ベリサーブナビゲーション

# Veriserve Navigation

## 未来への技術、これからの品質

2020 June Vol.20

# 未来への技術、これからの品質

未来には、どのような技術やシステムがあって、
どのように品質が担保されるのでしょう。

未来にある技術と、今後の品質への取り組みについて考えます。

# Contents

## 特集 Ⅰ



## 特集 Ⅱ



## 特集 Ⅲ



## 特集 Ⅳ



## 特集 Ⅴ



## サービス紹介

# よりよい未来の実現に向け、強い信念を持って活動するSkyDriveに迫る

今回、我々ベリサーブナビゲーション編集部が取材する株式会社SkyDrive（以下、SkyDrive社）の設立の経緯は、約8年前まで遡ります。2012年9月、さまざまな企業のメンバーが集まり、「空飛ぶクルマ」の開発に取り組む業務外有志団体であるCARTIVATORが発足しました。「空飛ぶクルマ」の開発をさらに加速させるべく、そのメンバーが中心となって、2018年7月に設立されたのが、SkyDrive社です。

現在、SkyDrive社は2つのテーマにおいて、事業化に向け、取り組んでいます。一つは、無人機であるカーゴドローン、もう一つは有人機である「空飛ぶクルマ」の開発です。今回、我々は、代表取締役の福澤 知浩氏、カーゴドローンの事業開発を担当する佐藤 剛裕氏、「空飛ぶクルマ」の開発を担当する山本 賢一氏の3名に、これまでのSkyDrive社の歩みや、それぞれが大事にしていること、「空飛ぶクルマ」の実現に向けてどんなことに取り組んでいるのか、お話を伺いました。

**想定仕様**

| 名称 | カーゴドローン | 「空飛ぶクルマ」コンセプトモデル |
|---|---|---|
| サイズ | 全長1.3m　全幅1.7m　全高1.0m | 全長4.0m　全幅3.5m　全高1.5m |
| 飛行速度 | 40km/h | 100 km/h |
| 飛行時間 | 15分 | 20~30分 |
| 用途 | 土木・建設現場での資材運搬　災害時の物資運搬<br>重農作物の運搬（積載荷重30kg） | 人を乗せての移動（最大搭乗人数2名） |
| 機体イメージ |  <br>©SkyDrive |  <br>©SkyDrive / CARTIVATOR |

©SkyDrive / CARTIVATOR

＊本記事に掲載されている機体および人物の写真は、株式会社SkyDriveより提供されたものです。

## SkyDrive社とCARTIVATOR

CARTIVATORは、自動車・航空業界・スタートアップ関係の若手メンバーを中心とした有志団体として発足しました。『モビリティを通じて次世代に夢を提供する』ことをミッションに、「空飛ぶクルマ」の技術開発と事業開発に取り組んでおり、2014年には5分の1スケールの無人試験機の飛行に成功しました。CARTIVATORの由来は『クルマ(CAR)でワクワクする体験を生み出す(CULTIVATOR)』という想いからきています。

当初はメンバーで費用を負担し合い、モーターやバッテリーなどを購入していましたが、2017年にスポンサーを募ったところ、数千万円の支援を受けることができました。これにより、これまで費用面で手が届かなかった資材調達が可能となりました。2018年、目標としている2020年のデモフライト実現への開発加速化と「空飛ぶクルマ」の事業化を進めるべく、CARTIVATORの一部メンバーがフルタイムで参加するSkyDrive社が設立されました。SkyDrive社の社名はCARTIVATOR時代に、機体の名前をSkyDriveとみんなで決めたことに由来します。SkyDrive社創業時に、この機体へのみんなの想いとリスペクトを込めて踏襲し、「空を飛ぶだけではなく、空を走り、クルマのように日常的に使う移動手段としての市場を創造していきたい」という意志を込めて、社名を決定しました。

現在、SkyDrive社とCARTIVATORは、2020年夏のデモフライトに向けて共同開発に取り組んでいます。

## TIME LINE

CARTIVATOR

2012.09 有志団体 CARTIVATOR 発足

2014.01 空飛ぶクルマ「SkyDrive」の開発に着手

2014.07 無人試験機 1/5 スケールの走行・浮上に成功

2017.09 SD-01 設計開始

SKYDRIVE

2018.07 株式会社 SkyDrive 設立

2019.06 豊田市と連携協定締結

2019.11 カーゴドローン実証実験

2019.12 空飛ぶクルマ有人飛行試験開始

2020.05 カーゴドローン 販売開始

2020.08 空飛ぶクルマ デモフライト

2023.XX 空飛ぶクルマ 販売開始

2030.XX 空飛ぶクルマ 自動運転化

# 「空飛ぶクルマ」でモビリティの歴史に変革をもたらしたい

代表取締役

**福澤 知浩**氏
ふくざわ ともひろ

東京大学工学部卒業。トヨタ自動車にて自動車部品のグローバル調達に従事。同時に多くの現場でトヨタ生産方式を用いたカイゼンをし、原価改善賞受賞。2014年にCARTIVTORに参画し、共同代表に。2017年に独立し、製造業の経営コンサルティング会社を設立。20社以上の経営改善実施。
2018年に株式会社SkyDriveを創業、代表に就任。

――SkyDrive社の現在の状況について、教えてください。

2019年12月に有人飛行試験を開始しました。直近では、2020年夏のデモフライト、カーゴドローンの販売、2023年の「空飛ぶクルマ」販売開始に向けて取り組んでいます。いずれもハードルが高いのですが、多くの人が欲しい、乗ってみたいと思う製品づくりに向けて、社員一丸となって取り組んでいます。

――重点的に取り組んでいることは、どんなことですか？

SkyDrive社の事業を加速させることが私のミッションだと思っています。例えば、採用面だと「最強」な人を連れてくることですね。具体的には、問題があるから諦めるのではなく、問題をどう乗り越えていくのか、解決することを楽しめるような人です。そんなマインドを持つ人を採用し、最大限力を発揮しやすい環境を作ることが私の役割です。また、政府や関係会社の方など、我々へのサポーターが増えてくるにつれて、メディアに取り上げてもらえる機会も増えています。それがSkyDrive社の認知につながり、これまでスタートアップに興味を持っていなかった、夢を実現させる志のある人に来てもらえることを期待しています。

また、スポンサーに認知してもらうことも重要です。ビジネス誌などを読んで、SkyDrive社を知り、出資を検討してもらえることもありますが、実はテレビの特集番組を見たスポンサーご自身の家族から


「空飛ぶクルマって知ってる？」と言われるほうが、出資を前向きに考えてもらうことに効果的なんです。そんな風に、知ってもらうきっかけを多くするために、海外の講演に登壇したり、メディアに出たり、興味を持ってくれた方の元へ自ら出向き、直接話したりするなど、認知度を高める活動を積極的に行っています。

現在は採用活動もスポンサー募集も行っていますが、直接の関わりがなくても、活動状況の進捗を、FacebookやTwitterなどのSNSで見て、知ってもらったり、話題にしてもらえたりするだけでも嬉しいです。

### ――仕事をする上で、大事にしていることはありますか？

設立当初からのメンバーも加えると、今では総勢50名以上となっています。コミュニケーションを円滑に進めるために、全メンバーとの1on1を行っています。[*1] 1on1をやることにより、色々な要望が出てくることもあるので、組織全体に反映できる場合には反映し、反映できなかったとしても、フィードバックしたり、議論したりする機会として活用しています。こういった場を設けることは、双方にとってとても意義のあることだと思います。

また、社内のコミュニケーションツールは、メールではなく、チャットツールを使っていて、タイムリーなやりとりを心掛けています。そうすると、社内全体でそれが普通となり、返信を早くすることで、判断の待ち時間が減り、結果的に物事が早く進捗します。

もう1点、心掛けているのは、あまり細かなことまでは口を出さないことです。基本的に、メンバーが考えたことを優先するようにしています。ただ、どんな考え方で物事を進めようとしているのか、その方向性だけは合わせるようにしています。判断する際に、判断するための材料をしっかりと揃えた上で、ロジカルに考えることを徹底してもらっています。

### ――「空飛ぶクルマ」を開発する意義と、SkyDrive社の目指す未来について教えてください。

歴史上、モビリティの変革の機会は滅多にありません。自動車においては、T型フォードが誕生し、大量生産が始まってから既に100年以上経っています。[*2]そして、現在、飛行機と自動車の良さを併せ持つ、「空飛ぶクルマ」が生まれようとしています。「空飛ぶクルマ」により、日常的に空を飛べるようになれば、生活に関しても相当大きなインパクトがありますし、100年後、200年後の未来にも「空飛ぶクルマ」が存在していると確信しています。そんな歴史に残るモビリティを事業化することにはやりがいを感じます。

我々のビジョンとして、2030年頃に「空飛ぶクルマ」の自動運転を実現させることを目標にしていますが、さらにその先の2040年頃には、乗り捨てサービスを実現させたいと思っています。携帯で「空飛ぶクルマ」を呼んで乗り、目的地までたどり着いたら、車が勝手に充電ポートまで戻っていくようなサービスですね。実際にそんなサービスが始まると、点から点への移動ができますし、その主要なブランドとしてSkyDrive社が存在している未来を実現したいと思っています。

＊1：1on1：ワンオンワン。定期的に上司と部下が1対1で話し合うことを指す。

＊2：フォード・モーター・カンパニーにより、1908年にT型と呼ばれる自動車のモデルが発売された。1927年に生産が終了するまでの生産台数は1,500万7,033台に上った。

続いて、SkyDrive社でカーゴドローンの事業開発を担当する、佐藤 剛裕さんへのインタビューです。

# より安全に、より簡単に物資が運べる未来に向けて、カーゴドローンの普及を目指す

カーゴドローン部 部長

**佐藤 剛裕**氏
さとう たけひろ

東京大学薬学部卒業。同大学院修士課程修了。総合系コンサルティングファームで、製造業・小売業・建設業などの業界における新規事業開発や業務改革プロジェクトを遂行。2017年7月よりCARTIVATORに参画し、総務HRリーダーとして、プロボノメンバー加入の運用設計や組織設計に従事。2018年10月より株式会社SkyDriveに参画し、事業開発担当として空飛ぶクルマの市場創造に取り組む。

**――SkyDrive社での佐藤さんの役割について、教えてください。**

私はSkyDrive社の設立時から参画しており、当初は、勤怠管理や経費精算など、会社の機能として必須である業務全般を担当していました。現在はカーゴドローンの事業開発を中心に取り組んでいます。SkyDrive社では、「空飛ぶクルマ」よりも先に、事業として市場にリリースできるものとして、当初からカーゴドローン事業を構想していました。設立当初はカーゴドローンに関わる人数も少なかったのですが、2019年7月頃から10名弱程度のメンバーで本格的に取り組みを始めました。これまで、プロトタイプ機を使っての実証実験を進めてきましたが、2020年5月から販売を開始していますので、今後は販売体制の強化を進めてまいります。


カーゴドローン：重量物運搬用のコンパクトなドローン。積載重量は30kg。高低差がある環境でも空を介して自動で運搬可能。資材の運搬に活用することで、労働負荷の削減、安全性・生産性の向上が期待される。

**――カーゴドローンの技術課題や、事業に関わる法整備の状況について教えてください。**

カーゴドローンの開発を進める上での課題は多くありますが、技術的な面で言えば、バッテリーの改善が挙げられます。カーゴドローンの稼働時間を増やすためには、バッテリーの改善が必要不可欠ですが、著しく改善がなされる技術分野ではありません。ただ、1年ごとに、約8%ずつバッテリー効率が改善してきた実績があるので、その実績を前提に我々も予測を立て、カーゴドローンの開発活動を進めています。産業が立ち上がれば技術が進化し、技術が進化すれば産業もさらに広がっていくと考えていますので、各省庁にも技術課題を認識してもらい、投資が積極的に行われれば、カーゴドローンに必要な技術開発のスピードも加速していくと思っています。

また、現在、政府がドローンの利活用推進を進めようとしており、2022年にドローンについての法改正が行われる見込みです。その法改正では、これまで人の上を飛ばしてはいけなかったドローンが、法規を満たしている場合に限り、人の上を飛行することが可能になるかもしれません。これは非常に画期的なことだと思っています。ユースケースの幅が広がり、ドローン活用の可能性が大きく広がるからです。我々としては、そういった法改正を見据えながら、「この時期にこんな機体を出すことのできるよう、開発を進めよう」と取り組んでいます。

――カーゴドローン事業における、今後の取り組みを教えてください。

カーゴドローンの販売に向けて、現在のプロトタイプ機に次ぐ新たなモデルの開発・製造を行い、実機としての使い勝手を改善していく必要があります。もちろん安全な機体を作ってはいますが、実運用では、一日に複数回の往復フライトをする可能性もありますし、よりハードな要求に耐えられるよう、品質をより向上させなければなりません。これからカーゴドローンを販売し、事業として成り立たせるためには、これまでの開発スピードは維持しつつも、安全な製品を納めていく品質管理の仕組みとルールづくり、その運用を行うことのできる体制構築にも取り組む必要があると認識しています。同時に、販売施策の検討も進めなければなりませんので、課題もやることも山積みですね。

SkyDrive社は、社会的な課題を解決するための事業というよりも、こんな技術が活用される可能性があるのではというプロダクトアウトから始まっています。今、皆さんがお使いのスマートフォンも、当初はどれだけ便利か分かりませんでしたが、使ってみたら便利で、今では世界中に普及していますよね。

ドローンも既に撮影用途などで、一般にも普及し始めていますが、「重いものを運搬する」という用途での使用方法は、あまりイメージが沸かないかもしれません。我々が開発するカーゴドローンは、物資や資材の運搬用途での普及を目指しています。現在、人が資材を運搬するのに時間がかかっている現場でも、カーゴドローンを導入できれば、生産性の向上につながります。「これまでドローンを使ったことがないし、必要ない」「事故が起きる可能性がある」との声もあるのですが、「カーゴドローンの導入により、これまでにない価値をもたらすことができる」という確固たる自信、信念をもって取り組んでいます。

©SkyDrive

続いて、「空飛ぶクルマ」の開発に挑む、技術渉外責任者の山本賢一さんへのインタビューです。

## 自らの目指す未来をつかむため、チャレンジしたからこそ、今がある

技術渉外責任者

**山本 賢一**氏
やまもと けんいち

東京工業大学大学院修了。
トヨタ自動車にてロボットの開発に従事(内1年はシリコンバレー赴任)。2015年、CARTIVATORに参画。実験担当と制御やインターフェースのMBD環境整備のチームリーダーを経て、現在はプロジェクトマネージャーを務める。株式会社SkyDriveでは、技術管理、機体開発プロジェクト管理など、技術渉外責任者を担う。

――SkyDrive社での山本さんの役割について、教えてください。

私は、技術渉外責任者という立場で、技術的見地からSkyDrive社の戦略を考えることが役割です。共同研究先との研究において、どのようにシナジーを生み出していくのかを考えたり、資金調達に携わったりと、さまざまな業務に幅広く関わっています。元々、工学部出身で、前職では医療機器の開発を担当しており、リスクアセスメントを強みとしていたことから、協力会社との製品の安全性解析業務の窓口も担当しています。

――2019年12月に有人飛行試験を開始されていますが、「空飛ぶクルマ」の開発で、特に苦労してきた点はどんな点ですか?

「空飛ぶクルマ」の開発においては、機体のスケールが大きくなるにつれて課題が増えていきました。まず、大きな機体を飛ばすことは難易度が高いのですが、機体が大きくなるに連れ、指数関数的に難易度が増します。小さな機体では、全体をほぼ剛体として捉え

られていたものが、大きくなると変形量を無視できず、機体制御を成功させるまで長く時間がかかりました。小さな機体を実験する際には、部屋で飛ばすこともできましたが、大きくなると、部屋で飛ばすわけにもいかず、場所の確保が必要になります。また、バッテリーの充電などの準備にも時間がかかるようになり、PDCAサイクルを回しにくくなっていきました。

次に、人を乗せるという課題があります。人を機体に乗せることを想定すると、外からラジコンのように機体をコントロールしていたものを、実際の運転ではどのようにコントロールをするかなど、小さな課題がどんどん顕在化してきて、解決に時間がかかるようになりました。

それでも開発スピードを維持するために、参考にできるものは積極的に取り入れていました。空飛ぶクルマは、これまでにない新しいモビリティではありますが、要素技術に分けると、世界中どこにも存在しないというわけではありません。例えば、航空機の椅子やコックピットの画面など、既に実現しているものを参考にしていました。


——今後の「空飛ぶクルマ」開発における技術的な課題を教えてください。

これまでは、大きな機体を安定して飛ばすことや、人を安全に乗せるということを目標としており、課題を認識しやすいフェーズでした。今では、そのフェーズが終わり、今後は実安全と計算上の安全、乗る人の安心感などを向上させていく必要があります。安心・安全を向上させるためには、安全に飛行できる「空飛ぶクルマ」を開発して終わり、というわけにはいきません。自動車でいう車検のようなメンテナンスをする技術者の育成も、安全を守り続けるには必要となります。さらに「空飛ぶクルマ」に安心して乗ってもらうためには、安全性解析の分野には入りませんが、飛行時の機体の振動なども、抑制していかなければなりません。

これまでは、機体が落ちてしまったら、その時に壊れてしまった部品が駄目だとか、ちゃんと飛ばないから駄目だという、飛行できるか否かに焦点が当たっていましたが、徐々にフェーズが変化してきたことを実感しています。具体的には、多くの選択肢の中から、何をやるのか、どんな風に進めるのかを決めて、取り組んでいく必要がある段階です。例えば、コスト面や操作性を考慮した際に、本当にこれまで通りの進め方でいいのか、進め方を変えるのであれば、どのように変えていくのかを検討しなければなりません。

——SkyDrive社に参画して、1年半が経った今、山本さんが感じていることを教えてください。

実は、私が今、SkyDrive社にいるのは、前職でのシリコンバレーへの赴任がきっかけでした。6年前、前職のトレーニングプログラムに立候補し、1年間シリコンバレーに行くことになったのです。現地では、技術系のミートアップが多数あり、そこでできたベンチャー企業の友人から、多くのインプットを得ることができました。

その楽しさが忘れられず、帰国後、自社技術者の有志団体のイベントに参加したのがきっかけで、CARTIVATORを知り、参画することになりました。その後、SkyDrive社に入社することを決めたのは、『今の仕事を続けるよりも、「空飛ぶクルマ」の開発に本気で取り組む方が面白そうだ』と思ったからです。

前職の仕事を続けていたとしたら、自分のアウトプットを実現しやすい立場で働くには、10年近く時間が必要だったと思います。会社を辞め、現在、SkyDrive社の技術渉外責任者という立場で、エンジニアリング以外の資金調達や企画にも携わることができ、自分自身の可能性がより広がったように感じています。

このような経験を経て、自分自身の未来に対する当事者意識を持ち、「こうであったらいいなあ」という未来を実現するために、チャレンジすることが重要だと感じるようになりました。私にとっては、会社を辞めて、「空飛ぶクルマ」の開発に専念するということがチャレンジでした。もし、やりたいこと、実現したいことがあるならば、考えるだけではなく、まずはアクションをとってみることが大事なのではないかなと思います。

## インタビューを終えて

今回はお忙しい中、3名の皆さまにインタビューにご協力いただきました。このインタビューを通して、皆さんが強い信念を持って、よりよい未来の実現に向けて取り組んでいるという印象を持ちました。SkyDrive社の「空飛ぶクルマ」や「カーゴドローン」が社会の中で活用され、大きな変革が起きる瞬間を、ぜひ見届けたいと思います。ご協力いただき、ありがとうございました。


SkyDrive メンバー 飛行試験場にて

本記事の内容に関するご質問、お問い合わせは、**ベリサーブ 広報・マーケティング部** Email:verinavi@veriserve.co.jp までお寄せください。


中尾 悠里氏
なかお ゆうり

株式会社富士通研究所
人工知能研究所
トラステッドAIプロジェクト

株式会社富士通研究所
人工知能研究所 研究員
専門は科学技術社会論。
人々の心理・法規制といった社会的要因によりテクノロジーが変化するプロセスに関心を持つ。
現在は、推薦システムや公平性配慮型機械学習など、人や社会との境界領域での工学研究に取り組む。
人工知能技術と社会の関係を議論する講演活動も行っている。


# 意思決定を真に支援するAIを目指して

## これからのAIとの付き合い方

### はじめに

人工知能（AI）技術一般について、社会的にさまざまな問題が指摘されていますが、同時にコンピュータサイエンスの分野からはその解決策が提案され続けています。本稿では、「説明可能AI」「公平性配慮型機械学習」といった先端的な技術開発の方向性を紹介すると同時に、今後のAIが流通する社会で求められる技術との付き合い方をご紹介します。

## AI技術が人間の意思決定に対してもたらす問題

現在、AI技術は我々の生活に無くてはならないものになっています。しかし、同時にさまざまな問題が指摘されています。その中には技術的な限界だけではなく、社会との接点におけるトピックも挙げられます。代表的なものとして、AI技術によって人間の思考が誘導されるという問題があります。フィルターバブル[*1]やソーシャルエコーチェンバー[*2]といった用語で指摘されているもので、AI技術を使った検索や推薦では、検索結果や推薦内容にユーザー自身の社会的な偏向が反映され、偏ったコンテンツだけが提示されるという現象を指します。我々が日常的に利用する検索エンジン、動画サイト、SNSといったサービスは裏で機械学習を利用したシステムが機能し、ユーザーの利用履歴から学習を行い、ユーザーが見たいであろうコンテンツを提示するように設計されています。そのため、ユーザーの嗜好を深く学習したシステムでは、次第にユーザーに都合のいい情報しか出なくなります。しかも検索エンジンなどのサービスのインターフェイスは、複雑な手順をなくし、使い慣れた操作で済む心理障壁を下げる設計がされているため、ユーザー自身がこの思考の誘導に気付くことはほとんどありません。

AI技術は
我々の生活に
無くてはならないもの
しかし、同時に
さまざまな問題が
指摘されている

次に、AIのモデルが性差別や人種差別といった差別的なバイアス（男性、白人といったある属性値に出力が偏ってしまうこと）を持った結果を出してしまう問題があります。過去、Googleの検索やアメリカの司法の文脈などで指摘されてきました。例えば、Googleの検索窓に

＊1「インターネットの検索サイトが提供するアルゴリズムが、各ユーザーが見たくないような情報を遮断する機能」（フィルター）のせいで、まるで「泡」（バブル）の中に包まれたように、自分が見たい情報しか見えなくなること。（「フィルターバブル」『フリー百科事典 ウィキペディア日本語版』より）

＊2 閉鎖的空間内でのコミュニケーションを繰り返すことによって、特定の信念が増幅または強化される状況の比喩である。（「エコーチェンバー現象」『フリー百科事典 ウィキペディア日本語版』より）

アフリカ系の名前を入れると、その人に全く逮捕歴がなくても逮捕歴に関する結果がサジェストされてしまうという事例があります。その他にも、2016年にアメリカにおいて、一度刑務所に入った犯罪者を保釈するかどうかを決定する際に再犯予測を行うCOMPASという機械学習ベースのシステムに人種差別的なバイアスがあることが指摘されました。同じ条件の人であれば、白人より黒人の再犯率を高く見積もってしまうというバイアスです。これらのバイアスは、データを基にしたAIを用いて意思決定を支援するというアプローチにおいて、過去の意思決定を集約した教師データをAIに学習させていることに起因します。教師データの中には社会的に行われてきた差別の結果が反映されてしまっているため、そのデータを用いて学習を行うことで、新たな意思決定を行うための機械学習モデルに差別的な偏向が含まれることになります。AIによる差別は主に欧米で指摘されている問題ですが、日本においても就職時や大学入試における男女差別が指摘されており、機械学習ベースの意思決定支援システムが広がれば広がるほど、顕在化する問題だと思われます。

## AIと社会の接点に関する問題への技術的解決策

上記で指摘された誘導的な判断や差別的な判断を是正する工学的な試みを二つ紹介します。

### 説明可能AI（XAI）

思考や行動の誘導のようにAI技術が人間に与える暗黙的な影響を是正するための取り組みとして、AIが出した結果を人間に対して説明可能（Explainable）にしようとする説明可能なAI（Explainable AI, XAI）の研究が挙げられます。この分野では、画像認識において画像のどの部分がAIの認識に効果を持つか、データのどの特徴量（あるデータを構成する特徴のこと）が分析結果に寄与しているかを説明する技術などが特に有名です。

「説明可能AI」の研究は、これまでの精度を競うことを第一目標にしていた工学研究と異なります。説明可能AIの研究において特徴的なことは、AIが出した説明に対し、人間が行う評価自体が技術革新の評価基準となる点です。結果に寄与している特徴量を特定できたとしても、人間とAI技術間の情報の授受を成り立たせるためには、特定した特徴をユーザー自身の判断基準に基づいて認識する必要があります。これにより、出力された説明が人間に適切に認識されたかを評価するために、人間側のリアクションを

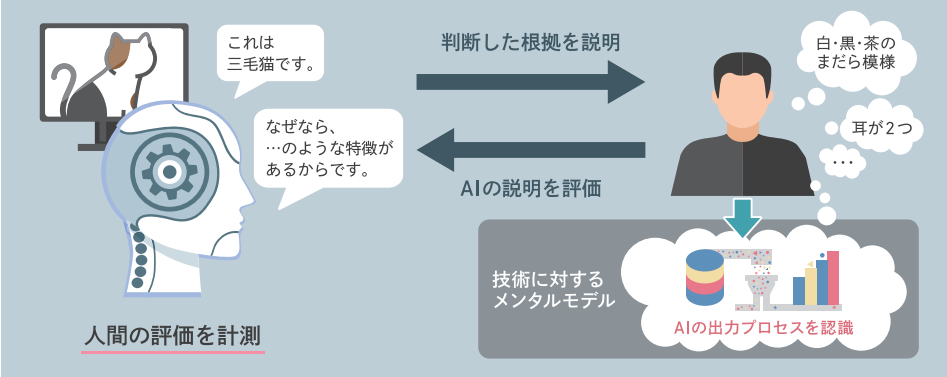



図1 説明可能AI（XAI）

適切に計測する必要が生じます。同時に、AIからの説明を理解するためには、人間側にもAIからの出力がどのようなプロセスで提示されてくるのかといった認識がある程度備わっている必要があります。この認識のことを「技術に対するメンタルモデル」といいます。「説明可能AI」は、理想的にはユーザーがシステムを利用する間に適切なメンタルモデルを構築できるようになっていることが求められています（図1）。

### 公平性配慮型機械学習

一方で、特に機械学習モデルに関する差別的なバイアスに対して、そもそもAIからの出力自体を公平なものにするために、データやモデルの中に含まれる差別を取り除こうとする「公平性配慮型機械学習」の研究開発が2010年前後から盛んになっています。学術界だけでなく、Google，Microsoft，IBMといった企業が最新の研究成果を公表し続けており、社会における公平性や透明性といった問題を取り扱った国際会議が複数設立され、毎年開催されるようになりました。

「公平性配慮型機械学習」の基本的な前提は、人間が行う意思決定は社会に存在するバイアスから自由ではないという事です。公平性配慮型技術を使うことで、差別をなくしたい属性（性別や人種など）を指定しさえすれば、機械的にデータや機械学習のモデルから人種差別や性差別を取り除くことが可能です。これにより、人間の側にあるバイアスをAI技術で正していこうとすることが基本的な方向性です。現在では、さらに発展して、「この技術をどういう文脈で使うべきか？」という文脈での議論が展開されています。例えば、ある属性の人たちに対して、少数派の人を優遇することで、同じくらい優秀である多数派が採用されにくくなる、という事が生じ得ます。優秀な人を採用すべきだという考え方からすると不当ですし、一方で、少数派というだけでこれまで蓄財や教育の機会を奪われてきたという観点からすれば、弱者を是正する改善（アファーマティブアクション）の一環として妥当な処置かもしれません。どのようなポリシーの下で差別を取り除くかについては機械的に決められません。現在では、情報系の学会が主催する国際会議においても、法学者や社会学者、哲学者が入り乱れてどのように技術を使うべきかについての議論を行っている姿が見られます。

AIは
印象や感情で
決定を左右される
ことはない

しかし、一旦ポリシーを決めてしまえば、AI技術は人間の専門家が行うよりも公平な意思決定を行うことができます。AIは印象や感情で決定を左右されることはないため、ある意味で人間より信頼度の高い意思決定を下すことができるのです。

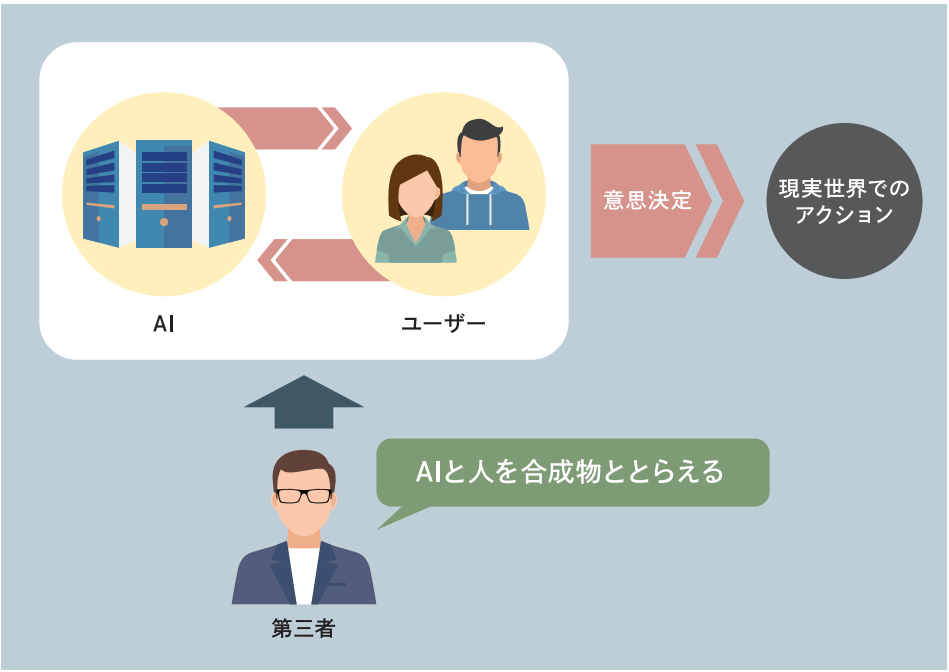



図2 第三者的な視点

## どちらか一方が正しいわけではない人とAI

「説明可能AI」や「公平性配慮型機械学習」は共に倫理的なAI（Ethical AI)や信頼されるAI（Trusted AI)と呼ばれます。ここまでに紹介した技術において前提となっているのは、AIの出した結果が基本的に正しいということです。そのため、従来では、人の側を評価する際に、どの程度AIの結果を信頼して行動に移れるかという事が考慮されてきました。

しかし、実際に意思決定を行わなければならないユーザーの視点に立つと、この前提は適切ではありません。なぜなら、ユーザーに結果を鵜呑みにさせることを前提として開発されたAIは、単に効果的な人間の誘導を狙っていることなります。この狙いはAIからの潜在的な影響を顕在化させて人間にとって自由な意思決定を実現するという目標と矛盾します。

では、AIを信頼せずに人間の思うとおりに意思決定をすればよいのかというと、これも正しくはありません。なぜなら、公平性配慮型機械学習の文脈からすると、人間の意思決定は社会的に望ましくない偏向を含んできた可能性があり、常に信頼の置けるものとは限らないからです。

## 不可分な合成物であるAIと人間、それをサポートするためのインタラクティブな「説明可能AI」

上で述べた人間がAIの誘導から自由になるべきだという考え方と、AIの方が合理的な判断を下せるので信頼すべきだという考え方は一見矛盾します。しかし、二つの考え方はどちらも合理的です。これらの立場を理論的、実践的に統合するために、技術哲学的な意味で第三者的な視点を導入する必要があります。第三者的な視点というのはAIとそれを使っている人間の二者から一歩引いた視点のことです（図2)。

AIは、ユーザーの目線から見ると技術的人工物という他者です。しかし、第三者的な視点から見ると意思決定を共同で行うとき、AIとユーザーは不可分な存在です。ユーザーの立場からするとAIに薦められたから意思決定をしたと感じるかもしれませんが、ユーザーは必ずしもAIが言ったことを鵜呑みにして決定したわけではないはずです。つまり、第三者的な視点に立てば、ユーザーは意思決定を行う際に単に「AI技術の結果から影響を受けた人」というだけの存在ではありません。AIのサポートを受けて行った意思決定はAIだけで行った決定でもないし、ユーザーがひとりで行った決定というわけでもないという意味で、近年の技術哲学の文脈では、意思決定時の人間とAI技術は不可分な「合成物」である、と言います。

人間とAI技術は
不可分な
「合成物」である

AIと人間の合成物による意思決定をサポートするためには、人間を説得することを狙う従来のAI技術の方向性だけでは不十分です。そこで必要になるのが、「説明可能AI」をインタラクティブ（対話的・双方向的）にすることです。インタラクティブにするとは、「説明可能AI」がユーザーに対して出した結果の根拠を説明し、ユーザー側は批判的に考え、納得できなければAIに修正を求めるフィードバックを返せるようにするという意味です。これにより、ユーザーがAIと共に熟考を重ねた上で意思決定を行うことが可能になります。ユーザーの説得を試みるのでなく、ユーザーとのインタラクションを誘発するための説明を行う、というのが我々の考える研究の方針です（図3）。

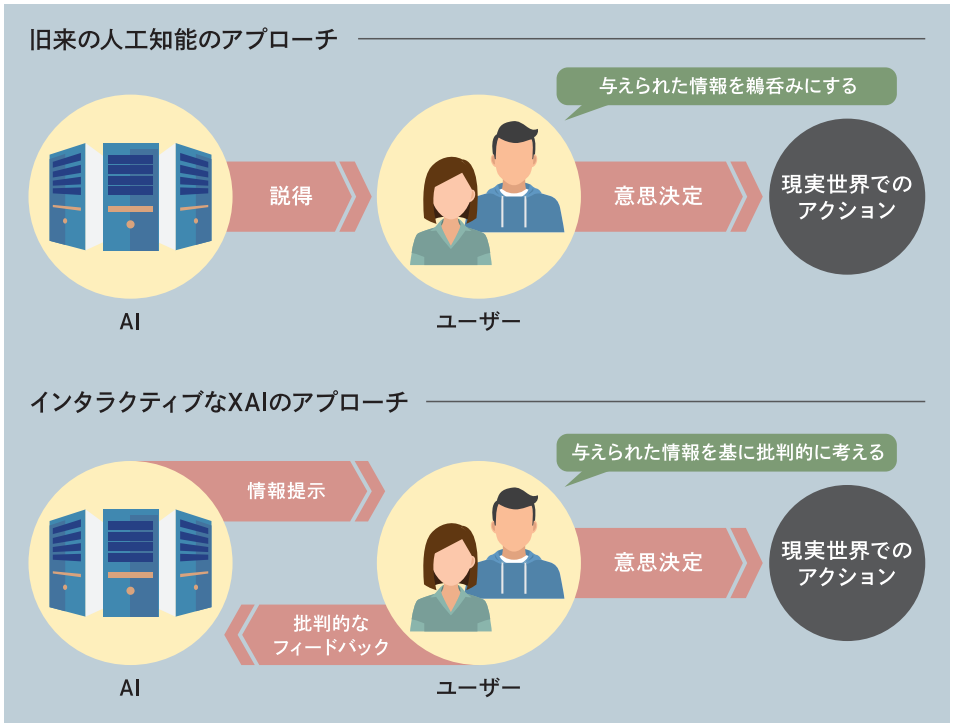



図3 インタラクティブなXAIのアプローチ

## ユーザーとインタラクションするマッチングAI：糸島市の事例

ユーザーとインタラクションするAIの一例をご紹介します。インタラクティブな「説明可能AI」のプロトタイプとして、私たち富士通研究所は福岡県の糸島市役所様と共同で、糸島市に移住したいと思う都市部に住む方に向けたマッチングAIを開発しました。都市部に住んでいる人は地方自治体の事前知識を持たないため、実際の現地の状況を学習しつつ自分が移住したい地域を探す必要があるのですが、このマッチングAIでは、糸島市内の行政区（町内会）ごとに、どの地域にユーザーが住むべきかをインタラクティブに推薦することが可能です。ユーザーへの地域推薦は、移住希望者・移住経験者の方へのインタビューに基づいて作成した病院・交通利便性・安全性といった地域指標のスコアに基づいて行われます。さらに、なぜユーザーにその地域がお薦めされているかが地域指標ごとの棒グラフとして表されます。そして、ユーザーは棒グラフを操作して値を変更することで、推薦地域のランキングを自分の希望に応じて変えていくことができます（図4）。

また、AIの側もユーザーが変えた地域指標の重視度を基にして、ユーザーの特性を学習していきます。このシステムでは、なぜユーザーにその地域が推薦されているかが示されると同時にその根拠自体をユーザーが変更することでAIと共同で意思決定を進めていくことが可能になっています。

近年のGDPR（EU一般データ保護規則、General Data Protection Regulation）などのデータ法制では、AI技術などで自動的に行われる意思決定については、意思決定プロセスを振り返ることができる材料をユーザーに対して提示する仕組みにすることが求められています。そのため、このマッチングAIに見られるような、意思決定においてユーザー自身がフィードバックを返すのに足る情報を与える技術を開発するという流れは、次第に一般的になってくると思われます。

しかし、技術が人間からのフィードバックを受け付けるようになっても、今度は人間側がAI技術の変化に対応していく必要があります。なぜなら、人間はAIからの応答を「答え」のように見なし、真に受けてしまうことがあるからです。上記の移住マッチングAIも、実際の移住相談イベントの会場で利用してもらった際、AIから命令されたように感じるという意見をいただきました。しかし、実際にはAIの出した結果を変えることができるので、ユーザーは答えを押し付けられているわけではありません。人間は意思決定を支援してくれるAIとどう向き合っていけばいいのでしょうか。

AIと共同で意思決定を進めていくことが可能に

## 技術の再編成の重要性

AIと向き合う際に人間側に必要になるのは、技術が出した結果を与えられた答えとして捉えるのでなく、自分で思うように改変するための仮設検証のループを回そうとする姿勢です。この姿勢はインタラクティブな「説明可能AI」が広がる前の現在であっても必要な態度です。データ駆動型の技術が主流であるからこそ、我々はAIに提示される情報を意図的に変えていくことができます。検索したり視聴するコンテンツを変えたり、トピックごとに異なるSNSのアカウントを運用することで、異なる性格をサービスごとに植え付けることが可能になります。また、このような態度を持つためには一定のリテラシーを持つことが必要です。どういうインプットを入れるとどのようなアウトプットが出てくるのか、技術を作る人であればどういう仕組み

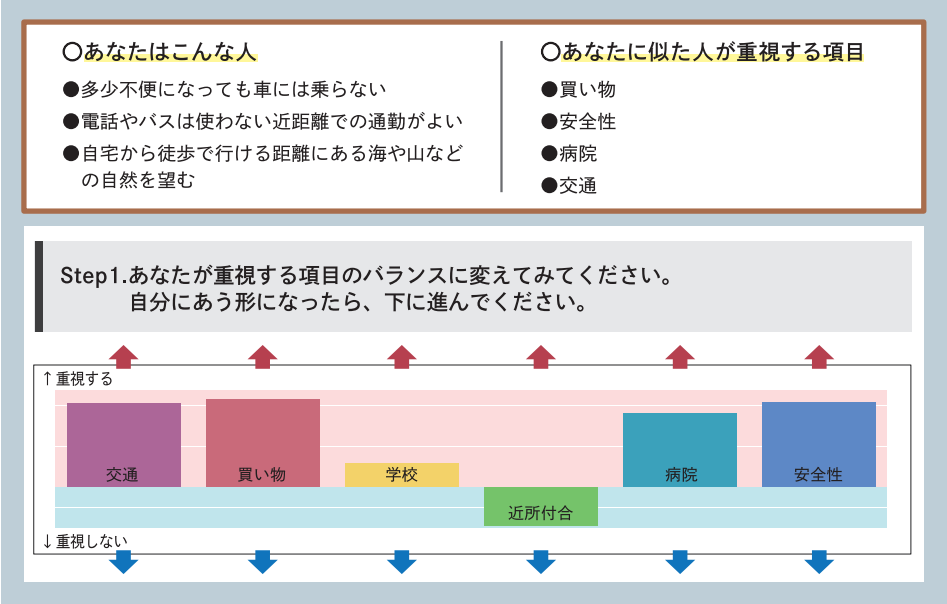



図4 移住先マッチングAI:
どの地域にその人が住むべきかをインタラクティブに推薦することが可能

でそのサービスが動いていそうか、技術を作らない人であっても、そのサービスがもたらす影響を想像して、それに対応してどのようなポリシーで技術を使うべきかを考えることが重要です。どんな情報にさらされたとしても、なぜその情報が自分に対して提示されているのかという事を常に検討していけば、技術とうまく付き合っていくことができるのではないでしょうか。

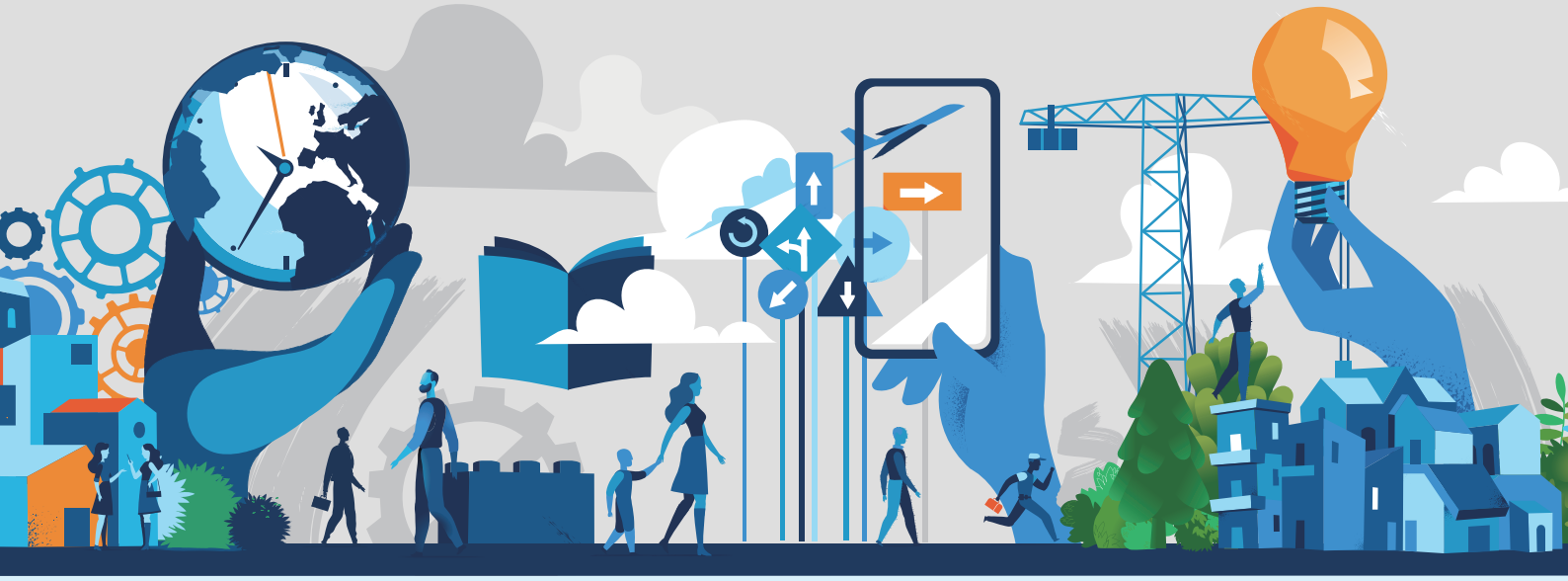
どんな情報にさらされても、技術とうまく付き合っていくことができる

## おわりに

AI技術の問題点は指摘され続けています。本稿で紹介した「説明可能AI」「公平性配慮型機械学習」のように、問題解決のために新たなAI技術が開発されています。開発が進んでも、ユーザーが何もしなくていいわけではありません。インタラクティブなAIのように、ユーザー側のフィードバックの必要な場面が出てきています。AIと離れられない現代、非専門家であっても、リテラシーを伴った仮説検証を回すことでうまくAIと付き合うことが求められます。本稿が、そうしたAIとの付き合い方を考える一助となれば幸いです。

本記事の内容に関するご質問、お問い合わせは、ベリサーブ 広報・マーケティング部 Email:verinavi@veriserve.co.jp までお寄せください。

# これからのソフトウェアをどうやってテストしよう?



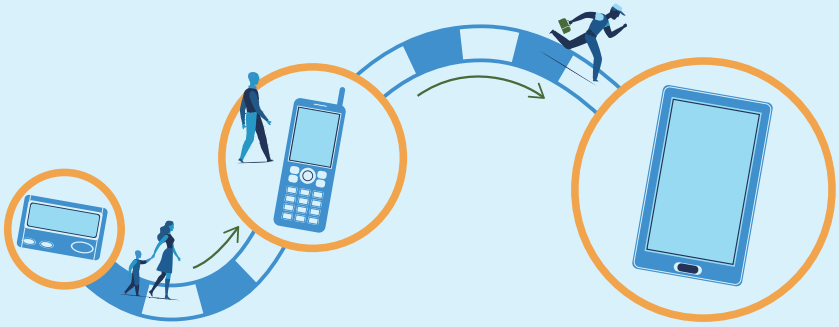

湯村 翼氏
ゆむら つばさ

国立研究開発法人情報通信研究機構
博士(情報科学)

株式会社東芝、クウジット株式会社、明治大学、フリーランスを経て2015年より情報通信研究機構に勤務。
北陸StarBED技術センターにてネットワークテストベッド、ユビキタスコンピューティングシステムの研究に従事。
著書に『iOSアプリエンジニア養成読本』。博士(情報科学)。


## はじめに

いつの時代も、ソフトウェアテストは重要です。継続的インテグレーションやテスト駆動開発といった、ソフトウェアテストを軸にした開発のパラダイムも生まれています。ソフトウェア開発が続く限り、ソフトウェアテストの重要性は失われないでしょう。初期のソフトウェアは単独のコンピューターで動作するものでしたが、コンピューターがネットワークを経由して他のコンピューターとつながるようになり、コンピューターシステムもソフトウェアも次第に複雑になっていきます。今では、周囲の物理的な状況をセンシングして挙動を変えるソフトウェアがあります。その代表例がスマートフォンアプリケーションです。GPSによって取得した位置情報に応じてユーザーがいる場所の付近の情報を提示するものなどがあります。また、加速度センサーや傾きセンサーなどの各種センサーのデータを用いるものもあります。ソフトウェアが複雑になる一方で、ソフトウェアテストの手法やツールも進化しています。Webインターフェースのテストのためのツールや、ソフトウェアが稼働するサーバー環境自体をテストするServerspec [https://serverspec.org/] というツールも出てきています。



## 20年後の未来を考えると……

20年後を考えるために、まず20年前を振り返ってみましょう(図1)。

20年前は、ポケベルが使われなくなり、PHSやiモード携帯が出始めた時期です。モバイルデバイスの進化が大きかったモバイルデバイスが急速に進化した時代ですが、進化したのはモバイルデバイスだけではありません。PCも、Windows98、Windows2000といったOSが登場し、一般家庭にどんどん普及していった時期でした。

それから20年後の現在、ガラケーを使う人は減り、スマートフォンが普及しています。

Fitbitのようなウェアラブル端末、Amazon Echoのような家に配置するデバイスが広がり、「ウェアラブルコンピューティング」「ユビキタスコンピューティング」といった技術が生活に密着した形で実装され始めました。バーチャルリアリティの技術を使ったヘッドマウントディスプレイでゲームするといったことも広がっています。

これらを実現するための画像処理や音声解析といった技術は、目には見えませんが、バックグラウンドで「機械学習」が支えています。

そして、20年後を予測したいところですが……正直どうなるかわかりません。

本稿では、ソフトウェアテストという切り口ではあまり語られないけれども、20年後にも存在していそうな技術分野・テーマを取り上げます。ユーザーインターフェース、センサーデータ、エンドユーザプログラミング、自動運転車、無線エミュレーションに関するテストについて、現状の課題を整理し、それらの解決の糸口となりうるツールや研究を紹介します。これらのツールや研究にはまだ実用に耐えうる段階ではないものも含まれるかもしれませんが、そのコンセプトに未来のソフトウェアテストを見据えるためのヒントがあるかもしれないと思っていただければ幸いです。

## 新たなユーザーインターフェース

コンピューターは、要素を単純化した5大装置というモデルで表されます。5大装置の中で人間とコンピューターの接点となるのが入力装置と出力装置です。コンピューターの代表的な入力装置はキーボードとマウス、出力装置はディスプレイで、これらは内部のしくみの変遷はあるもののコンピューターの

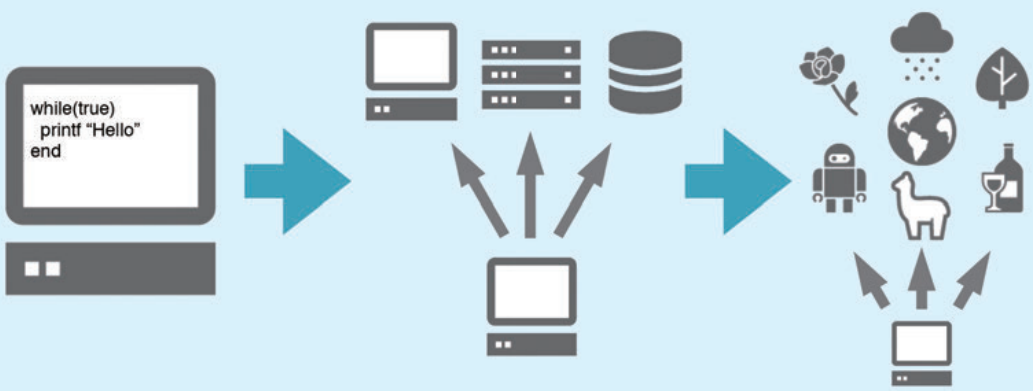
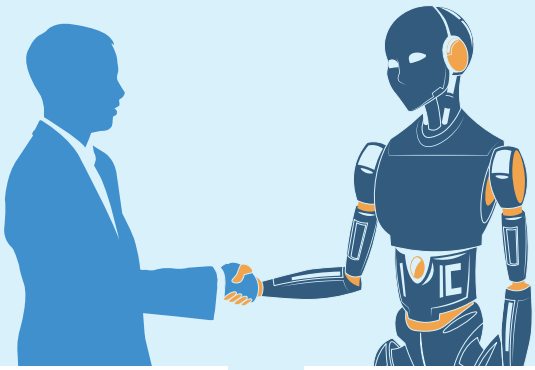



図1 ソフトウェアが扱う対象は次第に広がっていく



初期の頃から現在まで長い間利用されています。加えて、スマートフォンの普及により、入力装置と出力装置が一体となったタッチパネルディスプレイも広く利用されるようになりました。入力装置と出力装置を用いて、人間とコンピューターで情報を伝達するための境界はユーザーインターフェースと呼ばれます。物理的な入出力装置に加えて、出力装置に表示される画面もユーザーインターフェースに含まれます。ユーザーインターフェースのテストは、すべて手動で行われているわけではなく、さまざまな方法で自動化が進められています。例えばWebやアプリケーションの画面の任意の箇所をクリックする処理をソフトウェアとして行うSeleniumやAppiumというテストツールが使われています。

街中で情報を掲示するデジタルサイネージには、ジェスチャーによって操作する端末もあります。通常、ジェスチャー入力を用いるソフトウェアをテストするためには、開発者やテスターが体を動かしてジェスチャーを行い、それに応じた動作をチェックします。同じ動作を何度も繰り返すことになるため、これは肉体的にも非常に過酷な工程になります。これを解消するための提案がKinect Doll [https://www.youtube.com/watch?v=syY26n7ponM] です。Kinect Dollは、板を組み合わせて肘などの関節で結合した簡易的な人形です。これを用いて、ジェスチャー入力で用いられるボーン（人の骨格）を形作り、任意のポーズを取らせます。人がポーズを取る代わりに人形に肩代わりさせることによって、開発と検証の負担を減らします。

今後、より多種多様なユーザーインターフェースが登場すると予想されます。最近では、スマートスピーカーのような音声入力を用いた新しいインターフェースのデバイスも登場し、HoloLensやOculus Questといったヘッドマウントディスプレイも普及し始めています。このように、入出力装置というのは、技術の進化に伴ってどんどん増えていきます。それに合わせ、新しいインターフェースに適用したテストツールが出てくるのではないかと考えられます。

## 状況に応じて振る舞いを変えるアプリケーション

周囲の状況の情報を用い、状況に応じて振る舞いを変えるアプリケーションのことを「コンテキストアウェア・アプリケーション」と呼びます。コンテキストにはさまざまな種類があります。例えば、場所や建物などの空間的コンテキストや、時刻や曜日など時間的コンテキストがあります。コンテキストの分類は、サーベイ論文 Knappmeyer et al. 2012 [https://uwe-repository.worktribe.com/preview/936196/Survey_of_Context_Provisioning_Middleware.pdf] にまとめられています。

例としてスマートフォンやIoTデバイスについて考えてみましょう。これらに搭載されるソフトウェアは、センサーを用いて周囲の状況をセンシングし、コンテキストを用いて動作します。そのため、テストを行うには物理的な情報が必要となり、テストの自動化が難しくなります。スマートフォンのセンサーデータを用いたアプリケーションをテストするために、TestAware [https://github.com/cluo29/TestAWARE] というツールが研究開発されています。TestAwareは、取得したセンサーデータや擬似的に生成したセンサーデータをデータベースにため、テスト時にこのデータを再生してテストに用います。また、同様の目的のContextMonkey [https://github.com/manojrege/contextmonkey] というツールでは、カメラアプリのテストのために現地で撮った写真の代わりにGoogleストリートビューの画像を用います。

現在、身に付けるものや建物の中など、あらゆるところに小さなコンピューターが埋め込まれています。コンピューターの数は今後も増え続け、ソフトウェアはさまざまなものに搭載されるようになっていきます。それに伴い、周囲の状況を生成するテストツールも重要となっていくでしょう(図2)。

## ユーザー自身が開発したシステム

これまで、システム開発は、専門的な知識を持つ開発者が行うものでした。デバイスやアプリケーションにできることが増えると、それをユーザーが使いこなすことが期待されるため、専門的な知識を持たないエンドユーザが簡易的なシステム開発をするというケースも増えていくでしょう。

IFTTT(イフト)というアプリケーションがその一例です。IFTTTは、さまざまなWebサービスやIoT製品と連携して稼働します。サービスや製品は、トリガーまたはアクションという動作として定義されています。IFTTTのユーザーは、トリガーとアクションという2種類の動作を組み合わせ、自分だけのルールを決めることができます。例えば「会社を離れた時に家のエアコンをつける」や「家から離れたら家の鍵を自動で閉める」といったものです。IFTTTを使えば、自分だけのシステムを簡単に作ることができます。

他にはMESHという、簡単なIoTシステムを構築できる製品もあります。MESHは、人感センサーや温湿度センサーなどの機能を持った物理的なブロックを組み合わせてシステムの部品として使用します。iPadの画面上で線をつなぐことによって、システムの動作を決めます。いわゆるビジュアルプログラミングを行います。

IFTTTのようなアプリケーションは、専門的な知識を持つ開発者だけではなく一般のユーザーが使い、構築したシステムがうまく動作しない際には一般のユーザーがデバッグを行うことになります。このような一般ユーザーのデバッグを支援するために研究されているのがEUDebug [https://elite.polito.it/research/research-topics/488-eudebug] です。EUDebugは、IFTTTのようなTrigger-Action型のルールで起こりうる3種類の問題「ループ」「矛盾」「冗長」について静的解析し、問題があった場合にはユーザーに提示します。

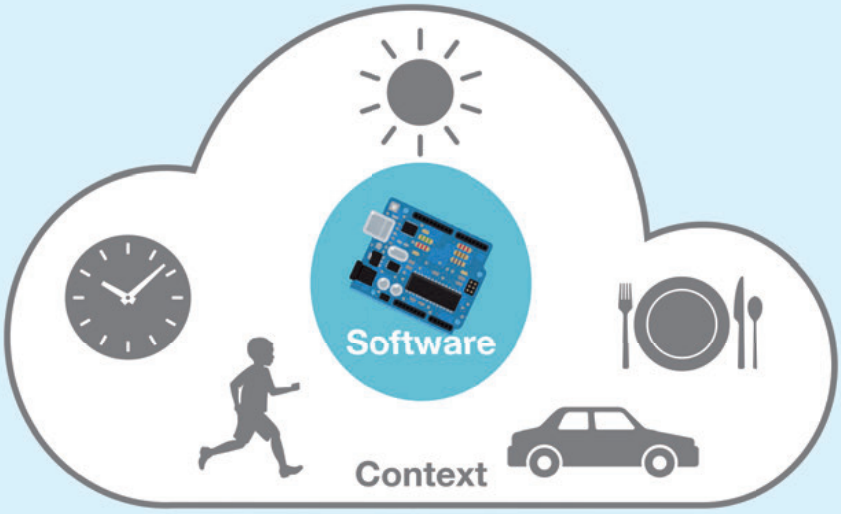



図2 コンテキストアウェアのイメージ

## 自動運転車

自動運転は、現在ソフトウェア技術の研究開発が非常に熱い領域です。Uberは、自動運転のテストを行うために、開発拠点のある米ペンシルベニア州ピッツバーグ近くに東京ドーム50個分のコースを建造中です。そのコースの中で自動運転車を走らせ、道路状況や標識の認識、車の前に擬似的な歩行者を飛び出させて車の安全機能を確認します。とはいえ、このような街スケールの大規模なテスト環境を用意できる企業や研究所は限られています。そこで、シミュレーションなど仮想環境を活用したテストが重要になってきます。

自動運転車の開発では、学習データを集めるために、コンピュータグラフィクス(CG)による仮想環境の構築が最近よく行われています。これには、UnityやUnreal Engine4といったゲームエンジンも活用されています。その仮想環境の中で車を走らせ、LiDARセンサー*1やカメラでの物体認識の学習を行います。Microsoftが開発するAirSim [https://microsoft.github.io/AirSim/] は、オープンソースの学習用仮想環境です。ゲームエンジンで構築した仮想環境の中で自動車を走らせて学習を行って運転の精度を高めていくというものです。現在、そのような目的のツールは、自動運転と機械学習を研究開発する企業や研究所でしのぎを削って開発が進められています。

仮想環境を活用したテストは、自動運転が大きなターゲットですが、今後はスマートシティ開発において、街中にセンサーやアクチュエータが多数設置された際の挙動を調べることなどが考えられます。そのような大規模なテストを現地で行うのは非常に大変なので、シミュレーションや仮想空間を用いたテストの有用性がより強調されていくでしょう(図3)。

ちなみに、CGによる仮想環境を用いたテストの歴史をさかのぼると、HPが2002年に行っていたUbiwise [https://www.hpl.hp.com/techreports/2002/HPL-2002-303.pdf] という研究にたどり着きます。これは、CGでつくった3次元仮想空間の中で、デジタルカメラのテストを行うというものです。デジタルカメラのバーコード読み取り機能を、角度などの条件を変えて読み取れるかどうかなどをテストするものでした。

*1「Light Detection and Ranging」の略。ライダー。レーザー光を走査しながら対象物に照射してその散乱や反射光を観測することで、対象物までの距離を計測したり対象物の性質を特定したりする光センサー技術。

## 仮想ネットワーク空間

ソフトウェアがさまざまな場所に点在するようになると、その環境を模した環境でテストが必要になってきます。例えば、店舗や自動販売機の来店検知や博物館などのナビゲーションといった屋内の測位を行うアプリケーションのテストです。

私が所属する国立研究開発法人情報通信研究機構(NICT)では、ネットワークテストベッドStarBEDを活用したソフトウェアテストの研究開発に取り組んでいます。テストベッドとは、大規模なシステム開発で用いられるテスト用プラットフォームの総称です。テストベッドは、現実世界で運用されているシステムを危険にさらすことなく、実際の運用体制に近い状況で稼動させる目的で使われます。StarBEDでは、1000台規模のPCノードを用意し、それらを接続してクローズドなネットワークを

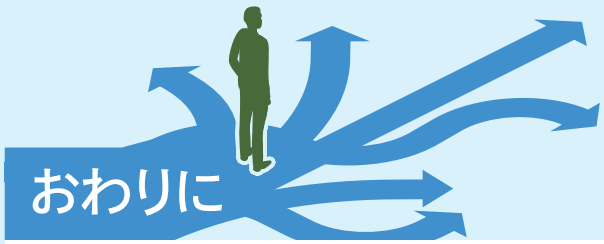
図3　無線エミュレーションのイメージ

構築しています。その中で、例えば仮想的なインターネットを作るなど、そこでしかできないネットワークの実験を行っています。現在は、コンピューター上のソフトウェアテストだけではなく、ネットワークに影響を及ぼす電波伝搬などの空間の情報をいかに取り入れてテストを行うかということを研究対象とし、空間情報を含めたネットワークのシミュレーションとエミュレーションの研究に取り組んでいます。

その一つがBluetooth Low Energy (BLE)のエミュレーションの研究です。Bluetoothのような無線通信は、通常は送信機と受信機を用意しそれらを用いてフレームを送受信することで実験や検証を行います。我々が研究開発しているエミュレーターはそれを送信機受信機の実機は用いずに全てコンピューター上で無線の送受信と同じことを行います。

無線を用いるシステムのテストには実機を用意したり設置したりするコストが伴いますが、エミュレーションを用いることでそのコストの削減に寄与できるのではないかと考えています。Bluetoothのような近距離無線通信であれば実機を用いた際の手間も小さいですが、LoRa[*2]やNB-IoT[*3]のような数km～数十kmの伝送距離を持つLPWA[*4]と呼ばれる無線では、一対一の通信を検証するだけでも非常に大変となります。そのため、エミュレーションを用いたテストの有用性はより大きくなるでしょう。

＊2 ローラ、少ない消費電力で広いエリアをカバーする無線通信方式の一つ
＊3 「Narrow Band-IoT」の略。スマートフォンなどで利用されるモバイル通信技術である「LTE方式」の中でも、IoT機器向けの規格。
＊4 「Low Power, Wide Area」の略。少ない電力で広いエリアをカバーする無線通信技術の総称。



## おわりに

仮想環境を用いたテストの話をすると「実地でのテストは無くなるんですか?」と聞かれることがありますが、シミュレーションや仮想環境があれば全てのテストを実施できるとは思っていません。あくまでも最終的には現地テストを行うという前提で、事前にソフトウェアの検証を進めておくことによって、現地での検証の手間を省くというものと考えています。

機械学習やIoTなどの技術の進歩や潮流の変化とともに、ソフトウェアの役割が広がっています。それに伴い、ソフトウェアテストの形は進化していくでしょう。

エンジニアは、この流れに乗っていく必要があります。

未来の技術がどうなっているか考えることは楽しくもありますし、その技術が登場したら、それをどのようにテストするか考えることも楽しいと思います。

本稿をきっかけに未来の技術とテストに想いを馳せてみてください。

本記事の内容に関するご質問、お問い合わせは、**ベリサーブ 広報・マーケティング部** Email:verinavi@veriserve.co.jp までお寄せください。

# 解説！
# ソフトウェアテストの国際標準
# ISO/IEC/IEEE 29119

ソフトウェアテストに関わるとき、目にするものの中に、ISO/IEC/IEEE 29119という規格があります。規格というととっつきにくい印象をお持ちの方も多いでしょう。しかし、ソフトウェアテストに関わる技術者として、コンサルタントとして、知っておいたほうがいい内容ですので、一通り説明したいと思います。


須原 秀敏
すはら ひでとし

株式会社ベリサーブ
オートモーティブ事業本部 オートモーティブ検証サービス開発部


2010年ベリサーブに新卒入社。車載ソフトウェアの品質保証業務に一貫して関わる傍ら、業界活動に積極的に参加。2018年テスト設計コンテストで「てすにゃんV3」の一員として優勝。
JSTQB技術委員、ASTER正会員、JTC1/SC7/WG26エキスパート

## はじめに

現在、さまざまなソフトウェアテストに関連する技術が開発され、またそれらの技術を知識としてまとめた文書が公開されています。これは他の技術分野でも同じですが、この技術情報を得ようと思った場合、例えば、書籍、国際/国内学会の論文、国際標準、資格試験等があります。本稿は、ソフトウェアテストの国際標準であるISO/IEC/IEEE 29119について解説します。

国際標準の世界は複雑に見える（本当は知ってしまえばそこまで難しくはない）ので、基本的なところから解説していこうと思います。本稿のゴールは、国際標準に関わる以下の文章を理解できるようになることです（内容は、すべて後述します）。

**POINT**

**JTC1/SC7/WG26がISO/IEC/IEEE 29119を書いている。**
**ISO/IEC/IEEE 29119にはすでにISとして発行されているPart1～5があり、今後複数のISやTRの発行が予定されている。**

## 01 ISO/IEC/IEEE 29119とは

ISO/IEC/IEEE 29119とは何かを理解するには、まずISO/IEC/IEEE 29119 Part 1のIntroductionの冒頭の説明を読むのがよいと思います。

The purpose of the ISO/IEC/IEEE 29119 series of software testing standards is to define an internationally-agreed set of standards for software testing that can be used by any organization when performing any form of software testing.[1]

簡単に言うと、国際的に合意されたソフトウェアテストの標準群、です。

ISO/IEC/IEEE 29119の策定自体は2007年に検討が開始され、Part1, 2, 3の初版が2013年に発行[2]されています。検討開始から初版の発行まで6年もかかっているのは、後述の重厚なプロセスがあるからです。現在は、Part1～5までが発行されています。各Partの内容はそれぞれ以下です。

Part1: Concepts and definitions（コンセプトと定義）[3]
Part2: Test processes（テストプロセス）[4]
Part3: Test documentation（テスト文書）[5]
Part4: Test techniques（テスト技法）[6]
Part5: Keyword-driven testing（キーワード駆動テスト）[7]

Part1が概要、導入の章です。ソフトウェアテストに関する考え方を記載しており、例えば、この標準群の特徴の一つであるリスクベースドテスト*1について説明がなされています。

Part2はこの標準群の中心的な考え方となるテストプロセスについて定義している章です。ポイントとしては、複数のレイヤのテストプロセスモデルを採用している点があります。各プロジェクトのテストプロセスの上位に組織的なテストプロセスを置き、そこで組織的に定めるテストポリシーなどをまとめる、という考え方に基づいたプロセスになっています。

＊1：“リスクのレベルに基づいてテストを設計し優先度付けする”テスト（ISTQBテスト技術者資格制度 Foundation Level シラバス 日本語版 Version 2018.J03）

Part3はPart2で定義したテストプロセスの中でどんな文書が必要となるか、また、その文書の構成要素としては何があるかを定めています。例えば、テスト完了レポートという文書があり、内容としては実施したテストのサマリ、テスト完了評価、テストのメトリクスなどが必要である、といったことが記載されています。

Part4は主にPart2のテスト設計プロセスにおけるテスト(設計)技法について列挙されています。仕様ベースのテストとして同値分割や境界値分析が挙げられていたり、構造ベースのテストとしてステートメントテストや改良条件判定カバレッジ(MC/DC)[*2]テストが挙げられていたりします。

Part5はテスト設計からテスト実装に関係する手法である、キーワード駆動テストを取り上げています。

上記の内容から、多忙な方は、まずPart2から読むとよいと思います。

## 02 国際標準の概要

国際標準を発行している組織には代表的なものがいくつかあります。本稿で取り上げているISO/IEC/IEEE 29119は、後述する代表的な3つの組織(ISO、IECおよびIEEE)が合同で発行した国際標準です。

国際標準は、世界中のエキスパートが知恵を絞り合って策定した「国際的に合意」された文書であるため、技術的に妥当なことが書かれていることが多い、という強みがあります。そのため、企業または組織は自分たちの活動の参考にできる可能性があります。

ただ、国際標準はそれ以上に大事な意味を持ちえます。それは、WTO(World Trade Organization)加盟国が守るべきルールになりうる、という点です。WTO協定の中にTBT(Technical Barriers to Trade)協定というものがあり、「TBT協定は、工業製品等の各国の規格及び規格への適合性評価手続き(規格・基準認証制度)が不必要な貿易障害とならないよう、国際規格を基礎とした国内規格策定の原則、規格作成の透明性の確保を規定」[8]しています。簡単に言うと、「国際規格を使って各国の足並みをそろえよう」ということです。この「国際規格」に該当するものが国際標準です。そのため、国際標準は企業または組織が従わなければならないルールになる可能性があります。

このように、国際標準は各主体の活動の妨げになりうるので、各国の代表組織が参画して策定やレビュー等を行って、自分たちの不利益にならないようにしています。日本では、ISO/IECの代表組織をJISC(日本産業標準調査会)が担っています[9]。

ただ、国際標準は各国の代表が策定を行っているものではあるものの、賛成意見ばかりではありません。ISO/IEC/IEEE 29119の策定に際してもAssociation for Software Testing[10]やInternational Society for Software Testing[11]などから反対意見が出されました(一部の議論は日本語訳されて公開されています[12])。このような意見があることを知った上で、企業または組織にとって有用な標準を自ら選んで活用するという立場が大切だと考えています。

*2:改良条件判定カバレッジ(MC/DC)(modified condition decision coverage):判定結果に対して独立に影響する全単一条件結果のうち、テストスイートが遂行した条件結果のパーセンテージ。(ソフトウェアテスト標準用語集(日本語版)Version 2.3.J01)

## 03 ISO/IEC/IEEEとは

「ISO/IEC/IEEE 29119」の先頭に付いている「ISO/IEC/IEEE」は、前述の通り、発行している組織です。それぞれ、ISO（International Organization for Standardization：国際標準化機構）、IEC（International Electrotechnical Commission：国際電気標準会議）、IEEE（Institute of Electrical and Electronics Engineers：電気電子技術者協会）です。ISOとIECは欧州の標準化団体で、IEEEはアメリカの標準化団体です。

ISOとIECは多くの分野で標準を発行していますが、特に情報技術に関してはJTC1（Joint Technical Committee 1）という合同専門委員会を作り、共同で標準の発行活動を進めています[13]。

また、IECとIEEEは協力して標準を策定していくことに合意しており[14]、IEEEはISOの標準化に協力できるため[15]、定められたプロセスを経て双方の発行物としてリリースできる手段が用意されています。似たような標準をそれぞれが出していると混乱するので、相互が合意できる内容であれば一本化してくれた方がわかりやすいでしょう（とはいえ、同じ分野であってもいろいろな意見があってしかるべきなので、複数の団体がそれぞれの意見を持った標準を出すことも価値があると思います）。

ISO/IEC/IEEE 29119は、ISO/IEC JTC1/SC7/WG26（詳細は後述）内で策定された標準であり、IEEEの承認プロセスを経て、三者合同の標準として発行されています（下図）。

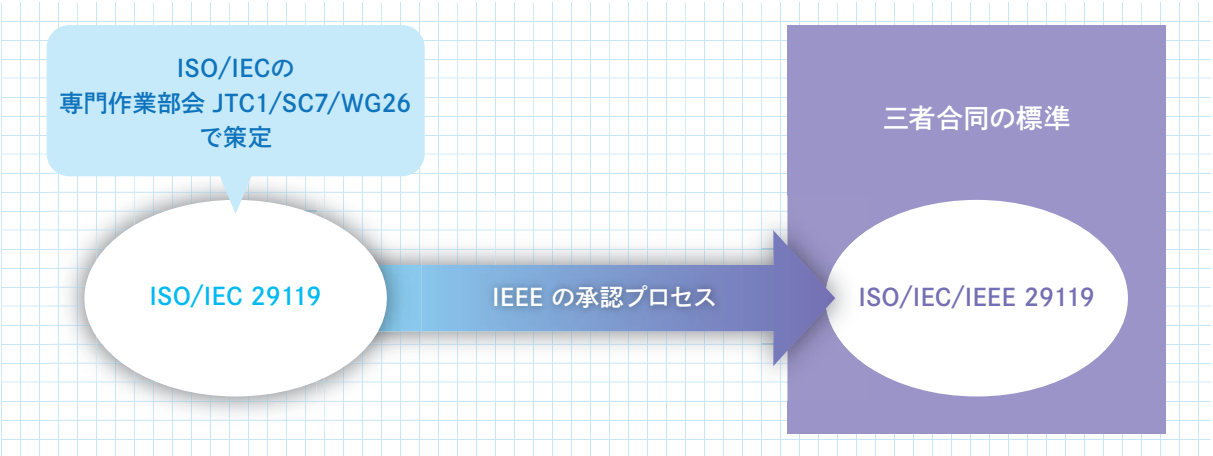
## 04 JTC1/SC7/WG26とは

JTC1/SC7/WG26は、ISO/IECの中のソフトウェアテストの専門作業部会です。Convenor（議長のようなもの）はStuart Reid氏が務めています[16]。ご存じの方もいらっしゃるかもしれませんが、Stuart Reid氏は2014年にJaSST Tokyo（ソフトウェアテストのカンファレンス）で基調講演をされました[17]。

JTC1は前述の通りITに関する合同専門委員会であり、その中に複数のSC（Sub Committee）があります。SC7はSoftware and systems engineering（ソフトウェア/システム工学）のSCです。SC7の中に複数のWG（Working Group：作業部会）があり、WG26はソフトウェアテストのWGです。その他のSC7に所属するWGとしては、WG6：Software product and system quality（ソフトウェアプロダクトとシステムの品質）、WG10：Process assessment（プロセスアセスメント）[16]などがあります。WG6は25000シリーズ（SQuaRE）と呼ばれるソフトウェア/システムの品質特性に関する標準群を策定しており、WG10は33kシリーズ（SPICE）と呼ばれるプロセスアセスメントに関する標準群を策定しています。33kシリーズ（SPICE）は自動車分野で有名な「A-SPICE」の元になっている標準群です。

WG26としては、ISO/IEC/IEEE 29119がメインの標準ですが、静的テストの一つであるレビューに関する標準であるISO/IEC 20246も策定しています。

POINT!




図：ISO/IEC/IEEE三者合同標準の承認プロセス

## 05 ISOが発行する文書

ISOの発行する文書には前述の通り種類があります。本稿で説明してきている国際標準は、IS（International Standards：国際規格）です。ISは、ルールやガイドラインとなりうるものです。文書のタイトルに特に何も書いていなければ、ISであると思っていただければ大丈夫です。例えば、”ISO 9001:2015”はとても有名で、ISの代表格と言えます。

TS（Technical Specification：技術仕様書）は、将来的にISに昇格させる目的で発行されるものです。技術的に発展途上であるが故にまずは途中段階で発行し、その後フィードバックを受ける目的もあります。あまりTSで有名なものはありませんが、自動車分野の方であれば”ISO/TS 16949:2009”という規格名に見覚えがあるかもしれません。今はIATF（International Automotive Task Force）16949となっている自動車版のISO 9001です。IATF 16949の前身がISOのTSだった、ということです。

PAS（Publicly Available Specification：公開仕様書）は、マーケットの要請により急いで文書を発行する必要がある際に使われる方式です。期限は6年であり、TSと同じようにフィードバックを受けつつ、その後にISへの昇格または破棄となります。最近だと、”ISO/PAS 21448:2019”が発行されています。こちらも自動車分野なので、分野が近い方はご存じかもしれません。SOTIF（Safety Of The Intended Functionality）という自動車の安全に関する文書です。この文書が発行された背景としては、自動運転が広がりつつある中でその安全性の保証の仕方を急いで定めて公開しよう、ということがあります。

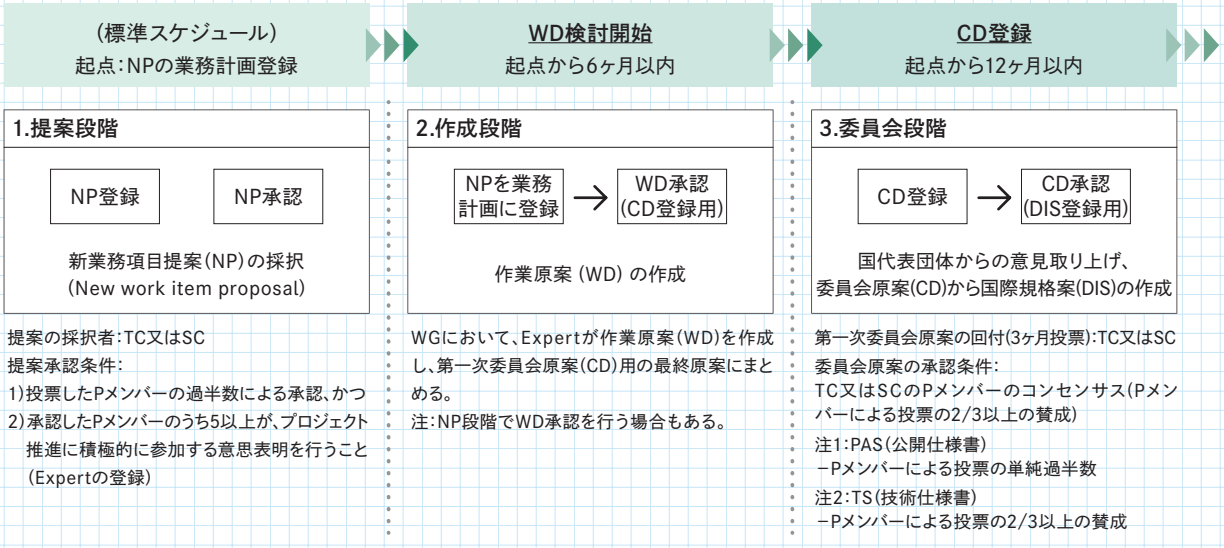
TR（Technical Report：技術報告書）は、informative（情報提供）の意味が強い文書です。ISに関連する調査データや、最先端の情報を含められますが、情報提供の範囲を超えないものです。TRにはその性格からISへの昇格の仕組みがありません[18]。

## 06 ISO標準の発行プロセス

ISOが発行する文書には、主にIS、TS、PAS、TRという文書があります。一般的に標準と呼ばれる文書はIS（International Standards：国際規格）です。そのISを発行するまでには、いくつものステージが必要となります（下図参照）。そのプロセスの

### ISO規格の策定手順

ISOの開発プロセスは、ISO/IEC Directives（専門業務指針）によって定められており、そのプロセスは次のようになります。



中で、各国からの意見を提出し、それに対して修正を行ったり、投票を行ったりして、最終的に発行まで進みます。WD/CD/DIS/FDISという文言を見たら、標準の開発ステージであると理解いただければ問題ありません。詳細なステージはISOのWebサイト[19]をご確認ください。

ここで詳しく発行プロセスをご理解いただくのは難しいと思いますので、お伝えしたいこととしては、一つの標準が発行されるまで長い道のりがある、ということです。

## 07 ISO/IEC/IEEE 29119の今後

まず、ISはシステマティックレビューという制度があり、5年に一度レビューを実施し、必要があれば更新しなければなりません。ISO/IEC/IEEE 29119のPart1～5も同じです。Part1～3の発行は2013年であるため現在まさにレビュー中であり、近いうちに更新版が発行されることになります。更新内容について詳細を書くことはできませんが、もしISO/IEC/IEEE 29119を参照してプロセスや文書を作っている企業や組織の方は、改めて更新版の標準を見ていただく必要があると思います。

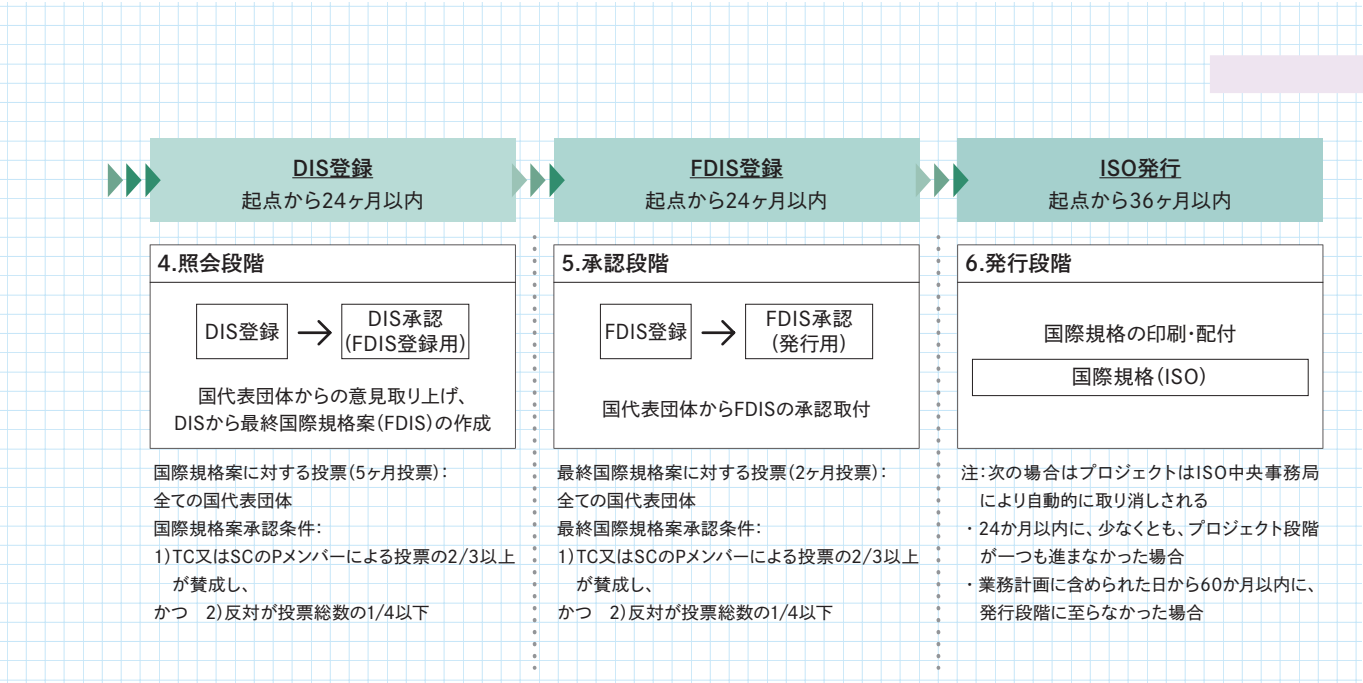
なお、更新前後は、例えば更新版が2020年に発行される場合は、

既存版"ISO/IEC/IEEE 29119-1:2013"

更新版"ISO/IEC/IEEE 29119-1:2020"

のように、末尾につく西暦年で見分けることができます。逆に、ISOの文書を論文や開発文書等で参照する場合は、西暦年まで書く必要があります。

また、ISO/IEC/IEEE 29119は、いくつかのPartの追加を予定しています。ISまたはTRという位置付けでの発行となる予定です。一つはISO/IEC TR 29119-11 Testing of AI-based systems[20]です。まだステージとしては初期ですが、TRということでドラフトが作られれば発行まで比較的早く進む可能性があります。性格としては、上述の通り最先端の情報をいち早く提供する、という意味合いが強いため、この文書がルールや規制になることは考えにくいです。その他には、アジャイルテスト、インシデント管理、パフォーマンステスト、テストアーキテクチャ等のISまたはTRの策定が進んでいます。




出典「ISO規格 の策定手順」|日本船舶技術研究協会　http://www.jstra.jp/images/standard/iso/iso_iec_kikaku_tejyun.pdf

## 08 ISO標準の策定に参画する方法

これまで書いてきた通り、ISO/IECの国際標準は各国の代表組織によって策定、レビュー、投票が行われて、最終的に発行されます。JTC1としての日本の代表組織はJISCですが、情報技術分野に関しては情報処理学会 情報規格調査会に委嘱が行われています[21]。そのため、ISO/IECの標準の策定に参画するためには、情報規格調査会の賛助員となることが条件になります。ベリサーブも2019年度から賛助員となり、この活動に参画しています。もしご興味がある方は情報規格調査会のWebサイト[22]でご確認いただければと思います。

## 09 ISO標準策定に参画することのメリット

ベリサーブでは、2019年度から情報規格調査会の賛助員となり、私自身がWG26のエキスパートとして活動しています。月に一度開催される国内委員会の他、半年に一度開催される各国のWGメンバーとの国際会議に参加し、上述のPart1～5の更新に対するレビューや新しいPartの策定などに関わっています。

活動に参画することによるメリットがあります。

国内や各国のメンバーから国際標準の内容の趣旨や意図、解釈を直接聞くことができ、適切に国際標準を利活用・運用することができるようになることです。さらに積極的には、自らの不利益になることを防ぐように、標準を修正、策定できます。もちろん、参画しているWGメンバーは技術的に正しいことを追い求めて標準を作っているため、「技術的に正しい」の範囲の中で、利益を求め、不利益を除外することができるということです。

結果、ベリサーブでは、適切な情報をタイムリーに、自社開発のツールやベリサーブ独自のテストプロセス標準に適用できます。実際に、ベリサーブで開発を行っているテスト設計支援ツールであるTESTRUCTUREや、当社独自のテストプロセス標準でテストの作業手順や成果となるドキュメントをまとめたVSMethod（Veriserve Standard Method）へのISO/IEC/IEEE 29119の活用を進めています。

標準に合わせ、使われる単語、ドキュメント、完了基準の統一などを行うことは、ツールを利用するベリサーブのお客様のテスト効率化などといったテストプロセスの改善にも貢献できる活動だと考えます。

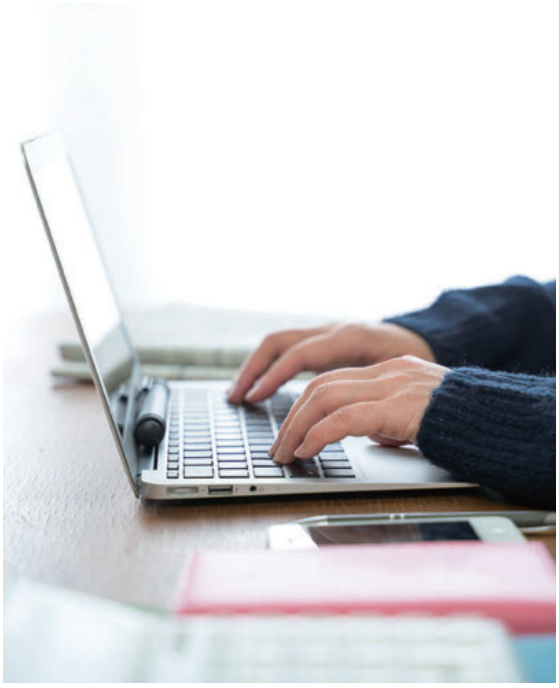
## まとめ

本稿では、ISO/IEC/IEEE 29119の解説を中心に、国際標準の仕組み等をご紹介しました。

最初に紹介した文章は理解できるようになったでしょうか。

**JTC1/SC7/WG26がISO/IEC/IEEE 29119を書いている。**
**ISO/IEC/IEEE 29119にはすでにISとして発行されているPart1～5があり、今後複数のISやTRの発行が予定されている。**

本稿を読んだことで、少しでも国際標準へのハードルが下がり、国際標準を読んでみようと思っていただければ幸いです。



## 参考文献

[1] https://www.iso.org/standard/45142.html
[2] Reid, S. (2012). "The New Software Testing Standard". In Dale, C.; Anderson, T. (eds.). Achieving Systems Safety: Proceedings of the Twentieth Safety-Critical Systems Symposium, Bristol, UK, 7-9th February 2012. Springer Science & Business Media. pp. 237–56. ISBN 9781447124948.
[3] ISO/IEC/IEEE International Standard. 2013. “Software and systems engineering --Software testing --Part 1: Concepts and definitions," in ISO/IEC/IEEE 29119-1:2013(E), pp.1-64. IEEE.
[4] ISO/IEC/IEEE International Standard. 2013. “Software and systems engineering --Software testing --Part 2: Test processes," in ISO/IEC/IEEE 29119-2:2013(E), pp.1-68. IEEE.
[5] ISO/IEC/IEEE International Standard. 2013. “Software and systems engineering --Software testing --Part 3: Test documentation," in ISO/IEC/IEEE 29119-3:2013(E), pp.1-138. IEEE.
[6] ISO/IEC/IEEE International Standard. 2015. “Software and systems engineering --Software testing --Part 4: Test techniques," in ISO/IEC/IEEE 29119-4:2015(E), pp.1-149. IEEE.
[7] ISO/IEC/IEEE International Standard. 2016. “Software and systems engineering --Software testing --Part 5: Keyword-driven testing," in ISO/IEC/IEEE 29119-5:2016(E), pp.1-69. IEEE.
[8] https://www.jisc.go.jp/cooperation/wto-tbt-guide.html
[9] https://www.jisc.go.jp/international/index.html
[10] https://www.associationforsoftwaretesting.org/2014/09/05/the-iso29119-debate/
[11] https://www.ipetitions.com/petition/stop29119
[12] http://qualab.jp/category/29119/
[13] https://www.jisc.go.jp/international/iso-tc.html
[14] https://standards.ieee.org/content/dam/ieee-standards/standards/web/documents/other/iec_ieee_coop.pdf
[15] https://www.iso.org/organization/11326.html
[16] https://www.iso.org/committee/45086.html
[17] http://jasst.jp/symposium/jasst14tokyo/report.html
[18] https://www.iso.org/deliverables-all.html
[19] https://www.iso.org/stage-codes.html
[20] https://www.iso.org/standard/79016.html
[21] https://www.itscj.ipsj.or.jp/faq/index.html#faq_10
[22] https://www.itscj.ipsj.or.jp/katsudo/

本記事の内容に関するご質問、お問い合わせは、**ベリサーブ 広報・マーケティング部** Email:verinavi@veriserve.co.jp までお寄せください。

# 高信頼性システムを開発するために

## 日本発の国際規格「IEC 62853 Open systems dependability」のご紹介


町田 欣史氏
まちだ よしのぶ

株式会社NTTデータ 技術革新統括本部
エグゼクティブR&Dスペシャリスト


株式会社NTTデータにて、テストプロセス改善、テスト自動化、テスト技法などの研究開発、技術支援、教育に従事する。近年はアジャイル開発における品質保証を担当している。著書に「ソフトウェアテスト教科書 JSTQB Foundation 第4版シラバス2018対応」など。JSTQB認定テスト技術者資格 技術委員、ソフトウェア品質シンポジウム実行委員、他。


# Introduction

IoTに代表されるように、現在のシステムは、他のシステムと連携していることが当たり前になっています。品質を担保したサービスやシステムを継続的に提供していく必要がある中、どのようにシステムに関わっていけばよいのか、「IEC 62853 Open systems dependability」という規格を通じてご紹介します。

## はじめに

ソフトウェア開発には、関連する国際規格がいくつかあります。有名なものにはISO 9000シリーズがあります。これは品質マネジメントシステムに関する国際規格で、ソフトウェア開発以外にもさまざまな業界で使われています。また、ソフトウェア開発に特化した規格としては、以下に挙げるものが代表的です。

- ISO/IEC/IEEE 12207 Systems and software engineering -- Software life cycle processes
- ISO/IEC/IEEE 29119 Software and systems engineering -- Software testing
- ISO/IEC 25010 Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models

(注)発行年は省略しています。

ISO/IEC 25010は、JIS X 25010として日本の国家規格にもなっています。これらの規格には、それぞれ、ソフトウェアのライフサイクルプロセス(開発プロセスなど)、テスト、品質に関する標準が書かれています。ソフトウェアを開発する際には、標準的なやり方を参考にして、自らのプロジェクトに合ったやり方を考えることが必要になります。

ソフトウェア開発に関する新しい国際規格として、「IEC 62853 Open systems dependability」が2018年に6月に発行されました。この規格の大きな特徴は、日本で考えられた理論が基になっていることです。筆者自身はこの規格の制定に関わったわけではありませんが、関連する団体で活動している立場から、この規格の基になった理論と規格の概要を紹介し、活用における課題についての見解を述べたいと思います。

# オープンシステム ディペンダビリティ

本章では、規格の名称にもなっている「オープンシステムディペンダビリティ」について解説します。

## 1 ディペンダビリティとは？

ディペンダビリティは、品質に関して近年よく使われるようになった概念です。「ソフトウェア品質知識体系ガイド第2版（SQuBOK® Guide V2）」[*1]では、『ディペンダビリティは、広い意味での品質にかかわる概念と捉えることができる。ISO 9000では、製品やシステムを使用している時間の経過や使用状況の変化の中での品質という意味合いで使われている。』と定義しています[1]。品質特性を使ったより具体的な定義として、IFIP WG10.4 on Dependable Computing and Fault Tolerance[*2]では、ディペンダビリティと類似の概念であるセキュリティとの関係を図1のように整理しています[2]。

ここで定義されたディペンダビリティに対応する5つの品質特性の定義を表1にまとめました。安全性以外の品質特性の説明はJIS X 25010から引用し[3]、安全性のみSQuBOK® Guide V2の定義を参考にしています[1]。なお、保全性の定義はセキュリティの副特性であるインテグリティの定義を引用していますが、より広い意味では、システムやデータの整合性、無矛盾性、一貫性などを指します[4]。

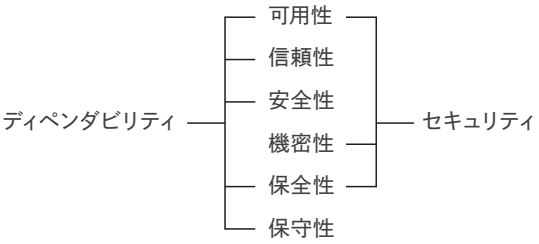



図1 ディペンダビリティとセキュリティの関係

これらを総合して、書籍「DEOS: 変化しつづけるシステムのためのディペンダビリティ工学」では、『ディペンダビリティとは利用者が期待するサービスをシステムが継続的に提供する能力である。』と定義しています[5][6]。

## 2 オープンシステムとは？

オープンシステムと聞くと、メインフレームのようなホスト系に対するオープン系をイメージする方が多いかもしれませんが、ここでの定義は異なります。ここではオープンシステムとクローズドシステムという視点でシステムを分類します[5][6]。

| 品質特性 | 説明 |
|---|---|
| 可用性<br>Availability | 使用することを要求されたとき、システム、製品又は構成要素が運用操作可能及びアクセス可能な度合い。 |
| 信頼性<br>Reliability | 明示された時間帯で、明示された条件下に、システム、製品又は構成要素が明示された機能を実行する度合い。 |
| 安全性<br>Safety | システムが、障害や危険事象が発生しても、人間の生命を損なったり、身体に害を及ぼしたり、社会に広範な悪影響を与えたりしない性質や回避できる性質、及び、そもそも障害や危険事象の発生を抑制できる性質。 |
| 保全性<br>Integrity | コンピュータプログラム又はデータに権限をもたないでアクセスすること又は修正することを、システム、製品又は構成要素が防止する度合い。 |
| 保守性<br>Maintainability | 意図した保守者によって、製品又はシステムが修正することができる有効性及び効率性の度合い。 |

表1 ディペンダビリティに含まれる品質特性

＊1：SQuBOK（すくぼっく）は、ソフトウェアに関する品質保証や品質向上に携わる技術者に求められる知識を体系的にまとめた知識体系。日本科学技術連盟 SQiPソフトウェア品質委員会と日本品質管理学会 ソフトウェア部会が共同で設置した「SQuBOK策定部会」が策定。

＊2：IFIP(International Federation for Information Processing)は1960年4月、国連ユネスコの提案で組織された情報処理国際連合

クローズドシステムは文字通り閉鎖型のシステムで、システム外からの影響を受けないシステムです。そのため、利用者や利用方法を特定、あるいは限定できますので、システムに起こり得ることが事前に想定できるというメリットがあります。

それに対して、オープンシステムはシステムの境界を定義できないものを指します。最近で言えばIoTが典型的な例で、システム・オブ・システムズも該当します。オープンシステムは、その機能や構造が時間とともに変化することがあります。そのため、想定外の入力や動作に対応できず、当初は十分であった品質が時間の経過とともに劣化することがあります。

## 3 オープンシステムディペンダビリティ

IoTに限らず、現在のシステムの多くは他のシステムと連携して動作しますが、システム間の境界が曖昧なものが多く、オープンシステムと言えます。オープンシステム、あるいはそれを使ったサービスを利用者に継続的に提供する上で、ディペンダビリティをいかに保証するかを考えなければなりません。このオープンシステムにおけるディペンダビリティが、DEOSプロジェクト[7]、およびディペンダビリティ技術推進協会（DEOS協会）[8]で研究されてきました（以降では、まとめてDEOS協会と呼びます）。

DEOS協会では、オープンシステムディペンダビリティを次のように定義しています[5][6]。

1. システムの目的や環境の変化に（継続的に）対応するための能力を有する。
2. 説明責任の遂行を継続的に支援するための能力を有する。
3. 利用者が期待するサービスを継続的に提供する能力を備える。

これらを実現するための技術についてもDEOS協会で研究、開発されました。その中核を成す概念・技術として、DEOSライフサイクルモデルとD-Caseについて、次に解説します。

# DEOS ライフサイクルモデル

DEOSライフサイクルモデルは、オープンシステムの開発および運用において、ディペンダビリティを実現するためのプロセスを定義したものです。IEC 62853では、このライフサイクルモデルを基に、ディペンダビリティの達成に必要な活動を定義しています。

## 1 モデルの全体像

古典的な開発は、図2に示すように、要求分析やリスク分析が最初にあり、開発、リリースの後は通常運用が（廃棄まで）続く、という直線的な見方をするものでした[9]。


図2 古典的な開発

この方式は、クローズドシステムに対しては有効な工学的アプローチでした。しかし、オープンシステムでは、要求や環境の変化への対応や、想定外の事象に起因する障害発生への対応が必要になります。それらに対応するために、サイクルを回して開発、さらには運用をすることが必要と考え、作られたのが、次頁図3に示すDEOSライフサイクルモデルです[9]。

このプロセスでは、変化対応と障害対応という2つのサイクルを含んでいるのが特徴です。システムの通常運用と並行してこれらの2つのサイクルを実行し、システムの状況を常に監視することで、システムのディペンダビリティの保証、維持に努めることができます。

また、開発の前後や障害対応時に行う「合意形成」と「説明責任遂行」が組み込まれていることがもう一つの特徴です。これらの活動を経て、顧客と密にコミュニケーションを取り、システムの状況に関する認識を擦り合わせていくことが重要と言えます。

## 2 変化対応サイクル

変化対応サイクルは次頁図3の外側のループを指します。このサイクルは、合意形成プロセス、開発プロセス、説明責任遂行プロセスの3つから成ります。また、障害対応サイクルの結果、再発防止のためにシステムに変更要求が発生した場合は、再び合意形成プロセスから開始することになります。

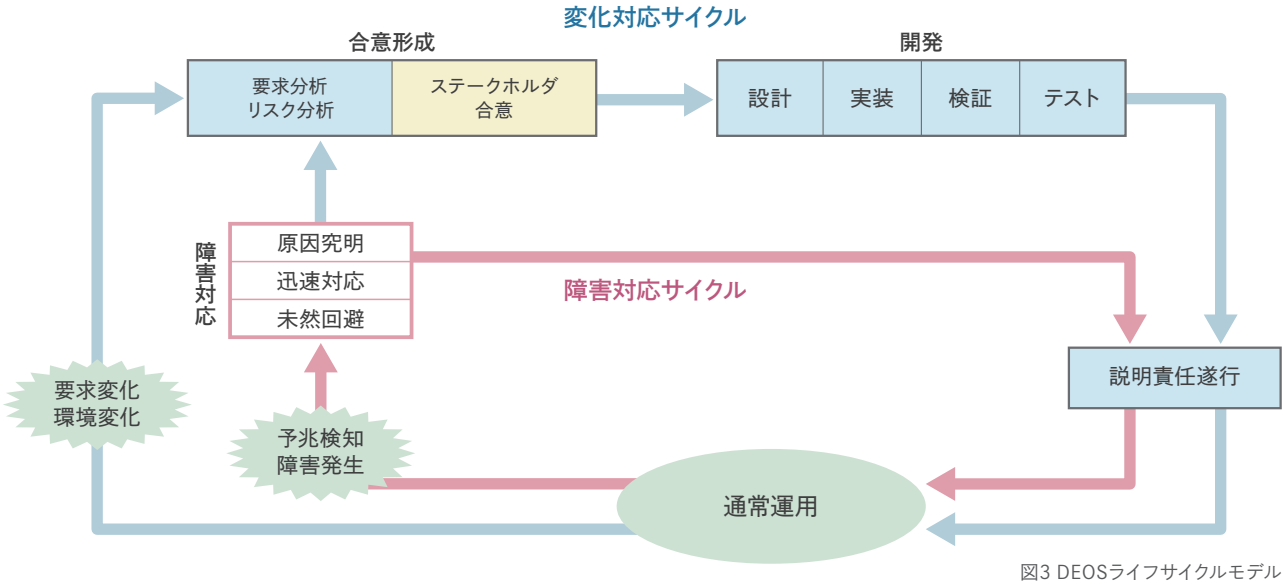



図3 DEOSライフサイクルモデル

まず、開発前にシステムのディペンダビリティに関する要件とその実現方法について合意します。その後の開発プロセスでは、当初のゴールが実現されていることを設計やテストで確認し、システムのディペンダビリティを保証します。さらに、説明責任遂行プロセスでは、想定した前提条件の下でシステムに障害が発生しないと言える理由をステークホルダーに説明し、合意を得る必要があります。

### 3 障害対応サイクル

障害対応サイクルは図3の内側のループを指します。このサイクルは、説明責任プロセスと障害対応プロセスから成ります。通常運用中に障害の予兆あるいは発生を検知したとき、障害対応プロセスとして、障害の未然回避、迅速対応、原因究明を行います。システムに障害が発生していても、障害対策が十分であれば、適切に対処してビジネスが継続できることを示します。ただし、システムに想定外の障害が発生した場合は、原因を究明して再発防止策を説明するとともに、事前に想定できなかった理由も説明する必要があります。

## D-Case

D-Caseは、オープンシステムがディペンダブルであることの論理的な説明を可視化するための表記法です。IEC 62853では、D-Caseを使って議論の過程を証跡として残すことが推奨されています。ここでは、その基となるアシュアランスケースから解説します。

### 1 アシュアランスケース

アシュアランスケースは、システムがディペンダブルであることをステークホルダーが確信するための議論および証拠と定義されています。

アシュアランスケースが生まれた背景は、1988年の北海油田事故をはじめとして、1970年以降に多くの深刻な事故が起こったことでした。このとき、欧米では規格認証の際に安全性が満たされていることの議論および証拠の提示が義務付けられました。この議論および証拠はセーフティケースと呼ばれましたが、それをより一般化した概念としてアシュアランスケースが生まれました。

アシュアランスケースでは、システムがディペンダブルであることを次のように示します[5][6]。

- 説明すべき主張を明示的に定義する。
- 主張が成立する根拠となる証拠を明示的に定義する。
- 説明の前提となるコンテキストを明示的に定義する。
- 証拠によって主張を論理的に説明する。

これらを分かりやすく、かつ統一的な形式で示すために、いくつかの表記法が作られています。

## 2 GSN

GSN（Goal Structuring Notation）は、アシュアランスケースの最も代表的な表記法です。GSNでは、表2に挙げた要素を用いてアシュアランスケースを記述します[10]。

これらの要素を用いて記述したアシュアランスケースの例を図4に示します。これは、山本修一郎氏による連載「要求工学」の中の「要求に基づくテストの十分性」で例として挙げられているものです[11]。この図では、以下のことが示されています。

- ゴールは「システムが要求を満たす」ことを示すことです。
- 要求仕様に基づいて、ゴールを2つの要求（R-1、R-2）に対するゴールに分けて説明します。
- 要求R-1に関するゴールは、テスト項目に基づいて、正常系と異常系の2つに分けて説明します。
- 要求R-1の異常系に関するゴールは、R-1の要求逸脱分析に基づいて、3つの逸脱種別に分けて説明します。
- 要求R-2に関するゴールは、未達成です。

そして、この図には示されていませんが、末端にある3つの逸脱種別に関するゴールについて、それが保証できればエビデンスを付与します。

このように記述することにより、ゴールが達成できていることをシステムの利用者や顧客に分かりやすく説明できます。また、説明のロジックにおける問題も明らかになります。この表記法は、オープンシステムディペンダビリティに特化したものではなく、ゴールを論理的に説明するあらゆる場面で適用できます。

| 名称 | ノード | 解説 |
|---|---|---|
| ゴール | システムは ディペンダブルである | 対象システムに対して、議論すべき命題 |
| 戦略 | ハザードごとに 議論する | ゴールが満たされることを、サブゴールに分割して詳細化するときの議論の仕方 |
| 前提 | ハザードリスト | ゴールや戦略を議論するとき、その前提となる情報 |
| エビデンス | テスト結果 | 詳細化されたゴールを最終的に保証するもの |
| 未達成 | ◇ | ゴールを保証するための十分な議論もしくはエビデンスがないこと |

表2 GSNの主な記述要素

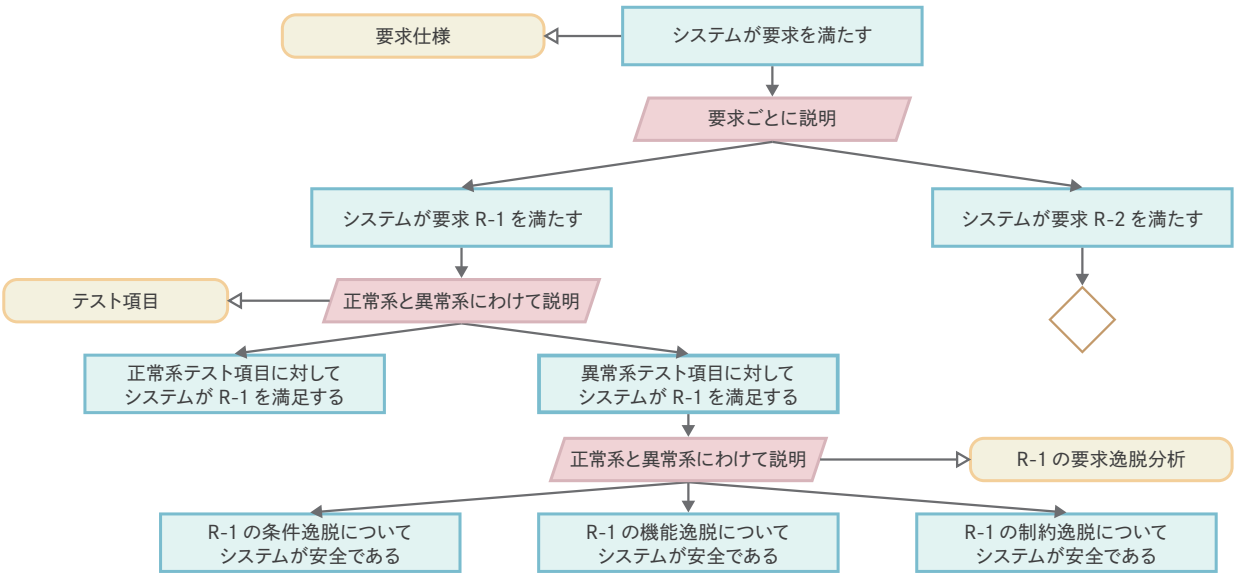



図4 GSNで記述したアシュアランスケースの例[10]

### 3 D-Case

D-Caseは、GSNをさらに拡張した表記法です。D-Caseでは、システムの変化と障害未然防止に対応していることを明確に示すため、GSNに表3の要素を追加しました[12]。ただし、D-Caseの公開事例には、これらの追加要素を使ったものは少なく、GSNの基本要素だけを使って書かれたものが多いのが実状です。

| 名称 | ノード | 解説 |
|---|---|---|
| モニタ |  | 運用中のシステムより取得可能なエビデンス |
| 外部接続 |  | 他のシステムの D-Case へのリンク |

表3 D-Caseで追加された主な記述要素

GSNやD-Caseは単なる表記法にすぎませんが、これらの表記法を用いてアシュアランスケースを記述することで、ゴールが達成できることを議論した過程を残せる利点があります。そのため、仮にシステム障害が起こっても、議論のどこに問題があったかを振り返り、障害を防げなかった原因を容易に突き止めることができます。また、顧客と議論をしながらアシュアランスケースを作成すれば、議論と合意形成を同時に行うこともできます。

## I IEC62853の概要

ここまで、IEC 62853を理解する上で必要な考え方について解説してきました。本章では、これらの考え方とIEC 62853との関係を説明します。

### 1 IEC 62853の構成

IEC 62853は、次のような構成になっています。

1章 適用範囲
2章 引用規格
3章 用語及び定義
4章 オープンシステムディペンダビリティ
5章 適合性
6章 オープンシステムディペンダビリティを達成するための プロセスビュー

メインとなる4章以降は、本稿でここまでに紹介した技術や考え方と密接に結びついています。4章では、オープンシステムディペンダビリティの定義が記載され、その目的や課題、および達成するためにするべきことが書かれています。5章、6章は、それぞれD-Case、DEOSライフサイクルモデルに関係しますので、それらについて次に解説します。

### 2 4つのプロセスビュー

IEC 62853では、DEOSライフサイクルモデルにある「合意形成」、「説明責任遂行」、「変化対応」、「障害対応」の4つの活動を「プロセスビュー」と呼びます[9]。規格の6章では、それぞれのプロセスビューについて、目的とアウトカムを定義しています。アウトカムとは、プロセスビューが適切に遂行されるときに期待される、目に見える結果を意味します。

例えば、プロセスビュー「合意形成」では、その目的とアウトカムの一部は、表4のように定義されています[9]。アウトカムが具体的に

| プロセスビュー名 | 合意形成 |
|---|---|
| プロセスビューの目的 | システム、システムの目的、目標、環境、性能、ライフサイクル、及びこれらの変化に関する共通理解と明示的合意を確立し、維持することである。 |
| プロセスビューのアウトカム（一部） | a)システム等に関して、ステークホルダー間で共通の理解と明示的合意が確立されている。<br>3)枠組みの中で、各ステークホルダーはシステムの目的等とそれらの変化について共通の理解を持つ。これには、システムに関する仮定やステークホルダーの責任についての理解が含まれる。<br>5)明示的合意がa)3)の理解に基づいて作られ、記録されている。記録には、合意形成過程の説明と、合意事項を適切かつ実現可能であるとする理由づけが含まれる。<br><br>b)ステークホルダー間で共通の理解と明示的合意が維持されている。<br>5)合意達成の事実と合意内容、合意形成過程の説明、合意内容を適切かつ実現可能である理由づけがディペンダビリティケースに記録されている。 |

表4 プロセスビューに関する記述内容の例

示されているため、それぞれのプロセスビューでは何をすべきかが明確に分かります。

なお、IEC 62853の規格本編には、プロセスビューの記載があるのみで、それらを使用するライフサイクルモデルには言及されていません。規格の付録Aには、ライフサイクルモデルの例として、DEOSライフサイクルモデルが、その解説とともに紹介されています。

### 3 D-Caseの活用

IEC 62853の5章「適合性」では、オープンシステムディペンダビリティが達成できていることを示すために、D-Case*3が提供されることを推奨しています。それは、プロセスビューのアウトカムの一部で活用されます。表4で示したプロセスビューの例では、D-Caseが関係するアウトカムを挙げました。

また、規格の付録Bとして、D-Caseの例が示されています。この例では、システム開発におけるディペンダビリティではなく、この規格で定められた内容そのものが、ディペンダビリティに関する議論、説明として妥当なものであることを、D-Caseを使って表しています。

## IEC62853の活用と課題

ここまで、IEC 62853とそれに関連する技術や考え方について解説してきました。最後に、それらの具体的な活用や課題について触れたいと思います。

### 1 IEC 62853の活用事例

IEC 62853の活用は、DEOS協会の自動車応用部会で検討されていますが[13]、全面的に活用して開発した事例は、筆者の知る限りではまだありません。ただ、IEC 62853の基になっているDEOSプロセスやD-Case(GSN)を活用した事例、あるいは参考にした事例はあります。

DEOSライフサイクルモデルについては、一例として、理化学研究所の医科学イノベーションハブ推進プログラムで、「開放系総合信頼性*4工学」を導入してシステムを開発しています。ホームページ、および

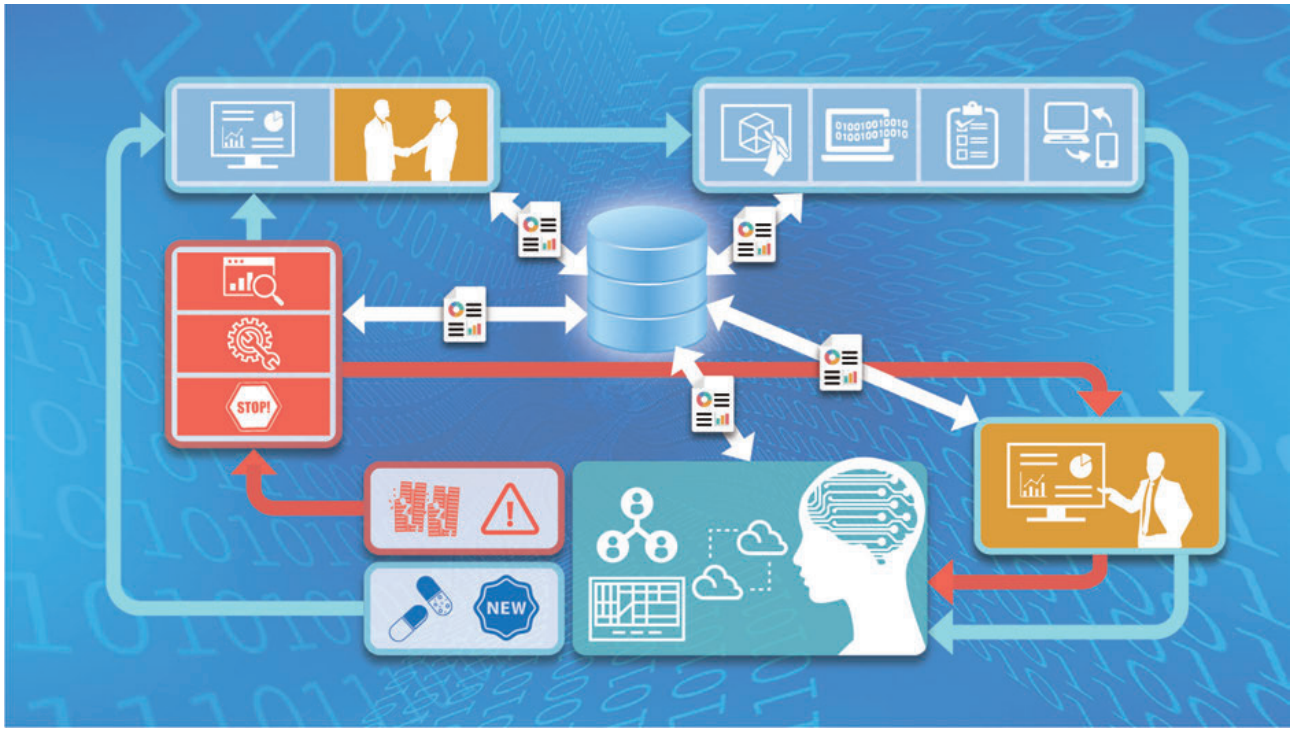



図5 理化学研究所 医科学イノベーションハブ推進プログラムでの開放系総合信頼性工学の図 (ホームページより引用)[14]

*3:本文中では「Dependability Case」となっています。

*4:開放系総合信頼性はOpen systems dependabilityの日本語訳です。

そこで公開されている動画によると、DEOSライフサイクルモデルを基にしたサイクルが示されています[14]。その中には、障害発生時の対応や各方面への説明、合意といった手順が含まれています。

また、D-Caseや、その基となるGSNについては、数多くの事例があります。中でも積極的に活用しているのが宇宙航空研究開発機構（JAXA）です。JAXAは、ソフトウェア品質シンポジウムではGSNを活用した事例をたびたび発表しています[15]。また、リスクシナリオの作成（2019年）、FMEA*5における故障モードの導出（2018年）、IV&V*6に必要な技術者の能力の表現（2016年）と、さまざまな場面で活用しています。JAXA以外では、D-Case研究会で年に1〜2回の研究会が開催され、事例や研究成果が発表されています[12]。

DEOSライフサイクルモデルを活用している理化学研究所や、D-Case（GSN）を活用しているJAXAのように、これらの技術や考え方は高品質が求められるシステムの開発で必要とされていることが分かります。

## 2 IEC 62853の課題

一方、IEC 62853の適用においては課題があると考えます。最も大きい課題は、前述のとおり活用事例がないことでしょう。事例がないと、規格を理解してもどうやって使うのが正解なのかを判断できない人が多いと思います。そのため、規格の適用に踏み出せず、事例が増えないという悪循環に陥ります。IEC 62853を適用して効果を得られた事例が望ましいですが、うまくいかなかった事例であっても、規格を実践的なものに洗練させるためには有効であると考えます。

事例がないことも関連しますが、IEC 62853は理論的に正しいと考えられることをまとめたものであり、理想的な世界を示しています。ただ、理想と現実が違うことは誰もが体験しています。理論上は正しくても、現実世界ではうまくいかないことはしばしばあります。規格が「絵に描いた餅」にならないよう、身近なシステムの例などを用いて実際の開発プロセスや運用プロセスの中での使いどころを示すことを、規格の策定側に期待したいです。

最後に、IEC 62853の中で使うD-Case（GSN）についての課題を挙げます。本稿で紹介した中で、D-Caseは比較的使いやすいものだと思います。ただ、実際にD-Caseの図を描いてみると、迷う点がいくつかあります。例えば、ゴールをどうやって分解するのがよいのか、

ゴールをどこまで詳細化していけばよいのかといった点です。このように判断に迷う点があると、同じゴールを目指すものであっても、作る人によって異なるD-Caseが出来上がります。つまり、D-Caseの書き方は属人性が高いと言えます。組織内などで使用するときは、組織内での作成方針や標準的な書き方などを決めて、人による違いが出ないような工夫が必要になります。

## 3 IEC 62853の導入

最後に、本稿を読んでIEC 62853に興味を持たれて、導入してみたいと思った方に、筆者がお勧めする導入方法を紹介します。

前述のとおり、IEC 62853を本格的に活用した事例がないため、いきなり導入するのは難しいと思います。そこで、まずはD-Case（GSN）を使うところから始めてみるのがよいでしょう。D-Caseは、形としてはマインドマップに似ていますが、単にアイデアを書き連ねるだけではなく、アイデアの裏にある根拠を明示するため、他者に説明するだけでなく、自らの振り返りにも使えます。ゴール指向のロジカルシンキングをする手法として、活用できる場面は幅広くあります。

そして、D-Caseを描くことに慣れてきたら、それを合意形成や障害対応の場面で使うことで、DEOSライフサイクルモデルで示している世界に近づいていきます。つまり、いきなりIEC 62853やDEOSライフサイクルモデルの世界を実現しようとするのではなく、できることからボトムアップで実施していき、全体のプロセスを整備していくのがよいと考えます。

＊5：「FMEA」は、“Failure Mode and Effect Analysis”の略称で、故障および不具合の未然防止を目的とした安全分析手法の一種

＊6：ソフトウェア開発組織や開発委託組織から独立した組織の指導と管理に基づいて、検証（ベリフィケーション）と妥当性確認（バリデーション）を実施すること。

IEC 62853やDEOSライフサイクルモデルでは、要求の変化に対応しようとする点、ステークホルダーとの合意形成や説明責任といったコミュニケーションを重視している点、障害が発生した際に迅速な対応をするという点など、アジャイル開発と似ている点があります。また、開発だけでなく、リリース後の運用も視野に入れて、運用中の変化対応や障害対応を想定している点はDevOpsに似ています。この規格やモデルは、アジャイル開発やDevOpsを意識して作られたものではありませんが、期せずしてそれらとの類似性や親和性が高いものになっています。近年のトレンドになっているこれらの開発手法においても、十分に参考になると思います。例えば、アジャイル開発のスプリントが終了する時点での品質についての説明や、リリース後に障害が見つかってしまった場合の原因や対策の説明などで、D-Caseを活用できるでしょう。

## まとめ

本稿を通して、「IEC 62853 Open systems dependability」という規格の存在を知っていただけたと思いますが、その内容についての説明はまだまだ足りないと感じています。さらに詳しい情報を知りたい方は、DEOS協会がホームページで発信している情報などをご確認ください[16]。ただし、前の章でも述べたとおり、具体的な事例が少なく、この規格自体の真の有効性が示されていないのが実状です。しかし、この規格の基になった理論は日本で考えられたものであることから、日本のソフトウェア開発のスタイルや品質保証の考え方に適合するものと考えます。そこで、事例が多く使いやすいD-Caseを手始めに使っていただき、この規格が日本的品質保証の一助になることを期待しています。

本稿をきっかけに、ディペンダビリティを保証するための技術や考え方に興味を持っていただければ幸いです。

参考文献

[1] SQuBOK策定部会, ソフトウェア品質知識体系ガイド 第2版 –SQuBOK Guide V2-, オーム社, 2014
[2] IFIP WG10.4 on Dependable Computing and Fault Tolerance, http://www.dependability.org/wg10.4/
[3] 日本工業規格, JIS X 25010:2013 システム及びソフトウェア製品の品質要求及び評価(SQuaRE)
－システム及びソフトウェア品質モデル, 日本規格協会, 2013
[4] IT用語辞典e-Words, http://e-words.jp/w/インテグリティ.html, アクセス日: 2020年5月1日
[5] 所眞理雄, DEOS: 変化しつづけるシステムのためのディペンダビリティ工学, 近代科学社, 2014
[6] M. Tokoro, ed., "Open Systems Dependability - Dependability Engineering for Ever-Changing Systems", CRC Press, 2012.
[7] DEOSプロジェクト, http://www.jst.go.jp/crest/crest-os/osddeos/index-j.html
[8] ディペンダビリティ技術推進協会, http://deos.or.jp/index-j.html
[9] 一般社団法人ディペンダビリティ技術推進協会 技術部会、標準化部会, はじめてみる62853の実装,
http://deos.or.jp/link/obj/pdf/Introduciton62853Implementation-DEOS20180605.pdf
[10] GSN Standard, http://www.goalstructuringnotation.info/
[11] 山本修一郎, 要求に基づくテストの十分性, ICTソリューション総合誌　月刊ビジネスコミュニケーション,https://www.bcm.co.jp/site/youkyu/youkyu108.html
[12] D-Case研究会, http://www.dcase.jp
[13] 一般社団法人 ディペンダビリティ技術推進協会(DEOS協会) 自動車応用部会, https://www.daddcar.deos.or.jp/
[14] 理化学研究所 医科学イノベーションハブ推進プログラム 研究開発テーマ 開放系システム総合信頼性(DEOS),
https://www.mih.riken.jp/thema/dependability.html
[15] ソフトウェア品質シンポジウム, https://www.juse.jp/sqip/symposium/
[16] DEOS協会, http://deos.or.jp/index-j.html

本記事の内容に関するご質問、お問い合わせは、ベリサーブ 広報・マーケティング部 Email:verinavi@veriserve.co.jp までお寄せください。

## トレーニング概要

ベリサーブでは、2020年も2月19日(水)から21日(金)までの3日間、IoT(Internet of Things)デバイスに対するハッキング方法やセキュリティを破るアプローチに関するトレーニングを行いました(2019年に続き2回目)。

逐次通訳を交えての座学講義の後、ベリサーブのスタッフが、適宜サポートしながら、トレーニングキットを使い、実際に攻撃者がIoTデバイスに侵入する手法などを学びます。IoTデバイス・IoTシステムのセキュリティを学びたい方を対象としています。前提知識は、一般的なOS、コンピュータ・アーキテクチャー、ネットワーキングに関わる基本的な知識をお持ちの方であればどなたでも参加できます。

# ベリサーブでIoTハッキング・セキュリティトレーニングを開催!

## IoTを囲む状況

IoT機器は、テレビ、冷蔵庫、エアコン、時計、自動車、フィットネス機器、産業用機械など、さまざまなモノに使われています。毎月のように新しいスマートデバイスが登場していることはもちろん、これまでアナログだった機器もどんどんデジタル化し、インターネットに接続することでデータ連携をしています。

2020年には、ネットワークに接続するIoT機器が530億個に増加すると予測されており、ネットワークを経由したIoT機器へのサイバー攻撃の脅威が増大することが懸念されています。[*1]

しかし、急に普及が進んだこともあり、利用者や開発者のセキュリティへの意識は低い状況です。

## IoTセキュリティが必要な理由

セキュリティを重要視するのは、いくつかの理由があります。

例えば、IoT機器は、ひとたび攻撃を受けると、機器単体にとどまらずネットワークを介して関連するIoTシステム・IoTサービス全体に影響を及ぼす可能性が出てきます。特に、自動車や医療機器に攻撃の影響が及んだ場合、利用者が生命の危険にさらされる場面も想定されます。

Aditya Gupta氏
100を超えるIoTデバイスの調査・テスト実績を持つ、
Attify社のファウンダー

カメラなどを通じて、プライバシーが漏洩する危険性もあります。

また、IoT機器には10年以上にわたって使用されるものも多く、システム構築・ネットワーク接続時に適用したセキュリティ対策が時間の経過とともに時代遅れとなり、セキュリティ対策が不十分になった機器がネットワークに接続され続けることになります。そして、それらの多くの機器へのパッチ適用も困難です。

実際に、上記の懸念の通り、IoTデバイスへの攻撃件数は、増加し続けています。

2019年の前半には、1億件を超える攻撃が検出されているというレポートもあります（2018年同期のほぼ9倍）。*2

特に有名な事例では、2016年10月、DNSサービスプロバイダ「Dyn」が、DDoS攻撃*3を受けるという事件がありました。セキュリティ対策が不十分なIoT機器に感染した不正プログラム「Mirai」が、世界中で何十万台ものビデオレコーダーやネットワークカメラに感染。一斉に攻撃を始め、インターネットを支えるインフラストラクチャの重要部分を機能停止させ、多くの大手サイトが利用不能となりました。

## トレーニング内容

このようなIoTの危険性を把握した上で、商品・サービスを考えることが必要になっています。

しかし、防御一辺倒の思考では不十分です。IoTデバイスには、さまざまな技術、アーキテクチャ、プロトコルがあり、攻撃手法も常に変化していることを考えると、最新の防御技術を知るだけでなく、侵入後の検知、対処、回復までの一連のプロセスを視野に入れたセキュリティ対策の実現が急務です。

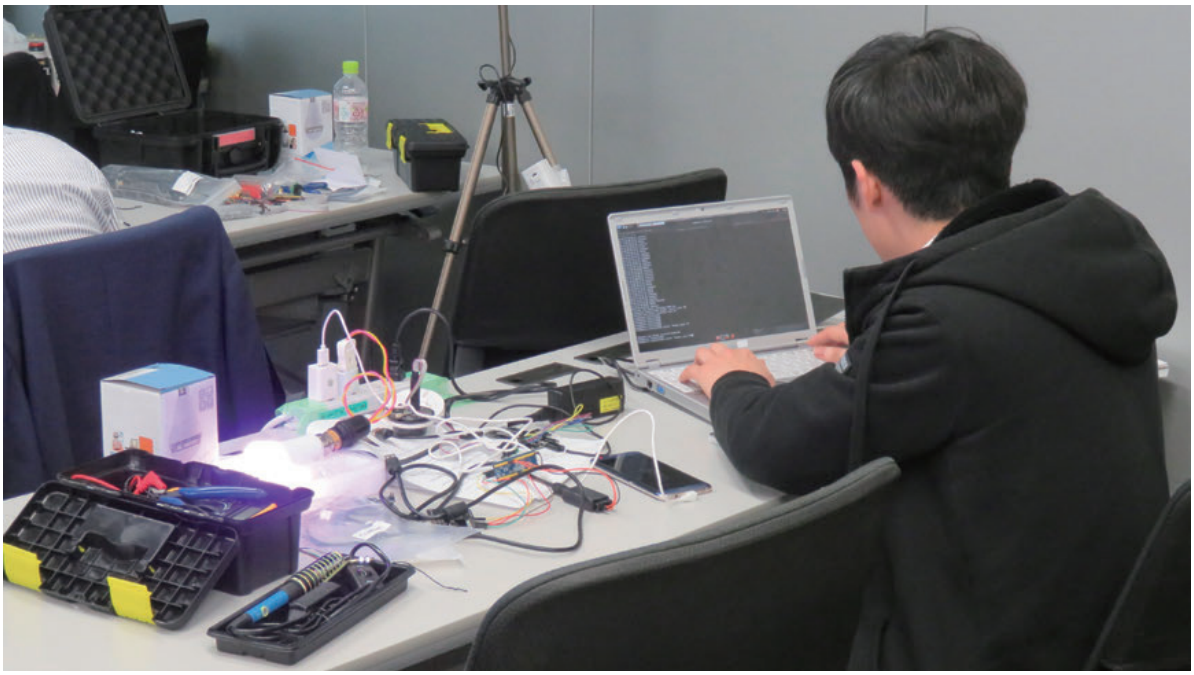
今回のトレーニングでは、デモとハンズオンを通じて、いかに我々が攻撃を受ける側の常識が通じないか意識するとともに、セキュリティ担当者が企業の事業活動を守り抜くために、どのような考え方が必要なのか、考えるきっかけをつくります。

＊1：IoTセキュリティガイドライン（METI/経済産業省）
https://www.soumu.go.jp/main_content/000428393.pdf
＊2：Over 100 Million IoT Attacks Detected in 1H 2019 (Infosecurity Magazine), https://www.infosecurity-magazine.com/news/over-100-million-iot-attacks/
＊3：DDoS攻撃（Distributed Denial of Service attack）は、トロイの木馬などのマルウェアを使って複数のマシンを乗っ取った上で、攻撃目標であるサイトやサーバに対して大量のデータを送り付けることで行われる攻撃。

ベリサーブで
IoTハッキング・セキュリティトレーニングを開催!

## トレーニングの流れ

| | |
|---|---|
| 1日目 | 以下の実践を通して、ARMやMIPSアーキテクチャのエクスプロイト、ファームウェアの抽出とデバッグ、ファームウェアのエミュレーションなどの概念を学びました。<br>● IoTセキュリティアーキテクチャの内部概念<br>● 既知のIoTデバイスの脆弱性およびケーススタディ<br>● IoTデバイスのファームウェアを取得して、リバースエンジニアリング<br>● セキュリティ上の問題を発見し、実際に悪用される可能性の認識 |
| 2日目 | 実際のIoTデバイスを分解して、回路基盤の持つコンポーネントを理解し、デバイスのルートを取得しました。<br>● UARTのエクスプロイト<br>● JTAGのデバッグ<br>● デバイスからフラッシュチップの内容をダンプする<br>利用するツールや配布資料と共にこれらのノウハウを学習し、<br>さまざまなIoTデバイスにも適用することができます。 |
| 3日目 | デバイスをリモートから攻撃するために必要なノウハウを学びました。<br>● BLEを使用したデバイスを盗聴して攻撃<br>● 攻撃に利用するカスタムラジオを作成<br>ツールと演習を組み合わせることで、実際に攻撃者がIoTデバイスに侵入するために<br>必要なことを学習しました。 |



2019 年のトレーニングの様子

## トレーニング後

トレーニングが終わるころには、以下のようなことができるようになります。

- 特定のIoTデバイスに関連するセキュリティリスクを理解する
- 侵入テストを実行し、IoTデバイスの脆弱性を見つける
- ハードウェアやラジオなどのさまざまなテクノロジーを迅速に利用する
- IoTデバイスの評価に使用されるツールと手法に精通する
- IoTデバイスに対して、新しい悪用戦術を思いつくようになる

トレーニングでは、IoT機器へのアプローチの考え方を重視しますので、コマンドが覚えられないということで焦る必要はありません。また、トレーニングで使用するキットはお持ち帰りいただけますので、トレーニング後も継続的に学習いただけます。

## おわりに

本トレーニングは、企業のセキュリティ担当者だけでなく、商品を販売し、サービスを運用に関わる方にとって有意義なものとなっています。時期は未定ですが、今後、トレーニングを実施することがあると思いますので、ご興味がある方は、以下のページからお問い合わせください。

**本トレーニングに関するお問い合わせ先 株式会社ベリサーブ　ソリューション事業部 イベント窓口**
**EMAIL：sol_event@veriserve.co.jp　URL（トレーニング詳細）：https://www.veriserve.co.jp/service/detail/iothacking.html**

品質を創造する企業

VERISERVE

Veriserve Navigation 2020年6月号（ベリサーブナビゲーション）

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9F

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