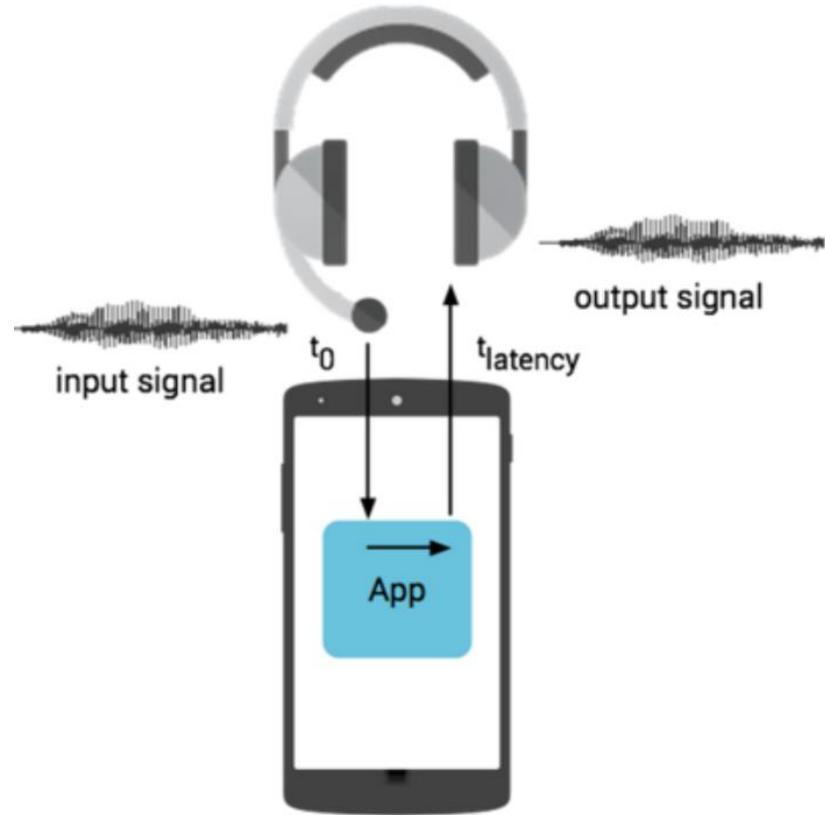
このリコールは、ソフトウェアの更新ですか、それともコンポーネントの交換が必要ですか？

その両方です。このリコールの第1段階の対応は、お客様がソフトウェア リリース2020.48.12またはそれ以降のリリースを確実にインストールし、潜在的な安全上の問題に対処することです。第2段階では、車両の利用可能なメモristorageを**8GBから64GB**に無償でアップグレードしていきます。

車を安全に運転することはできますか？

この症状に起因する事故や怪我については報告されていません。書き換え摩耗により当該症状が発生した場合、タッチスクリーンを再起動してもセンターディスプレイがブランクのまま回復しない場合や、サービスへの連絡を指示するメモristorageデバイスの劣化通知アラートが表示される場合があります。eMMCの当該症状は、加速、ブレーキ、ステアリング操作に影響を与えることはなく車両の運転性能に問題はありますが、2020.48.12以前の古いソフトウェアを使用していてeMMCの症状が発生した場合、リヤビューバックアップカメラにアクセスできなくなったり、外部方向指示器が点灯しなくなったり、フロントガラスのデフォグおよびデフロスト コントロールが使用できなくなる場合があります。

厳しい性能要件



Actual latency data for Google Nexus 9 tablet

モデル	プラットフォームバージョン	ビルド番号	サンプルレート (Hz)	バッファサイズ (フレーム)	バッファサイズ (ミリ秒)	ラウンドトリップレイテンシ (ミリ秒) ±1 のバッファ
Nexus One	2.3.6	GRK39F	44100	768	17.4	345
Nexus S	2.3.6	GRK39F	44100	1024	23.2	260
Nexus S	4.0.4	IMM76D	44100	1024	23.2	260
Nexus S	4.1.2	JZO54K	44100	880	20	210
Galaxy Nexus	4.0.1	ITL41D	44100	976	22.1	270
Galaxy Nexus	4.3	JWR66Y	44100	144	3.3	130
Nexus 4	4.2.2	JDQ39E	48000	240	5	195
Nexus 4	5.1	LMY470	48000	240	5	58
Nexus 10	5.0.2	LRX22G	44100	256	5.8	36
Nexus 10	5.1	LMY47D	44100	256	5.8	35
Nexus 7 (2013)	4.3	JSR78D	48000	240	5	149
Nexus 7 (2013)	4.4	KRT16S	48000	240	5	85
Nexus 7 (2013)	5.0.2	LRX22G	48000	240	5	64
Nexus 7 (2013)	5.1	LMY470	48000	240	5	55
Nexus 7 (2013)	6.0	MRA58K	48000	240	5	55
Nexus 5	4.4.4	KTU84P	48000	240	5	95
Nexus 5	5.0.0	LRX210	48000	240	5	47
Nexus 5	5.1	LMY47I	48000	240	5	42
Nexus 5	6.0	MRA58K	48000	192	4	38
Nexus 9	5.0.0	LRX21L	48000	256	5.3	35



OSSの活用

teslamotors / buildroot

Watch 117 Star 573 Fork 129

Code Pull requests 9 Actions Security Insights

buildroot-2019... buildroot / configs / ap-hw3_model3_defconfig Go to file ...

Tesla OpenSource internal-ce1ef36ccf6377873fc3fac3d5e4d2a493f865f6 (2019.40.50) Latest commit 7dd296c on 3 Jan 2020 History

0 contributors

152 lines (152 sloc) | 4.57 KB Raw Blame

```
1 BR2_aarch64=y
2 BR2_DL_DIR="$(BASE_DIR)/dl"
3 BR2_ENABLE_DEBUG=y
4 BR2_TOOLCHAIN_BUILDROOT_VENDOR="tesla"
5 BR2_TOOLCHAIN_BUILDROOT_GLIBC=y
6 BR2_KERNEL_HEADERS_4_14=y
7 BR2_TOOLCHAIN_BUILDROOT_CXX=y
8 BR2_TOOLCHAIN_BUILDROOT_FORTRAN=y
9 BR2_TARGET_OPTIMIZATION="-g2"
10 BR2_TARGET_GENERIC_HOSTNAME="ap3"
11 BR2_TARGET_GENERIC_ISSUE="Autopilot ///"
12 BR2_INIT_NONE=y
13 BR2_ROOTFS_DEVICE_CREATION_DYNAMIC_EUDEV=y
14 BR2_TARGET_TZ_INFO=y
15 BR2_TARGET_LOCALTIME="America/Los_Angeles"
16 BR2_LINUX_KERNEL=y
17 BR2_LINUX_KERNEL_CUSTOM_GIT=y
18 BR2_LINUX_KERNEL_CUSTOM_REPO_URL="https://github.com/teslamotors/linux.git"
```

https://github.com/teslamotors/buildroot/blob/buildroot-2019.02/configs/ap-hw3_model3_defconfig

OSSの脆弱性対応

Search CVE List

You can search the CVE List for a [CVE Record](#) if the [CVE ID](#) is known. To search by keyword, use a specific term or multiple space. Your results will be the relevant CVE Records.

View the [search tips](#).

Search Results

There are **361** CVE Records that match your search.

Name	Description
CVE-2021-3613	OpenVPN Connect 3.2.0 through 3.3.0 allows local users to load arbitrary dynamic loadable libraries via an OpenSSL configuration file if present, which allows the user to run arbitrary code with the same privilege level as the main OpenVPN process (OpenVPNConnect.exe).
CVE-2021-3606	OpenVPN before version 2.5.3 on Windows allows local users to load arbitrary dynamic loadable libraries via an OpenSSL configuration file if present, which allows the user to run arbitrary code with the same privilege level as the main OpenVPN process (openvpn.exe).
CVE-2021-3450	The X509_V_FLAG_X509_STRICT flag enables additional security checks of the certificates present in a certificate chain. It is not set by default. Starting from OpenSSL version 1.1.1h a check to disallow certificates in the chain that have explicitly encoded elliptic curve parameters was added as an additional strict check. An error in the implementation of this check meant that the result of a previous check to confirm that certificates in the chain are valid CA certificates was overwritten. This effectively bypasses the check that non-CA certificates must not be able to issue other certificates. If a "purpose" has been configured then there is a subsequent opportunity for checks that the certificate is a valid CA. All of the named "purpose" values implemented in libcrypto perform this check. Therefore, where a purpose is set the certificate chain will still be rejected even when the strict flag has been used. A purpose is set by default in libssl client and server certificate verification routines, but it can be overridden or removed by an application. In order to be affected, an application must explicitly set the X509_V_FLAG_X509_STRICT verification flag and either not set a purpose for the certificate verification or, in the case of TLS client or server applications, override the default purpose. OpenSSL versions 1.1.1h and newer are affected by this issue. Users of these versions should upgrade to OpenSSL 1.1.1k. OpenSSL 1.0.2 is not impacted by this issue. Fixed in OpenSSL 1.1.1k (Affected 1.1.1h-1.1.1j).
CVE-2021-3449	An OpenSSL TLS server may crash if sent a maliciously crafted renegotiation ClientHello message from a client. If a TLSv1.2 renegotiation ClientHello omits the signature_algorithms extension (where it was present in the initial ClientHello), but includes a signature_algorithms_cert extension then a NULL pointer dereference will result, leading to a crash and a denial of service attack. A server is only vulnerable if it has TLSv1.2 and renegotiation enabled (which is the default configuration). OpenSSL TLS clients are not impacted by this issue. All OpenSSL 1.1.1 versions are affected by this issue. Users of these versions should upgrade to OpenSSL 1.1.1k. OpenSSL 1.0.2 is not impacted by this issue. Fixed in OpenSSL 1.1.1k (Affected 1.1.1-1.1.1j).
CVE-2021-3446	A flaw was found in libtpms in versions before 0.8.2. The commonly used integration of libtpms with OpenSSL contained a vulnerability related to the returned IV (initialization vector) when certain symmetric ciphers were used. Instead of returning the last IV it returned the initial IV to the caller, thus weakening the subsequent encryption and decryption steps. The highest threat from this vulnerability is to data confidentiality.
CVE-2021-3423	Uncontrolled Search Path Element vulnerability in the openssl component as used in Bitdefender GravityZone Business Security allows an attacker to load a third party DLL to elevate privileges. This issue affects Bitdefender GravityZone Business Security versions prior to 6.6.23.329.

サイバーセキュリティ対応

テンセントのハッキングチームがトヨタ車の脆弱性を報告、トヨタは既に対策済み

🕒 2020年03月31日 06時30分 公開

[齊藤由希, MONOist]

🖨️ 印刷する

📌 クリップする

📧 通知する

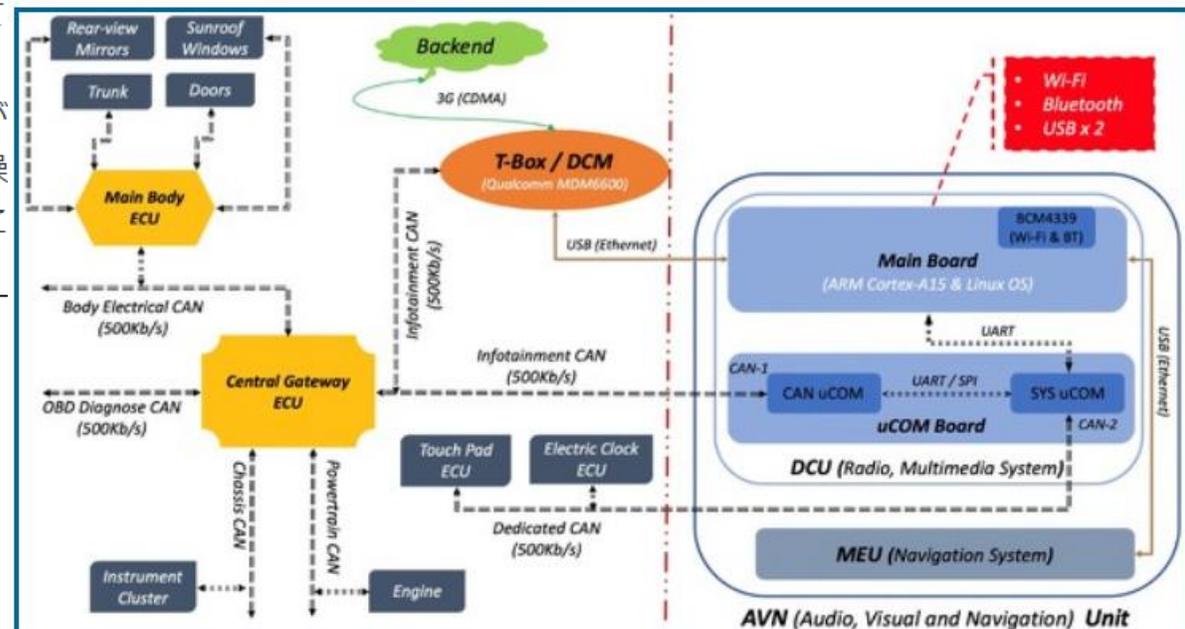
🐦 21

👤 Share

📌 B! 6

トヨタ自動車は2020年3月30日、既に販売された一部のレクサス車とトヨタ車について社外から脆弱性の指摘があり、対策を施したことを発表した。

サイバーセキュリティを研究するTencent Keen Security Labによるレポートを受けてトヨタ自動車側でも再現評価を実施。一部車種のマルチメディア製品のBluetooth通信を介して、一部の機能を遠隔操作できることを確認した。この脆弱性を利用した走る・曲がる・止まるの制御に関する遠隔操作はできないという。また、この脆弱性を突いた遠隔操作には極めて高度なプログラムや、対象車両と近い距離を保ち続けることが必要となることから、実現性が限定的で困難だとしている。

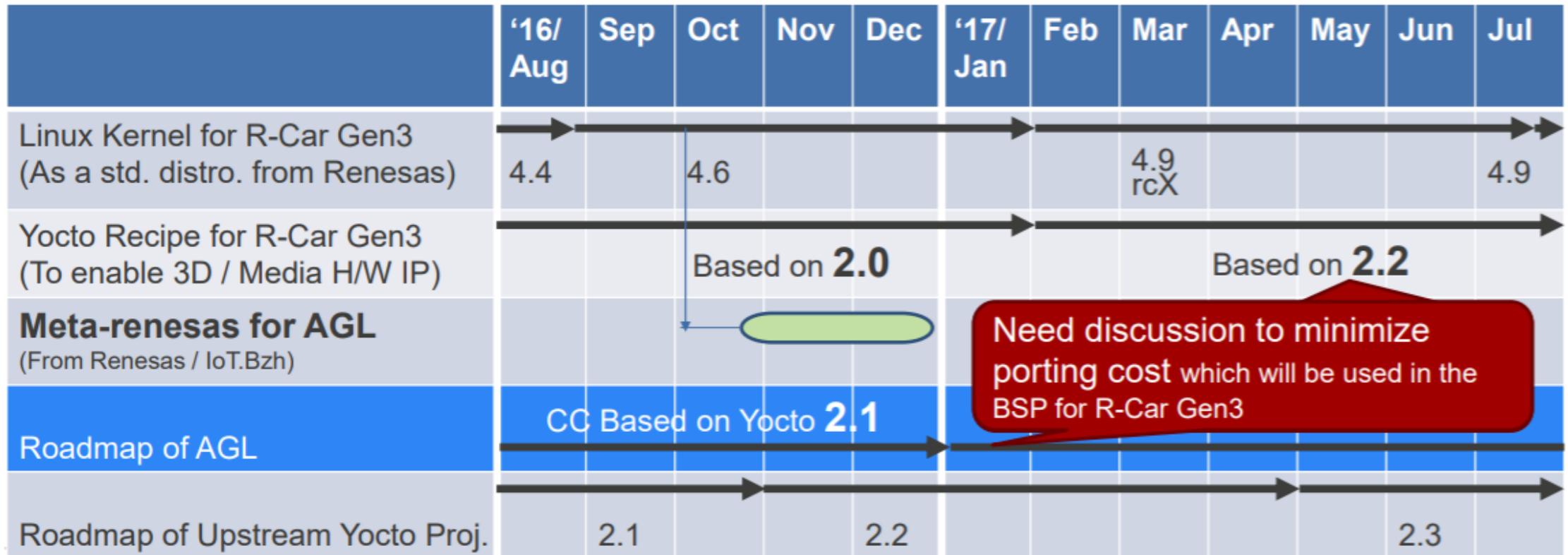


2017年モデルのレクサスNX300のシステム概要図（クリックして拡大） 出典：テンセント

BSPスケジュールの考慮

ROADMAP OF AGL BSP FOR R-CAR GEN3 RELEASE SCHEDULE OF THE BSP

- Renesas will release BSP and “meta-renesas for AGL”
- Renesas and partners will care the gap between Yocto 2.1 and BSP of R-Car Gen3



開発スケジュールの違い

試作 イベント	24ヶ月前 (試作)	18ヶ月前 (試作)	12ヶ月前 (正規工程)	6ヶ月前 (本型本工程)	2ヶ月前	1ヶ月前	量産
トヨタ	AS (試作)	FS (試作)	CV	1A	量確	品確	SOP
イベント	★納入イベント	★納入イベント	★納入イベント	★納入イベント			
仕様	★仕様書発行	★仕様変更(随時)					
SoC	ES →	CS →	MS →				
Kernel	4.14 →	→	5.4 →	セキュリティFix →			
開発	AS →	CV →	1A →	サポート →			

複数HWへの対応

- Renesas R-CAR H3、 M3、 M3N、 E3
- DDR : 8GB、 4GB
- eMMC : 128GB、 64GB、 32GB
- 解像度 : 外付けディスプレイ、 内蔵ディスプレイ
- AMP : 外付けAMP、 内蔵AMP
- カメラ : バックカメラ、 周辺カメラ
- 仕向 : HD、 DAB、 XM、 RDS

新規チーム/大規模開発の課題

- 新規に立ち上げたチームの場合、経験値/スキルの違い
- 300～400名近いエンジニアのマネジメント
- システム設計/全体最適の難しさ
- 人数が増えると、車載未経験の場合も

外の世界を知る

Top Companies in 2016

Company	Commits	Company	Commits
IoT.bzh	872	Witz	14
Fujitsu-Ten	273	Renesas	13
Linux Foundation	147	Mitsubishi Electric	11
Mentor Graphics	92	Intel	10
Qt Company	86	Samsung	6
Konsulko	68	Collabora	5
Panasonic	57	Toyota	5
Advanced Telematics Systems	42	ADIT	3
Microchip	22	Cogent Embedded	3
ALPS	20	Hitachi	2
AisinAW	18	Igalia	2
TI	15	Radio Sound	2

1791 Total Commits
45 Committers
24 Companies

- 01 Jan 2016 – 31 Dec 2016
- Commits to master

LINUX
FOUNDATION

Slide 4

AUT
GRA

Steps in Man*Month, Linux

FUJITSU

50

9

Copyright 2014 FUJITSU

生産性の違い

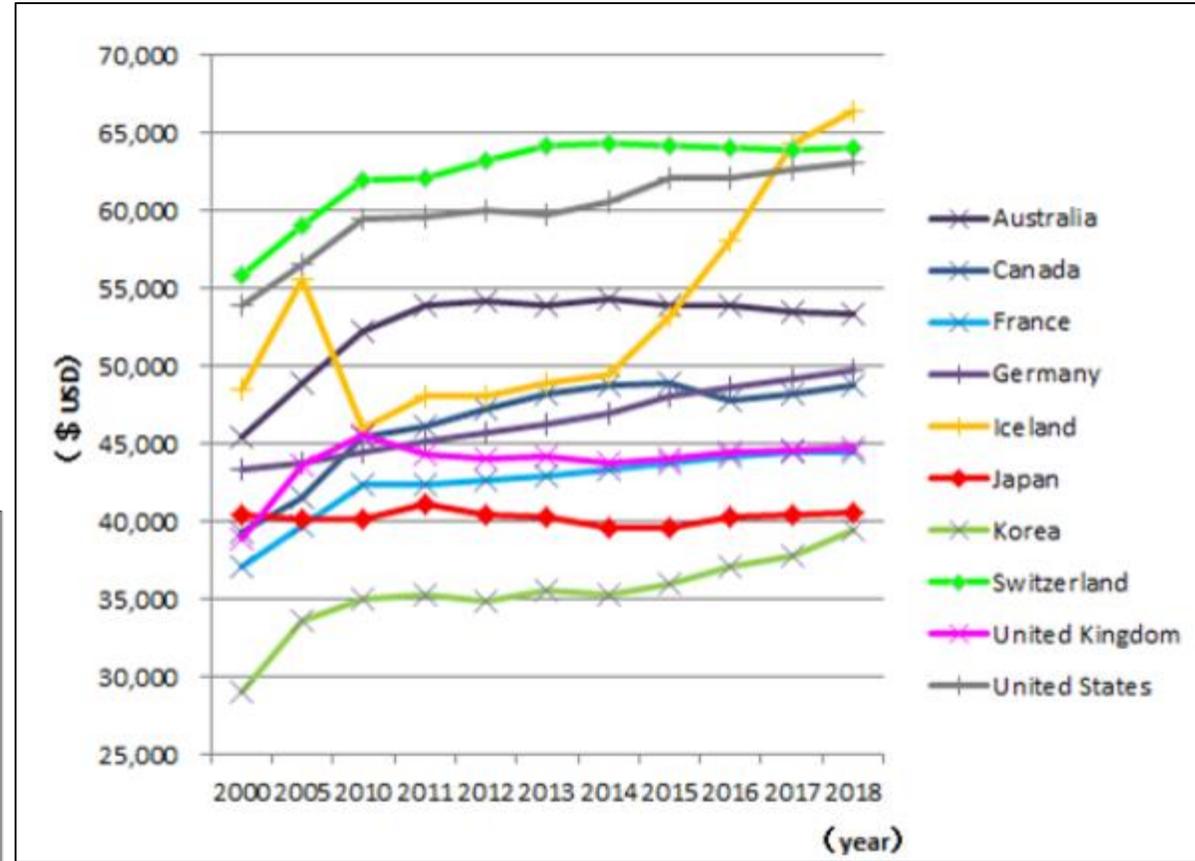
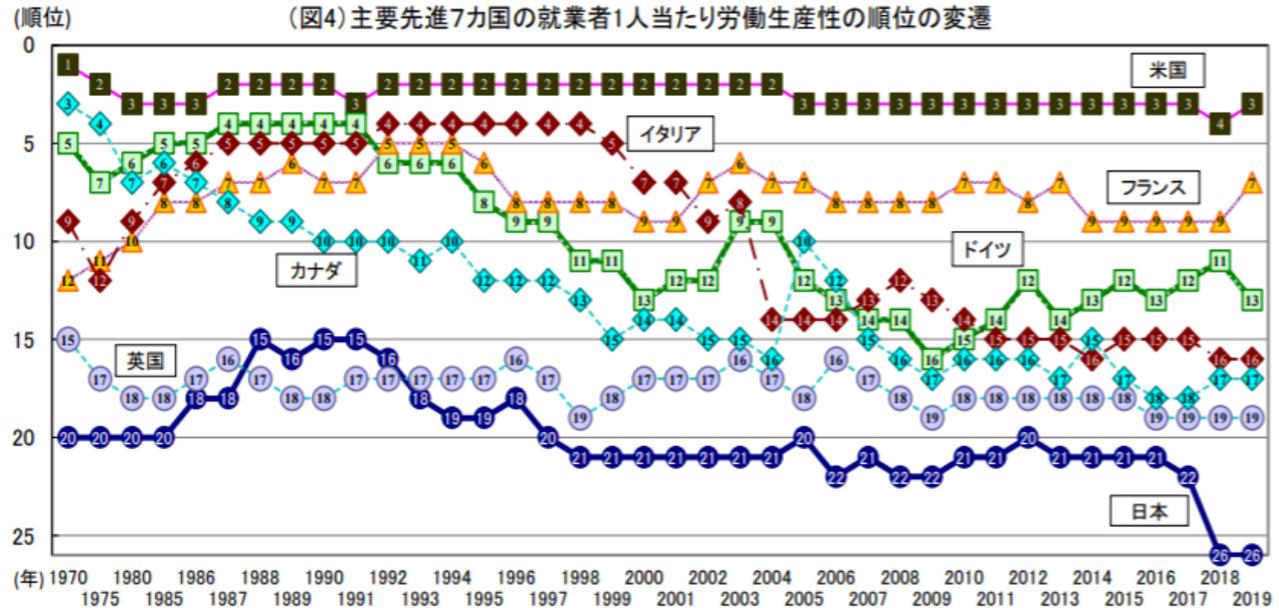
[要約]

1. 日本の時間当たり労働生産性は、47.9ドル。OECD加盟37カ国中21位。

・OECD データに基づく 2019 年の日本の時間当たり労働生産性(就業1時間当たり付加価値)は、47.9ドル(4,866円/購買力平価(PPP)換算)。米国(77.0ドル/7,816円)の6割の水準に相当し、順位はOECD加盟37カ国中21位だった。名目ベースでは前年から5.7%上昇したものの、主要先進7カ国でみると、データが取得可能な1970年以降、最下位の状況が続いている。

2. 日本の1人当たり労働生産性は、81,183ドル。OECD加盟37カ国中26位。

・2019年の日本の1人当たり労働生産性(就業者1人当たり付加価値)は、81,183ドル(824万円)。韓国(24位・82,252ドル/835万円)やニュージーランド(25位・82,033ドル/832万円)とほぼ同水準。名目ベースでは前年水準を3.4%上回ったが、順位でみるとOECD加盟37カ国中26位で、1970年以降最も低くなっている。



エンジニアに必要なスキルとは

- 車両開発/各種法規に関係する知識
- 先行開発と量産開発の違いを正しく理解
- チームビルディング/マネジメント
- 英語でのコミュニケーション
- サイバーセキュリティ/セキュアコーディング(CERT/CWE)
- HWの知識(APIをコールしてから、0/1の信号が出力されるまで)
- システム全体を把握出来るスキル(全体最適設計)
- OSSに関係する知識(ライセンスや脆弱性等も)
- 課題/問題解決(不具合解析含む)の力(洞察力)
- 外の世界にも目を向ける(自分の実力を正しく知る)
- C, C++, Java, Python, Docker, WSL2

何故ソフトウェアを内製化するのか？を正しく理解する

END