特集

# 来るべきSDV時代を支える電子制御開発の基盤技術

## The Foundation Technologies for Electronic Control Development Supporting the Upcoming SDV Era

宮田 八郎[*1] Hachiro Miyata
小澤 尚之[*1] Naoyuki Ozawa
前田 洋信[*1] Hironobu Maeda
福田 仁志[*1] Hitoshi Fukuda
後藤 宏之[*1] Hiroyuki Goto
藤井 英樹[*1] Hideki Fujii
小林 貢[*1] Mitsugu Kobayashi
犬塚 浩之[*1] Hiroyuki Inuzuka

＊1 EC開発部

## 1 はじめに

### 1.1 モビリティ向け電子制御基盤技術動向

自動車および産業車両などモビリティでは、カーボンニュートラル実現に向けた世界的な取組みの加速を受けクリーン・ゼロエミッションにつながる電動化技術、安全・安心・快適な社会を目指すための自動化技術、および新たなサービスやスマート製品を実現するコネクティッド技術など電子制御技術の進化が加速してきている。

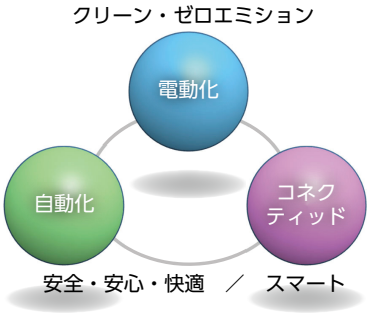



図1 モビリティ向け電子制御技術
Fig.1 Electronic Control Technology for Mobility

こうした電子制御技術の進化をソフトウェアが担うようになり、ソフトウェアでクルマの機能・性能が決まるソフトウェアデファインドビークル(以下、SDV)への移行に伴いソフトウェアの重要性が高まっている。

SDV時代のソフトウェアは、電動化、自動化、コネクティッドといった電子制御技術を統合して車両で機能させるため、従来の開発に比べてさらに制御の複雑性が増している。

このような高度に複雑化されたソフトウェアを開発するには、新しい基盤技術である仮想化技術やセキュリティ、オープンソフトウェアの活用などにより安全性と開発の効率を向上するとともに製品品質を確保していく必要がある。

また、品質と安全性、およびセキュリティを確保できるようにするための法規、規格・標準に対応できる開発プロセスの整備やソフトウェア開発技術者の育成が重要となってきている。

#### 1)SDV時代の新しいソフトウェア基盤技術

(1)仮想化技術

自動車の制御が機械式から電子制御に変わっていくことで、車両に搭載されるECU[*1]が増加しつづけ、従来の分散型ECU構成のままでは車両全体を一つのシステムとして成立させることが困難になってきた。そこで、同機能(ドメイン)軸でECUの機能を集約したドメイン集約型ECU構成が採用されるようになった。さらに、自動運転システムや先進運転支援技術の高度化に伴い、セントラルECUという多数の機能が統合された中央集中型ECU構成に移行してきている(図2)。

注:＊1　ECU(Electronic Control Unit):電子制御装置

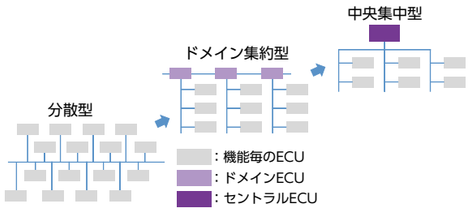



図2 ECU構成の変遷
Fig.2 The Evolution of ECU Architecture

複数のECUに実装されたソフトウェアを一つのセントラルECUに統合するためには仮想化技術が活用される(図3)。これにより、ECUのリソース(CPU、メモリなど)を効率的に利用でき、ECUのコストを削減し、車両全体を一つのシステムとして成立させやすくすることができる。さらに、ECUが少なくなることで、システム全体の保守が容易になる。

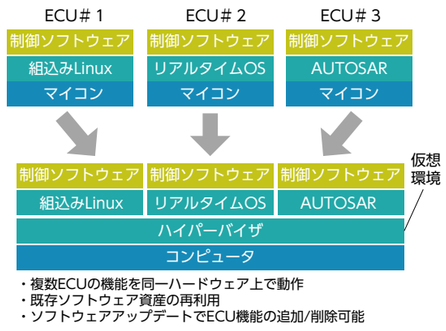



図3 仮想化技術
Fig.3 Virtualization Technology

(2) セキュリティ

コネクティッドサービスの普及により遠隔から悪意ある車両のハッキングによって引き起こされる事故などサイバー攻撃を受けるリスクが高まっている。このようなリスクに対応するために、強固なセキュリティ対策技術とともに車両のセキュリティを常に最新状態にするためのソフトウェアアップデート技術が必要となる。その中で、OTA[*2]は、短時間に多数の車両へ最新のセキュリティ対策を提供することができる。また、ユーザーに対しても販売店に車両を入庫することなくアップデートすることができる重要な技術となる。

(3) オープンソースソフトウェアの活用

自動化・自律系システムなどの高度化・知能化したサービス機能を効率的に実現するためにROS[*3]、画像認識、機械学習などのオープンソースソフトウェア(以下OSS[*4])が広く使われている。ただし、OSSはLinux[*5]環境で動作するものが多く、組込みLinux(組込み機器向けに最適化されたLinux)(図4)を導入する必要がある。また、OSSは無償かつ開発効率向上に有用である一方、ライセンス規約を遵守した上での活用が必須である。

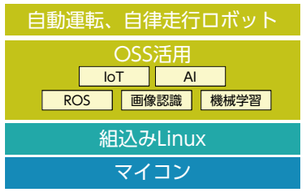



図4 組込みLinux
Fig.4 Embedded Linux

**2) 法規、規格・業界標準化動向**

ソフトウェアの役割が増加するに伴い、従来の品質と開発効率はもとより、安全とセキュリティの確保もソフトウェアの重要な課題となってきた。これに対応するため、国際基準に基づく法規、規格・標準が制定され、自動車メーカや政府認証機関からはこれらの基準への遵守が要求されている。代表的な規格・標準として、品質に関するAutomotive SPICE、安全に関するISO26262、セキュリティに関するISO/SAE21434が制定されている(図5)。

## 1.2 電子制御基盤技術の強化

ソフトウェアの品質、安全、セキュリティに関する法規、規格・標準への対応が必須となるなか、当社では、2011年10月から全社的な電子制御基盤技術の強化を目的としたECU開発推進プロジェクトを発足し、開発体制の整備を開始した。そして、2014年6月にEC開発部として本格的な活動に移行した。

EC開発部では、車載電子制御ソフトウェアの標準化団体であるJASPAR[*6]、および自動車のサイバーセキュリティに関する情報共有や対策向上を推進するJ-AUTO-ISAC[*7]など、業界の標準化を推進する様々な団体に参加している。こうした団体での活動の成果やそこで得られた最新の情報を社内に取り込むことで、電子制御開発のやり方(開発プロセス標準化)、電子制御技術開発(ソフトウェアプラットフォーム開発)、電子制御開発人材の育成(組込みソフトウェア技術者の育成・強化)の3軸をコア技術として、全社の電子制御開発の基盤技術を強化している(図6)。

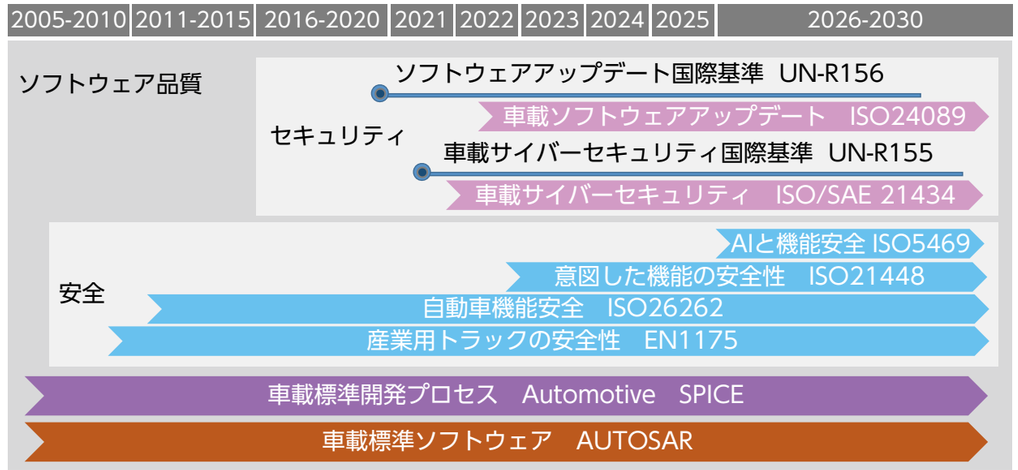



図5 電子制御技術動向(法規、規格・標準)
Fig.5 Trends in Electronic Control Technology (Regulations, International/ Industry Standards)

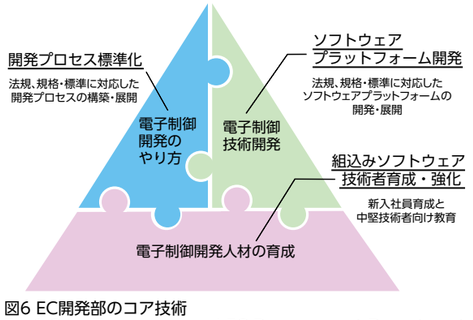



図6 EC開発部のコア技術
Fig.6 Core Technologies of EC Development Department

以降にて、当社の事業の継続および発展のために必要となってくるソフトウェアの品質、安全性、セキュリティおよび組込みソフトウェア技術者育成の取組みについて説明する。

# 2 品質向上

## 2.1 ソフトウェア品質について

**1) 開発プロセス**

欧州の自動車メーカが中心となり、車載ソフトウェアの品質向上を目的とした国際的な業界の標準開発プロセスであるAutomotive SPICE[*8]が策定され、2006年より本格的な運用が開始された。Automotive SPICEでは、ソフトウェア開発で実施すべきプロセスの目標と成果を定義し、開発能力を5段階レベルで評価できるようにしている。多くの自動車メーカは、サプライヤに対し、組織標準プロセスに従った開発ができるレベル3に達していることを要求している(図7)。

注:＊2 OTA(Over The Air):無線による遠隔からの各種サービスの総称。ここでは無線によるソフトウェアアップデートのことを意味する
＊3 ROS(Robot OS):ロボットの制御機能を開発するためのソフトウェアライブラリ
＊4 OSS(Open Source Software):ソースコードを公開し、使用、修正、再配布などが可能なソフトウェアの総称
＊5 Linux:OSSとして公開されているオペレーティングシステム

注:＊6 JASPAR(Japan Automotive Software Platform Architecture):一般社団法人JASPAR。日本の自動車業界で車載ソフトウェア・電子システム開発の効率化・標準化を行うことを目的として設立
＊7 J-AUTO-ISAC(Japan Automotive ISAC(Information Sharing and Analysis Center)):一般社団法人J-AUTO-ISAC。自動車に関するセキュリティについて、関係企業が協力して情報共有やセキュリティ対策を行うことを目的として設立
＊8 Automotive SPICE:Software Process Improvement and Capability dEtermination

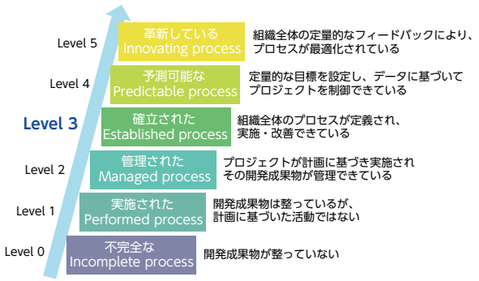



図7 Automotive SPICE開発能力レベル
Fig.7 Automotive SPICE Process capability levels

Automotive SPICEで定義されるプロセスは当初、ソフトウェア開発を中心としたプロセス(図8)とシステム開発プロセスが標準化対象であったが、その後、ハードウェア開発プロセス、メカニカル開発プロセスへと対象を拡張している。現在、サイバーセキュリティやアジャイル開発のプロセスにも拡張中である。

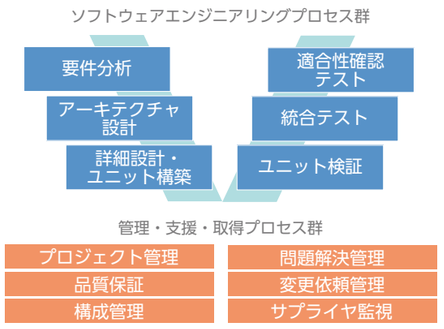



図8 Automotive SPICE定義ソフトウェア開発プロセス
Fig.8 Automotive SPICE Defined Software Development Processes

また、ソースコードの品質に関しては、プログラムの可読性、保守性、移植性、および信頼性を向上させることを目的としたC言語コーディングルールMISRA-C[*9]や、セキュリティを向上させることを目的としたC言語コーディングルール CERT-C[*10]が策定されている。

### 2)車載ソフトウェアプラットフォーム

ソフトウェアの品質と開発効率の向上を目的とし2003年に設立された車載ソフトウェアの標準化団体AUTOSAR[*11]により車載標準ソフトウェア仕様“AUTOSAR”が策定された。

AUTOSAR仕様ではさまざまなECUのソフトウェアに共通して搭載される機能を集約したソフトウェアプラットフォームをBSW[*12]と定義し、BSW仕様や制御ソフトウェアおよびハードウェアとのインターフェース仕様を標準化している(図9)。

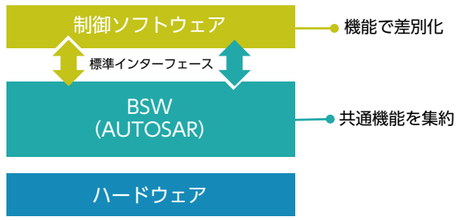



図9 AUTOSAR標準ソフトウェア構成
Fig.9 AUTOSAR Standardized Software Architecture

共通機能としては、車両内通信、診断、マイコン制御、システム、セキュリティなどの機能があり、これら機能を標準化することでBSWを協調(非競争)領域、制御ソフトウェアを各社製品の差別化(競争)領域と位置づけている。

インターフェース仕様を標準化しているためハードウェアの変更による影響を吸収することや、制御ソフトウェアを複数の開発プロジェクトで再利用できるという製品展開時のメリットがある。

このようなことから、BSWを導入することで制御ソフトウェアの開発に注力することができるようになり、品質と開発効率の向上が期待できる。

### 3)組込みLinux

OSSとして開発されてきたLinuxは無償で利用でき、豊富なライブラリが存在するため製品開発期間を短縮することができる。また、OSSコミュニティの中で多くの開発者が継続的に機能および品質向上のための活動を行っているため、機能進化が早く品質も高い。しかし、無保証であるため組込みLinuxを製品に適用する時には、利用者側での確実な品質の確保が必要となる。

## 2.2 これまでの品質向上の取組み

### 1)開発プロセス標準化

Automotive SPICEに準拠した全社標準ソフトウェア開発プロセスの構築を2011年度から開始し、2014年3月に国際認証機関によるAutomotive SPICEプロセス レベル3の認証を取得した。このプロセスを基に、各開発部署の開発のやり方を考慮したプロセスの適応展開を進めている。認証取得時にはソフトウェア開発プロセスのみの対応であったが、その後システム開発プロセス、ハードウェア開発プロセスへと全社標準を拡張してきている。さらに、機能安全、モデルベース開発、セキュリティにも対応できるようなプロセスも構築している(図10)。

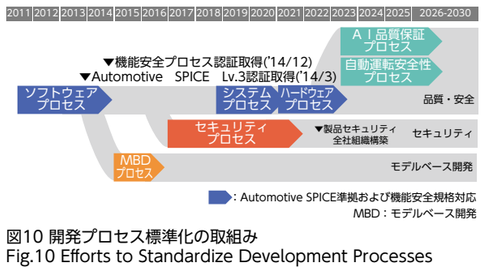



図10 開発プロセス標準化の取組み
Fig.10 Efforts to Standardize Development Processes

全社標準ソフトウェア開発プロセスの構築においては、仕様書や設計書などの帳票(テンプレート)を整備し、作成手順書やチェックリストを用意することで、規格書読解の難しさや個人の解釈ミスによる問題を防ぎ、一貫した品質で開発を進めることができるようにしている。

### 2)車載ソフトウェアプラットフォーム開発

当社はEC開発部設立後の2014年にAUTOSARに加入し、本格的にAUTOSAR対応BSWの開発を開始し、社内標準のBSW(以下、TICO_PF)として製品への適用を行っている。そして、2015年以降、機能安全やセキュリティ要求に対応した機能拡張と、コネクティッド化への対応としてOTA機能の開発を進めている。さらに、自律系システム開発を支える組込みLinux導入のための環境整備に2019年から着手した。今後はこれまで蓄積してきたソフトウェアプラットフォーム技術を活用して統合ECUに適用できる仮想環境の整備を進めていく(図11)。

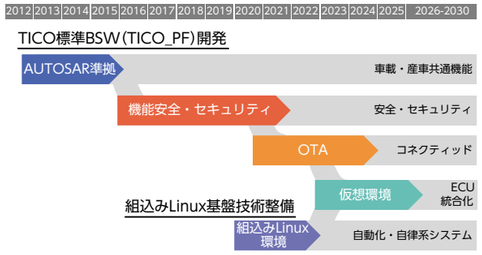



図11 ソフトウェアプラットフォーム開発の取組み
Fig.11 Efforts to Develop Software Platforms

(1)TICO_PFの開発

当社製品に共通的に使用されるマイコンや機能を絞り込み、AUTOSAR仕様をベースに開発を行っている。AUTOSARは自動車向けの仕様であるため、産業車両用の仕様に関してはAUTOSAR仕様に設計を合わせることで、自動車・産業車両共通に使える構成としている。

TICO_PFの品質基準に関しては、全社標準ソフトウェア開発プロセスをベースに開発し、MISRA-C:2012、CERT-C:2016への準拠、およびカバレッジテスト[*13]においてMCDC[*14]網羅率100%などを設定し、これらの基準をクリアした状態で社内リリースしている。また、主要自動車メーカの品質監査基準もクリアしており、これらの活動を通じて得られた知見も含め、当社製品の効率的な品質確保に貢献している。

(2)組込みLinux基盤技術整備

組込みLinuxを実装した製品開発特有の課題に対する基盤技術整備を行っている。

**① ディストリビューション[*15]の選定**

非常に多くのディストリビューションが存在するが、不具合修正や機能追加に関する情報やサポートを迅速に受け取れることが重要であるため、長期保守が可能であり組込み製品やエッジデバイスに適したディストリビューションを選定することにした。

注:＊9 MISRA(Motor Industry Software Reliability Association)-C:自動車分野の安全性を向上させるためのC言語コーディングルール
＊10 CERT(Computer Emergency Response Team )-C:セキュリティを向上させるためのC言語コーディングルール
＊11 AUTOSAR:AUTomotive Open System ARchitecture
＊12 BSW:Basic SoftWare

注:＊13 カバレッジテスト:テスト対象となるソフトウェアのコードのうち、どの程度の範囲がテストされたか(テスト網羅性)を評価するための手法
＊14 MCDC(Modified Condition/Decision Coverage):カバレッジテストの一種。機能安全対応の要求事項となる場合がある
＊15 ディストリビューション:Linuxカーネルをベースにして必要となるソフトウェアを組み込んでパッケージ化したもの

**② 選定したディストリビューションの最適化**

組込み製品は、リソースが限られているため、効率的なメモリ管理やパフォーマンスの最適化が求められる。そのため、当社製品で使用する必要な機能だけに絞ってカスタマイズを行うことで複雑性を解消させながら、使用するリソースを最小限に抑えた組込みLinux環境を整備した。

**③ 組込みLinux設計ガイドライン**

Linuxを採用したシステムでは、動的メモリ利用時のメモリ解放漏れ(メモリリーク)や突然の停電などによるファイルシステム破損といったLinux特有の問題が起こりうるため、これらの課題に対応するための設計ガイドラインを策定した。

**④ 組込みLinux社内コミュニティサイト**

組込みLinuxを用いた開発では、最適化のためのカスタマイズの手法が重要であり、これらを間違えると製品品質に悪影響を与えることがある。そのため、社内の開発者間で専門的な製品化のノウハウを共有し、情報交換できるように、社内コミュニティサイト「Ticopedia」を立ち上げ、整備してきたさまざまなノウハウを部門サイトで全社公開している。

**3) OSS活用ガイドライン構築**

OSSを活用した製品開発では、訴訟リスクがあるため、OSSの活用状況を把握し、ライセンス規約を遵守した上で、ソフトウェア品質を担保する必要がある。そのため、OSSのライセンスコンプライアンスの標準化を行うためのOpenChainプロジェクトで得られた知見を基にOSSを活用する際に必要な活動(図12)を明確にしたOSS活用ガイドラインを策定した。このガイドラインは知的財産部の部門サイトで全社公開している。

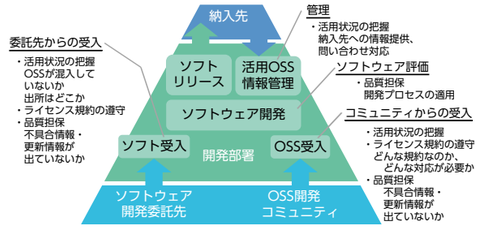



図12 OSS活用時に必要な活動
Fig.12 The Necessary Activities for Utilizing OSS

また、OSS活用時の受入確認作業は膨大なソースコードの調査や、外部から広く不具合情報を収集する作業に多くの工数が必要となる。そこで、この確認作業を効率化するために、OSS管理ツールの整備を行った(図13)。

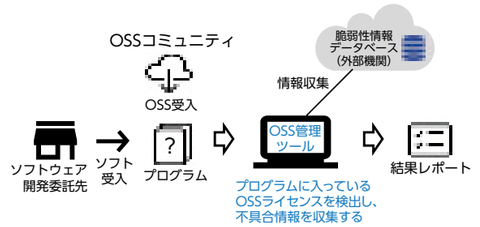



図13 OSS管理ツール
Fig.13 OSS Management Tool

OSS管理ツールは当部の部門サイトで全社公開しており、導入支援も行っている。

## 2.3 今後の品質向上の取組みについて

**1) AI品質プロセス**

AI(人工知能)技術の進歩により、自動化・自律系システムなどのさまざまな分野にAI技術が活用されてきている。そのため、これらのAI活用製品に対する品質保証の必要性が高まり、説明責任や不確実性への対処を目的として、AIシステムの機能安全性を向上させるためのガイドラインとしてISO5469などの規格化が進められている。今後、安全な製品開発のために、国立研究開発法人産業技術総合研究所が公開しているAI品質ガイドラインなどを参考に当社のガイドライン構築を行い、品質保証するための解決手順を提供していく。また、AI品質プロセスに関連する規格化に合わせ、全社標準ソフトウェア開発プロセスを拡張していく。

# 3 安全性向上

## 3.1 機能安全について

電子制御システムの高機能化/複雑化に伴い、異常や故障に対する更なる安全方策の確立が必要になり、機能安全プロセスが策定された。機能安全とは、製品やシステムが意図された機能を安全かつ信頼性高く実行するための取組みのことである。特に複雑なシステムにおいて、部品の故障が車両システム全体に波及しないようにするため、重大な事故や障害のリスクを最小限に抑え、システムの安全性を確保することを目的としている(図14)。

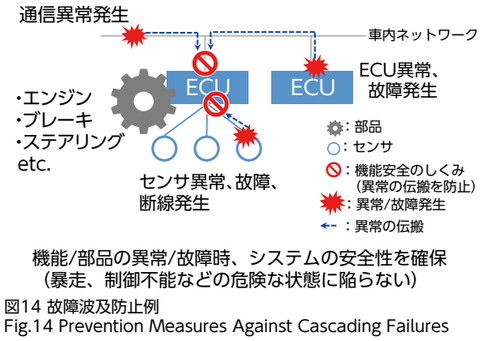



図14 故障波及防止例
Fig.14 Prevention Measures Against Cascading Failures

機能安全規格として、ISO26262(自動車分野)や欧州地域規格EN1175(産業車両分野)などが定められている。機能安全規格には、機能安全に対応するための開発プロセスや要件に関するガイドラインが含まれており、製品やシステムの設計・開発において必要な手順が定められている。また、ISO26262では安全分析手法に基づいた自動車用安全度水準(以下、ASIL[*16])(水準はA ～ Dまであり、Dが最も高い水準である)が定義されており、ASILに応じた安全設計手法と開発エビデンスの作成が必須となる。EN1175に関しても同様の対応が必須となり、欧州では法規要求事項としている国もある。

## 3.2 これまでの安全性向上の取組み

**1) 開発プロセス標準化**

ISO26262の要求事項に準拠した機能安全対応全社標準ソフトウェア開発プロセスを構築し、2014年12月に国際認証機関による機能安全プロセス認証を取得している。プロセスの構築にあたっては、Automotive SPICEに準拠した全社標準ソフトウェア開発プロセスに機能安全で要求されるプロセスを追加し拡張する形で構成した(図15)。


図15 追加した機能安全プロセス
Fig.15 Additional Functional Safety Process

機能安全対応全社標準ソフトウェア開発プロセスを用いて、各開発部署の開発プロセスに合わせた形で機能安全規格への適用支援に加え、機能安全アセスメントの実施を行うなど、開発部署の機能安全活動全体をサポートをしている。

現在は、電気回路等のハードウェア故障に対する機能安全対応に関する開発部署からの支援要望の高まりを受けハードウェア開発プロセスの構築を進めている。

**2) 車載ソフトウェアプラットフォーム開発**

(1) TICO_PFの開発

ソフトウェアでの機能安全対応として、故障の監視(検出/通知)と故障による影響範囲を抑えるための保護機能を実装している。表1に主な機能安全対応のBSW機能を示す。また、代表例として、メモリアクセス監視・保護機能の詳細を図16に示す。

注:＊16　ASIL(Automotive Safety Integrity Level):自動車用安全度水準

表1 機能安全対応BSW機能
Table1 BSW Functions of 'Functional Safety'

| 機能 | 異常 | 代表的な機能の概要説明 |
|---|---|---|
| 実行タイミング監視・保護 | 実行時間超過 | 規定実行時間超過の監視、処理順序監視・保護 |
| メモリアクセス監視・保護(図16) | 不正メモリアクセス | 書き込み禁止メモリ領域へのアクセス監視・保護 |
| ハードウェア故障監視・保護 | ハードウェア故障 | マイコン・周辺機能／監視機能を用いた保護、故障検出・通知 |

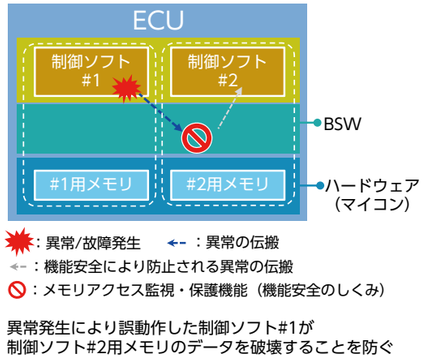



図16 メモリアクセス監視・保護
Fig.16 Memory Access Monitoring and Protection

なお、ハードウェア故障監視・保護機能についてはAUTOSARでは仕様化されていないため、TICO_PFの独自機能として、マイコンに備わっているメモリプロテクション、エラー訂正回路などを活用して設計・実装している。

(2) 組込みLinux基盤技術整備

LinuxはOSSであるため開発エビデンスが十分整っておらず、製品に要求されるASILの要件を満たせない場合がある。このことから、ASIL要件を満たすLinux適用製品を開発していくためには、機能安全認定を受けたLinux互換OSを採用していく必要があると考え、互換OSに関する調査および評価を行った。評価としては、移植性や処理遅延など制御ソフトウェア開発に影響するLinuxインターフェースとの互換性および実行性能確認を実施することとした。これら調査と評価の結果は製品開発時に採用する際の検討に役立つように、Ticopediaで社内公開している。

## 3.3 今後の安全性向上の取組みについて

### 1) 非故障時の安全性対応

自動運転や先進運転支援など、高度な自動化技術を備えたシステムでは、これまで人が行っていた操作をシステムが担うようになってきており、人とシステムの役割が変化してきている。これに伴い、認知・判断などを行うシステムが持っている性能の限界を超えた状態での動作や、人が誤った操作をした場合の動作などに関する安全性など、故障時の安全を担保する機能安全規格ISO26262ではカバーできていない“故障によらないリスク”に対する安全性基準(非故障の安全性)に関するフレームワークの定義が必要となった。このことを背景とし、意図した機能の安全性(SOTIF*17)の規格としてISO21448が2022年6月に発行された(図17)。

ISO21448では天候や道路状況などの様々な条件下での機能の十分性や、運転時の誤操作などに対する安全性に関する製品ライフサイクル(企画、開発、生産、運用)にわたってのプロセスを定義している。全社標準ソフトウェア開発プロセスには製品ライフサイクルのうち、開発フェーズに対応したプロセスを追加していく。


図17 自動車の安全規格
Fig.17 Automotive-Safety Standards

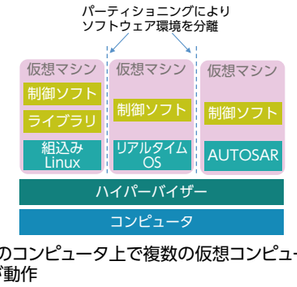
また、ISO26262の改訂第3版も予定されているため、ISO21448と合わせてJASPARなどを通して情報の収集に努めていく。

### 2) OSSの機能安全対応

OSSの機能安全フレームワークとして、ISO-PAS8926*18やELISA*19プロジェクトなどで安全性基準が策定されている。これを受けて、これら国際規格や業界標準を活用することにより、安全性を確保するためのノウハウを蓄積し、OSSを活用する製品開発に適用できるガイドラインを策定していく。

### 3) ECU統合化対応

既存ソフトウェアの統合を実現するために、ハイパーバイザー*20を活用した仮想環境の構築を行う。これにより、仮想マシン間のパーティショニング*21が確保されるため一つの仮想マシンの不具合やセキュリティ攻撃に対する他の仮想マシンへの影響を与えることなく、システム全体の安全性およびセキュリティの向上を図ることができる(図18)。




一つのコンピュータ上で複数の仮想コンピュータやOSが動作

図18 ハイパーバイザー型の仮想環境
Fig.18 Hypervisor type virtual environment

## 4 セキュリティ向上

### 4.1 サイバーセキュリティについて

遠隔からの不正なアクセスによるデータ盗聴や車両制御の乗っ取りなどの深刻な問題を引き起こすサイバー攻撃に対処するために、安全性と信頼性を確保するための新たな枠組みが世界的に求められるようになった。そのため、国際連合の自動車基準調和世界フォーラム(WP29)は、2021年に車両のサイバーセキュリティに関する国際基準UN-R155を発効した。さらに、サイバーセキュリティの国際規格であるISO/SAE 21434が2021年8月に発行され、UN-R155で実施すべき具体的な遵守事項が示された。

ISO/SAE 21434は、製品ライフサイクル全体(企画、開発、生産、運用、保守、廃棄)にわたってのプロセスを定義している。

UN-R155を法規として施行した国においては車両を生産・販売するために、サイバーセキュリティ認証を受ける必要がある。具体的には、国土交通省などの許認可権を持つ政府認証機関による審査を受け、法規を遵守していることが認められた場合に認可を受けることができる。

サイバーセキュリティ認証はCSMS*22認証と車両型式認証とで構成されている(図19)。

【CSMS認証】

サイバーセキュリティを確保できるプロセスおよび体制が整備されていることを審査し、3年ごとに認証を更新する必要がある。

認証の対象は自動車メーカであるが、自動車メーカはサプライヤがISO/SAE 21434に準拠した活動が行えることを確認する責務がある。このため、サプライヤも自動車メーカと同等のしくみを有することが必要である。

【車両型式認証】

これまでの車両型式認証手順に加えて、CSMS認証を取得したプロセス通りに開発した開発エビデンスおよび、実施したサイバーセキュリティ対策とその有効性、対策の評価結果を審査する。

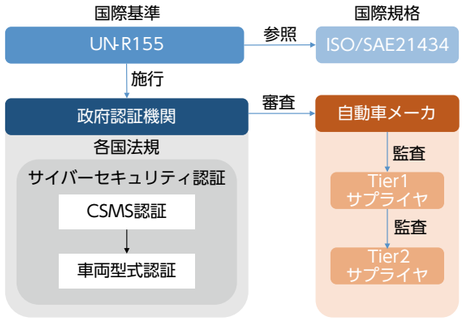



図19 自動車のサイバーセキュリティ法規
Fig.19 Automotive Cybersecurity Regurations

注：＊17 SOTIF：Safety Of The Intended Functionality

注：＊18 ISO-PAS 8926：ISO26262に従って開発されていないLinuxなどの既存ソフトウェアの認定フレームワーク
＊19 ELISA(Enabling Linux in Safety Applications)：Linux Foundationが発足したLinuxの機能安全を推進するプロジェクト
＊20 ハイパーバイザー：物理的なコンピュータ上で複数の仮想マシンに物理リソースを割り当てることで独立して動作させる仮想化技術
＊21 パーティショニング：複数のソフトウェア環境を分離し、それぞれの環境が互いに影響を与えることなく動作できる状態
＊22 CSMS(Cyber Security Management System)：サイバーセキュリティ管理システム

## 4.2 これまでのセキュリティ向上の取組み

### 1)開発プロセス標準化

サイバーセキュリティ対策を講じるため、ISO/SAE 21434の開発フェーズの要求事項に準拠したセキュリティ対応全社標準ソフトウェア開発プロセスの構築を2021年に完了した。

プロセスの構築にあたっては、機能安全対応プロセスとの整合性を保つよう、これまでの機能安全対応全社標準ソフトウェア開発プロセスにISO/SAE 21434で要求されるプロセスを追加する形で構成している(図20)。


図20 追加したセキュリティプロセス
Fig.20 Additional Security Processes

また、構築したセキュリティ対応全社標準ソフトウェア開発プロセスを用いて、各開発部署の開発プロセスに合わせた形でセキュリティ規格や法規への適用を進めている。そのため、開発部署の脆弱性分析活動にアドバイザーとして参加し、より網羅的な分析を行うなどセキュリティ対策活動全般にわたり開発支援を行うとともに、開発者の育成を行っている。

### 2)車載ソフトウェアプラットフォーム開発

ソフトウェアプラットフォームに実装しているサイバーセキュリティ対策のための主な機能を以下に示す。

(1)通信メッセージのセキュリティ

なりすましなどの不正アクセスを防止するため、車両内の通信メッセージに認証コードを付加する。受信側ECUでは認証鍵を使って、正しいECUからのメッセージかどうかを確認し、セキュリティを確保する(図21)。

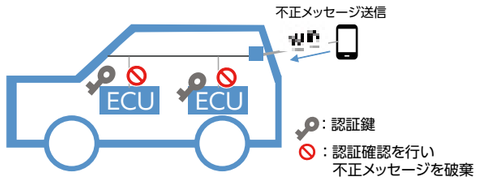



図21 通信メッセージのセキュリティ(認証)
Fig.21 Communication Messeage Security (Authentication)

また、データの盗聴を防ぐために、通信メッセージの暗号化についても開発している。

認証・暗号をマイコンで実現する方法には、マイコンに搭載された専用の処理回路を使う方法と、ソフトウェアで処理する方法があり、TICO_PFでは両方に対応している。ソフトウェア処理の方法は、マイコンを変更する必要がなく、すぐに現行システムに実装することができる利点があるが、処理速度が遅いことが課題であった。これに対してTICO_PFでは軽量暗号方式を採用し処理の軽量化を図ることで解決した。

認証鍵や暗号鍵は、外部からのアクセスができないように管理しなければならない。また、鍵管理の方法自体も秘匿性が求められるため、各社それぞれ独自の方法で定義している。当社ではJASPARの考え方を基本にして、独自の秘匿性の高い鍵管理仕様を定義した。

(2)ECUソフトウェアのセキュリティ

ECUのソフトウェアアップデート時のプログラムの改ざんを防ぐため、セキュアリプログラミング機能を開発した。また、ソフトウェアアップデート中に予期せず電源が切れるような問題が起こった場合でも、更新前のプログラムを保持しておくことで、元の状態に戻せる機能を追加し、安定性を保つ設計にしている。

セキュアリプログラミングは、サービスツールを使った有線接続方式と無線接続方式(OTA)(図22)の両方に対応している。有線接続方式は、当社の自動車分野や産業車両分野の両製品で広く使われており、今後OTAも導入されていく予定である。

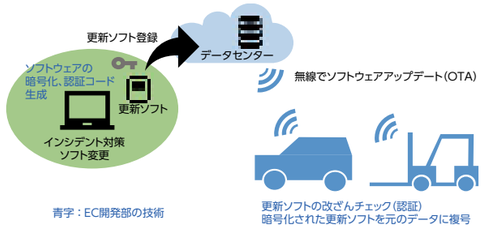



図22 ECUソフトウェアのセキュリティ(OTA)
Fig.22 ECU Software Security (OTA)

### 3)製品セキュリティ活動

サイバーセキュリティ認証では、製品の運用・保守フェーズにおいても対策が求められる。そのため、製品販売後に新たな攻撃方法が出てきた場合に対応できるように、脆弱性情報の収集・共有・分析・対策を行う全社組織PSIRT*23をITデジタル推進部、品質統括部と立ち上げた。図23にPSIRTの体制、図24にPSIRTの役割と責任を図示する。




開発後のインシデント活動としてPSIRTを中心とした体制を構築

図23 PSIRT体制
Fig.23 PSIRT Framework

| 担当 | 役割 | 責任 |
|---|---|---|
| PSIRT | インシデント情報社内展開・全社規定管理 | [脆弱性情報収集]<br>フィールド監視&トリアージ<br>新しい脆弱性が出荷済みの製品に影響ないかを確認する<br><br>[情報提供]<br>脆弱性情報の提供<br><br>[規定管理]<br>インシデント対応関連全社規定の維持管理<br><br>[社内状況把握]<br>事業部内の展開状況・対応状況を把握し、必要な対応を進言する |
| 事業部 | 法規・規格動向の開発計画への織り込み | [インシデント対応]<br>万が一、対策前の脆弱性が悪用されてしまった場合の対応<br>・顧客、取引先への情報公開<br>・市場製品の改修<br>・開発へのフィードバック |

図24 PSIRTの役割と責任
Fig.24 The Roles and Responsibilities of PSIRT

脆弱性情報収集では、ITデジタル推進部が当社製品に影響を及ぼす可能性がある脆弱性情報を抽出し、当部にて情報の分析・見える化を行い開発部署へ展開している。

## 4.3 今後のセキュリティ向上の取組みについて

### 1)ソフトウェア管理システムの整備

ソフトウェアアップデートを実施する場合に必要な情報の提供や管理方法に関する国際基準としてUN-R156が制定された。具体的な遵守事項としては自動車のソフトウェアアップデート規格ISO24089で規定されている。ソフトウェア管理システムとしては、今後特にOSSや商用ソフトウェアを対象とした脆弱性に関するリスクを特定するためのしくみが重要となってくる。これに対応するためJ-AUTO-ISACのサブワーキング活動に参加し、OSSや商用ソフトウェアを対象としたしくみの構築と運用ガイドラインの検討を進めている。

### 2)進化するサイバー攻撃への対策

車両内の通信メッセージを監視し、不正メッセージの侵入などのインシデントの兆候を検知し、新たな攻撃への対策を早急に行っていくためのしくみとして侵入検知技術を確立していく(図25)。

注:*23 PSIRT(Product Security Incident Response Team):製品のインシデント対応組織

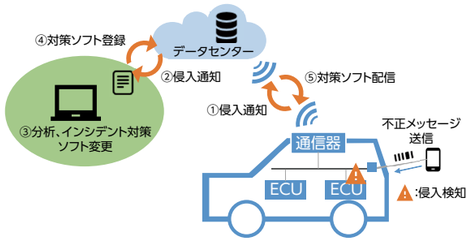



図25 侵入検知技術を活用したインシデント対策例
Fig.25 Example of Incident Countermeasure Using Intrusion Detection Technology

最新のサイバー攻撃対策に関しては、現在参加しているJASPARの情報セキュリティワーキングでの標準化活動で、世の中の最新動向を把握し対策を進めていく。

## 5 組込みソフトウェア技術者育成の取組み

### 5.1 エキスパート新人教育

組込みソフトウェア技術者の育成・強化のため、新入社員の中から将来のソフトウェア開発を主導する人材(エキスパート新人)を集め、組込みソフトウェア開発の教育をした後、各開発部署への配属を行っている。

**1)教育体系**

教育カリキュラムはETSS*24のフレームワークをベースとし、当社製品の開発に必要な技術内容にアレンジし策定した。このカリキュラムは、常に最新の技術動向に合わせて見直されるとともに、教育の実施状況や育成結果を元に継続的に改善されている。

教育カリキュラムは、組込みソフトウェア開発実務者に必要かつ基礎的な知識習得を目的とする座学による「基礎教育」「専門教育」と、そこで学んだ知識を腹落ちさせるために実際に開発を経験させ確実に身に付けさせることを目的とする「実践教育」と「OJT教育」で構成されている(図26)。

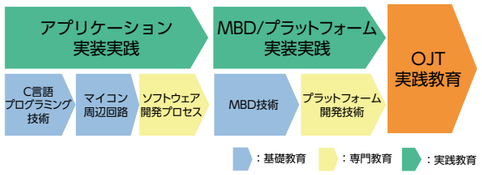



図26 エキスパート新人教育
Fig.26 Training for New Recruits to Become Experts

実践教育はアプリケーション開発とMBD/プラットフォームをテーマとしており、Automotive SPICEに準拠した開発プロセスに従ったソフトウェア開発スキルを習得させている。また、チームで開発をすることにより、メンバーとのコミュニケーションスキルや与えられた日程で課題を達成していくためのマネジメントのスキルも積み上げられるようにしている。

教育講師としては、主に当部の現役技術者が担当することで、実践的な技術の使い方についても直接学ぶことができる。

こうした教育カリキュラムで身に付けたソフトウェア開発と開発プロセスのスキルを、実際の開発現場で行う「OJT実践教育」でさらに現場力として高めた上で全社の開発部署へ配属させている。これらの教育体系を取ることにより短期間で、即戦力となる組込みソフトウェア技術者として活躍できるレベルまで育成している。また、ソフトウェア開発プロセスを理解し、幅広い技術力を持った人材を各開発部署に送り出すことで全社的に正しいソフトウェア開発の方法を広める役割も担っている。

**2)情報共有会**

エキスパート新人と育成後に開発部署に配属された社員(卒業生と呼ぶ)を一堂に集めた情報共有会を当社のグローバル研修センター(幡豆アカデミー)で毎年開催している。

情報共有会の主な目的は次のとおりである。

- 卒業生がお互いの業務内容を共有することで、自業務のヒントとする。また、協業の可能性を探る。
- 他部署の開発テーマを知ることで、組込みソフトウェア技術者としての知見を広げる。
- エキスパート新人の配属先検討に向けた、情報収集／提供の場として利用する。
- エキスパート新人の縦(先輩後輩)と横(同期)のつながりを維持・強化することで、技術交流の円滑化を図り、会社全体のソフトウェア開発スキルを向上させる。

写真1 情報共有会
Photo1 Information Sharing Meeting Among Former Trainees

また、情報共有会後にお互いの親睦を深めるため、車座で懇親会を開催している。

### 5.2 中堅技術者向け教育

**1)全社向け教育講座**

当社の技術技能ラーニングセンターでは、全社の技術者向けに様々な教育講座を開催している。その中で、組込みソフトウェアに関する専門的な知識が必要な講座に関しては、当部がカリキュラムの企画から教材開発、講座開催までを担当している。以下に当部主催の講座概要を示す。

(1)ソフトウェア開発プロセス系

エンジニアリングプロセス群、およびプロジェクト管理プロセス、サプライヤ監視などプロセス毎に講座を開設。

グループディスカッションを活用することにより各開発部署の好事例となる開発活動を受講者全員と共有し、同時にプロセス改善の機会を提供している。

(2)リアルタイムOS系

知識の定着と実践スキルを身に付けることを目的として、座学と演習を組み合わせて実施している。

(3)OSSライセンス系(知的財産部と共同開催)

初級編、開発者編、審査者編の3つのレベルに講座を分けることで、受講者が必要なレベルに応じて受講できる。

講座終了後の受講者アンケートに基づく改善と、技術の進歩に合わせた最新の専門的な知識・技術を習得できるように内容を見直すことで毎年教育の質を向上させている。

**2)リスキリング教育**

ソフトウェア開発の重要性の高まりを受け、今後の組込みソフトウェア技術者強化の取組みとして、エキスパート新人教育に加えさらに、機械・電気系など非ソフトウェア開発領域に従事する中堅技術者を対象とした、リスキリング教育の体制を構築していく。

エキスパート新人教育は組込みソフトウェア開発実務者の育成を中心としたカリキュラムを構築しているが、リスキリング教育においては、より上流工程となるソフトウェア要件分析やプロジェクト管理などの教育カリキュラムを充実させる計画である。

これにより、中堅技術者自身が持つ専門技術をコアとしたソフトウェア開発領域へ業務範囲を拡大し、組込みソフトウェア技術者の更なる強化を目指す。

## 6 まとめ

自動車への電気・電子部品の搭載が進み、ハードウェアだけでなくそれを制御するソフトウェアを含んだ不具合によるリスクを許容可能なレベルに抑えることを目的として、2011年11月に自動車用の機能安全規格ISO26262が発行されることになった。ISO26262でソフトウェア品質が本格的に要求されるようになったことを契機とし、当社でもソフトウェア技術の強化が重要課題となり、それを解決するためにEC開発部が設立された。EC開発部ではソフトウェアの品質を高めるべく、開発プロセスやソフトウェアプラットフォー

注:*24 ETSS(Embedded Technology Skill Standard):独立行政法人情報処理推進機構が策定した、組込みシステム技術者認定スキル標準

ムを整備し、各開発部署に提供するとともに、開発した基盤技術を普及させるための人材育成を同時に進めてきた。エキスパート新人教育の卒業生は現在では100人を超え、各開発部署のソフトウェア開発の中心となって、当社のソフトウェア開発を支えてくれている。

また、本文中で説明したように、ソフトウェアへの課題も品質確保から機能安全、さらにはセキュリティ対応など開発すべき基盤技術の重要性や規模がより一層増大してきている。今後は、自動車が現在のスマホ以上の機能を持ち、販売後もOTAでのソフトウェアアップデートによる新機能追加が当たり前になろうとしている。従来は、ハードウェアに組み込んで一緒に製品化されてきたソフトウェアも単独で部品化・製品化され、現状よりもさらに重要な位置づけとなってくる。最先端を走っているテスラでは、OTAにより定期的にソフトウェアでの機能追加が行われ、自動運転ソフトウェアが高価格で販売されているという事実もある。このような来るべきSDV時代に向け、ソフトウェアの重要性はさらに高くなり、最先端のソフトウェア技術なしでは製品開発が成り立たなくなるであろう。我々はソフトウェア技術者の更なる増強とモデルベース開発などの開発手法を含む最先端の基盤技術開発を進め、各事業部とともに製品を開発し、当社の事業拡大をソフトウェア技術で支えていきたいと考えている。

**■著者紹介■**



宮田 八郎


小澤 尚之


前田 洋信


福田 仁志


後藤 宏之


藤井 英樹


小林 貢


犬塚 浩之