

Veriserve Navigation

ベリサーブナビゲーション

進化するモビリティと、
深化する品質創造。

ベリサーブ オートモーティブ カンファレンス 2021
レポート Vol.2



進化するモビリティと、 深化する品質創造。

ベリサーブ オートモーティブ カンファレンス 2021 レポート Vol.2

CONTENTS

特集1 06

「電費」時代の自動車産業～クルマは仮想空間で稼ぐ

中山 淳史氏

特集2 14

これからの車載ソフトウェア開発と品質保証

森崎 修司氏

特集3 22

車載システムの分散・分担開発における
プロジェクトマネジメントの課題解決に向けて

冬川 健一

特集4 30

車載セキュリティテストに関するベリサーブの取り組み

長谷川 邦幸

特集5 38

トレーサビリティ管理ツールをもっと実践的に使いこなすには
～規格対応のため「だけ」から、品質・生産性向上のために～

横田 浩行



Top
トップページ

Conference Hall
カンファレンス会場

Exhibition Booth
展示ブース

FAQ
よくある質問

Overview
開催概要

進化するモビリティと、深化する品質創造。

Veriserve Automotive Conference 2021

ベリサーブ オートモーティブ カンファレンス

Concept Movie



オンライン開催

2021 6.24 Thu 10:00 ▶ 7.14 Wed 18:00 視聴 無料

進化するモビリティと、深化する品質創造。

Veriserve Automotive Conference 2021



ベリサーブ オートモーティブ カンファレンス

ベリサーブ オートモーティブ カンファレンス 2021 レポート

品質への挑戦を続けるベリサーブは、2021 年 6 月 24 日から 7 月 14 日までの 21 日間、「ベリサーブ オートモーティブ カンファレンス 2021」をオンラインで開催しました。ベリサーブナビゲーションでは、このカンファレンスでの講演の模様を 2 号にわたってお届けします。

講演プログラム

モビリティ産業はいま、百年に一度と言われる変革期を迎えています。MaaS/CASE といった言葉に代表されるように、過去に例のないサービスやプロダクトが世の中に登場してはさらなる進化を続け、その開発手法もアップデートを繰り返しています。そうした時代に求められる「品質」とはどのようなものなのでしょうか。

本カンファレンスでは、「進化するモビリティと、深化する品質創造。」をテーマに、MaaS/CASE を前提とした世の中の動きや社会的インパクト、新時代のモビリティに求められる品質保証の考え方など、第一線の有識者・技術者がそれぞれの知見を披露しました。

前号（23 号）に続き、今号（24 号）でご紹介する講演プログラムは右ページの通りです。



Atsushi Nakayama



Shuji Morisaki



Kenichi Fuyukawa



Kuniyuki Hasegawa



Hiroyuki Yokota



Keiichi Yoneya



Hidetoshi Suhara



Hirokazu Igarashi



Kazuhiko Makimura

Vol.24

基調講演

「電費」時代の自動車産業～クルマは仮想空間で稼ぐ

中山 淳史 氏 | 株式会社日本経済新聞社 コメンテーター

これからの車載ソフトウェア開発と品質保証

森崎 修司 氏 | 名古屋大学 大学院情報学研究所 准教授
株式会社ティアフォー 技術顧問

技術講演

車載システムの分散・分担開発における
プロジェクトマネジメントの
課題解決に向けて

冬川 健一 | 株式会社ベリサーブ

車載セキュリティテストに関する
ベリサーブの取り組み

長谷川 邦幸 | 株式会社ベリサーブ

トレーサビリティ管理ツールを
もっと実践的に使いこなすには
～規格対応のため「だけ」から、
品質・生産性向上のために～

横田 浩行 | 株式会社ベリサーブ

Vol.23

基調講演

自動運転社会の実現に関する取組について

五十嵐 大和 氏 | 総務省 総合通信基盤局電波部移動通信課
新世代移動通信システム推進室長 高度道路交通システム推進室長(当時)

基調講演

MaaSが都市を変える～移動×都市DXの最前線～

牧村 和彦 氏 | 一般財団法人 計量計画研究所 理事 兼 研究本部 企画戦略部長

技術講演

車載ECUの開発現場から見たテストと自動化の現状

米谷 恵一 | 株式会社ベリサーブ

技術講演

車載ドメインにおけるモデルベースドテストの実践例のご紹介

須原 秀敏 | 株式会社ベリサーブ



「電費」時代の自動車産業 ～クルマは仮想空間で稼ぐ



中山 淳史氏
なかやま あつし

株式会社日本経済新聞社
コメンテーター

1989年上智大外国語学部卒、日本経済新聞社入社。編集局産業部、米州総局（ニューヨーク）、証券部などを経て2010年に産業部編集委員兼論説委員、13年から企業報道部次長兼編集委員。自動車、電機、運輸などを担当。17年2月から現職。

はじめに

長い間、自動車と電機は日本経済を支える二本足と言われてきました。一部を除き凋落傾向にある電機業界と比べ、自動車産業は今も隆々とした勢いを保っていますが、昨今ではここにも大きな荒波が押し寄せてきています。これからの時代、クルマは単に走るだけではなく、走りながら稼ぐものになっていきます。これが今後の自動車産業をどう変えていくのか、世界と日本の動向を踏まえながら考察してみたいと思います。

本講演では、

- 変わる「ものさし」
- 覇権をめぐるゲームチェンジ

という二つのポイントからお話させていただきます。



燃費から電費へ-変わる「ものさし」

》 47万円の「ウーリン」

昨年夏、中国の中堅自動車メーカー、上汽通用五菱汽車が「ウーリン」ブランドより「宏光 MINI EV」という小型電気自動車（EV）を発売しました（図1）。発売直後の瞬間風速ではあるものの、米テスラ社の Model S を抜き中国で最も売れたEVとなりました。47万円という低価格ながらよくできていて、1回の充電で120km程度は走るそうです。日本の軽自動車よりもやや小ぶりのサイズですが、若いカップルの街乗りやMaaSなどのシェアリングサービスには十分で、合理的な中国の消費者からは高く評価されています。



図1 上汽通用五菱汽車の小型EV

引用：上汽通用五菱汽車 Web サイト <https://www.sgmw.com.cn/e50.html>

脱炭素の取り組み

このクルマが登場した背景には、欧米を中心に進む「脱炭素」への動きがあります。これに関しては、以下のようなキーワードがさまざまなメディアで頻出しています。

- カーボンニュートラル
- カーボンフリー
- カーボンプライシング
- グリーンエコノミー
- ライフサイクルアセスメント

カーボンニュートラルとカーボンフリーは、文字通り脱炭素のことです。カーボンプライシングは価格、つまり二酸化炭素の排出枠取引や炭素税、炭素国境調整措置などをひとくくりにした言葉です。グリーンエコノミーは地球温暖化を防止する経済活動、そしてライフサイクルアセスメントは、部品作りから廃車に至るクルマの一生を通じて脱炭素を目指す取り組みです。こうした状況の中で生まれたのが「宏光MINI EV」というわけです。

電費 = 「走行距離 + α 」

本講演のタイトルにある「電費」、これは従来の自動車産業における「燃費」と対比する言葉です。ご存じの通り、燃費はガソリン1L当たりの走行可能距離を指します。1886年にドイツ・ダイムラー社のカール・ベンツ博士がガソリンで動く自動車を開発、世界で初めて商品化しました。そこから実に135年間、メーカーの競争力を図る指針として、燃費の時代が続いてきたわけです。

では、「電費」とは何か。当然ですが、EVにおいても1回の充電で何km走れるかは重要な要素です。先ほどのウーリンは120kmですが、東京の郊外から都心までの往復ぐらいいはできそうです。しかし、電費が内包する意味はそれだけではありません。そこには、「走行距離 + α 」の新しい価値が存在するのです。

この「+ α 」によって、クルマを評価する「ものさし」が変わります。燃費の時代、ものさしが「1台当たりの利益」だったとすれば、電費の時代には「走行当たりの利益」というものが問われるようになります。通信やITなどの課金型ビジネスを展開する企業の業績では、ARPU（ユーザー平均単価）という

指標がよく使われますが、これは一ユーザーにつき、どれくらいの売り上げがあったかを意味します。これを自動車に置き換えて考えると、ソフトウェアのアップデートによる機能追加などによりクルマを販売した後にどれだけ稼ぐことができるか、そこが重要になってきます。

走行データの売買

実は「+ α 」にはもう一つ別の考え方があります。EVが走る、それ自体が利益を生む可能性があるということです。EVでは、それまでハードウェアが行っていた制御の多くをソフトウェアが行うようになるため、動力源である電気がいつでも作られたか、どこで充電したか、どこからどこまでどのように走ったか、その全てをデジタルデータとして容易に追跡、蓄積することが可能です。メーカーはユーザーに同意を得た上で、このデジタルデータを資産として活用できるかもしれません。例えば、ある区間を走行した時の道路や人通りの状況、立ち寄り先など、メーカー自身だけではなく他の企業も欲しがれる情報は無数にあります。こうしたデータは排出権取引などと同様に売買され、莫大な利益につながる可能性があります。

電費を抑える運転

また、先に示した排出枠と同じ考え方で、電費を抑える運転という行為が経済価値を持つということです。つまり、CO₂を排出しないクルマをさらに効率よく走らせると、その報酬として仮想通貨が支払われる。EVを通じて、こうしたやり取りが仮想空間上で始まる可能性があるのです。

デジタルツイン（電子の双子） ～リアルとサイバーのマーケットプレイス

これを可能にするのが、デジタルツイン技術（電子の双子）です。図2は伊藤忠総研の深尾三四郎氏によって作成されたもので、左に人・モノの輸送、右側にエネルギー・マネジメント、中心にはこれらを巡る脱炭素などの取り組みがあります。面白いのは、この二つの円の上下には現実世界とサイバー空間（仮想空間）があり、同じ事象がサイバー空間上にも存在することです。先述の電費を抑える運転で生じる取引はサイバー空間側で行われることになります。

これをさらに詳しく説明したのが図3です。左が現実世界、右側がサイバー空間ですが、一番下にはどちらの世界も同じクルマが存在します。道路や周囲の街並み、店舗などもサイバー空間に作れば、現実世界と同期させてクルマを走らせることができます。電費を抑える運転のより良い方法をサイバー空間でシミュレート、現実世界にフィードバックし、そこで得た報酬を再びサイバー空間に蓄積する。こうしたやり取りがリアルタイムで行われる、それが電費の時代なのです。

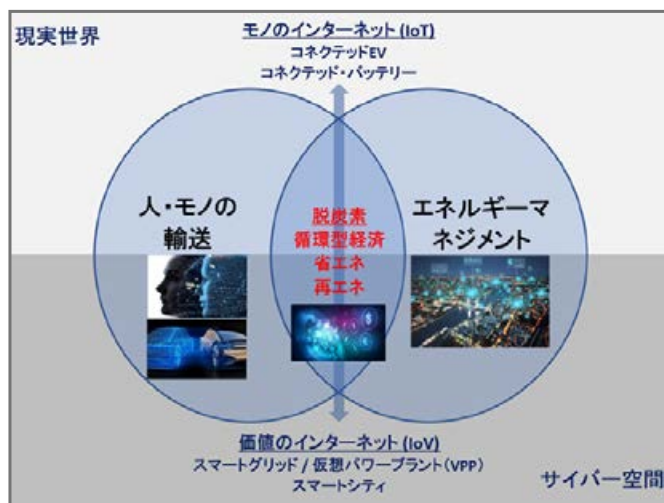


図2 EV・車載電池の提供価値

出所：深尾 三四郎氏より

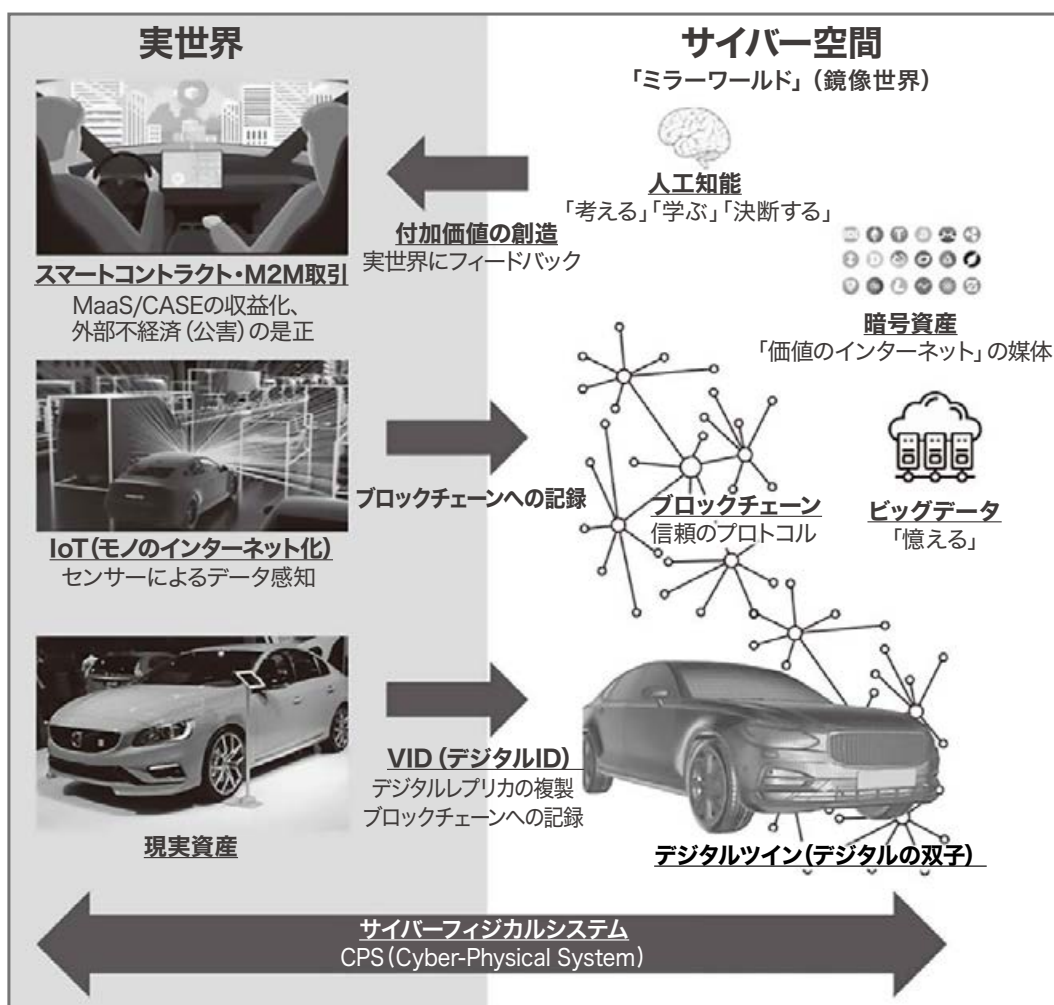


図3 車のデジタルツインと信頼のプロトコル

出典：深尾 三四郎・クリス・バリンジャー（2020）、モビリティ・エコノミクス ブロックチェーンが拓く新たな経済圏 日本経済新聞出版

》 電費の時代はいつやってくるのか？

EVはいつ爆発的に普及すると考えられるのでしょうか？ その普及にはエネルギー効率の高い電池技術が不可欠です。しかし、電池はそう簡単に性能を上げることはできません。現在、主流となっているリチウムイオン電池は、半導体のように1年半に1回の割合で性能が倍増するようなことはまずありません。

ガソリンエンジン車を超えるエネルギー効率を得るためには、ブレークスルー（障壁を突破する革新的な電池技術の開発）が必要とされています。現在、メディアで取り上げられているのは「全固体電池」という技術で、これが確立されると走行可能な距離は何倍にもなると言われています。こうなると、EVは黙っていても（または「必然的に」）普及するでしょう。それがいつになるのか？というのが自動車業界最大の関心事となっています。

自動車や電池業界の予測では、期待も込めて2025年には全固体電池の量産化技術が確立されているとも言われています。しかし、日本の関連企業の投資状況をヒアリングしてみると「リチウムイオン電池」への投資割合が多く、すぐには量産化が実現する時は来ないだろう、というのが私の見解です。ただ、それが少し先、例えば2035年頃になるとしても、のんびりとはしてられません。わずか15年足らずで電費の時代がやってくると考えた方がいいでしょう。



覇権を巡るゲームチェンジ

》 遅れが目立つ日本

ここまでお話しした通り、これからの自動車産業は全く違うものにアップデートされていく可能性があります。これをにらみ、次代の主役の座を狙う国や企業がすでに活発な動きを見せています。脱炭素化に向けた各国および主な企業の動向は図4の通りで、いろいろな思惑が読み取れます。

法制化は少し先になりますが、欧米は2030～35年にはガソリン車を全廃する方針です。中国はマーケットが巨大なこともあってそのスピードが少し緩めで、2035年に全てを省エネ車、新エネルギー車にするとしています。そして日本は2030年代半ばにHVを含めて半数を電動車にするということで、中国の目標のさらに半分となっています。ガソリン車という大きな資産を抱えている事情はあるにせよ、対応が遅すぎるという感はやはり否めません。

企業レベルでは、米ゼネラルモーターズ社が2035年にガソリン車を全廃すると発表しています。この会社はHVが弱いので、EVに完全移行すると見てよいでしょう。自動車販売台数が世界第1位のトヨタは2025年に半数を電動化するとしています。同社はHVも電動車と見なしていて、これを含めての半数です。EV「リーフ」を擁する日産も現在注力しているのはe-POWERというHVの一種となる方式で、発電専用のガソリンエンジンと蓄電池、モーターを組み合わせたものです。両社ともEV化という点では遅れ気味と言えるでしょう。

》 デファクトか、デジュールか

タイムスケジュールではほぼ足並みを揃えている欧米ですが、そこに向かう戦略は分かれる可能性があります。米国が得意なのは「デファクト」、つまり企業間の自由競争によって事実上の標準を生み出すというものです。スマートフォンにおける米アップル社のiPhoneが代表的な例です。

一方、EUが好むのは「デジュール」で、国家が法制度によって標準を決める戦略です。EV関連では「バッテリー指令」、「炭素国境調整」、「炭素税」、「排出枠取引」、こうしたものを法制化し、遵守しない相手とは取引禁止、あるいは高い関税をかける。これを域外に広げつつ、自国企業には補助金で支援し、あわよ

くばEUの製品を世界に拡散しようとしします。WTO（世界貿易機関）協定との整合性や補助金の高騰といった問題をはらむものの、ヨーロッパの戦略はこれが昔からの基本です。

中国の方針はまだ定まってははいないようですが、どうやらEUと協調していく方向のようです。こうした各国の動向に日本がどう対処していくのが、今後の大きな課題になっていくと思われます。

》》 テスラの10年計画

次は、企業の例としてEVのリーダーと言えるテスラの状況を見てみましょう、同社には「マスタープラン」と呼ばれる10年スパンの経営計画があり、図5はその要旨です。

国・自治体	
 アメリカ カリフォルニア州	2035年にガソリン車の新車販売禁止
 イギリス  オランダ	2030年に脱ガソリン車
 中国	2035年に全て省エネ・新エネ車
 日本	2030年代半ばに電動車化（ハイブリッド車：HV含む）、50年温暖化ガス0

自動車各社の計画	
ゼネラルモーターズ	2035年にガソリン車全廃
トヨタ	2025年に半数を電動化
日産	2030年代に新車を電動化

図4 脱炭素化に向けた各国および主な企業の動向

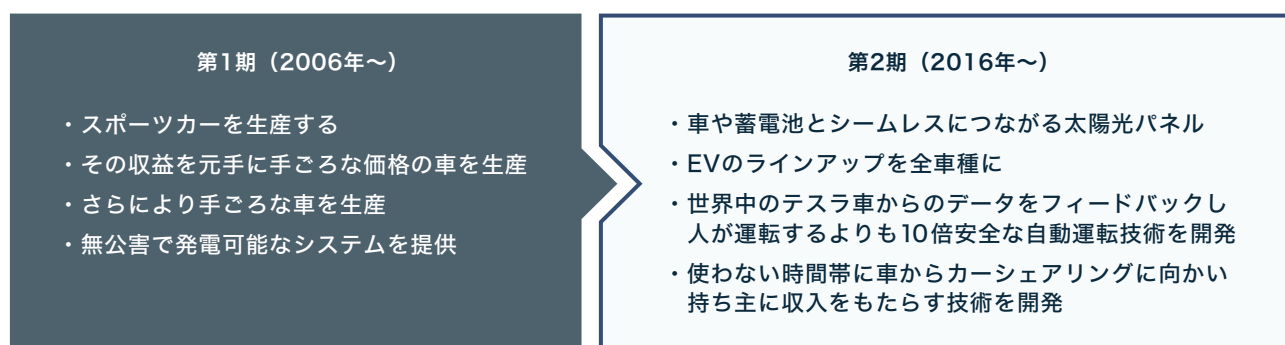


図5 テスラのマスタープラン

（出所）テスラ、アーサー・D・リトル・ジャパン

2006年に発表した第1期、上から3番目までについては、2008年のロードスターを皮切りに、Model S、Model 3と、ほぼ計画通りに進んでいます。4番目がテスラの特徴を表すもので、これは建物の屋根に設置する太陽光パネルを指しています。つまり事業の根幹は持続可能エネルギーのマネジメントで、EVもその一つというわけです。

2016年からの第2期の1番目は先ほどの延長線上で、太陽光パネルの充実です。2番目が車種拡充で、小型車やSUV、トラックなどにもEVを広げていく。そして3番目が先ほど説明した走行データに関わるもので、世界中で走るテスラ車から得たデータを元に、人間が運転するよりも10倍安全な自動運転技術を開発すると言っています。

4番目が非常に興味深いのですが、テスラ車のオーナーがマイカーをシェアリングサービスに登録しておく、空き時間にクルマが自動でお客様のところへ出掛けていき、目的地に送り届け、また家に帰ってくる。要するに、自動運転技術を使ってクルマが勝手に稼いでくれるというわけです。

無人運転が法律で認可されることが前提ですが、第1期の計画が全て達成されたことを考えれば、技術的には2026年までに実現する可能性が高いとみています。おそらく市場もそう判断していて、それが高い株価に反映されていると言えるでしょう。

》アップルの動向

もう一つの注目の企業はアップルで、おそらく自らクルマを作るのではなく、欧州などの企業にOEMで生産を依頼し、そこに搭載するITやサービスで稼ぐビジネスモデルを考えていると思います。彼らはウォレットやデジタルツイン、ブロックチェーンなど、いろいろな技術を持っています。クルマを売った後に稼ぐ、そこだけに賭けようという企業がアップルなのです。

図6は、株式時価総額の比較です。トヨタの時価総額は日本の他の自動車メーカー全部を足しても及ばないほどなのですが、テスラの株価は実にその3倍です。これが何を意味するかというと、仮にテスラが生産能力を増強したいと考えたなら、株式交換を使えば日本の中堅メーカーを買収することが理論上は可能なのです。

いわんやアップルです。時価総額はトヨタの10倍で、この会社がどのようにEVに参入してくるか。固定資産を持たないという方針ゆえに自動車メーカーの買収はないでしょうが、時価総額の実力を見ればさまざまな方策が考えられるでしょう。

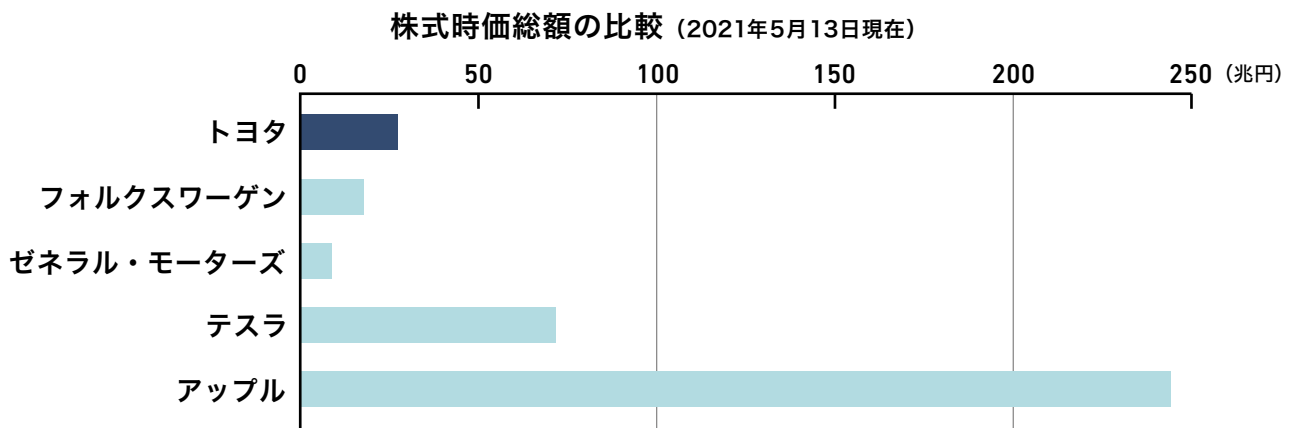


図6 株式時価総額の比較

》 迎え撃つトヨタ

2020年1月にミラスベガスで開催された世界最大の家電・IT見本市「CES 2020」において、トヨタは実証都市「ウーブン・シティ」のプロジェクト概要を発表しました。この計画に当たりトヨタは5000億円という大規模な社債を発行しましたが、1カ月足らずで完売したようです。また、事業母体としてウーブン・プラネット・ホールディングスという持株会社を作り、こちらには豊田章男社長も個人で出資をしています。

豊田社長の弁では、ここからトヨタは新旧に分かれていく。現在のトヨタを旧として、新トヨタはクルマのOSをはじめとしたソフトウェアの開発やサービスを事業として展開し、いずれは新トヨタが旧トヨタの時価総額を抜いて欲しいと言っています。ウーブン・シティはトヨタの新たな技術やデータ活用のショーケースであり、特斯拉やアップルの猛攻を黙って見てはいないということでしょう。

おわりに

図7は、企業の衰退プロセスのイメージで、米国の経営学者であるジェームズ・コリンズ氏が著書で提示したものです。実はトヨタの豊田社長は2009年に日本記者クラブの場でこれを示し、今のトヨタは第二段階にいる、と発言しています。その時期はリーマンショックの直後で、トヨタは世界中で工場の建設ラッシュを進めていたために一気に赤字に転落していました。「規律なき拡大路線」に走ってしまったわけです。

では今はどうなのか。想像するに、第三段階に陥りかけていたのではないのでしょうか。先に述べた通り、他国と比べて日本は国もメーカーも脱炭素には消極的で遅れが目立ちます。EVの

衰退の五段階 (イメージ)

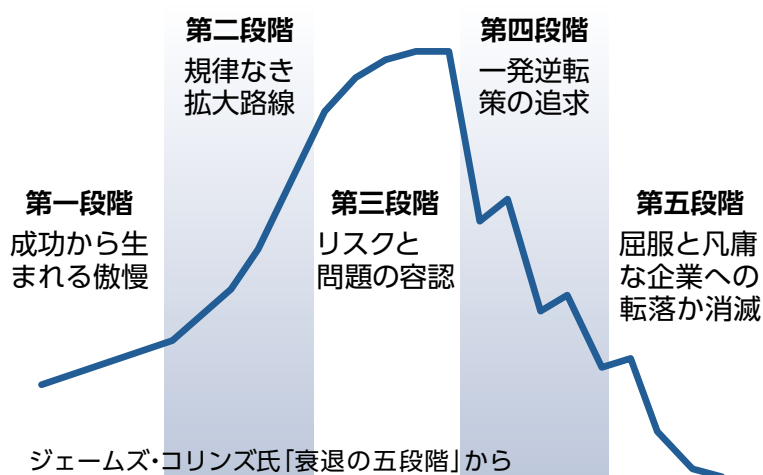
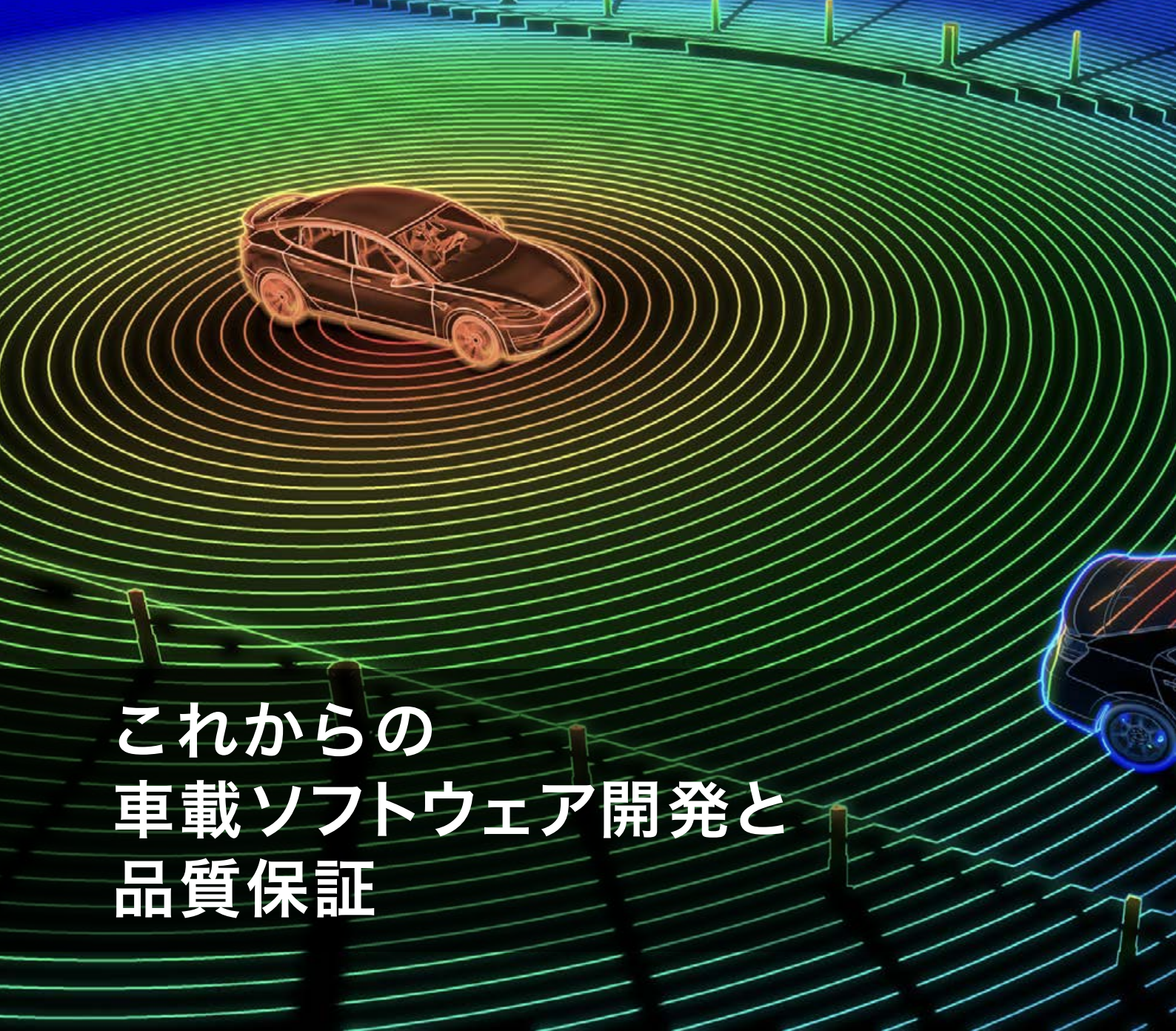


図7 米国の経営学者、ジェームズ・コリンズ氏が提唱する「衰退の五段階」のイメージ図

時代など簡単には来ない、売った後に稼ぐなどありえない、そんなふう考えていた節があります。まさに「リスクと問題の否認」です。豊田社長は明確には述べていませんが、ウーブン・シティや新旧トヨタの構想は、そこに気付き方針の転換を図ろうとする意志の表れであると感じます。

今のところ、日本の自動車産業は世界一の座に君臨していますが、このままあぐらをかいては特斯拉やアップルのような企業に簡単に足をすくわれてしまう危険があります。そうならないためにも、燃費から電費へと戦略のマインドをすぐにもシフトし、新しい経営に踏み出すことを期待しています。



これからの 車載ソフトウェア開発と 品質保証



森崎 修司 氏
もりさき しゅうじ

名古屋大学 大学院情報学研究科
准教授
株式会社ティアフォー 技術顧問

2001年 奈良先端科学技術大学院大学 情報科学研究科 博士後期課程修了。通信事業者にてオンラインストレージサービスの企画・開発に従事し、2013年10月より現職。実証的ソフトウェア工学の研究に従事。独立行政法人 情報処理推進機構 (IPA) にて、つながる世界の品質指針検討ワーキング・グループ等の主査を務める。日本、ドイツの車載ソフトウェア開発に携わる社会人博士の学位取得の支援や審査に従事。

はじめに

自動運転や高度運転支援への対応、より高度なセキュリティの実現など、車載ソフトウェアへの要求が多様化する中、ODD (Operational Design Domain : 運行設計領域) やシナリオベースの品質保証など、国内外のさまざまな取り組みが発表されるようになりました。ソフトウェアの開発方法やその品質保証の方法もこれらの取り組みや技術に合わせて変えていかなければならない部分があります。

本講演では、そうした車載ソフトウェアを取り巻く環境の変化や背景を整理した後、これからの開発方法や品質保証に求められる要件を解説します。

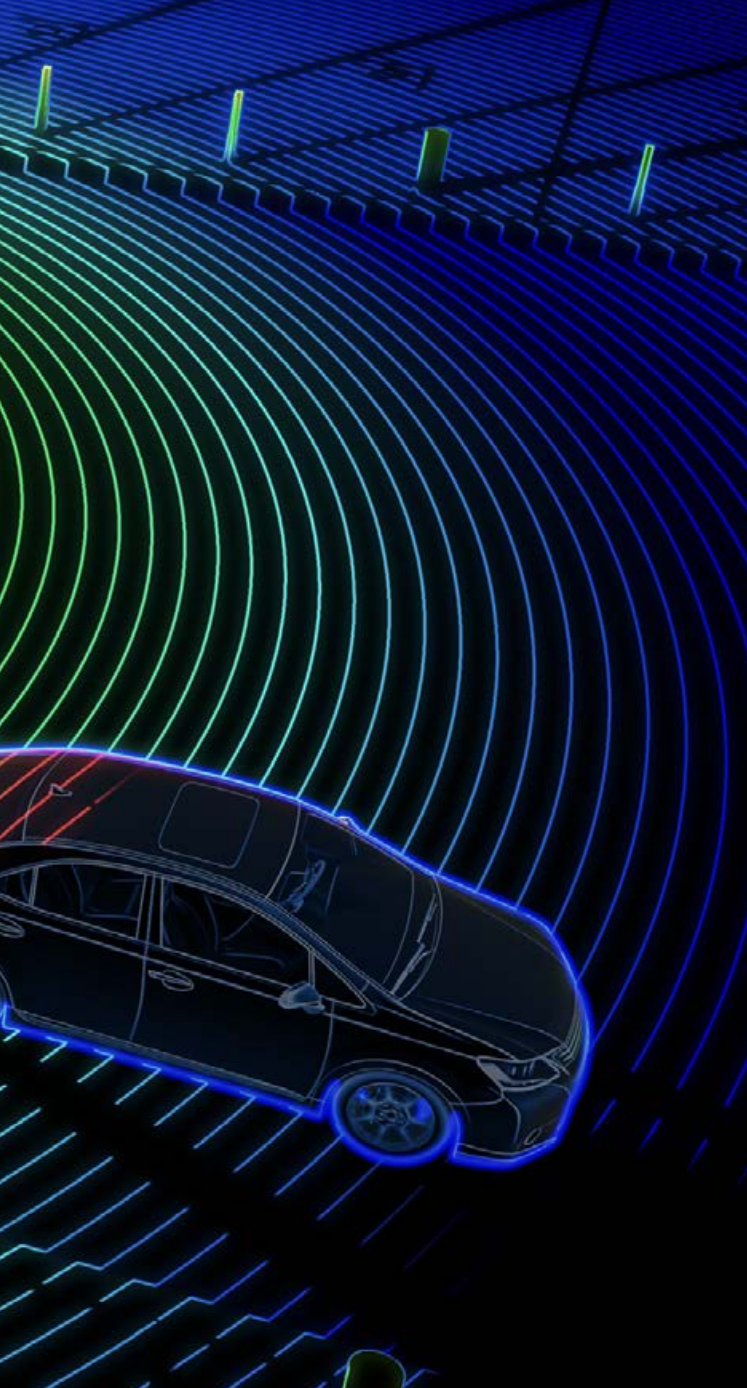


図1は、環境想定^{*}の難易度を表したものです。環境想定や制御が比較的容易な左側の「車体系」から、右側の「高度運転支援」や「自動運転」に向かって環境想定^{*}の難易度が高くなっていきます。そこには「他システムとの連携」や「ドライバーへのテイクオーバー¹⁾」「路上障害物の判断」といった複雑な状況判断が求められることを示しています。今後、必要とされる環境想定は右側、より想定が難しい方へ向かっていくと考えています。

ODDによる条件の明確化

環境想定が難しくなる中で求められているのが、ODDによる条件の明確化です。ODDとは、自動運転システムが正常に動作するための前提となる条件を定めたもので、「この前提が満たされないとこの機能は使えません」という条件をドライバーに確実に伝えるようにするためや、型式認定にも用いられます。

ODDの例として、国土交通省が公開している資料には次のような条件が示されています。

- 道路条件（自動車専用道路、一般道 など）
- 地理条件（山間部、設定済みジオフェンス など）
- 環境条件（天候、夜間制限 など）
- その他（速度制限、信号情報等のインフラ協調の要否 など）

^{*}1: 自動運転システムによる対応が難しい場合に、システムが運転をドライバーに要求すること

車載ソフトウェアの変化

➡ 求められる環境想定^{*}の精緻化

まず車載ソフトウェアをめぐる状況として、利用環境を想定することの難しさについてお話しします。そもそも環境や利用状況についての想定は以前から簡単なものではありませんでした。現在、環境想定はいよいよ複雑さを増しています。場所（国）、環境、法令、走行時期（季節や時間帯）といった条件について、それまで人が行っていた判断や操作を、ソフトウェアでも担うようになってきたためです。

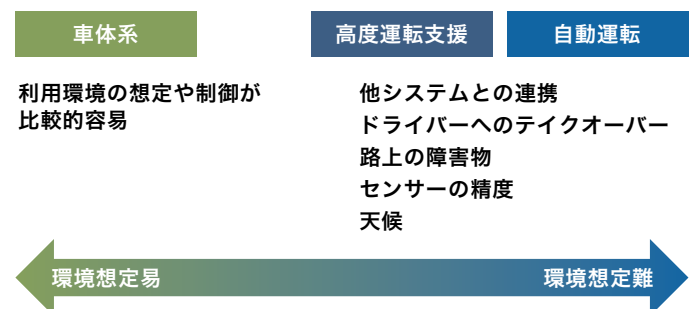


図1 環境想定^{*}の難易度

例えば、車線逸脱警告やアダプティブクルーズコントロール（定速走行・車間距離制御装置）は、センサーがうまく稼働しないときにメーターなどのインジケータで「機能しない」ことをドライバーに示します。これと同じように、想定されていない条件・場所である場合にもドライバーに伝えることが求められつつあります。

将来的にはこのODDが利便性や商品競争力につながっていくと思われます。無数にある条件の組み合わせから、適切にODDを定義することは非常に難しい判断を要します。テスト技術者や品質保証技術者が昔から取り組んできた絞り込みの技術が、複雑化する環境想定を支える技術として重要になります。

表 ODDをベースにしたシナリオ（ユースケース）の例

Road section No	ODD use case	Use case name	Use case No
1	発車	発車する（前方に車両がない）	UC-007-0004
2	同一車線内走行	通常走行	UC-001-0001
3	左折	T字路交差点（信号なし・合流）	UC-005-0001
4	同一車線内走行	通常走行（対向車線なし）	UC-001-0001
5	左折	T字路交差点（信号なし・合流）	UC-005-0001
6	回避	路駐・障害物回避	UC-003-0005/ UC-001-0024
7	左折	T字路交差点	UC-009-0001
8	同一車線内走行	通常走行	UC-001-0001
9	左折	T字路交差点（信号なし）	UC-009-0001
10	同一車線内走行	通常走行（対向車線なし）	UC-001-0001
11	左折	T字路交差点（信号なし）	UC-009-0001
12	同一車線内走行	通常走行	UC-001-0001
13	停車	停車する（停車予定の場所に何もいない場合）	UC-006-0001

出典: Tier IV Tech Blog <https://tech.tier4.jp/entry/2021/04/07/160000>



図2 シナリオ（ODDユースケース）と走行コースの対応（1～13の番号は表の番号に対応）

出典: Tier IV Tech Blog <https://tech.tier4.jp/entry/2021/04/07/160000>

最初は確実に安全を確保できるような条件のみを環境想定として定義し、徐々に範囲を広げていくアプローチが一般的と言えるでしょう。これにはプロダクトを作る側より、品質保証する側が積極的にODDの設計に関与していく必要があります。

シナリオベースの品質保証

ODDで示された条件内で問題がないかどうかを確かめる方法として、シナリオベースの品質保証について見ていきます。シナリオを作成し、それを前提として開発したり、品質保証したりする手法が今後、主流になることが予想されます。

表は、ティアフォー社が公開しているODDをベースにしたシナリオ（ODDユースケース）の一部です。前方に車両がない場所から発車する、同一車線内を走行する、路上駐車車両や障害物を回避する——といった具合に表の1行ごとにシナリオと対応する走行コースが示されています（図2）。開発も品質保証もこのシナリオとその組み合わせを前提として進めます。

図3は、シナリオを階層化し、シナリオの組み合わせによってさまざまな状況を想定できるようにするためのものです。これは、自動運転を実社会で利用する際に考えなければならないことを整理するため、ドイツで発足した産官学共同のPEGASUSプロジェクトが公表した資料の一部です。

変化の少ないものを1層目として、下から、道路配置（道路の状態や道幅、車線数など）、標識・路側物、一時的な保守や事象（工事や一時的な通行止めなど）、可動物体（他の車両など）、環境条件（天候など）、V2X^{*2}と、6つの層に分けて条件を考えます。これらの層を組み合わせながら、さまざまなパターンでクルマが期待通り動くかをテストしていきます。

*2: Vehicle to Everything：車車間通信、歩車間通信をはじめとして車両と他のあらゆるモノとの間の相互連携を総称したもの

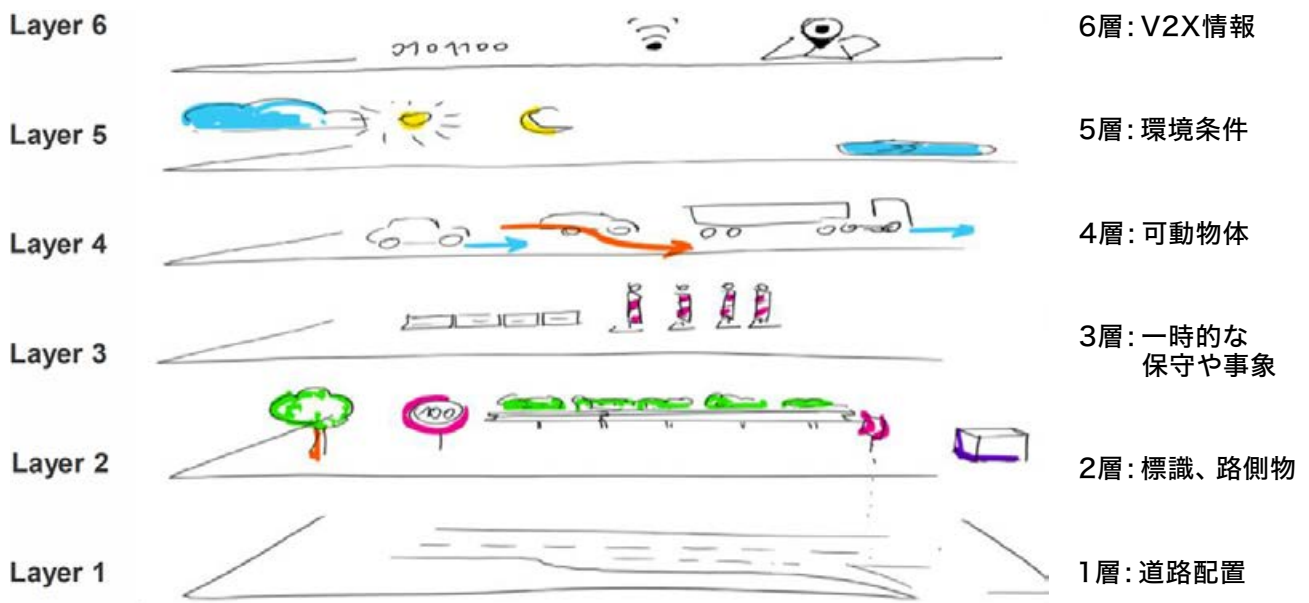


図3 シナリオの階層化

https://www.pegasusprojekt.de/files/tmpl/Pegasus-Abschlussveranstaltung/15_Scenario-Database.pdf を元に講演者が一部加筆

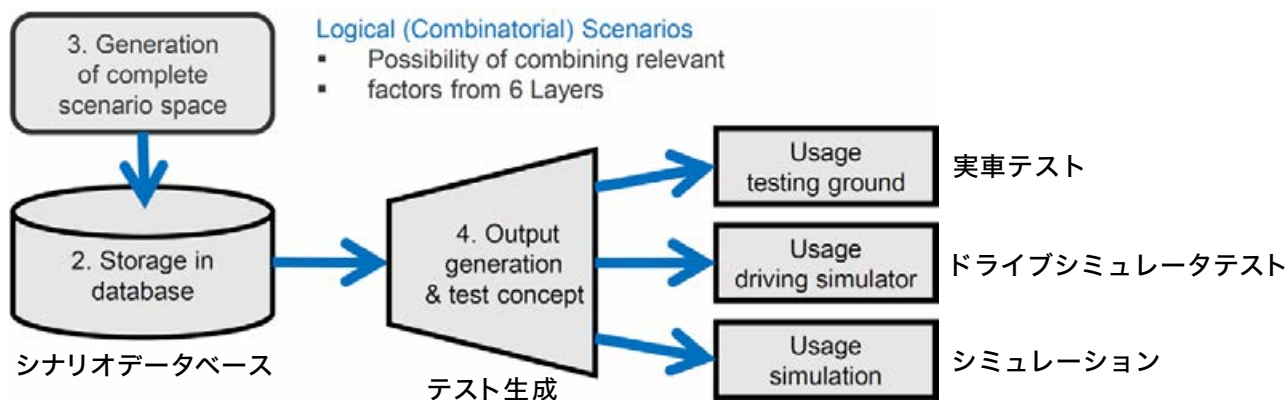


図4 シナリオの条件からテストケースを作成し、テスト実行するフロー

https://www.pegasusprojekt.de/files/tmpl/Pegasus-Abschlussveranstaltung/15_Scenario-Database.pdf を元に講演者が一部加筆

これらシナリオの条件を基にテストを生成し、テスト実行する流れを表したものが図4です。左から、シナリオデータベースにさまざまな各層の条件を入れておきます。条件の組み合わせによってクルマの振る舞いが変わることがありますので、それぞれのパターンについてテストを生成していきます。これについて実車テストやドライブシミュレータ、計算機上のシミュレーションでテストを実行します。

➡ 多岐にわたる要素技術

車載ソフトウェアの開発および品質保証の在り方が変わりつつある背景には、環境想定の大規模化のほか、自動運転や高度運転支援を実現するために多岐にわたる要素技術が使われるようになったことが挙げられます。主にセンサー、アルゴリズム、車外サービスとの連携などです。それぞれについて順番に説明します。

1 センサー

センサーについては、LiDAR(Light Detection And Ranging: レーザー光を用いたリモートセンシング技術) やアダプティブクルーズコントロールなどでよく使われるミリ波レーダー、さらにカメラ(単眼・双眼)などがあります。

このうち比較的新しいセンシング技術がLiDARで、センシング結果は点群あるいはポイントクラウドと呼ばれる粒の集合によって得られます(図5右側)。遠距離にある対象物を捉えられる一方で、カメラのように色を識別できないこと、また悪天候

時のセンシングが弱点とされています。図5の右側の画像を人間の目で見ると形状などからクルマに見えるのですが、これをシステムにクルマだと認識させるのは難しい場合があるため、現状ではこのLiDARの情報にカメラやミリ波レーダーから得られた情報を組み合わせて対象物を判断するセンサーフュージョンという技術を使って推測することがあります。



図5 センシングの結果(左:車載カメラ、右:LiDAR)

出典: Kumar, G A.; Lee, Jin H.; Hwang, Jongrak; Park, Jaehyeong; Youn, Sung H.; Kwon, Soon. 2020. "LiDAR and Camera Fusion Approach for Object Distance Estimation in Self-Driving Vehicles" Symmetry 12, no. 2: 324.

2 アルゴリズム

センサーから得られた情報を基に物体を認識するアルゴリズムを図6に示します。

LiDARの画像に見られた粒の集合(点群)から対象を推測することをクラスタリングと言います。例えば、その形状から対象物がクルマなのか歩行者なのかをクラスタリングにより推測します。そこにLiDARよりも高い解像度と色の情報を持つカメ

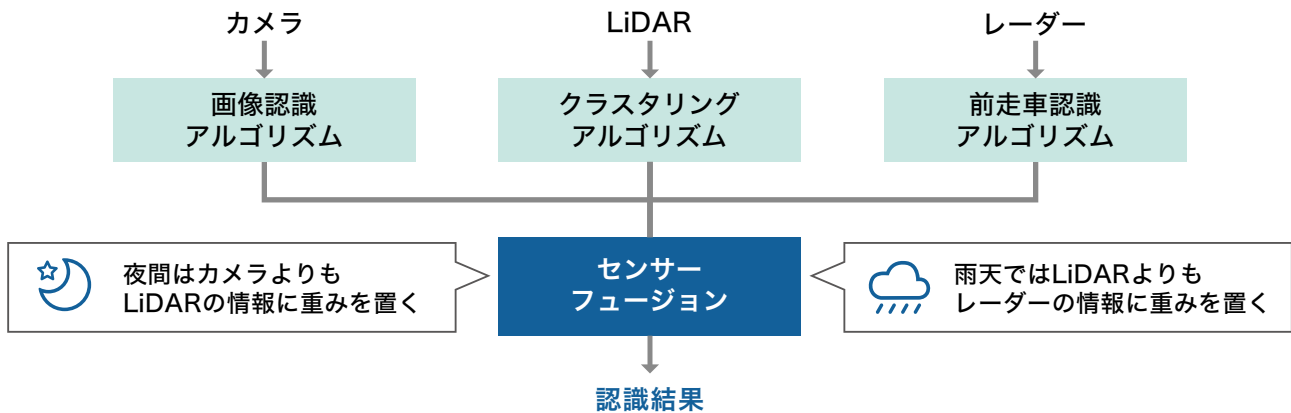


図6 センサーフュージョンの仕組み

ラを用いた画像認識アルゴリズムや、レーダーによる前走車認識アルゴリズムを加え、これら複数のセンサーの結果を統合して判断していきます。

また、センサーごとに明るさ（時間帯）や天候などへの弱点や適性があるため（例えば、夜はカメラよりLiDARの方が正確な情報を収集でき、雨天時や雪の日にはLiDARよりレーダーの方が有効）、状況に応じてどの情報に重きを置いて判断するかを決めていく必要があります。こうした統合的な判断を行うのがセンサーフュージョンです。実際には非常に難しいところですが、効果的なアルゴリズムが開発されれば、高い価値を生む可能性を秘めています。

して期待されています。現在はISO14296に向けて標準化が進められています。

➡ 新しいタイプの欠陥とその対応

ここまで車載ソフトウェアをめぐる環境や要素技術について見てきました。これまでと同様に要素技術単体に対して故障モード影響解析（FMEA）のような分析をして、単体では問題がないことを確認していても、特定の組み合わせだけで不具合が出たり、考慮すべき点が増えたりということが起こっています。ここ3、4年ほどの間で、こうした不具合や考慮すべき点をサイバーフィジカルギャップと呼んで、これに対する研究も進められています。

③ 車外サービスとの連携

3番目の要素技術「車外サービスとの連携」として、ここではダイナミックマップに触れます（図7）。ダイナミックマップは、静的情報である3次元地図情報に通行規制などの準静的情報、事故情報などの準動的情報、周辺車両や歩行者といった動的情報を組み合わせたデジタル地図であり、これをネットワークから得ることで自動運転や高度運転支援に活用しようというものです。全体が4つの階層に分かれており、階層ごとに最新情報が反映されます。ダイナミックマップは、センサーが収集する情報より多くの、より詳細な情報をもたらすことで、自動運転をさらに安全で精緻にするものと



図7 ダイナミックマップの概念

出典: ダイナミックマップの概念/定義および、SIP-adusにおける取り組みに関する報告
https://www8.cao.go.jp/cstp/gaiyo/sip/iinkai/jidousoukou_30/siryo30-2-1-1.pdf

現実世界をセンシングしてコンピュータ上にモデル化し、モデルの上で分析して最適な結果を導き出し、それを現実世界にフィードバックするシステムを「サイバーフィジカルシステム」と呼びます。この現実（フィジカル）とモデル（サイバー）の間に「差」が生じることをサイバーフィジカルギャップと呼んでいます。

例えば、自動運転中のクルマの前方にクルマがある場合のサイバーフィジカルギャップを次の①～③で説明します（図8）。

- ①自動運転システムはセンサーから情報を取得します
（1. センサーによる情報取得）
- ②この情報を基に前方の物体をクルマと判断します
（2. 物体認識）
- ③対象の位置を時系列で捉えて、前に進んでいるか、止まっているかを判断します。前方の車両が信号待ちなどで一旦停止しているとして後ろについて停車すべきか、駐車しているとして追い越すべきか判断し経路を選択します（3. 経路選択）

この経路選択を間違えることもサイバーフィジカルギャップと呼ばれる欠陥の一つです。サイバーフィジカルギャップを防ぐには、まず、センサーやアルゴリズム単体での検証が求められます。センサーが壊れたことをどのように自己診断するか、物体認識の精度、経路選択の妥当性などを、これまでの車載ソフトウェアと同じように検証することになります。

「単体での妥当性確認」は既存の品質保証の方法である程度対応できますが、難しいのは、個々では問題なく見えるにもかかわらず、情報取得、物体認識、経路選択といったフィードバックループの中に一つ不具合が生じることによって問題がドミノ倒しのように後続のフィードバックループに悪影響を与えるケースです。例えば、センサーの一つが故障して自己診断機能が動き始めている間に別のセンサーの精度が落ちると、センサーフュージョンの物体認識アルゴリズムがそれを想定しておらず、結果として後続の経路選択で実現されている正しい経路選択が行われなくなるようなケースです。

これらはサイバーフィジカルギャップの中でも特に手ごわい不具合で、品質保証の現場ではこのようなコーナーケースをひたすら探して対処する必要があります。私もこうした研究に取り組んでいます。

➡ セキュリティ

セキュリティは基本的に安全性の検証と似ていますが、いくつか異なる部分もあります。

悪意ある攻撃者は次々に新たな脆弱性、新たな手口で攻撃してくるため、製品として出荷した後も継続的にこれらに対応していかなければなりません。スマートフォンやPCで随時行われているセキュリティアップデートのような頻繁なプログラムの更新が求められます。

また、攻撃対象となる機能の重要性により攻撃のレベルも変わってくるため、対策をどの程度に見積もるのかも難しい問題



図8 前方にクルマがある場合の自動運転中のクルマの制御

で、自動運転や高度運転支援のセキュリティを考える上では重要な課題になると予想されます。

どのように開発と品質保証を変えていくべきか

➡ より精緻な品質保証と開発プロセス

これまでお話ししたように、自動運転や高度運転支援を前提として車載ソフトウェアは大きく変化しています。品質保証や開発についても、同時に仕組みを変えていく必要があります。

まず、プロダクト開発と品質保証をこれまで以上に独立させる必要があります。品質保証は品質保証として、プロダクトはプロダクトとして社会需要やコストに応じた開発をしていくことが重要です。図9に示した通り、今まではプロダクトに対してその仕様が満たされているかといった品質保証が中心でしたが、これからはよりユーザーや現実世界の要求に対して品質保証がどうあるべきかを考慮することが求められます。

次にテスト（テストケース）の増加を抑えることも重要です。例えば、派生型の開発をする場合、ある車種に対してテストを行う、次の車種には今までのテストに機能追加、機能変更分を加えてテストを行う、といった手法でテストしてはテストが著しく増えてしまいます。従って、まずプロダクトを共通化し、共通化されたプロダクトの使用を前提とした全体のプロダクト群を定義して、それに合わせてテストする「プロダクトライン」の考え方を取り入れていかなければテストが肥大化していきます。

最後に開発プロセス（活動）についてですが、単体テストで検出可能なバグを、実車のテストで検出するようなことがあれば大変な手戻りにつながってしまいます。開発プロセスや開発活動ごとに、どのような欠陥が混入される可能性があって、それらをどのように検出するべきかという想定をこれまで以上にしっかりと持つ必要があります。

先ほどシナリオベースの品質保証について説明しましたが、他の車両の動きや天候などさまざまな条件を組み合わせでテストケースを作り出していきます。これを全て手作業でやるのは現実的ではありません。つまりテストケースの自動生成機能を前提としたテスト実行の自動化プラットフォームを作る必要が

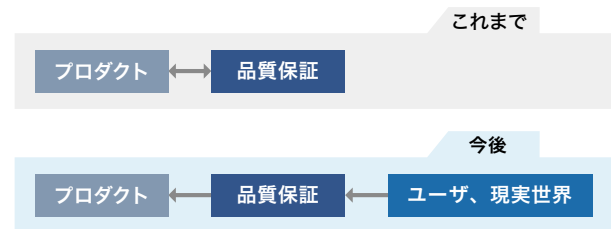


図9 これからの品質保証のイメージ

あります。ハードウェアの開発に携わる企業では、すでに検査の自動化はかなり進んでいると思いますが、これからはソフトウェアでも当たり前になっていくと思われます。

➡ 継続的な開発の必要性

車載ソフトウェアの変化に伴い、継続的に開発・品質保証を続け、製品（ソフトウェア）に問題がない状態を維持することが求められるようになります。その典型例は、新たな脆弱性に対するセキュリティアップデートが必要になることです。

こうしたソフトウェアのアップデートを積極的に活用して次のようなことが実現できます。

- クルマ購入後に特定の機能を追加購入できるオンデマンド機能
- 交通シナリオデータベース(ダイナミックマップなど)の随時更新

このような継続的なリリースを前提として、開発や品質保証のプロセスも変化させていかなければなりません。

おわりに

ここまで、自動運転や高度運転支援を支える技術や課題を通じて、開発や品質保証に対する新たな要求について俯瞰してきました。技術の進化に応じてさまざまな工夫が求められると同時に、継続的な開発・品質保証の重要性も浮き彫りになってきたと思います。

継続的な開発をするには、前提としてV字型開発からアジャイルに代表されるイテレーティブな開発にシフトしていくことが大切であり、私自身も現在ベリサーブ社と共同でこうした開発に適した品質保証の在り方について研究しているところです。成果などについてはブログやSNSでも発信していますのでご参照いただければと思います。



車載システムの分散・分担開発における プロジェクトマネジメントの 課題解決に向けて



冬川 健一
ふゆかわ けんいち

株式会社ベリサーブ
東日本オートモーティブ事業部 技術部長

1990年代に株式会社CSK（現SCSK株式会社）にてPCシステムの開発や検証業務に従事した後、2001年に株式会社ベリサーブへ移り、家電や携帯電話など組み込みシステムの検証およびプロジェクトマネジメントを担当。2010年以降は一貫して、車載システムの開発プロセスやプロダクト検証における品質と効率の改善提案、業務マネジメントに注力。社外活動では、IT検証産業協会にてIT検証標準工法ガイドの作成などを推進している。

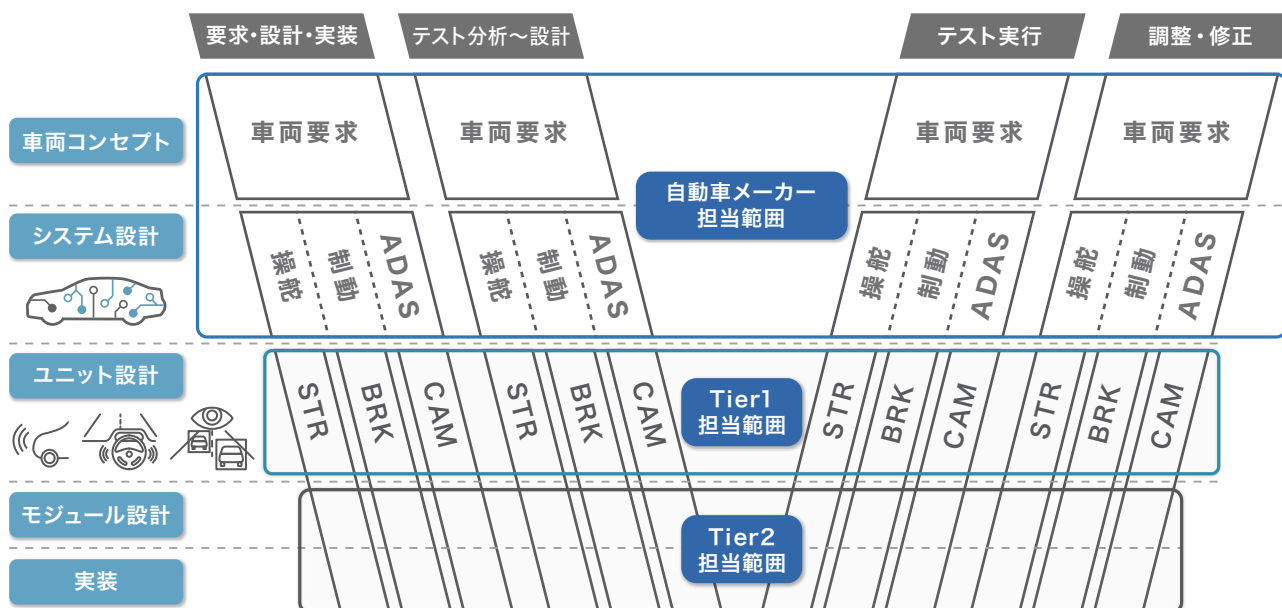
はじめに

車載システムやその機能が複雑化するに伴って、複数のシステムユニットやモジュールを組み合わせ、統合して開発されるケースが増えています。各ユニットやモジュールの開発は分散した組織によって役割を分担して行われますが、プロジェクト全体のマネジメント方法に正解はなく、手探りで開発を進めているというケースが多いのが実情ではないかと思われます。本講演ではこのような開発における課題分析の進め方について、ベリサーブの事例を基に、各現場で推進可能な取り組みをご紹介します。

車載システム開発プロジェクトの現状

システム開発の多層化と複雑化

図1は、車両・システム開発のレイヤーを図示したものです。最上段の「車両要求」を実現するため、各種システムで役割分担する設計を行い、そこからユニット設計のために、各ユニットに要求を出します。それぞれの担当範囲として「自動車メーカー」「Tier1」「Tier2」とありますが、最近は自動車メーカーが下位のユニット設計を行ったり、Tier1サプライヤーが車体開発に進出したりと、いわゆる水平分業体制になってきており、システム開発の多層化と責任分担の複雑化が進んでいます。



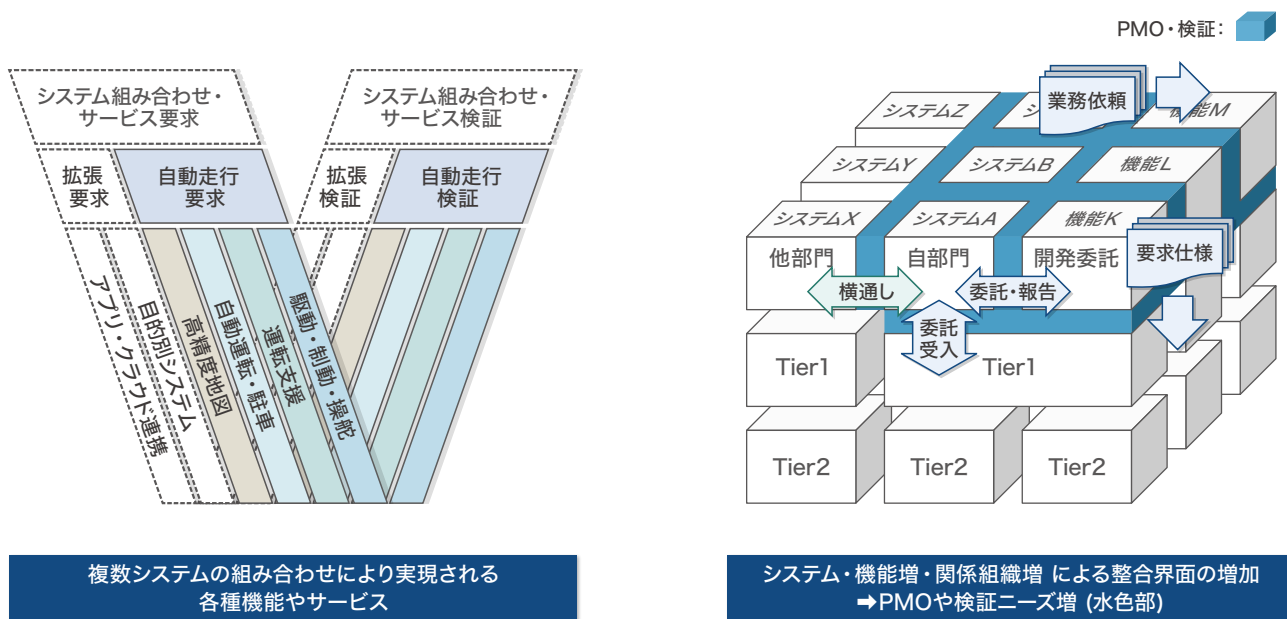
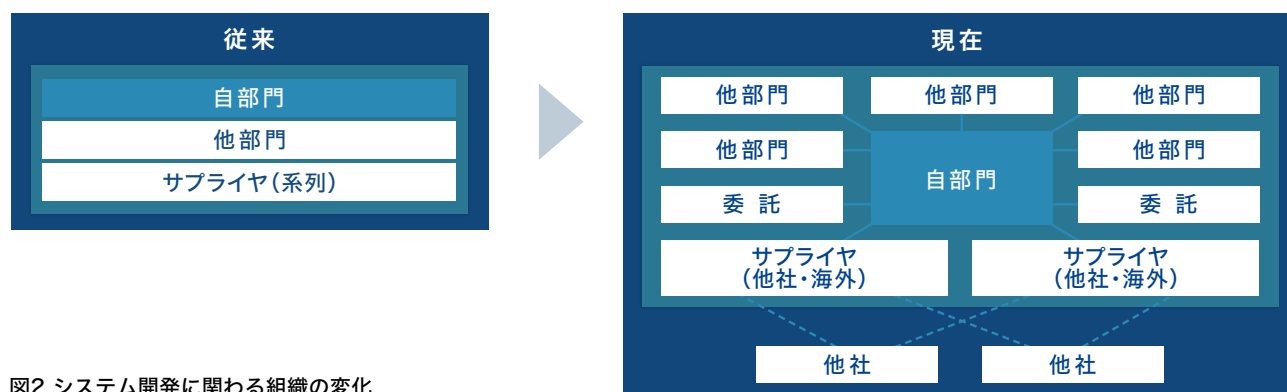
システム開発の多層化と、責任分担の複雑化

図1 車両・システム開発における開発のレイヤー

また、現在はさまざまな部品にソフトウェアが搭載されるようになり、多くの部門が新たにシステムを開発するようになりました。そのため、システム開発に関わる組織・企業の複合化が顕著になっています（図2）。

図3は、自動車開発の「システムズエンジニアリング化」についてまとめたものです。システムズエンジニアリングとは、システムの実現を成功させるための複数の分野にまたがるアプ

ローチおよび手段を指します。左のV字開発プロセスを例に見れば、自動走行要求を満たす「駆動・制動・操舵」「運転支援」「高精度地図」などのシステムを組み合わせ、さらに「目的別システム」「アプリ・クラウド連携」といった新たな拡張要求を積み上げています。また右の図は、システム・機能・関係組織の増加による整合界面の増加を示しており、結果として分散組織で分担開発するプロジェクトマネジメントや検証の重要性が高まる状況になっています。



品質確保に必要な4領域

図4は、プロダクトとプロセス、それぞれの品質を向上するための Validation (妥当性確認) と Verification (検証) をまとめたものです。図右上の「プロダクトの Verification」(仕様通りに作っているかの検証) はベリサーブが従来から取り組んできた業務ですが、最近では同左上の「プロダクトの Validation」(要求を実現できる仕様であるかなどを確認) の重要性が高まっています。

そのためには「プロセスの Validation」(同左下、目的を実現できるプロセスであるかなどを確認) や、「プロジェクトの Verification」(同右下、プロセスを手順通りに実施しているかの検証) を併せて行う必要があります。これが、ベリサーブがたどり着いた品質確保に必要な4領域で、これら全てが満たされないとうつりや不具合が生じることになります。

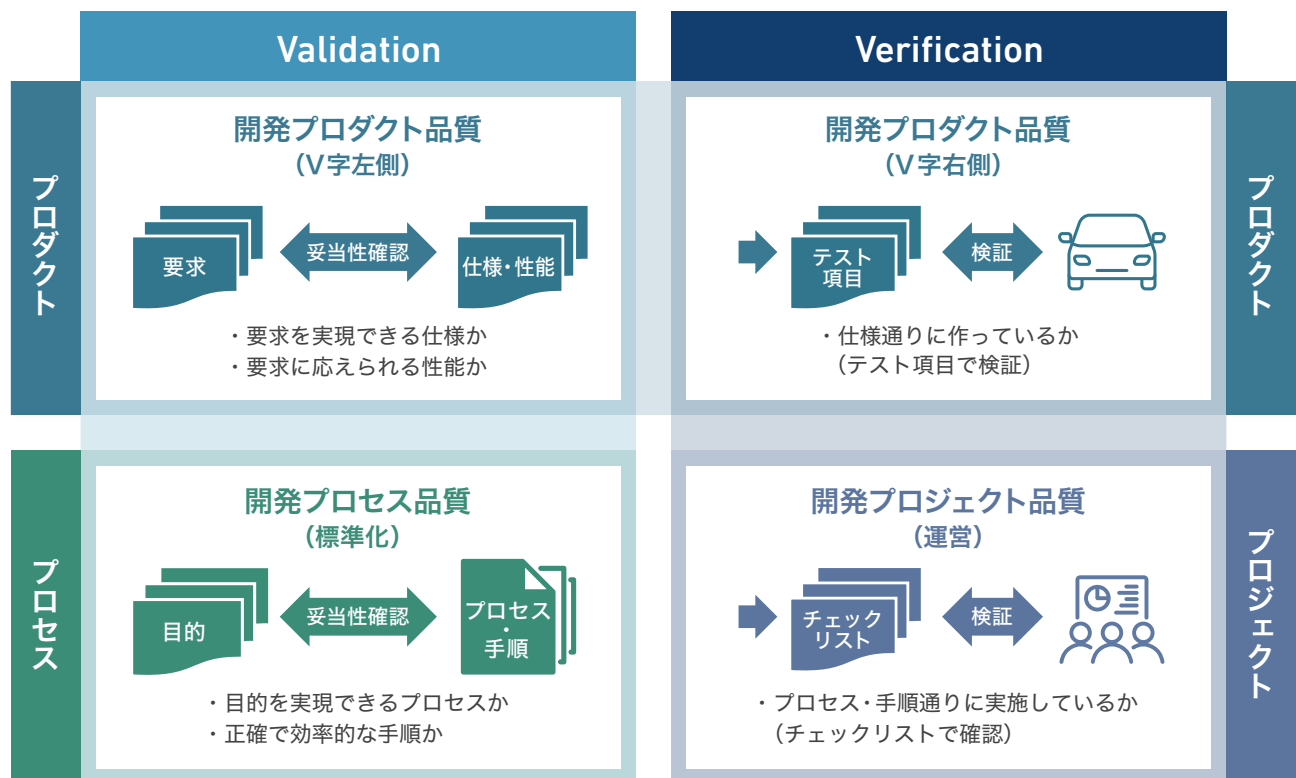


図4 品質確保に必要な4領域

プロジェクトタイプの分類

プロセス状況とプロジェクトタイプ

以前、ベリサーブ主催のセミナーにおいて聴講者アンケートを取ったところ、現状として「開発プロセスが整備されている／されていない」にかかわらず、「困っている」という回答が大半を占めました。私がこれまでに見た現場の事例を分類したところ、整備されているプロセスに対し「古い」「分かりづらい」といった声がある他、新たにプロセスを整備することへの懐

疑があるなど、異なる課題があることが明らかになりました(表1)。プロセスの在り方については、部門・チーム・個人によって考えが異なるため、組織や人数が大きくなるほど意見がまとまらない状況が発生します。

組織・プロセス形態による差異

組織・プロセス形態による差異もまた、プロジェクトマネジメントの在り方に大きく関わります。図5の「石垣型」「レンガ・ブロック型」分類は、日本とアメリカ・シリコンバレーそれぞれの組織特性の違いとして、しばしば紹介されるものです。

プロセス状況	タイプ分類	状況
プロセス整備 不十分	分析していない	困っているのはプロセス整備が不十分なせいだと思っている 実際にはそれだけではないが、絞り込みができていない
	プロセスに懐疑的	プロセスを整備したところでうまくいくか疑問だと考えている 複雑な事情が絡み合っている
	必要だが大変	何をどこまでプロセス定義すべきか難しいと思っている 全部詳しい必要はなく、人によって必要なレベルも異なる
プロセス整備 されている	古い・合っていない	プロセスが現状の業務にマッチしていないと思っている 更新が必要だがメンテナンスされていない
	分かりづらい	経験の浅い人は、プロセスガイドをただけでは理解できない 熟練者には理解できるが、熟練者はガイドを必要としない
	運用できていない	人によって実施のレベルに差異がある 皆がきちんと考えてやればいだけであるが、それが難しい
上記混合	上記複合	やるべきことが多く、切迫しているところから対応している

表1 プロセスの整備状況に応じたプロジェクトの課題

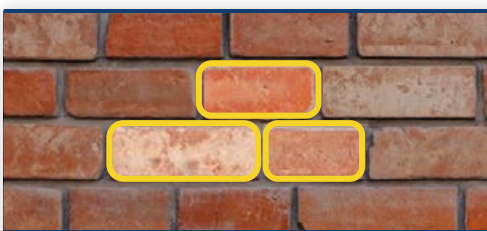
	
石垣型	レンガ・ブロック型
- 目的に対する体制や要員数を定義 - 各人の能力に応じて役割をアサイン	- 目的に対するプロセスやジョブを定義 - ジョブディスクリプションに従い人をアサイン
日本的 組織構成	欧米的 組織構成
国際標準プロセスを適用しづらい	国際標準プロセスを適用しやすい
ジョブのデータ化や生産性計測が難しい	ジョブのデータ化や生産性計測がしやすい
生産性が低くなりがち	生産性を高めやすい
担当者の協力で隙間(ポテンヒット)を埋めやすい	ジョブ定義の隙間が担当者から放置されやすい
ハイコンテキスト(文脈共有前提の文書化)	ローコンテキスト(伝えることは全て文書化)

図5 組織・プロセス形態による差異

黄色の線で囲まれている楕円（だえん）や四角が各要員の業務領域を表します。石垣型では、目的に対する体制や要員数を定義し、各要員の能力に合わせて役割をアサインします。一方、レンガ・ブロック型では目的に対するプロセスやジョブを定義し、それらに合わせて要員をアサインするのが特徴です。

日本的な組織構成と言われる石垣型と、欧米的なレンガ・ブロック型、それぞれにメリット・デメリットがあり、単純に優劣を語ることはできません。日本の伝統的な企業では、石垣型のまま連携を強化した組織を作り上げ、企業文化に昇華している例もあります。大切なことは、欧米で生まれた標準プロセスを適用する場合、組織の形態や現状を意識し理解した上でなければ、十分な効果が発揮できないということです。組織構造と課題を見極めた上で対策を検討・適用することが重要です。

プロジェクトマネジメントの 課題に対応した事例

ここからはベリサーブの事例を基に、プロジェクト運営レベルとそのステップアップについて説明します。取り上げるのは、分担・分散開発の体制でシステム設計を行っている自動車メーカーに対して、プロセスのValidationとプロジェクトのVerificationという2領域で支援した際の事例です。

まず、プロジェクトの運営状況を理解しやすくするため、運営のレベルを便宜的に0～4までの5段階に分類します（図6）。

新規システムやプロジェクトの立ち上げ時、支援を必要とする開発現場のほとんどはレベル0、1からスタートします。以下は、お客様から多く相談が寄せられる0～2の各レベルからのステップアップ事例です。

レベル0 ▶ 1

関係者で共通認識可能な
標準プロセスを作り上げる



この段階では主に、リファレンスプロセス（関係者で共有された標準プロセス）が整備できるよう支援します。

● 課題抽出と対策立案

課題分析は三つのステップで行います。まずは問題や課題、要望といった事象を洗い出し、システム開発全体をカバーするプロセスの枠組み（Automotive SPICEなど）にマッピングします。次に、事象の背景や経緯を確認し、原因の分析と、対策を講じる領域の特定を行います。最後に、対策を実行するためのプロセスの構築や仕組み作りの計画を立案します。

レベル 0	リファレンスプロセスが存在しない	標準的なプロセスの認識が関係者により異なる。 運用も管理も属人的で安定しない。
レベル 1	リファレンスプロセスは存在するが、 運用の仕組みが弱い	標準的なプロセスの認識は共有されているが、 運用の仕組みが整備されていない、または適切に運用されていない。
レベル 2	リファレンスプロセスが存在し、 それと比較検証する運用ができている	標準的なプロセスはあるが、運用が属人的であるため 体制規模が大きくなるとうまくいかないことがある。
レベル 3	プロセス運用がシステム化 されて管理されている	プロセスの運用状況が可視化され、体制規模が大きくなっても 管理しやすい。
レベル 4	プロセスとプロダクトの関係が 紐付けされて管理されている	変更点に影響するプロセス導出が可能で、業務目的も理解しやすい 理想的な形。

図6 プロジェクト運営レベルのステップアップ

● 開発プロセスフローの作成

現場へのヒアリングなどを通じて開発プロセスの現状を調査し、課題への対策を踏まえた上で、新たなフローの全体像（開発プロセス概要）を作ります。その際、主要業務間の関連や実施のタイミングが見えるようにするのがポイントです。次に、開発プロセス概要で定義した業務区分別に、作業の流れや各業務のINPUT・OUTPUTを理解しやすい形で明記し、プロセスを詳細化した詳細業務フローを作成します。

レベル 1 → 2

標準的なプロセスが運用できるようになる



この段階ではリファレンスプロセスを運用する仕組みの整備について支援します。

● 共通・汎用プロセスの運用

会議のファシリテーションや議事録作成、課題一覧の管理、情報の一元管理（各チームへの状況ヒアリング）、期日監視など、比較的一般的な業務支援を行います。これら汎用的な運用支援はどの運用レベルに対しても必要ですが、運用レベルが低い現場においては高い効果が得られる傾向があります。

● 作業概要／ガイドライン／手順書の作成

プロジェクトで実施する作業の一覧や目的、成果物との関連性を俯瞰できる資料、ガイドラインや詳細な作業手順書を作成します。作業領域ごとにこれらの資料を作成することで、組織標準が作られていない石垣型組織でも各領域の作業を標準化しやすくなります。

● 教育プログラム作成／教育事務局／運用・改善

リファレンスプロセスに関する資料や、テストに関連の深い「システム開発」部門などに存在する資料を基に、教育資料を作成します。その後、それらの資料を使い、教育事務局として新規参画メンバーに対する「集合教育」を実施することで、メンバーのプロセスに対する理解の底上げを図ります。

レベル 2 → 3

プロセス運用がシステム化され、適切に管理されるようになる



この段階ではプロセス運用のシステム化を進め、適切に管理された状態を目指します。

● プロジェクト状況／課題の見える化

残された課題の分類・分析や進捗状況のサマリーなどの可視化、DR（デザインレビュー）の指摘内容分析、業務負荷の分析・管理を行います。

● プロジェクト運用効率化・業務改善

計画作成や業務改善、進捗管理、管理ツールの導入、開発成果物の管理、管理運用ルールの改善に加え、実際に開発業務を進める上で必要となる自動化ツール（比較・検証ツールなど）の作成などを行い、効率化を支援します。

またレベル2→3の取り組み例として、表2のようなシステムやツールを積極的に活用します。

プロセス・プロジェクト領域の活用ツール例（当社例）	
要求管理・ALM 系	Integrity, DOORS, ConTrack
構成管理	Git, SVN
タスク管理	Jira, Redmine, iQUAVIS
成果物管理	SharePoint, Confluence, Lotus Notes
不具合管理	Redmine, QualityForward
BIツール	Qlik Sense

表2 運用レベルアップに役立つシステムやツールの一覧

さらに Automotive SPICE のアセッサーや TUV-SUD の機能安全の資格保有者、PMI プロジェクトマネジメントの資格保有者などによるプロジェクト支援は、上記レベルのどの段階にあるかに限らず有効です。

期待される取り組み効果

このような支援により実施された対策によって得られた効果の例を挙げていきます。

1 組織・グループ別の課題が見えてくる

プロセスの標準化に取り組むこともできないような「忙しすぎる」組織においては、課題が見えづらくなっています。このような組織が抱える問題も課題分析などを通して見えやすくなります。

2 課題領域を絞り込める

対策の実施前はQCDの「品質」と「成果物納期」がともに良くない状況だったのが、レビューの仕組み改善などにより品質の網羅性が高まったケースがあります。つまり課題が「期限」に絞り込まれた（課題を寄せることができた）ことになります。

3 プロセス運用支援チームによるトレーナー的活動で、開発者の負荷が軽減できる

トレーナー的活動では、主に次の工程・組織への移行断面での課題の取りまとめや対策の推進支援を行います。

4 標準規格への適合が容易になる

プロセスが整備されると業務エビデンスをトレースしやすくなるため、標準規格への適合が容易になります。

5 関係する会社・組織間で、それぞれのプロセスの差異が埋めやすくなる

他社・他組織のプロセスと差異がある場合でも、相違点を可視化し、整合方法の検討・運用が可能になります。

おわりに

ここまで見てきた対策は、決して全てのケースに対応するわけではありませんが、課題に対して偏りの少ない適切な分析を可能とするために、ひととおりを検討範囲とすることは重要です。また、組織の階層や役割の垣根を越えた課題に対応するには、第三者による客観的・俯瞰的な課題の「見える化」が有効です。最後に、多くの方は過去の経験の延長で今日の業務を行うため、新たに整備したプロセスがプロジェクトでの日々の経験として定着するよう業務に活かすことが重要です。以上を踏まえながら、プロセスの妥当性とプロジェクト運営のレベルをステップアップしていくことをお勧めします。





車載セキュリティテストに関する ベリサーブの取り組み



長谷川 邦幸
はせがわ くにゆき

株式会社ベリサーブ

ソリューション事業部 サイバーセキュリティ第一ビジネスユニット

2019年 株式会社ベリサーブ入社後、企業ネットワークの脆弱性テストやIoT機器のセキュリティテスト、リスク診断などに従事。現在はISO/SAE 21434のプロセス構築の支援を担当している。CISSP、認定ホワイトハッカー (CEH)、情報処理安全確保支援士。



はじめに

自動車業界はCASE (Connected, Autonomous, Shared & Services, Electric) の時代を迎え、クルマとその周辺システムの開発は複雑化が進み、セキュリティに対するリスクも増大しています。

ベリサーブでは2005年からセキュリティ事業を展開しており、車載システム関連についても日々充実を図っています。図1は、クルマの開発を取り巻く状況と当社のサービスを図式化したものです。品質向上と効率向上というお客様のご要望に対して、プロダクト品質支援、プロセス品質支援、そして三つ目のサービスとして、セキュリティ支援を提供しています。

このようなサービスを提供するに当たり、経済産業省の情報セキュリティサービス基準審査登録制度に申請し、審査登録委員会の審査を経て一定以上の品質を満たした脆弱性診断サービスの提供ベンダーとして登録しています。またセキュリティに関するサービスを提供する部門において、情報セキュリティマネジメントシステムの認証規格であるJIS Q 27001:2014 (ISO/IEC27001:2013) 認証 (ISMS認証) を取得しています。

自動車業界については、最近の傾向として、サイバーセキュリティに関する協定規則 (UN-R155) への適合およびISO/SAE 21434への準拠に関するご相談を多くいただいています。

2020年6月、国際連合欧州経済委員会自動車基準調和世界フォーラム (WP29) においてUN-R155が採択されました。日本においても、UN-R155に適合するための法改正があり、無線によるソフトウェアアップデートに対応した新型車両については2022年7月から適用されます (継続生産車は2024年7月から適用)。

また、車両電気電子 (E/E) システムの開発に関わるOEM、サプライヤー向けのサイバーセキュリティのガイドライン規格として、ISO/SAE 21434:2021が2021年8月31日に発行されています。

本講演は、UN-R155認証取得の手段として多くの企業が適合を進めている自動車のサイバーセキュリティに関するISO/SAE 21434という規格の概要より、セキュリティテストの位置付けを確認し、その具体的なテストの手法であるペネトレーションテストとファジングについて、当社の実践例をご紹介します。

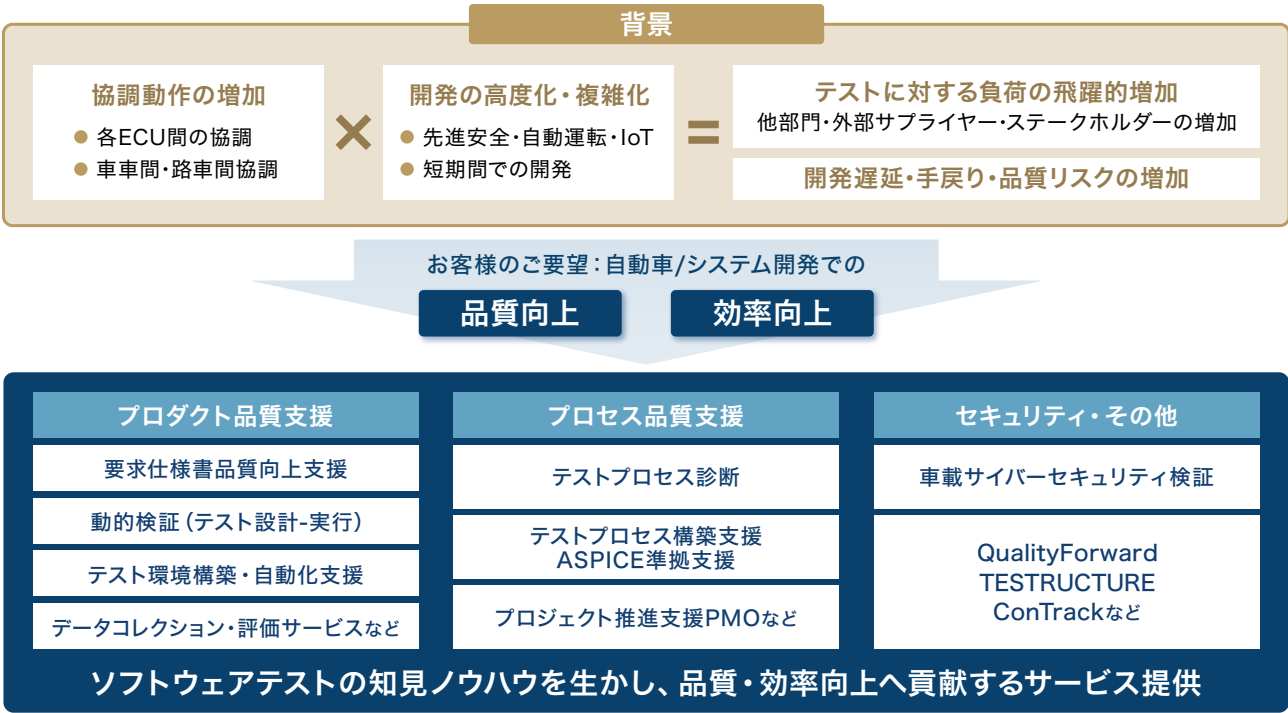


図1 車載開発に対してベリサーブが提供するサービス



ISO/SAE 21434の概要

ISO/SAE 21434^{*1}では、クルマの企画から製品開発、運用、廃棄に至るライフサイクル全般に及ぶサイバーセキュリティ要求が定義されています。1～4章が概要、5～15章が要求事項という構成になっており、これを実際のライフサイクルとの関係性に合わせてマッピングしたのが図2です。

開発のフェーズに特化した部分を少し掘り下げてみましょう(図3)。まず9章のコンセプト(Concept)には、OEMがクルマを企画する際の要求が定義されています。「9.3 アイテム定義」では、車両の中のアイテム、例えば自動運転などを定義し、その機能やアーキテクチャを考えます。次の「9.4 サイバーセキュリティゴール(Cybersecurity goal)」では、アイテムに対して脅威分析・リスクアセスメントを実施してリスクに対する必要な

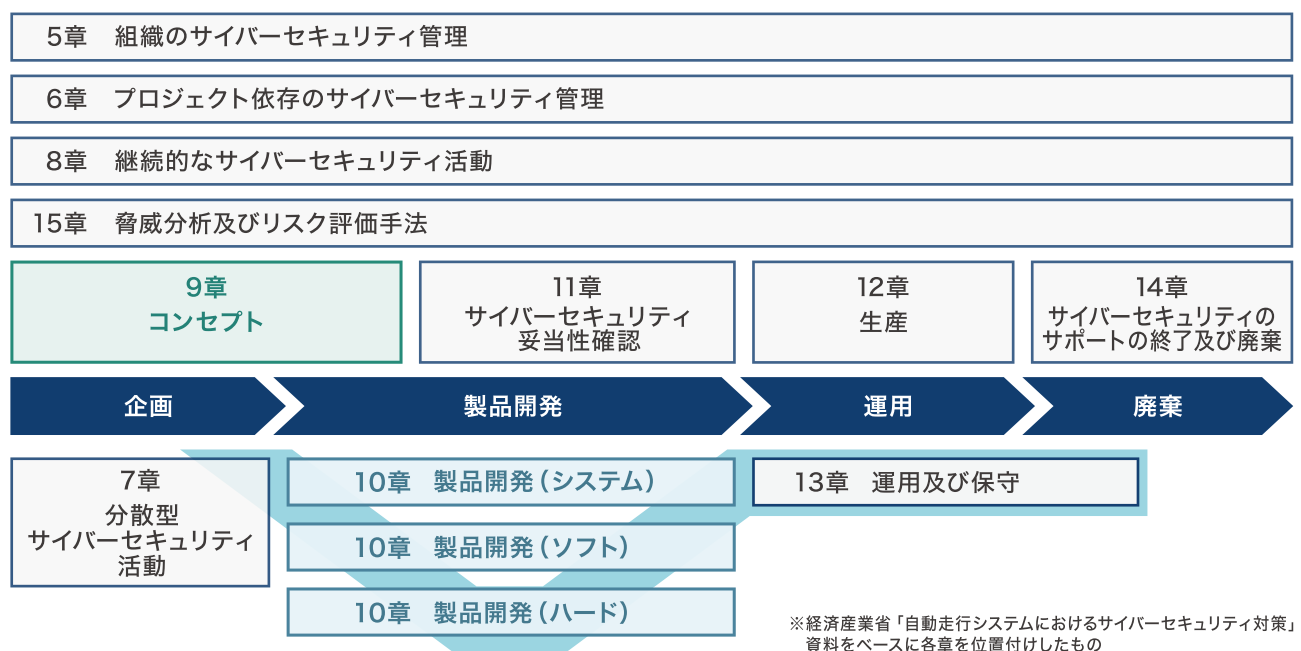


図2 ISO/SAE 21434の章立てと自動車ライフサイクルの対応

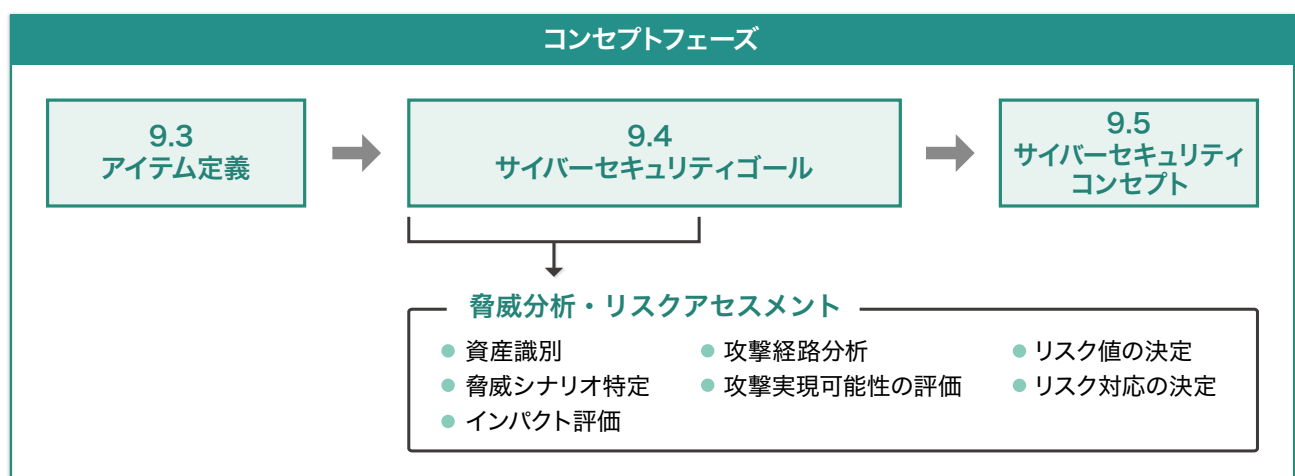


図3 ISO/SAE 21434におけるコンセプトフェーズでの要求事項

*1: 本記事のISO/SAE 21434の日本語訳については、以下の日本規格協会の訳を参考している。

ISO/SAE 21434:2021 自動車-サイバーセキュリティエンジニアリング https://webdesk.jsa.or.jp/books/W11M0090/index/?bunsyo_id=ISO%2FSAE+21434%3A2021

対策と目標を設定し、最終的に「9.5 サイバーセキュリティコンセプト(Cybersecurity concept)」で機能サイバーセキュリティ要求 (FCR: Functional cybersecurity requirement) を作成します。

OEMが作成したFCRが自社の開発部門もしくはサプライヤーに渡し、10章の製品開発フェーズに入ります (図4)。「10.4.1 設計」では、FCRの実装を考慮した技術サイバーセキュリティ要求 (TCR: Technical cybersecurity requirement) を作成しますが、この際に脆弱性分析を実施します。ここでは、攻撃の経路を検証する攻撃経路分析、攻撃実現可能性の評価などを実施し、リスクに応じて設計の見直しを行います。

設計においてサイバーセキュリティ面で問題がないと判断したら実装に移り、「10.4.2 統合及び検証」でセキュリティテストを実施します。ここでは5種類のテストが定義されていますが、本講演では名前だけではその内容が想像しづらいと思われるペネトレーションテストおよびファジングについて詳しく解説します。



ペネトレーションテスト

ペネトレーションテストは攻撃者の視点からシステムを調査し、実際に侵入を試みるテスト手法です。図5は、アメリカ国立標準技術研究所 (NIST) の「SP 800-115 Technical Guide to Information Security Testing and Assessment」からの引用で、テストのプロセスを示しています。最初に攻撃の計画を立案、システムの弱点を探索して攻撃を行い、その結果を分析して攻撃を繰り返し、脆弱性を検証します。

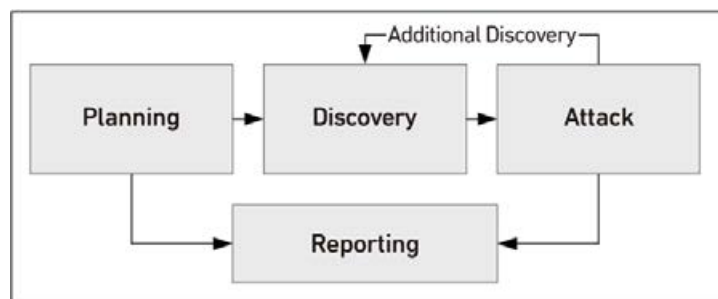


Figure 5-1. Four-Stage Penetration Testing Methodology

図5 ペネトレーションテストのプロセス

出典：アメリカ国立標準技術研究所 情報セキュリティテスト、アセスメントの技術的ガイド (<https://csrc.nist.gov/publications/detail/sp/800-115/final>) を元に一部加筆

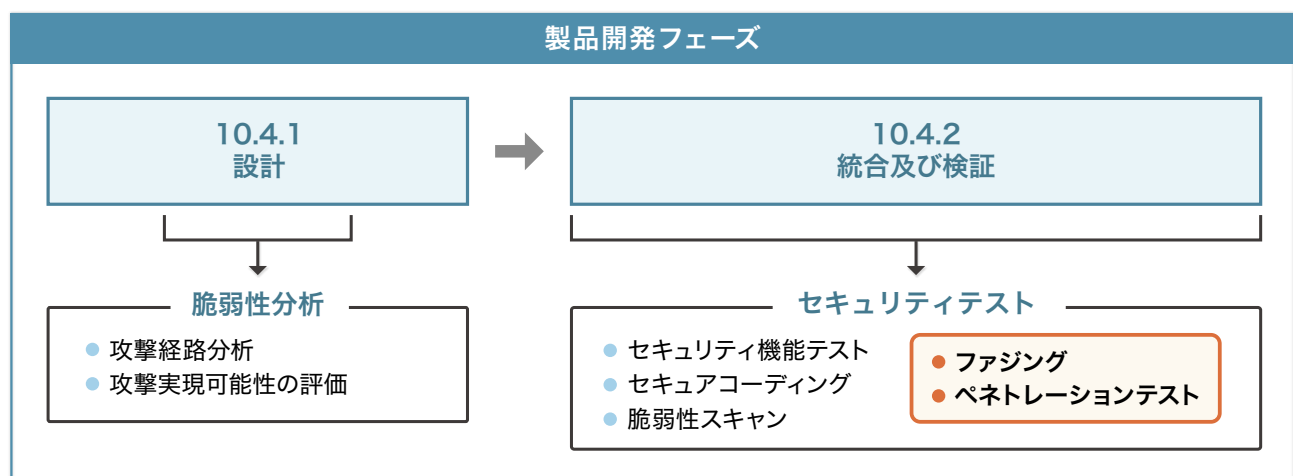


図4 ISO/SAE 21434における製品開発フェーズでの要求事項

ペネトレーションテストは、必ずしもブラックボックスで行う必要はありません。「ISO/SAE 21434 Annex E サイバーセキュリティ保証レベル」では、攻撃者はターゲットに対してある程度の知識を持っていると想定しています。また、攻撃者側にはある意味無限に時間がある一方、お客様やベリサーブのようなベンダーはテストの時間が限られています。そのため、攻撃者が時間をかければ見つけれられるレベルの情報は事前に開示し、その上で攻撃を実施するグレーボックステストというのが、一番費用対効果が高いと私たちは考えています。

ペネトレーションテストの実践例

ここからは、車両のECUを対象としたペネトレーションテストの一例をご紹介します。クルマからECUを抜き取り、さらに中の基板からチップを外してファームウェアを抽出し、ファームウェアから得た情報をもとに攻撃を行うことを想定しています。

ファームウェアの抽出

チップを外すための道具としてヒートガン、はんだごて、ペンチ、ピンセットを用意します。まずヒートガンで基板に熱を加えてハンダを溶かし、チップを分離します。チップの接合面に汚れが付いていると読み取りに支障が出るため、なるべくキレイにしておくのも一つのポイントです（図6）。

チップをアダプターにセットし読み取り装置に取り付け、USBでPCと接続し、ツールを利用してファームウェアを抽出します（図7）。



図6 ECUの基板よりチップを取り出す様子

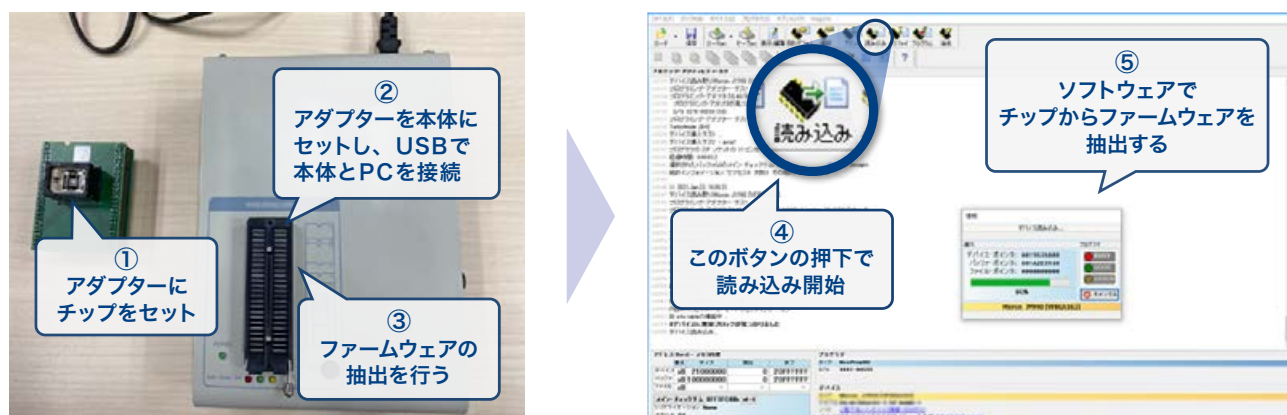


図7 取り出したチップよりファームウェアを抽出する様子

ファームウェアの解析

図8は、セキュリティテストによく使われる「Kali Linux」の画面で、ここに含まれる「binwalk」というツールでファームウェアのイメージファイル²を解凍しファイル・ディレクトリの一覧として表示しています。この図では重要な設定ファイルなどが配置されることの多い「/etc」というディレクトリの内容を表示しています。当該ディレクトリに存在するテキストファイルに機微な情報が含まれていればこの時点で読めてしまいますし、また、バイナリファイルを入手して中身を解析することも可能です。

²: 外部記憶装置に保存されているすべての内容を、ディレクトリ構造を保ったまま一つのファイルとしてまとめて保存したものを。

脆弱性調査とエクスプロイトコード

ファームウェア内のファイルが入手できたら、脆弱性を見つけてエクスプロイト（攻撃）コードを作成し、実際に攻撃を試みます。システムの脆弱性調査の場合、スクリプトファイルを解析して得られた暗号化アルゴリズムから復号用のスクリプトの作成や、パスワードを暗号化して記述しているシャドウファイルが見つければブルートフォース（総当たり）攻撃を実行するなどして、パスワードの解析を試みます。

バイナリファイルに対しては、同じアーキテクチャ上で動作させデバッガで解析することもできます。また、Hex-Rays社

の「IDA Pro」というツールを使ってアセンブリコードから脆弱性を探したり、バイナリファイルに含まれるOSS（Open source software）をSynopsys社の「Black Duck」などのツールで特定し、既知の脆弱性について検索したりして、見つかった脆弱性を利用した攻撃を行います。

ファジング

既知の脆弱性というのは氷山の一角に過ぎず、実際にはそれをはるかに上回る数の未知の脆弱性が存在します。ファジングは、こうした未知の脆弱性、潜在的な脆弱性を発見するためのテスト手法です。

具体的には、ファズ（予測不能な入力データ）をシステムに与えて意図的に例外処理を発生させ、その挙動を確認します。ファズの例としては、プロトコル上の標準や製品の仕様を満たさない、あるいは違反するデータで、極端に長い文字列や上限を超えた数値、許可されていない特殊文字などが挙げられます。

ファジングの種類と検出できる問題

ファジングで代表的なのはプロトコルファジング・ファイルファジング・APIファジング・UIファジングの四つで、いずれも要求された仕様を逸脱するデータを投入するものです。このうちプロトコルファジングについては、業界ごとにさまざまな通信プロトコルが使われているため、製品でどのようなものが使われているのかを把握しておくことが大事です。

図8 ファームウェア内のファイル・ディレクトリの一覧

図9は、Ari他(2018)の「Fuzzing for Software Security Testing and Quality Assurance, Second Edition」p.29のFigure1.10に記載されたファジングで検出できる問題の一覧です。実際のテストではcrashes（クラッシュ）やDenial of Service（DoS、サービス拒否）、つまりプログラムやシステムが異常終了もしくは無反応になる、といったものが多くを占めます。

● ファジングを実施する前に検討すべき事項 ●

ファジングについては、対象とするインターフェース、送信するファズの内容、結果の判定方法が重要なポイントとなります。

🎯 対象インターフェース&プロトコル

機器の持つインターフェースを列挙し、検討漏れがないようにします。また、そのインターフェースで使われるプロトコルに違反し例外を出す必要があるため、プロトコルの仕様も確認します。

🎯 ファズデータの種類

ファズデータはツールでの生成も可能で、車載関連ではSynopsys社の「Defensics」などが多く使われています。適切なデータを生成するために、ツールに任せるだけではなく、人の観点での補足も大事です。

🎯 監視方法

どのような問題を検出するのかを事前に決めておくことが重要です。監視にもツールを利用できますが、自動での判定には限界があるため、やはり人による確認が必要になります。一般的には三つの方法があります。検出する問題について最適なものを組み合わせることをお勧めします。

- **システムモニタリング**
(システム内プロセス、ログの監視)
→ 検出する問題：クラッシュ
- **リモートモニタリング** (ネットワーク応答の監視)
→ 検出する問題：サービス拒否、性能低下、
応答遅延、機密情報の漏えい
- **アプリケーションモニタリング**
(デバッガなどによるアプリ動作の監視)
→ 検出する問題：改ざん、破棄、異常

🎯 脆弱性スキャンとの違い

脆弱性スキャンは既知の脆弱性を発見するのが目的で、入力インターフェースやコンソール内でのスキャンを実行します。一方、ファジングは未知の脆弱性の発見を目的に、入力インターフェースに対して例外データを送って反応を確認します。この点が二つのテスト手法の違いです。

カテゴリ※	問題の概要
crashes[クラッシュ]	ファジング対象のプログラム、システムが異常終了する
denial of service(DoS)[サービス拒否]	ファジング対象のプログラム、システムの一部機能が無反応となる
security exposures[機密情報の漏えい]	ファジング対象のプログラム、システムファズに対するレスポンスに、機微な情報が含まれる
performance degradation[性能低下]	ファジング対象のプログラム、システムの処理能力が低下する
slow responses[応答遅延]	ファジング対象のプログラム、システムのレスポンスが遅くなる
trashing[改ざん、破棄]	ファジング対象のプログラム、システムの設定が意図せず変更、削除される
anomalous[異常]	ファジング対象のプログラム、システムで、上記以外の異常な振る舞いが発生する (例えば、スタックオーバーフロー)

図9 ファジングで検出できる問題

※参考：書籍「Fuzzing for Software Security Testing and Quality Assurance, Second Edition」p.29

ファジングテストの実践例

図10は、実際のファジングテストの様子です。「Defensics」が動作するPCをターゲット機器とアダプター経由で接続し、ファズを送信して反応を確認します。アダプターを交換することで、さまざまな機器のファジングが可能です。

おわりに

図11は、ISO/SAE 21434の要求とベリサーブが提供するサービスの対応表です。全体的なコンサルティングからPSIRT支援、個別の脅威分析・脆弱性分析やテスト支援まで幅広く対応していますので、ご興味ございましたら、ぜひお声がけいただければ幸いです。当社の35年以上に及ぶ検証分野での実績と技術を生かし、今後も品質・安全・セキュリティの三つの要素を守るため、より良いサービスを継続的に開発・提供していきます。

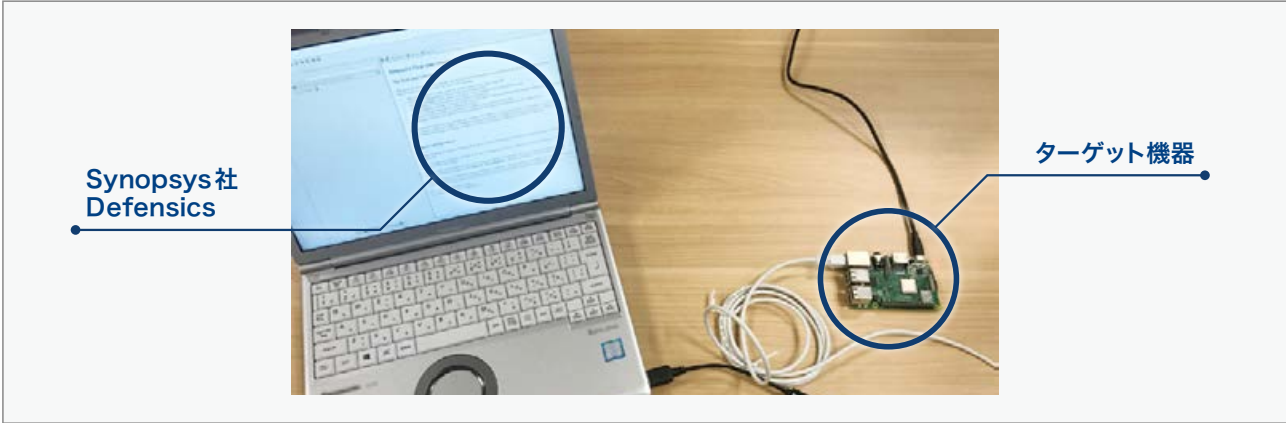


図10 ファジングテストの様子

No.	サービス名	システム提供	実務支援	ISO/SAE 21434 要件
1	ISO/SAE 21434コンサルティング (規程策定・運用)	△	○	規格全体をカバー
2	PSIRT支援	○	○	8.3 サイバーセキュリティ監視 8.4 サイバーセキュリティイベントの評価 8.6 脆弱性管理 13.3 サイバーセキュリティインシデント対応
3	脅威分析／脆弱性分析	×	○	8.5 脆弱性分析 15. 脅威分析及びリスク評価手法
4	セキュリティテスト	△	○	[RQ-10-09] 実装・統合のサイバーセキュリティ仕様充足を検証
5	脆弱性スキャン	△	○	[RC-10-12] 未確認のウィークネス(弱点)・脆弱性の確認 [RQ-11-01] 妥当性確認活動
6	ファズテスト(ファジング)	○	○	[RC-10-12] 未確認のウィークネス(弱点)・脆弱性の確認 [RQ-11-01] 妥当性確認活動
7	ペネトレーションテスト	×	○	[RC-10-12] 未確認のウィークネス(弱点)・脆弱性の確認 [RQ-11-01] 妥当性確認活動
8	セキュアコーディング準拠支援	○	○	[RQ-10-05] 設計・コーディングルール

図11 ISO/SAE 21434の要求・推奨とベリサーブの提供する支援サービスとの対応



トレーサビリティ管理ツールを もっと実践的に使いこなすには ～規格対応のため「だけ」から、 品質・生産性向上のために～



横田 浩行
よこた ひろゆき

株式会社ベリサーブ

プロセスエンジニアリングサービス開発部 プロダクトサービス開発課 課長

2009年 株式会社ベリサーブ入社。大手電機メーカーのプロジェクトなどでQAエンジニアを務めた後、新規事業の企画・推進に従事。その後はトレーサビリティ管理ツール「ConTrack」の企画・開発を主導するなど、システム開発全体のQCD向上のためのソリューションの開発に注力。この間、ソフトウェア開発に関する独立行政法人情報処理推進機構（IPA）の標準化・制度検討部会にも参画している。



はじめに

車載システム開発では、この10年で要求管理ツール（いわゆるトレーサビリティ管理ツール）の導入が進みました。しかし、規格への対応という当初の目的に沿った利用にとどまり、品質改善や生産性向上といった効果につなげられていないケースが少なくありません。本講演では、代表的な要求管理ツールを比較・分析した上で、より本質的な改善に導くツールの活用方法について、事例を交えて解説します。併せて、ベリサーブが提供するトレーサビリティ管理ツール「ConTrack」についてもご紹介します。

● 要求管理ツールとは

要求管理ツールとは、定義された要件がどのような工程を経て設計やソースコードへ落とし込まれているかを追跡、可視化するものをいいます。要求管理の主な目的は次の三つです。①要件と設計の整合性の確認（一貫性）、②要件の全てが設計に反映されているかの確認（完全性）、③設計の変更による影響の明確化（影響分析）。一般的な要求管理ツールは、これらを実現する機能を備えています。

● 日本での要求管理ツール普及の背景

日本で車載システム開発に要求管理ツールが利用されるようになったのは、2011年のISO 26262（自動車機能安全規格）の発行がきっかけです。それまでは表計算ソフトで要求を管理するケースが一般的でしたが、規格対応の必要に迫られ、さらにはシステムの大規模化・複雑化という事情も相まって、専門ツールの導入機運が高まりました。ただ当時はツールの大半が海外製品だったため、WordやExcelで設計書を作成する日本の開発文化にはなじまず、規格対応のエビデンスとなる成果物の“保管庫”の扱いにとどまっていた。

2010年代後半に入ると、シンプルで使い勝手の良いタグベースの要求管理ツールが登場しました。使い慣れたWordやExcelがそのまま使えることもあってそれなりに普及しましたが、導入の主目的はやはり規格への対応でした。

その規格対応もこの2～3年で一段落したようで、最近では品質や生産性の向上といった、より本質的な改善効果を期待して要求管理ツールの導入を検討するお客様が増えています。以下ではまず、代表的な要求管理ツールの種類と特徴を見ていきます。

● ALM観点からの要求管理ツールの分類

要求管理ツールを分類するに当たり、今回は、要求管理より少し広い概念であるALM(Application Lifecycle Management)の観点から考えてみることにします。

ALMは、システム開発で作成されるドキュメントなどを、システムのライフサイクル全体にわたり統合的に管理していくという考え方です。図1は、ALMを実践するためのツールをイメージしたもので、要求管理の他に構成管理、変更管理、要求・仕様記述などのモジュールから成り立っていることが分かります。

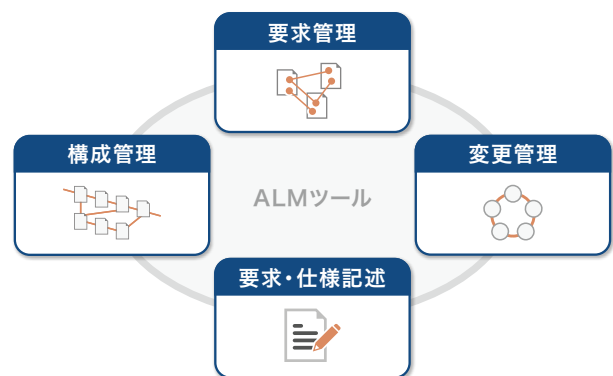


図1 ALMツールの構成イメージ



Excel

たいていのPCにインストールされているExcel（図2）は、導入や習得コストが事実上ゼロである点が最大のメリットです。要求管理ツールを提供するベンダーにとっては最強のライバルといえます。

しかし、Excelはそもそも要求管理の専門ツールではないため、できることはマトリクスや展開表の記述がせいぜいで、N対Nのような複雑な関係性の表現は困難です。構成管理や変更管理といった他のシステムとの連携も皆無なため、要求の追加や変更などの追従が全て手作業となり、大変な手間がかかります。

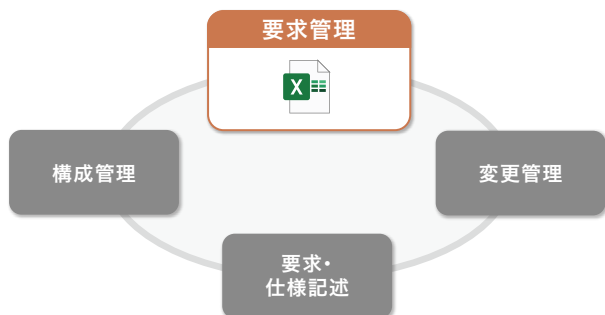


図2 Excelを要求管理に用いる場合のイメージ

フルALM型

Excelとは逆にALMに必要な全ての機能を含むものがフルALM型（図3）で、きちんと使いこなせば大きなメリットが得られます。

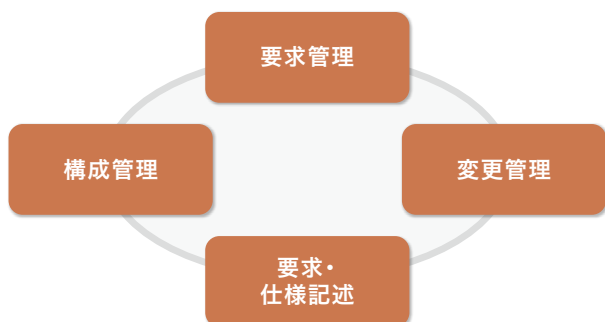


図3 ALMに必要な全ての機能を備えたフルALM型ツール

しかし、多機能なだけに「きちんと使いこなす」ことが非常に難しいのが実情です。開発プロセスやスタイルもある程度ツールに合わせる必要があります。特に、要求・仕様の記述にツール内の機能を使わなければならないことが、日本で普及する上で高いハードルになっているようです。

要求管理特化型

先ほど触れたタグベースのツールは、ドキュメント内に埋め込んだIDとタグをトレース情報として可視化するもので、この要求管理特化型に分類されます（図4）。要求管理に絞り込んだシンプルさが強みで、要求・仕様はWordやExcelで記述できるため、日本の開発現場との親和性が非常に高いのも特徴です。

一方で、機能を変更・追加するとIDの情報が変更になるため、差分開発やバージョンアップ開発が進むほどにタグをメンテナンスする負担が大きくなります。また、トレースの可視化に特化しており、他のツールとはデータ連携にとどまり緩やかな結合になっています。従って、ALMを実践するには運用プロセスを工夫する必要があります。

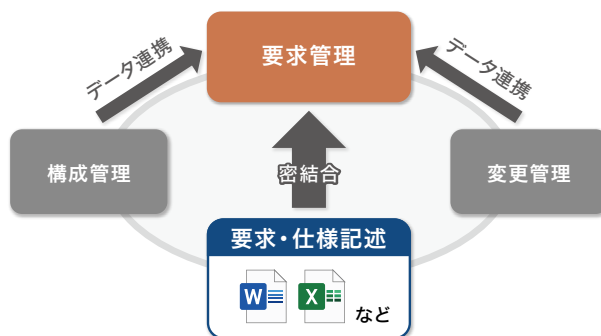


図4 要求管理に絞り込んだシンプルさが特徴の要求管理特化型ツール

● 軽量ALM型

ツール自体の機能は要求管理のみですが、構成管理や変更管理はオープンソースソフトウェア（OSS）とのシームレスな連携で統合的にALMを実現します。要求・設計はWordやExcelから取り込み可能で、慣れ親しんだ開発プロセスをそのまま維持できるので導入も容易です（図5）。ペリサーブが提供する「ConTrack」はここに分類されます。

あえて課題を挙げるなら、派生・保守開発を行わない単発のプロジェクトでは、導入時のフォーマット検討などの負担があり費用対効果が得られにくいことです。

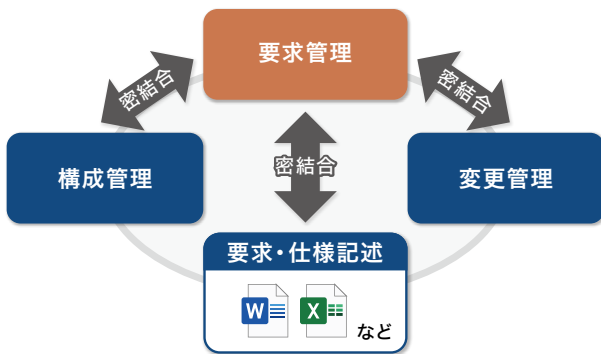


図5 他システムとのシームレスな連携でALMを実現する軽量ALM型ツール

● 開発における要求展開の類型

ここまで、要求管理に使われるツールを特徴ごとに分類し比較してきました。「さまざまな種類があることは分かったが、結局、自分たちのプロジェクトにはどれが適しているのか」と思う方もいらっしゃるでしょう。ツールの特徴の他に、もう一つの判断基準となるのが要求とそれを実現する機能の関係性に着目した、開発における要求展開の構造です。大きく次の二つに分類されます。

● ツリー構造型

システムの要求が機能へとシンプルに分解されて1対Nの関係になる構造です。新規開発やフルスクラッチなど、これから新しいものを作る場合に多く見られます（図6）。どの要求がどの機能で実現されるかが分かりやすく、小規模なプロジェクトであればExcelでも表現できます。

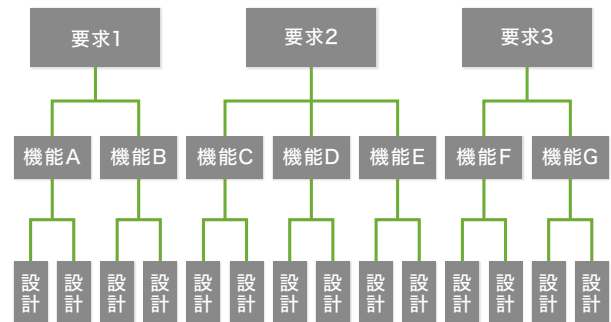


図6 ツリー構造型要求展開のイメージ

● ネットワーク構造型

要求と機能がN対Nの複雑な関係になる構造で、機能改良や派生開発、カスタマイズ開発に多く見られます（図7）。追加の開発を繰り返すたびに関係性はさらに複雑になるため、可視性や視認性に優れた専用ツールでないと追跡はほぼ不可能になります。

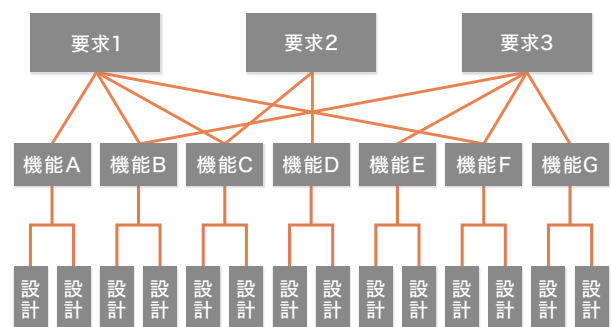


図7 ネットワーク構造型要求展開のイメージ

● 日本の開発事情を踏まえた要求管理ツールの適用マップ

システムのライフサイクルを縦軸、要求展開の構造を横軸とし、日本の開発事情も加味して各ツールをマッピングしたのが図8です。ツリー構造でライフサイクルの短い単発の開発であればExcelでも管理可能ですが、ライフサイクルが長くネットワーク構造になっていくと、使い勝手の面からも軽量ALM型が最適解になると考えられます。開発する製品の特徴を踏まえることが大前提となりますが、ツール選択の参考になりましたら幸いです。

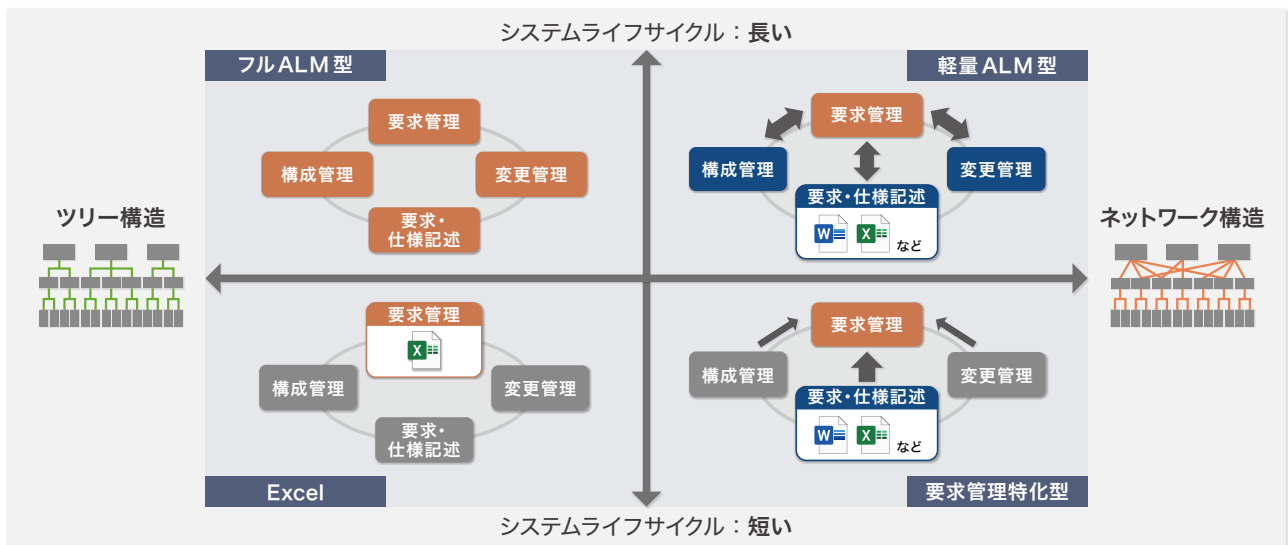


図8 要求管理ツールの適用マップ

📁 ConTrackの概要

ベリサーブが提供する「ConTrack」は、日本の開発現場にマッチした要求管理を実現するために開発したトレーサビリティ管理ツールです。要件管理機能と文書解析エンジンを中心に、Subversion・Redmine・Jira Softwareなどの実績あるツールと強力に連携して構成管理と変更管理を実施します（図9）。直感的で使いやすく、既存のドキュメントをそのまま活用できるなど、品質や生産性の向上に直結する要求管理ツールとして開発を進めてきました。



図9 ConTrackの機能概要

● ConTrack 事例紹介

ConTrackの導入により課題の解決に至った2事例を紹介します。

事例1 “超上流”での品質向上～自動車業界サプライヤー

<課題>

メーカーから最初に提示される要求仕様は曖昧で未確定な部分が多く、その後のコミュニケーションによって要求を固める形を取っている。ただ、仕様書・Q&A表・会議の議事録・メールなど、変更要求の形式がバラバラで情報が散在しているため、要求の矛盾に気付かないまま開発が進んでしまい、テスト工程で手戻りが起きることもあった。

<導入後>

さまざまな形式で提示される変更要求をConTrackで一元管理し、要件定義モデルや基本設計モデルとつなげていく。これにより、過去から現在までの変更要求の経緯が可視化でき、整合性が確保されて、後工程からの手戻りを大幅に減らすことが可能になった(図10)。

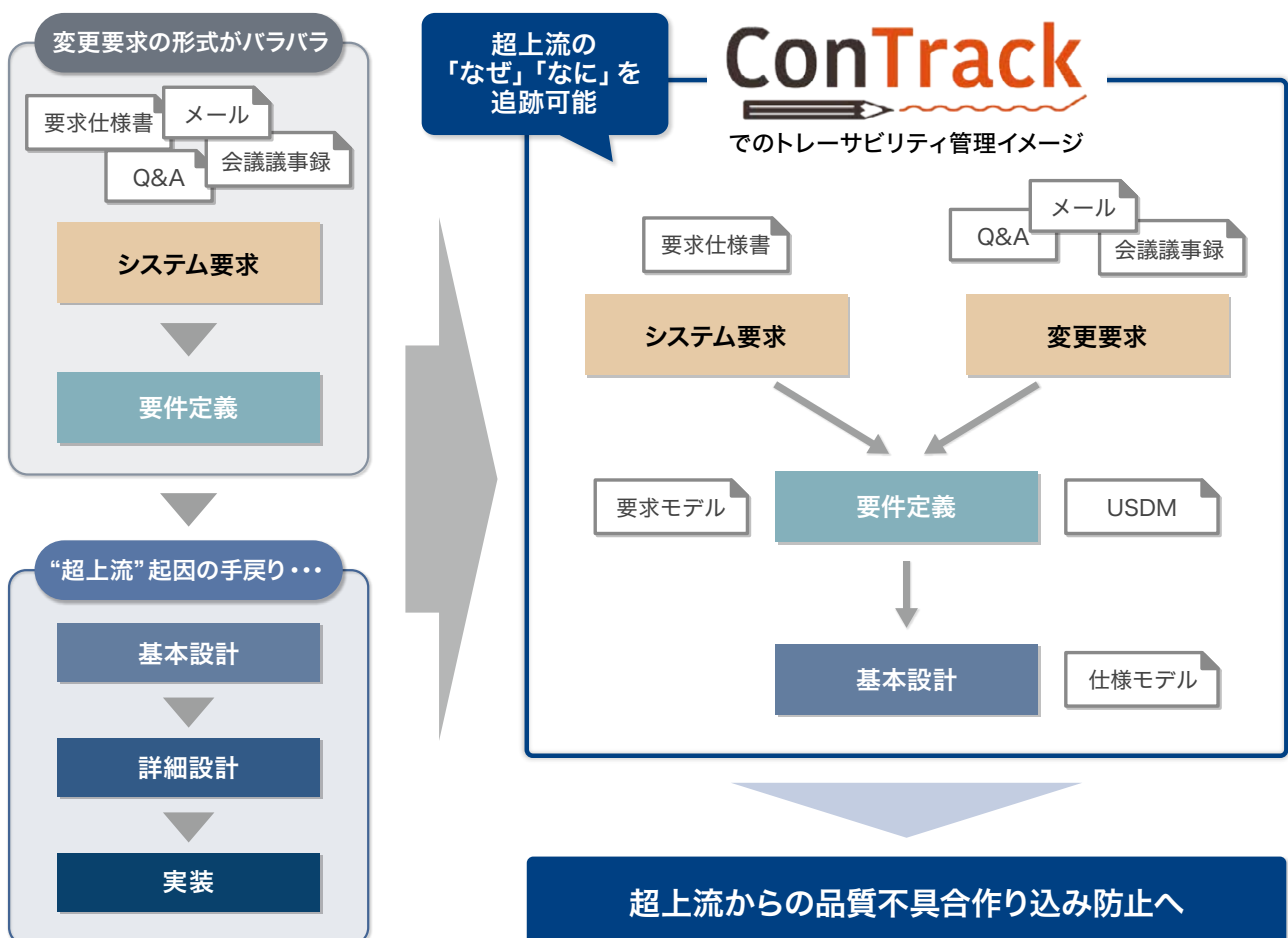


図10【事例1】“超上流”での品質向上にConTrackを活用

事例2 多品種開発での変更要求対応～組み込み産業機器開発

<課題>

ベースとなる汎用仕様の製品を顧客ごとにカスタマイズする業務形態で、開発は顧客別の派生ブランチで並行して進めている(図11)。複数の顧客から類似した変更要求があった場合にも各ブランチで個別に対応していたため、無駄が生じていた。さらに、共通部分に不具合が見つかり一つのブランチで修正したものの、別のブランチにはそれが反映されず品質問題が発生したケースもあった。

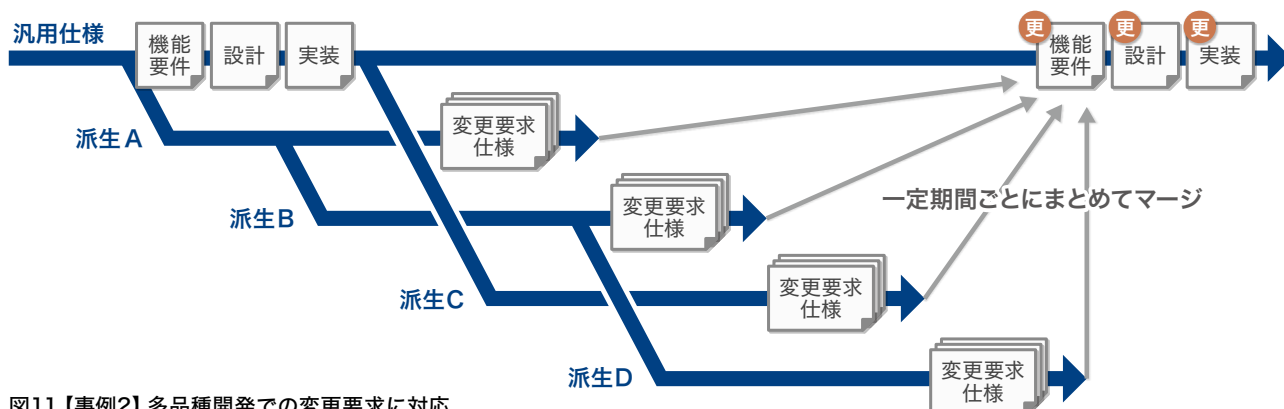


図11【事例2】多品種開発での変更要求に対応

<導入後>

汎用仕様の機能要件・設計・実装のドキュメントをConTrackで管理、これを各ブランチに置いて変更要求をトレースできるようにした。これにより、類似した変更要求に対応する重複作業が大幅に減少した。また、同じモジュールを流用しているブランチはどこかが即座に把握できるため、共通の不具合も迅速に解消できるようになった(図12)。

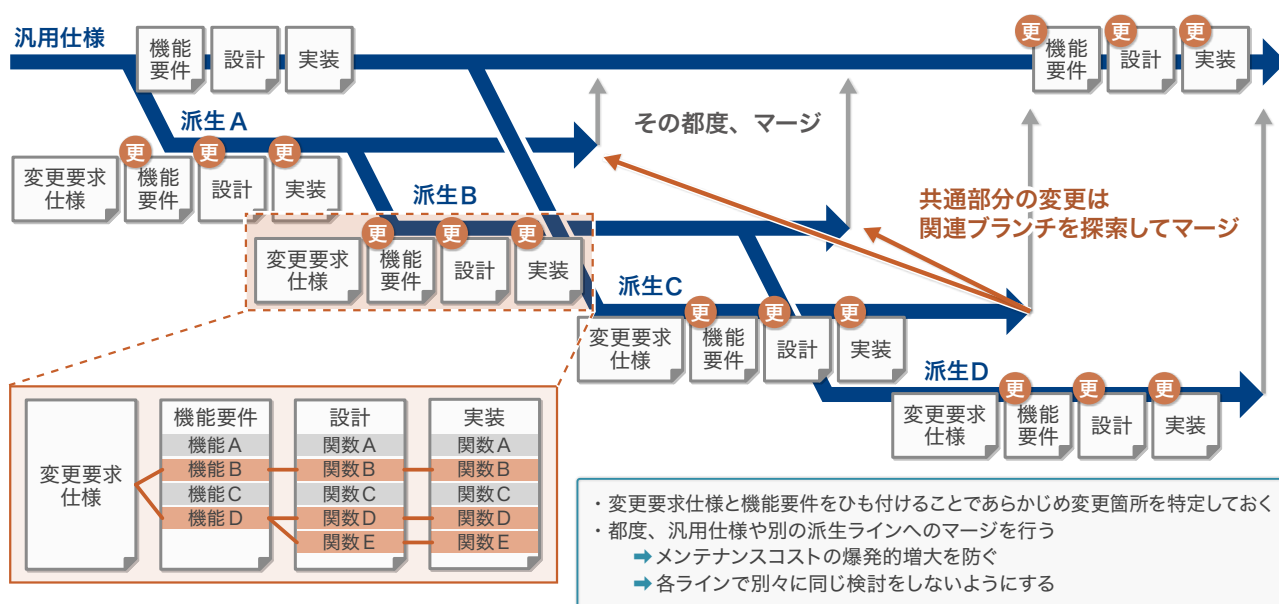


図12【事例2】多品種開発での変更要求対応にConTrackを活用

● ConTrackの新機能「ReqIF Adapter」

2021年6月にリリースしたConTrackの最新バージョンには、欧州の自動車メーカーやメガサプライヤーと取引する日本企業を支援する新機能「ReqIF Adapter」が追加されています。

欧州ではフルALM型要求管理ツールを採用する企業が多く、日本の取引先にも同じツールでのやり取りを求めるケースが増えています。取引先が1社なら指定のツールを導入すれば良いのですが、複数の欧州企業と取引する場合にそれぞれ別のツールを用意するのはコスト面で大きな負担になります。

「ReqIF」は米国の標準化団体による規格で、異なる要求管理ツール同士で要求や設計の情報をやり取りできるフォーマットです。ReqIF AdapterはConTrackでReqIFのインポート／エクスポートを実現する機能で、以下のようなことが可能になります（図13）。

- ReqIFの要求をConTrackに取り込み、それに基づいてWordやExcelで設計やテストを作成しトレースする
- ReqIFの要求が設計やテストに全て網羅されているかを確認する
- 開発終了後、結果を取引先に納入する際にReqIF形式で出力する
- 要求の変更や追加が発生した場合に、ReqIFの新旧バージョンを比較した差分分析をする。それにより、設計やテストの見直しに役立てることができる

📁 おわりに

ConTrackは、ほぼ年1回のペースでメジャーアップデートを行っており、2021年はReqIF Adapterの他にフローティングライセンスを追加、より柔軟な導入を可能にしました。フローティングライセンスとは、複数のメンバーで同一のソフトウェアを共用するための使用許諾契約（ライセンス）方式をいいます。ライセンスはサーバーで運用し、ネットワークで接続されたパソコンからライセンス数の範囲内でソフトウェアを同時に利用できます。これにより、利用するユーザ全員分のライセンスを購入する必要がなくなるメリットがあります。

日本発の製品という強みを活かし、日本の開発現場をより良くするための改良を今後も続けていきたいと考えています。

※ConTrackの詳細については、<https://www.veriserve.co.jp/service/detail/contrack.html>をご参照ください。

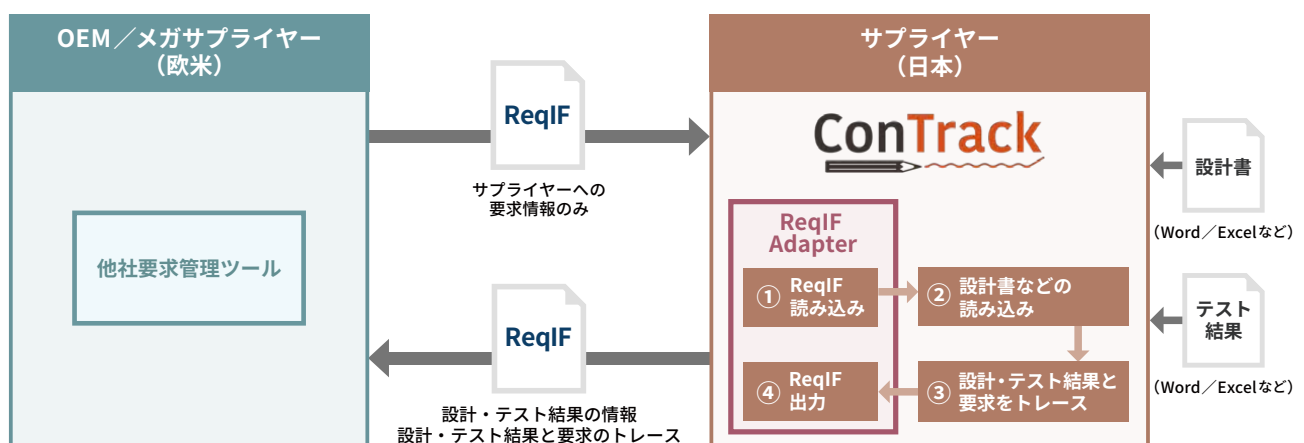


図13 ReqIF Adapterを活用した要求の交換イメージ

品質を創造する企業



Veriserve Navigation (ベリサーブナビゲーション) 2021年11月号

編集・発行：株式会社ベリサーブ 東京都千代田区神田三崎町3-1-16 神保町北東急ビル9階

お問い合わせ：広報・マーケティング部 TEL：050-3640-8194 MAIL：verinavi@veriserve.co.jp

＊本誌の記事中に掲載する社名または製品名は、各社の商標または登録商標です。