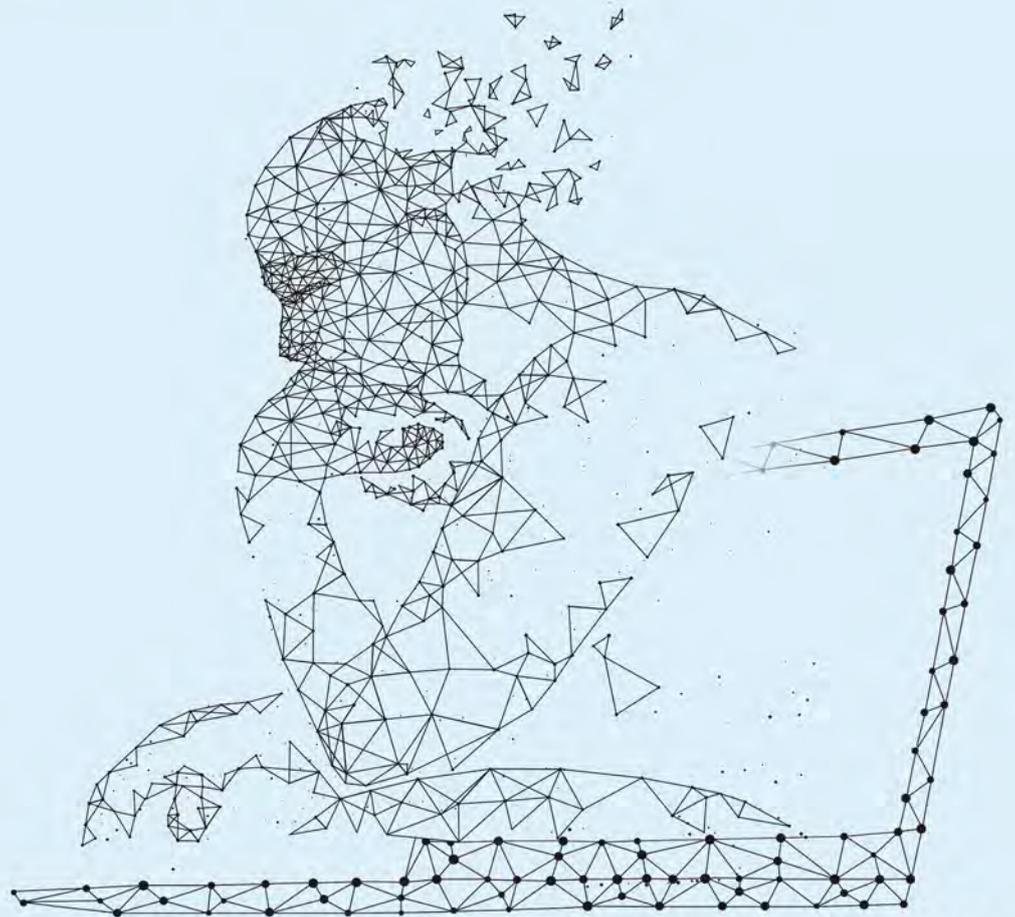


ベリサーブナビゲーション

Veriserve Navigation

DX時代に求められる ソフトウェア品質とは？

ベリサーブ アカデミック イニシアティブ 2020 DAY 1



2021 February Vol.21

DX時代に求められる ソフトウェア品質とは？

ベリサーブ アカデミック イニシアティブ 2020 特集

Contents

| | | | |
|----|---|--|----|
| 特集 | 1 | 正解がない時代のプロダクト開発 及川 卓也氏 | 06 |
| 特集 | 2 | ウェブサービス開発における品質改善の取り組み 2020 中野 直樹氏 | 14 |
| 特集 | 3 | テストエンジニアがテストベース視点で考察した 「Testing for 人工知能」 佐々木 方規 | 22 |
| 特集 | 4 | テスト自動化現場の「いま」と、次のステップに進むために 伊藤 由貴 | 28 |
| | | サービス紹介 | |
| | | テスト管理をシンプルに 快適、スケーラブルなテスト管理ツール 「QualityForward」導入事例のご紹介 | 32 |



ソフトウェア品質とは？
Heriserve Academic Initiative 2020
Heriserve アカデミック イニシアティブ
play Movie
2020 12.8 Tue 9 Wed 12:45-16:00 参加費：無料
〜品質向上の最先端がここに集結〜
品質向上のリーディングカンパニー、Heriserveがお届けするソフトウェア開発のフロンティアからの研究報告。
変革の時代に活路を見出すための最先端の研究や新たなチャレンジに立ち向かう現場からの
リアルなレポートなど明日の「品質の創造」を考える2日間です。

Veriserve Academic Initiative 2020

ベリサーブ アカデミック イニシアティブ

DX時代に求められるソフトウェア品質とは？

ベリサーブ アカデミック イニシアティブ 2020 レポート

品質向上のリーディングカンパニー・ベリサーブがお送りする「ベリサーブ アカデミック イニシアティブ」は、最先端の研究や現場からのリアルなレポートなど、ソフトウェア開発のフロンティアからの研究報告によって明日の「品質の創造」を考えるイベントです。

5回目となる今回は、新たな取り組みとして場をオンラインに移し、2020年12月8日、9日の2日間にわたって開催しました。ベリサーブナビゲーションVol.21では、初日の講演内容をダイジェストでお届けします。

2020年の 講演プログラムについて

DX時代への突入に伴い、あらゆる企業が変革の機会を迎え、時代のニーズに沿った、新しい観点でのソフトウェア品質と、その品質のつくり方が問われています。

ベリサーブアカデミックイニシアティブ2020では、「DX時代に求められるソフトウェア品質とは？」というテーマのもと、この分野の第一人者である研究者や開発者の方々4名にご講演いただいたほか、ベリサーブ社員4名が当社の考え方や取り組みをご報告しました。2日間のプログラムは以下の通りです。

12月8日(火)

基調講演

及川 卓也氏 (Tably株式会社)
正解がない時代のプロダクト開発

技術セッション

佐々木 方規 (株式会社ベリサーブ)
テストエンジニアがテストベース視点で考察した
「Testing for 人工知能」

技術セッション

伊藤 由貴 (株式会社ベリサーブ)
テスト自動化現場の「いま」と、次のステップに進むために

ゲストセッション

中野 直樹氏 (株式会社LIFULL)
ウェブサービス開発における品質改善の取り組み2020



12月9日(水)

基調講演

伊本 貴士氏 (メディアスケッチ株式会社)

AI・IoT時代におけるプロジェクトマネジメントと品質管理のあり方

技術セッション

松木 晋祐 (株式会社ベリサーブ)

AIとソフトウェア品質技術

技術セッション

桑野 修 (株式会社ベリサーブ)

DX時代のITサービスに要求される「安心・安全な品質」とは？
～より安心・安全なITシステムの構築を支援する品質エンジニアとしてのアプローチ～

ゲストセッション

鷺崎 弘宜氏 (早稲田大学)

パターンQA to AQによるAgile Quality(アジャイル品質)への変革と事例



表紙デザインコンセプト
変革の時代に活路を見出すためにあらゆるパーツ(知識)をインプットし、解決への道を模索している「最先端の研究をする人」を表現しております。



正解がない時代の プロダクト開発



及川 卓也氏
おいかわ たくや

Tably株式会社 代表取締役 Technology Enabler

大学卒業後、外資系コンピューター企業に就職。ソフトウェアの研究開発に従事する。その後、外資系IT企業にて、プロダクトマネージャーやエンジニアリングマネージャーとして勤務の後、スタートアップを経て、独立。2019年1月、テクノロジーにより企業や社会の変革を支援するTably株式会社を設立。著書『ソフトウェア・ファースト～あらゆるビジネスを一変させる最強戦略～』（日経BP）



はじめに

社会や顧客のニーズが見えにくい時代において、プロダクト開発の仕方も変わりつつあります。本講演では、今私たちが意識したいプロダクト開発における注意点、プロダクト開発を支える組織のあり方、品質との向き合い方について、自著『ソフトウェア・ファースト』の内容を交えつつお話をさせていただきます。

プロダクトの時代

I 価値と需要の変化

この10～15年の間に世の中は大きく変化してきました。端的に言うと、顧客に対して提供する「価値」が変わってきている、逆に言えば顧客や社会が求めているものが大きく変わってきていると言うことができます。

その顕著な例を私たちは音楽産業の中に見ることができます。かつてアナログレコードからCDへと音楽メディアは大きく形を変えました。その後さらにメディアを必要としないMP3ファイルが主体となっていきました。しかし、レコード、CD、MP3はいずれも販売・購入を経て所有するという性格においては大きな違いはありませんでした。ところが現在では「所有」ではなく、「いくらでも利用できる」という価値が求められるようになりました。いわゆる聴き放題というスタイルが一般的になり、その結果、人と音楽の関わり方も変容しました。例えば昔の私なら好きなアーティストの曲だけを指名買っていたものですが、聴き放題が可能となった今は、知らない曲でも積極的に聴き、誰かが「この曲いいよ」と言えば試しに聴き、「今どきの若いアーティストもいいじゃないか」と、より幅広い体験を楽しめるようになりました。

「所有」から「利用」、「体験」へと価値が変化したのは音楽産業だけではありません。今や世の中のおもむきで同様のことが起きています。背景にあるのはビジネスモデルや収益体制そのものの変化です。かつての音楽は一度購入すればユーザーの所有となり、提供する側からすれば「買ってもらえるかどうか」を軸としてビジネスが成り立っていました。ところが聴き放題に代表されるようにサブスクリプションというスタイルが一般的になると、企業は顧客に「使い続けて」もらわないと収益を上げられなくなります。例えば、

ゲームの世界では特別な武器やアイテムを使いたい時、ユーザーは課金をしなければなりません。そこでメーカーは課金し続けてもらうために魅力あるアイテムを提供し続ける努力をする。そうすることでゲームソフトをいかに使い続けてもらうかというビジネスモデルが成り立っているのです(図1)。

一方、価値の変化とともに、人々や社会の需要を把握することも困難になりつつあります。

現代はVUCA (Volatility=変動、Uncertainty=不確実、Complexity=複雑、Ambiguity=曖昧)の時代と言われますが、人々の需要も複雑化、多様化しています。かつての高度成長期は世の中無いものだらけで、安く質の良いものを提供すればビジネスが成り立つ時代でした。人々の需要もシンプルかつ明確で、夏暑くなれば扇風機、技術が進化すればエアコンといった具合でした。例えば二十代男性ならこれ、三十代女性ならこれといった具合に、典型的なデモグラフィック属性に基づくことで比較的簡単に需要を予測することができました。ところが今、私たちの身の回りにはすでに物が溢れ、昔のように需要が見えなくなっています。時には自分の欲しいものすら分からない。欲しいものを見つけてもすぐに変動してしまう。まさにプロダクトを取り巻く世界にもVUCAの時代が到来しているわけです。

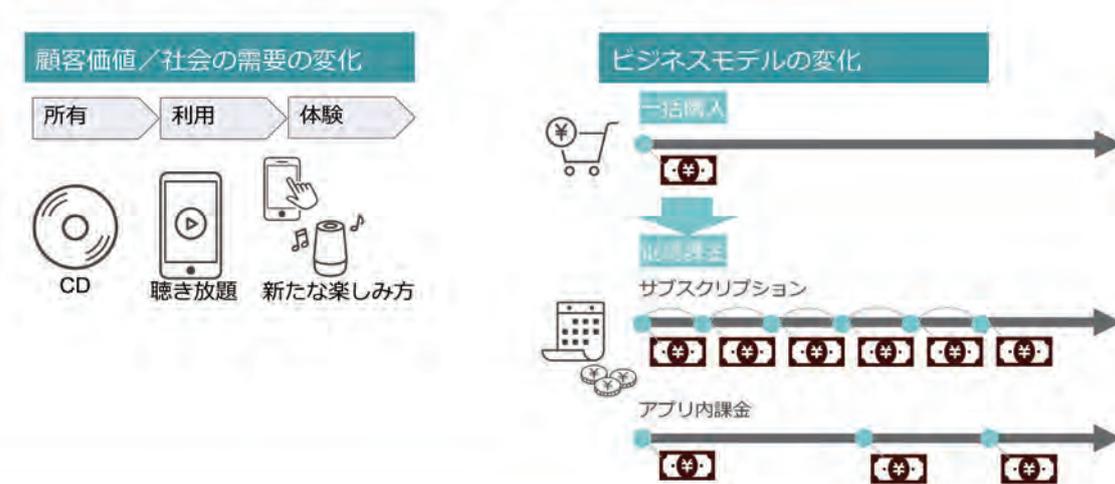
II 見えない需要を見つける

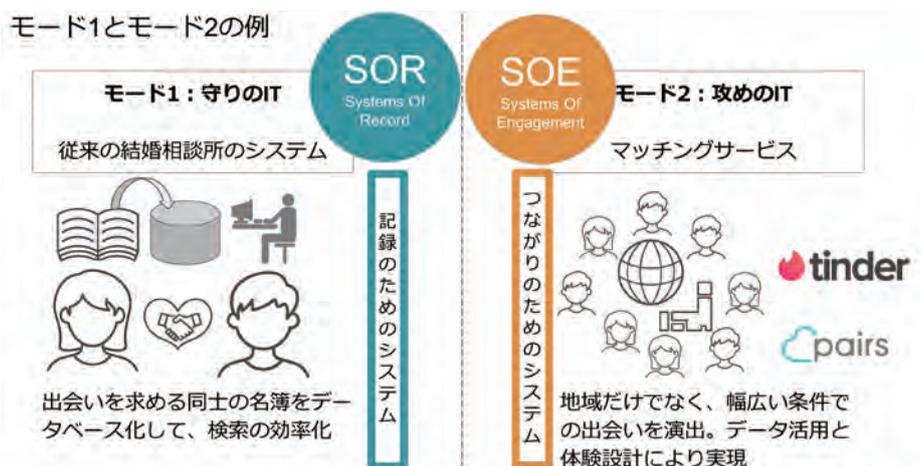
こういった時代において、人々が本当に欲しいと思うもの、あるいは継続的に使い続けてもらえる魅力的な商品やサービスを模索することは非常に難しくなっています。勘や直感で正解を導くことは天才的なヒットメーカーでなければ不可能です。結局のところ、一つ一つ仮説を立て、その仮説を何度も検証していく、この仮説と検証を素早く繰り返していくことが、現代のものづくりには大切になっていきます。

この時必要となってくるのが「ユーザー接点」です。例えば、リテール(小売)の分野ではキャッシュレス決済が普及しつつありますが、その目的の一つは、できるだけ多くの顧客、多くのショップに利用してもらうことによって膨大なユーザー接点を獲得することにあります。このユーザー接点の蓄積から顧客の行動情報を取得する、さらにそこから需要を読み取ることでプロダクトやサービスの将来の姿を考えていくわけです。

もう一つ大事な事実としてお話したいのが、ITの位置付けの変化です。かつて、ITは人がやっていたことを機械に置き換えるというものでした。しかし、今はそもそも人ができないことを行う、という位置付けに変化してきています(図2)。この流れについて婚活サービスを例に説明しますと、昔のお見合いは、その前提として町内の結婚適齢期の男女のデータ(趣味嗜好や学歴や年取など)というのは

顧客価値／社会の需要の変化とプロダクト





Copyright © 2020 Tably inc. All Rights Reserved

図2 ITの位置づけの変化

町内会の世話好きのおじさんやおばさんの頭の中にありました。時代が進むと、これをコンピュータでデータベース化してマッチングしていくようになる。しかし、これは従来人がやっていた作業をITに置き換えたもので「守りのIT」または「SOR=System of Record」と呼ばれるものです。

これに対して、データベースの照合レベルではなく、膨大なユーザー接点の中から、例えばフェイスブックへの何気ない投稿などから、投稿した本人も気付かないような微妙なシグナルを読み取ってその人の趣味・嗜好を分析し、そこから出会いをレコメンドするという方法が現代の婚活サービスに取り入れられています。このような顧客とのつながりや出会いのためのIT活用は「攻めのIT」[SOE=System of Engagement]と呼ばれ、CRM(顧客関係管理)やMA(マーケティングオートメーション)など、顧客とのつながりを強くし、いかに顧客への価値を最大化するかを考える上で役立てられています。また、膨大なデータをいかにして取得しどのように分析するか、そこからどのような洞察を得るかという「SOI=System of Insight」という仕組みも必要になります。

プロダクトの開発

I プロダクトとは何か

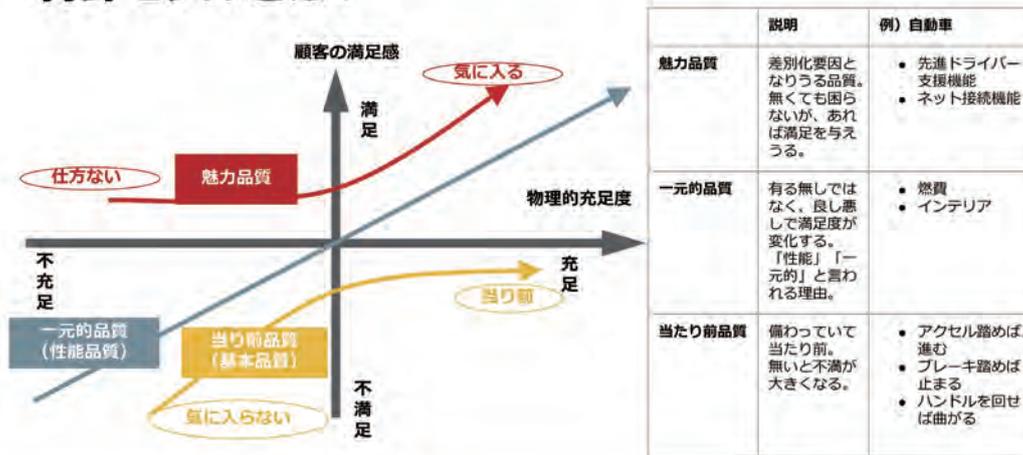
このように社会や顧客のニーズが見えにくい時代にあって、プロダクト開発の仕方も変わりつつあります。

プロダクトとは何かについてあらためて考えてみましょう。プロダクトにはユーザーの期待しているものに答えるという大きな使命があります。大前提として、プロダクトはそのプロダクトが考える未来像、どのような世界にしたいかという「プロダクトビジョン」を持っていることが非常に重要です。その世界の住民(ユーザー)が理想的な状態になるのを妨げている課題を解決する。これこそが、プロダクトの大きなゴールとなります。

II 品質をどのように考えるか

プロダクト開発のプロセスにおいてはさまざまな局面で意思決定が求められます。例えば「プロダクトの設計においてエラー処理をどこまで考えるべきか」「ソフトウェアの実装段階で、見つかったバグをどこまで修正するか」などは典型的な例です。そして、これら意思決定において重要な判断材料となるのが「品質」です。重要なのは、プロダクトチーム内における「品質」についての定義統一であり、その品質基準を事前に明確にしておくことです。また品質責任についてはプロダクトチーム全員で負うことも大切です。

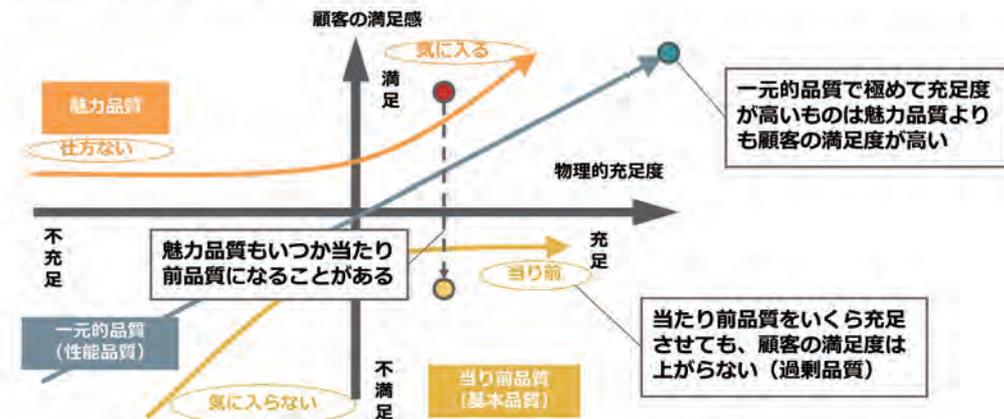
狩野モデルとは？



Copyright © 2020 Tably inc. All Rights Reserved

図3 狩野モデルにおける3つの品質

狩野モデルでの注意



Copyright © 2020 Tably inc. All Rights Reserved

図4 狩野モデルでの注意点

それでは改めて「品質」とは何でしょうか。それは「ユーザーが望む価値をどこまで提供できているかの尺度」と考えることができます。

ここではユーザーの求める価値に焦点を当てて品質をモデル化した東京理科大学の狩野紀昭先生による「狩野モデル」を紹介いたします。狩野モデルでは品質は3種類に分類されます(図3)。最初が「当たり前品質」と呼ばれるもので、文字通り「あって当たり前」の基本品質です。自動車で例えるなら「アクセル(踏み始め走る)」「ハンドル(回せば曲がる)」などがこれに当たります。次に「一元的品質」でこれは性能品質とも呼ばれ、図右上に向かって性能が良くなるほど顧客の満足度も上がります。自動車では燃費性能やインテリアなどが該当します。最後に「魅力品質」がありますが、これは一種の

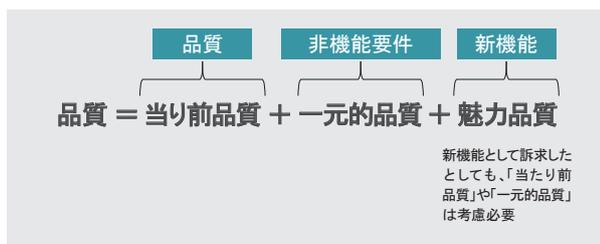
差別化要因であり、魅力品質が高いほど顧客満足感や購買欲求も高まります。

この3種類の品質にはそれぞれ注意すべき点があります(図4)。まず「当たり前品質」は「あって当たり前」なので、いくら充足させても顧客満足度は上がりません。ここにこだわり過ぎることは過剰品質につながる恐れがあります。「一元的品質」は重要で、この品質の充足度を高めていくと「魅力品質」よりも顧客満足度が高くなることもあり、しっかりコストをかけて作り上げていく必要があります。また「魅力品質」は非常に高い顧客満足度をもたらしますが、普及した瞬間に「当たり前」になってしまうことがあります。例えばかつては「魅力品質」であった自動車のABS(アンチロックブレーキ)なども、

現在では多くの自動車に搭載されるようになりました。「魅力品質」は常に変容するものだという事に留意する必要があります。

プロダクトの品質について狩野モデルを用いて考えるなら、「当たり前品質」+「一元的品質」+「魅力品質」と足していき、さらにプロダクトごとに重み(重点)がありますから、これらを掛け合わせた総和こそ「品質」となります(図5)。「求められている品質はどの品質であるか?」マーケット、顧客、プロダクトに応じて判断し、チーム全員で合意することが重要となります。

狩野モデルを用いてのプロダクト判断



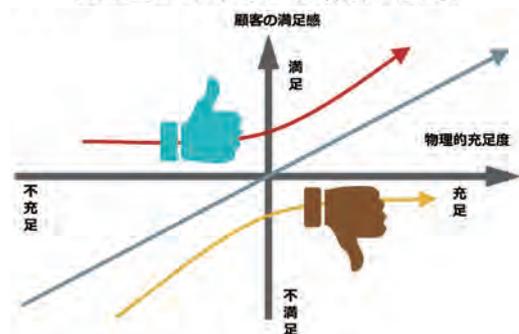
$$\text{品質} = \text{当たり前品質} \times \text{重み} + \text{一元的品質} \times \text{重み} + \text{魅力品質} \times \text{重み}$$

* どの品質が顧客に求められているかの判断

Copyright © 2020 Tably inc. All Rights Reserved

図5 狩野モデルを用いてのプロダクト判断

Yesという人にYesと言わせ、
Noという人には黙らせる



Copyright © 2020 Tably inc. All Rights Reserved

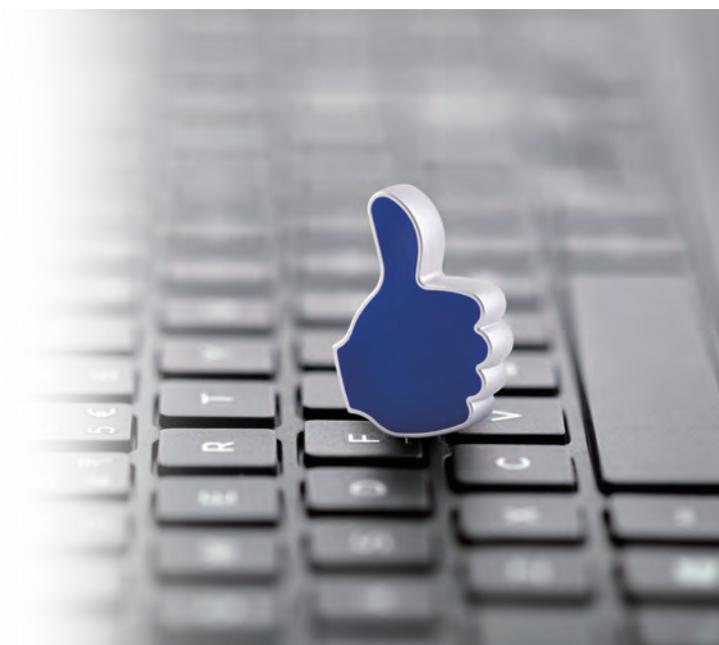
図6 3つの品質の充足度と顧客の満足度の関係

これと同じことをクレディセゾンのCTOである小野和俊氏もその著書『その仕事、全部やめてみよう』の中で、「谷を埋めるな、山を作れ」という見事な言葉で説明しています。つまり、足りないところを埋めていくことは分かりやすいので誰もがやりたくなるのですが、本来は山(魅力品質)を作らない限りプロダクトは人々の心に響かないということです。この山を作るということを意識して、私たちは「品質」を考えていくことが大事かと思えます。

III 2種類のユーザーへの対応

メモアプリで有名なエバーノートの創業者のフィル・リーゼンは、プロダクトに対する人々の反応には2種類あると語っています。一つは「良い体験が得られたことに対してYesという人」であり、もう一つは「不満に対してNoという人」です。前者はメリットに目を向け「これ欲しいね」と考え、後者は「あれがない、これがない、使いにくい」とデメリットに目を向けます。

これに対してフィルは狩野モデルと同じ図を用いて「Yesという人にはYesと言わせ、Noという人には黙っていてもらおう」と言っています(図6)。このNoという人というのは「当たり前品質」にこだわる人であり、この人たちに向けて一所懸命ケアしても「買う理由」は生まれません。一方、Yesという人は魅力品質に目を向けて評価をします。初代のiPhoneが発売された時、コピー&ペースト機能がないことに不満をならす人(Noという人)が大勢いましたが、他方にはiPhoneならではのインターフェース、あのタッチの快適な体験を「これ、いいね」と賞賛し購入する人々(Yesという人)がいました。



プロダクトの組織

では、プロダクトを作る時、いったい誰が何をすればいいのでしょうか。

まずプロダクトマネージャーという存在があります。これはプロダクト組織におけるCEOと呼ばれています。

もちろん本来のCEOではなく、人事権なども持たない「ミニCEO」です。開発から品質保証、法務、マーケティング、広報など組織横断的にチームを率い、プロダクトにおけるすべてのフェーズに関わりその都度必要な意思決定を下します(図7)。そして高いビジョンと熱意、全体を見通す能力によってプロダクトを成功に導くことが求められます。

プロダクトマネージャーがプロダクトの成功に責任を持ち、「何を作るか(What)」を決定するのに対し、エンジニアはそれを「技術的にどのように実現するか(How)」を担い、かつ責任を負います。さらにプロダクトには、エンジニアリングマネージャーが関わります。

こちらは直接プログラムを書くわけではありませんが、エンジニアの生産性と創造性を発揮するための組織作りを通してプロダクトの成功にコミットしている存在です。このように、プロダクトマネージャーとエンジニアリングマネージャー、そしてエンジニア、デザイナーなどが個々に、パーフェクトに仕事をする事で良いプロダクトは生み出されていきます。

プロダクトマネージャーの存在は日本でもここ4、5年の間に認知が広がっています。しかし反面で、これを曲解し、極端な分業制を進めてしまっている例も見られます。何を作り、何を目指すか、その最終決定を下すのがプロダクトマネージャーであったとしても、プロダクトの全員が事業の利益、ひいてはユーザーの体験をベストなものに仕上げるとい「プロダクト志向」の思いを共有することは重要であり、一人一人がプロダクトのオーナーであるという“オーナーシップ”を持つ、そのような組織が一番強い、と私は考えます(図8)。

プロダクトマネージャーはミニCEO

✓ プロダクトマネージャーは組織横断的なプロダクトチームを束ねる

✓ “ミニCEO”とも呼ばれる



単に分業制を進めるのではなく、

全員が「**プロダクト志向**」を高める



Copyright © 2020 Tably inc. All Rights Reserved

図8 強い組織を生むために必要となる「プロダクト志向」の共有

おわりに

エンジニアの中にはプログラミングが得意な方や実装が大好きという方がいます。とてもよく分かります。しかし、やはりその先にある「プロダクトを作る」というところを意識することで物作りの面白味は生まれます。コードを書く人であれば、そのコードの一行一行の向こう側にユーザーを意識し、どういったロジックで、どのような最適化を施していくかというようなことを考えながら書いていく。そのようにして書かれたコードの集合体であるプロダクトはとても魅力の高いものになると思います。

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。



ウェブサービス開発における 品質改善の取り組み2020

はじめに

経済産業省は平成30年9月に『DXレポート～ITシステム「2025年の崖」の克服とDXの本格的な展開～』（DXレポート）という報告書をまとめています。^{*1}

このレポートでは、DX（デジタルトランスフォーメーション）の重要性と共に、既存システムの複雑化・ブラックボックス化などに代表される技術的負債をそのまま放置しておくこと2025年以降、最大12兆円／年（現在の約3倍）の経済損失が生じる可能性がある。すなわち、2025年の崖という問題を提起しています。このような背景のもと、株式会社LIFULLにおいてもさまざまな品質改善の取り組みを行っています。本講演では、Webサービスの開発現場のQAマネージャーという立場から、弊社の品質に対する考え方や取り組みについてご紹介します。

中野 直樹氏

なかの なおき

株式会社LIFULL
テクノロジー本部 品質戦略部
品質改善推進ユニット
ユニット長



エンタープライズシステムの開発者、プロジェクトマネージャーなどを経験後、テスト技術者へ転向。

ネット証券のシステム開発部門のテストチームのリーダーを経験後、現在のLIFULLでソフトウェアテスト技術の展開など品質改善の仕組み作りに従事。

共著に「ソフトウェアテスト教科書 JSTQB Foundation 第4版」。
NPO法人ソフトウェアテスト技術振興協会（ASTER）理事/
JSTQB技術委員/JaSST Tokyo 実行委員など。



品質組織の移り変わり

はじめに、背景となるサービスや品質組織についてご紹介すると、弊社は不動産情報ポータルサイトを中核にさまざまな事業を行っています。Webサービス開発を中心に複数のプロジェクトが常時並行しており、月のリリース件数は200件を超えています。プロジェクトごとの人数は数名～数十名と幅広いのですが、ディレクター・デザイナー・エンジニアを基本構成とし、我々、品質に関する支援組織のメンバーも必要に応じて各プロジェクトに参加しています。それでは、品質組織について少しご説明します。

I. 品質組織の歩み

図1は、弊社の品質組織の移り変わりを表しています。品質組織が設置されたのは約10年前に遡ります。現場のニーズや我々の戦略

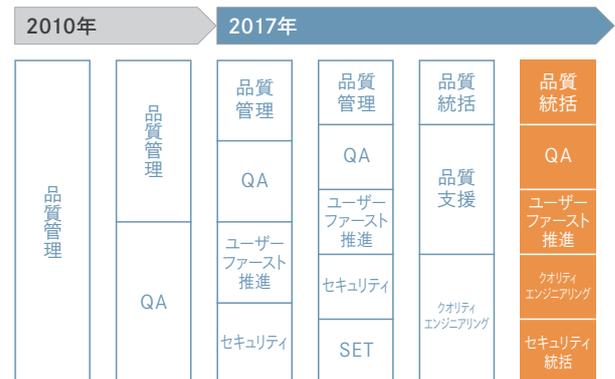


図1 品質組織の移り変わり

に合わせて組織は徐々に変化し、QAの独立をはじめ追加・分割・統合を経て、現在は一番右のオレンジ色の箱で示した五つのチームで構成されています。QAチームもこのうちの一つに属しています。本講演の内容は、2017年頃からの活動が中心となります。

II. 品質組織の役割

我々にはいくつかのミッションがあります。まずは、開発全体を通じてストレスの少ない効率的なプロセス・環境を作ること。また、QAやUX、セキュリティなどの専門領域からその開発チームにおける不足を補うためのサービスを提供すること。さらに、品質レベル向上の前提となる教育や啓蒙にも取り組んでいます。品質という文脈において、あらゆることを考え、企て、実践する中で、組織の構成も最適な形へと進化してきたということになります。

QAの変化

QAは、本番障害の発生率低減を目的に、日々プロジェクトや開発現場に寄り添って活動しています。具体的には、テストの最適化や支援、包括的なリスク低減のための情報確認、現場にフィットしたテスト関連ツールの開発・自動化などが挙げられます。開発プロジェクトにおけるテストは、開発チーム内で実施することを基本として、QAによる支援はテスト技術の不足を補う視点で決定しています。また、教育・啓蒙としての活動では、新卒者に対してテスト関連の研修を実施し、組織のテスト文化を育てるような取り組みも行っています。

* 1: https://www.meti.go.jp/shingikai/mono_info_service/digital_transformation/20180907_report.html

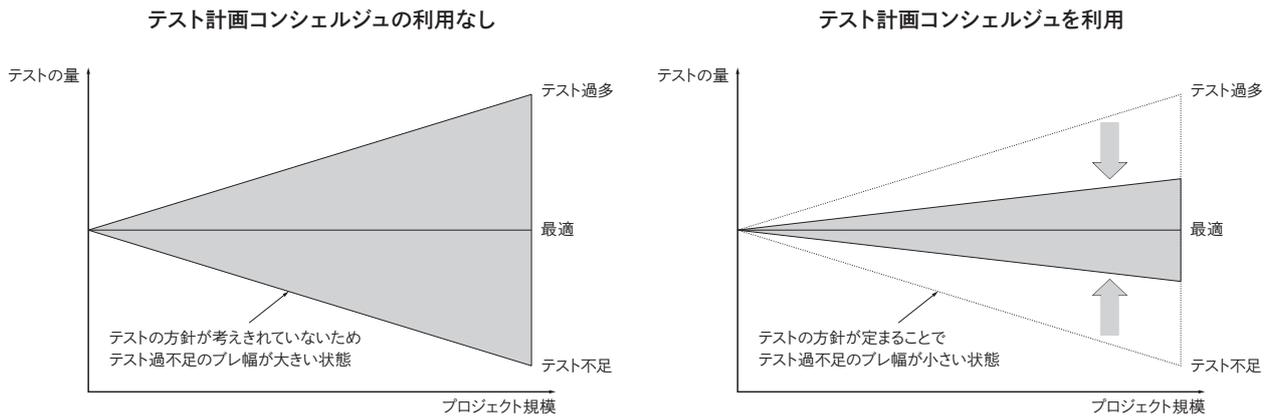


図2 テスト計画コンセルジュの役割と効果のイメージ

I テスト計画のタイムリーな作成と提供

QAの発足時から継続して行っているサービスの一つに「テスト計画コンセルジュ」があります。これは、開発現場に対してテスト計画の作成を支援もしくは代行するサービスです。

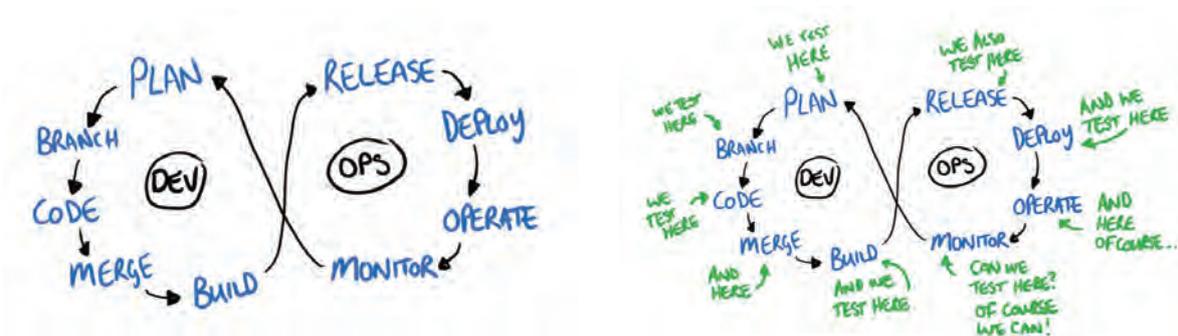
テスト計画は、開発チームでテストの方針の共通認識を持つためにあるものなので、素早くタイムリーに作成することが大切です。テスト計画コンセルジュでは、テスト計画の立案とチーム内での議論、合意形成を短時間のミーティングで実行します。

図2は、テスト計画コンセルジュの役割と効果のイメージです。左側のように、テスト方針が曖昧なまま進んでいくと、目的に相応しくないテストを作成したり、逆に不足したりということが起こり得ます。

テスト計画を素早く作成することで、この上下のブレ幅を抑制し、テストの効率化とプロダクトリスクの低減に貢献するのがテスト計画コンセルジュの狙いです。

II テストの可能性を追求

図3は、DevOpsのプロセスとテストの考え方を示したものです。ウォーターフォールの開発モデルでは、テストの工程としてシステムテストが重要視されがちですが、QAの目的である本番障害の発生率低減を考えた際には、あらゆるプロセスにテストの可能性があることを認識し、どこでテストをするべきかを常に思考し続ける必要があります。



出典: <https://danashby.co.uk/2016/10/19/continuous-testing-in-devops/>
 図3 DevOpsのプロセスとテストの在り方

Ⅲ 開発に対してQAが先回りして動く

図4は、プロジェクトに対するQAの活動を示したイメージになります。

- ①まず、同時に進行するすべてのプロジェクトで、開発に先んじて企画の段階からリスクを分析します。
- ②開発側からテスト計画の依頼が来た場合は、テスト計画コンシェルジュがサポートします。
- ③システムテストが終了し、間もなくリリースというタイミングで、QAがテストの不足などをチェックします。
- ④チェックした結果、必要に応じて探索的テストを実行してリスクをケアします。
- ⑤プロジェクトによっては、QAが企画から運用の段階まで伴走してサポートする場合があります。

このように、QAは常に開発と並行、もしくはそれに先んじて動き、成果物の品質確保とリスク低減に努めています。

Ⅳ DXへの対応と課題

最近のトピックスとして言及したいのが、DXへの対応です。DXとセットで語られるものとして「2025年の崖」問題がありますが、経済産業省のDXレポートではその対策として「DXシナリオ」も提示されて

いて、技術的負債を解消し人材や資金を新たなデジタル技術の活用に移す必要がある、と論じています。我々ユーザー企業においては、DXの変革と普段の業務を同時に進める必要があるわけで、QAが担う役割は大変難しいものになると感じています。

技術的負債の解消という文脈では、通常の開発業務とは異なる要望が出てきます。具体的な例で言えば、「データベースをAからBに置き換えたい。当然、サービスは止められません。どんなテストをしたら良いですか」、「モダンなアーキテクチャに刷新したい。影響範囲は不明ですが、もしかすると全部かもしれません。必要なテストは何ですか」、開発現場からは、こんな形のリクエストがQAに投げかけられてきます。

こうした要望に対して、どんなテストをすれば良いのか、いかに品質を担保すれば良いのか、我々QAは答えを持ち続けていなくてはならないのです。DXを推進する中で、現場では改善への意欲が日増しに高まっています。新しいことがどんどん始まり、粛々と状況は変化しています。QAはこの変化に食らいつき、テストを改善し続けなくてはなりません。「効率的なテストは何か?」「どのようにテストを実践すれば良いのか?」という問いに対して最適解を考え、現場が必要とするサポートができないと、まさに2025年の崖を転げ落ちることになります。このような課題に対して大きな責任を担うのが、テストやQAの分野なのだと感じています。

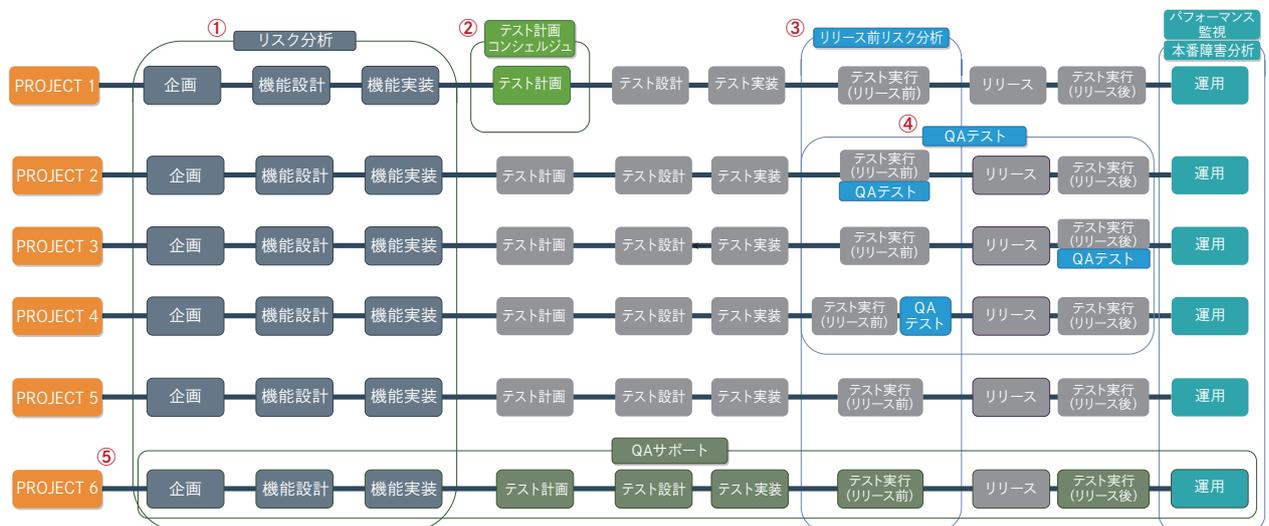


図4 プロジェクトに対するQAの活動を示したイメージ

V QAとしてどう成長し何を実現すべきか

QAは施策を検討する際、チームにフィットしたテスト戦略であるか、という観点を持つことも大事になります。「こうあるべき」を押し付けるのではなく、チームの構成や成熟度に合わせたテストを考え、実践できるようサポートすることを心掛けています。近年は、開発側もスクリプトテストだけでなく探索的テストも実施しているので、これらに対する評価やフィードバックなども求められてきています。

また、技術情報のアップデートも欠かせません。Webサービスの分野には新たな技術が次々と登場してきますが、こうした開発における優れたプラクティスを、理解した上でサポートする必要があります。その他にも、UI/UX・ユーザビリティ・アクセシビリティ・セキュリティ・パフォーマンスといった、専門分野のテストを支援するためにも、常に新たな知識のインプットが必要です。

SET誕生からQEへ

2018年、QAから派生する形でSET(Software Engineering in Test)チームが誕生し、後にセキュリティエンジニアリングチームと一体になってQE(Quality Engineer)というチームになりました。QEでは、テストや脆弱性診断の自動化、ツールの開発や導入などを実施しています。

I 自動テストツールの導入

テストの自動化については、10年前に品質組織が発足した時点から運用を開始しています。「Bucky」という自動テストツールを社内開発し、テスト運用などの効率化に活用しています。

このほか、「Gazo-san」という画像差分検出ツールも開発しています。これらは2つともOSS化していますので、興味があれば是非試してみてください(図5)。

重要なのは、コストパフォーマンスの良いテストが運用されている

環境を作ることです。プロジェクトで活用しやすいツールにするために、改修・改善を日々続けています。OSS化についても、さまざまな人が開発に携われるので、長い視点で見ればツールの改善スピードを上げるための一助になると考えています。



図5 「Bucky」と「Gazo-san」

II Developer eXperienceの改善

図6は、CTO協会が公開しているDX Criteriaという資料の引用ですが、DXの実現に必要な改善項目を示しています。この中のシステムに関する部分には、継続的インテグレーション(CI)や継続的デプロイ(CD)、セキュリティシフトレフト^{*2}などが含まれています。QA・テスト・開発のすべてを理解した上で、それをけん引していく人が必要なので、QEの役割はこれに該当します。

ただサービスを改善するだけでなく、開発プロセスの中でチームと協力し、効率化を促進していくということです。同じ資料の中に「2つのDX」、Digital TransformationとDeveloper eXperience(開発者体験)というものが出てきますが、後者を改善することも重要だと考えています。

現在、QEのけん引により、1. ソースコード品質の可視化、2. CI/CDのパイプラインの整備、3. 早い工程で手戻りの原因をなくしていくセキュリティシフトレフトという3つについて重点的に取り組んでおり、開発プロセスの最適化や自動化によるフィードバックループの高速化を目指しています。

* 2: セキュリティに関する問題がソフトウェア開発ライフサイクルの早い段階で対処されること(出典: “DevOps 技術: セキュリティのシフトレフト”, https://cloud.google.com/solutions/devops/devops-tech-shifting-left-on-security/?hl=ja#ways_to_improve_security_quality,2021/01/06参照)

| チーム | システム | データ駆動 | デザイン思考 | コーポレート |
|--------------|--------------|--------------|-------------------------|---------------|
| チーム構成と権限委譲 | バージョン管理 | 顧客接点のデジタル化 | ペルソナの設定 | スパン・オブ・コントロール |
| チームビルディング | ソースコードの明確さ | 事業活動データの収集 | 顧客体験 | 開発者環境投資 |
| 心理的安全性 | 継続的インテグレーション | データ蓄積・分析基盤 | ユーザーインタビュー | コミュニケーションツール |
| タスクマネジメント | 継続的デプロイ | データ分析 | 開発チームと協力して 取り組んでいく領域 | 人事制度・育成戦略 |
| 透明性ある目標管理 | API駆動開発 | データ分析 | 開発組織 | デジタル人材採用戦略 |
| 経験主義的な見積りと計画 | 疎結合アーキテクチャ | 機械学習プロジェクト管理 | プロトタイピング | モダンなITサービスの活用 |
| ふりかえり習慣 | システムモニタリング | マーケティング自動化 | ユーザビリティテスト | 経営のデジタルファースト |
| バリューストリーム最適化 | セキュリティシフトレフト | 自動的な意思決定 | プロダクトマネジメント | 攻めのセキュリティ |

出典: <https://cto-a.github.io/dxcriteria/>
 図6 CTO協会が公開しているDX Criteriaの一部

UXリサーチャー

UXリサーチャーは、ユーザーファースト推進チームの中でユーザビリティテストやコンセプト評価を行っており、我々はこれを検証型リサーチと呼んでいます。具体的には、ユーザーがサービスを開発者の意図通りに使えているか、スムーズにサービスを利用できているかを検証しています。

図7は、開発におけるユーザビリティテストの実行ポイントを示したものです。我々エンジニアは、ユーザビリティテストはシステムテストの後半に位置するものと思いがちですが、実際には開発プロセスのさまざまなフェーズにテストの可能性があります、専門家であるUXリサーチャーがこれを支援しています。

I ユーザー体験の最適化

図8(次ページ)は、UXピラミッドと呼ばれるもので、UXの品質評価を階層的に表現しています。UXリサーチャーはこうしたモデルを参考に、サービスを単に実用性の面だけでなく、心地良い体験になっているかどうかというところまで評価を行っています。

前述のDX Criteriaの中には、デザイン思考というカテゴリがあり(次ページ図9)、ここにはユーザビリティテストをはじめ、ユーザー体験に関わる要素が含まれています。これを見ると、正直なところエンジニアやQAにとってはあまり得意ではない分野が多いと感じられます。品質の向上という目的は同じでも、アプローチや考え方が全く違うので、専門知識を持つUXリサーチャーの支援によって、課題の発見や問題解決がよりスムーズに進められるようになります。

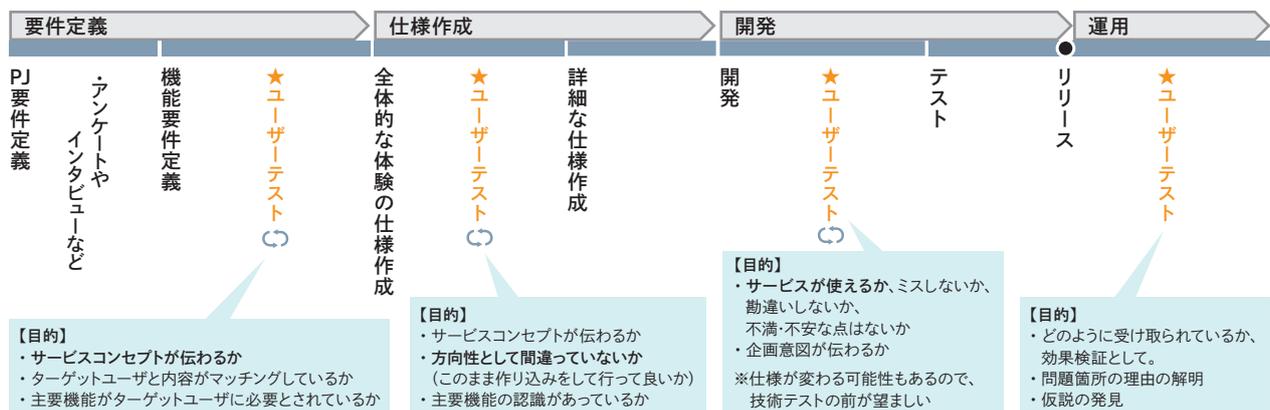
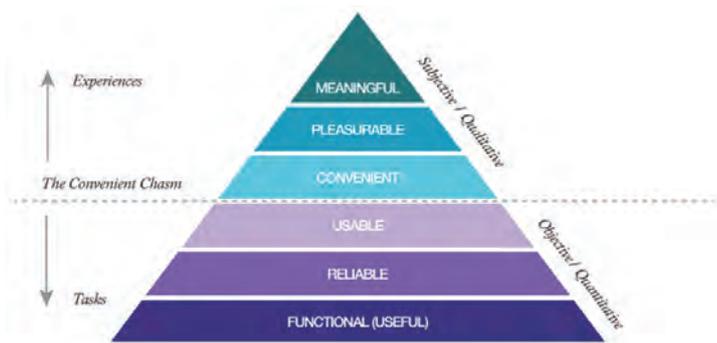


図7 開発におけるユーザビリティテストの実行ポイント



Experience – 心地良い体験(レベル4-6)

- Level 6: MEANINGFUL – 価値がある
- Level 5: PLEASURABLE – 楽しい・心地よい
- Level 4: CONVENIENT – 便利である

Task – 目的達成可能(レベル1-3)

- Level 3: USABLE – 使いやすい
- Level 2: RELIABLE – 信頼できる
- Level 1: FUNCTIONAL (USEFUL) – 機能的である

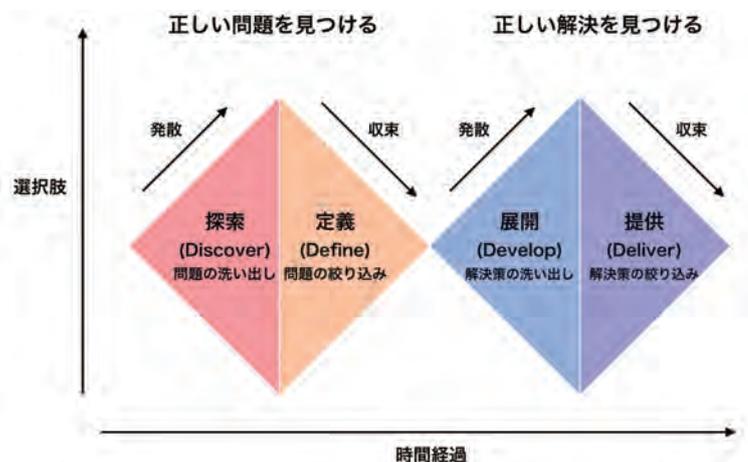
出典: <http://blog.btrax.com/jp/2017/08/21/ux-pyramid/>
 図8 UXピラミッド

| チーム | システム | データ駆動 | デザイン思考 | コーポレート |
|--------------|--------------|---------------------|-------------|---------------|
| チーム構成と権限委譲 | バージョン管理 | 顧客接点のデジタル化 | ペルソナの設定 | スパン・オブ・コントロール |
| チームビルディング | ソースコードの明確さ | 事業活動データの収集 | 顧客体験 | 開発者環境投資 |
| 心理的安全性 | 継続的インテグレーション | データ蓄積・分析基盤 | ユーザーインタビュー | コミュニケーションツール |
| タスクマネジメント | 継続的デプロイ | 開発チームと協力して取り組んでいく領域 | デザインシステムの管理 | 人事制度・育成戦略 |
| 透明性ある目標管理 | API駆動開発 | データサイエンス | デザイン組織 | デジタル人材採用戦略 |
| 経験主義的な見積りと計画 | 疎結合アーキテクチャ | 機械学習プロジェクト管理 | プロトタイピング | モダンなITサービスの活用 |
| ふりかえり習慣 | システムモニタリング | マーケティング自動化 | ユーザビリティテスト | 経営のデジタルファースト |
| バリューストリーム最適化 | セキュリティシフトレフト | 自動的な意思決定 | プロダクトマネジメント | 攻めのセキュリティ |

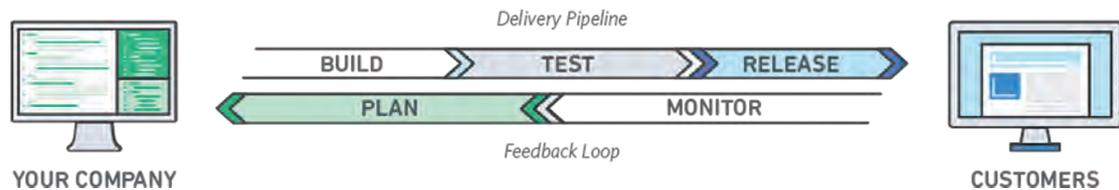
出典: <https://cto-a.github.io/dxcriteria/>
 図9 CTO協会が公開しているDX Criteriaの一部

II 探索的領域へのシフト

図10は、ダブルダイヤモンドと呼ばれる課題解決モデルで、UXの改善プロセスを示したものです。ユーザビリティテストは、主に右側の「サービスを作って提供する」という段階での評価になります。ただし、サービスを提供する前提には解決すべき課題というものが必ずあるはずで、それを発見することを我々は探索的リサーチと呼んでいます。まず正しく課題を見つけ、サービスがその課題を解決するものになっているかを検証する、つまり、左側へもっとシフトして、探索的リサーチの領域を広げていきたいと考えています。



出典: <https://uxdaystokyo.com/articles/glossary/doublediamond/>
 図10 ダブルダイヤモンドと呼ばれる課題解決モデル



出典: <https://aws.amazon.com/jp/devops/what-is-devops/>

図11 DevOpsの開発モデル

III DevOpsサイクルの精度向上

図11は、DevOpsの開発モデルを示したもので、中央にあるフィードバックループ^{*3}を高速かつ適切に回していくことが、品質向上のカギとなります。我々の組織で言うと、QEチームが開発プロセスの最適化・自動化によってこのループを高速に循環させていきます。一方、UXリサーチャーはユーザーファーストの観点からサービスを評価し、次の課題を適切に分析して、ループ1回の精度を上げていきます。この精度の向上が、結果的には、よりスピーディーなサービス改善にも有効で、そのためのリサーチを行っています。

おわりに

品質への取り組みは、各支援領域においてまだまだ広がっている状態です。直近4～5年は、「2025年の崖」問題を回避するためにさらなる品質改善の仕組みが必要になってくると思っています。技術的負債に足を取られず開発に集中するためには、より最適な技術の提供やプロセスの改善、新しい時代のニーズに合ったサービスデザインを行うための支援や、DXの実現を後押しする力が必要になってきます。

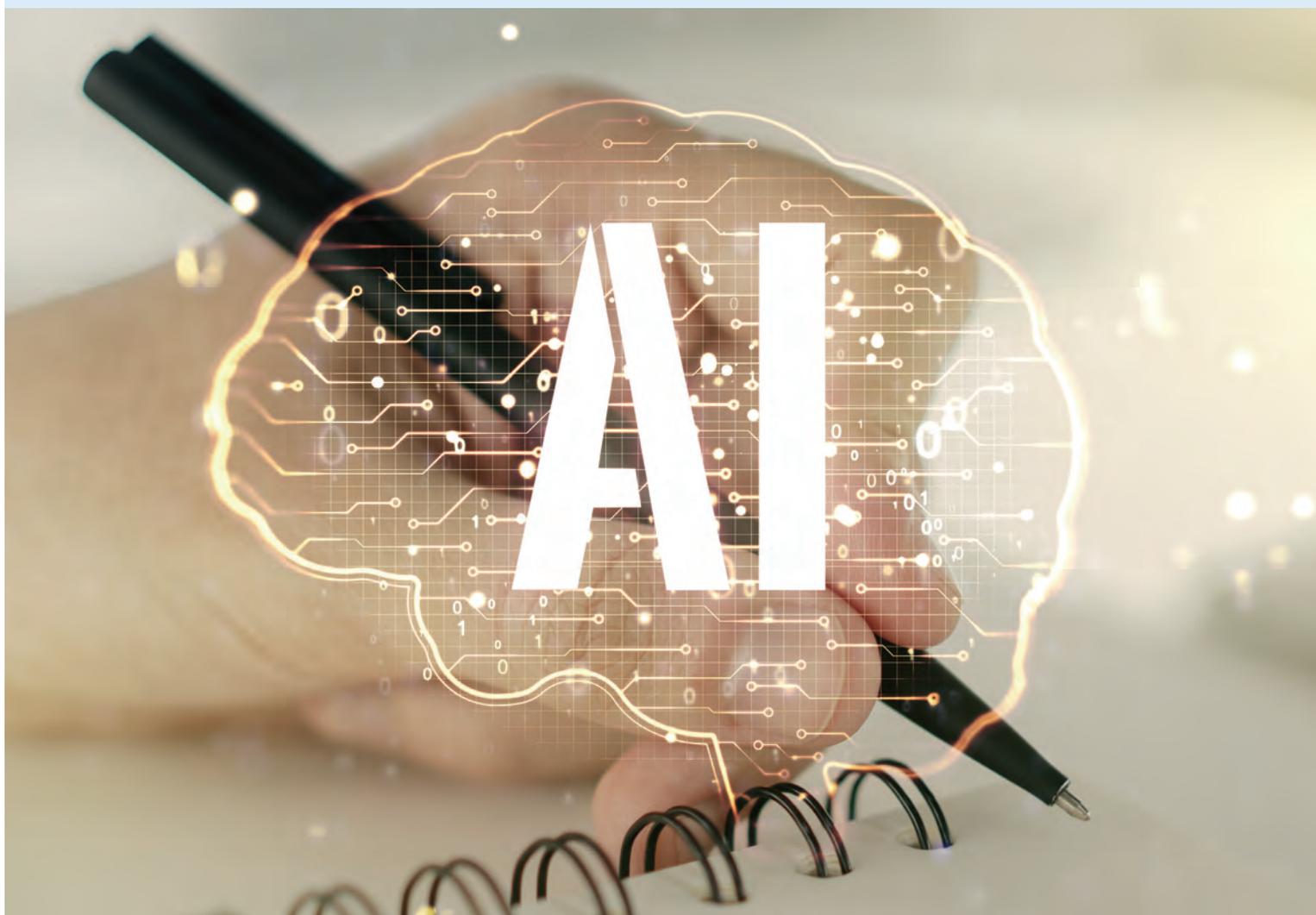
今後の展望としては、二つの軸が必要と考えています。一つは、開発チームのメンバーがそれぞれの仕事に対してきちんと納得感を持てる環境を作ること。自分たちのプロセスや組織を気に入っている人は生産性が高いということは、統計データでも示されています。

もう一つは、そうは言っても、理想の状態と現状には常にギャップがあることを理解し、それを埋めるための努力を続けていくことです。その際には、品質組織のメンバーがそれぞれの専門領域の視点を持ち寄り、継続的な改善を行うことが必要です。この二つの軸が、ハイパフォーマンスな開発プロセスと環境を整備する上で重要になると考えています。

*3: 開発チームからのプロダクトのデリバリーと、ユーザーからのフィードバックのサイクル。フィードバックループを高速に回すことで、仮説検証を用いて、プロダクトの改善を迅速に行うことが期待できる。

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。

テストエンジニアが テストベース視点で考察した 「Testing for人工知能」



佐々木 方規

ささき まさき

株式会社ベリサーブ 品質保証部 技術フェロー

1985年、金沢工業大学工学部卒業。同年、株式会社CSK(現SCSK株式会社)に入社、2001年ベリサーブに転籍し現在至る。開発組織から独立したテストチームに参加したことから、ソフトウェアテストの活動に35年以上従事している。多種のテスト領域を経験しテストチームのマネジメントを経て、現在は人材開発のための基礎研究やソフトウェアテストの普及振興を主としている。JSTQB技術委員会委員長、ODC分析研究会運営委員会委員長。

はじめに

昨今、人工知能は社会のさまざまなシーンに普及してきていますが、そのテスト技術に関しては未だ発展途上の段階にあります。人工知能の評価にはいくつかの分類が存在しますが、本稿では「Testing for 人工知能」、つまり「人工知能をテストする」というターゲットに絞り、テストエンジニアの立場から考察します(図1)。

Testingを主観とした

- ◆「Testing for 人工知能」本セミナーのターゲット
- ◆「人工知能 for Testing」

Quality Assurance を主観とした

- ◆「QA for 人工知能」
- ◆「人工知能 for QA」

※QAに関しては「AIプロダクト品質保証コンソーシアム(QA4AI:www.qa4ai.jp)」をご参照ください。

図1 人工知能の評価の分類

テストベースとして人工知能の理解

I 人工知能の歩み

現在、人工知能は第3次のブームを迎えていると言われていいます。第1次ブームは1950～60年代、コンピューターによる「推論」「探索」を中心としたもので、定理証明などの単純な用途が限界でした。第2次ブームは1980年代、主役は専門分野の情報を取り扱う「エキスパートシステム」で、特に自動翻訳などはかなり研究が進んだと認識しています。ただし、利用には人の手によるデータの正規化が必要でした。

第3次のブームにあたる現在の人工知能は、2000年代から普及したインターネット、さらにビッグデータの活用で利用シーンが大きく広がり、機械学習や深層学習の技術によって人を介さない学習も可能になりました。ただし、この状況をかつてのような一過性のブームに終わらせることなく社会に根付かせるためには、新たな課題も見えてきています。

II 人工知能(システム)をテストベースとして知るための整理分類

人工知能のテストを論ずるに当たり、まずはテストベースを考えてみましょう。人工知能システムの整理分類にはさまざまな捉え方がありますが、ここではその技術や機能を人間に置き換えてみます。

図2の通り、「認識」は目や耳などの五感に、「学習/推論」は脳にあたる部分です。ビッグデータを参照し、学習・推論するという一連の流れをここまでが司ります。もう一つは「制御」で、推論の結果を元にした操作・行動を指し、最近注目を集める自動運転などが該当します。

このモデルで考えた場合、人工知能システムはそれまで人が意思決定し行ってきた仕事を、人に代わって実行している、ということになります。ここで問題になるのが、その結果責任をどう考えるかということです。

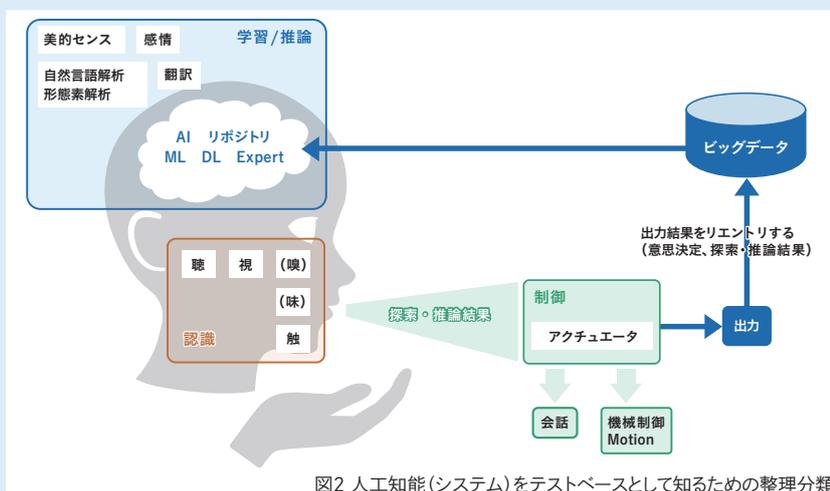


図2 人工知能(システム)をテストベースとして知るための整理分類

Ⅲ 社会と共存する人工知能とは

AI

現在、自動運転技術はかなり進歩を遂げ、自動車保険の料率に影響するまでになっています。人工知能が搭載された自動車は事故率が低い、という社会的認識があるわけです。ただ、もしも自動運転車が事故を起こした場合、人々はそれをどう捉えるでしょうか。人工知能は欠陥品であるというように大きな落胆を感じたとしたら、過去2回と同様に現在の第3次ブームも終焉を迎える結果になるかもしれません。

こう考えれば、人工知能は単なるプロダクトではなく、もはや社会インフラの一つとして認識すべきものであると言えるでしょう。だとすれば、そのテストも仕様への対応を確認するといった単純なものではなく、利用した結果が社会に与える影響も十分に考慮することが必要ではないでしょうか。

同時に、社会の側にも人工知能が導いた結果を冷静に受けとめる姿勢が必要かと思えます。これには人々の意識変化とともに、法的整備なども今後の課題になっていくと考えています。

人工知能のテストアプローチ

ここからは、具体的なテストアプローチについて言及します。テストには「期待値」というものが必要であり、これを「テストオラクル」として定義しています。これには、類似したリファレンスモデル、成果物の内容・仕様、利用者モデルや品質特性などが用いられます。ところが、人工知能の場合はこのテストオラクルの定義が非常に困難です。そこで今回は、テストオラクルを「責務モデル」という形で定義するアプローチをご紹介します。

※QA4AIの活動の中では、AIプロダクトの品質保証技術として「疑似オラクル」というものも紹介されていますので、こちらもご参照ください。
(<http://www.qa4ai.jp/QA4AI.Guideline.201905.pdf>)

I 人工知能のテストの難しさ

AI

図3は、一般的なテストで用いられるIPO(Input Process Output)モデルに人工知能を当てはめたものです。中央のプロセス部分に学習結果が反映されるためアウトプットが予想しにくく、事前に期待値を定義するのが困難であることを示しています。これを詳しく検証するため、プロセスの中身を展開してみます(図4)。人工知能プロダクトは、認識→学習→判断→推論を学習データに基づいて実行しますが、この段階でデータの不足や未正規化、さらに過学習・誤学習などが発生する場合があります。これによってアウトプットがかなり影響を受けるため、想定と異なる期待値が出る可能性を考慮する必要があります。

Ⅱ 人工知能の役割を「責務」として捉える

AI

ここで、人工知能の価値というものを考えてみましょう。まず、深層学習によって人を介さずに膨大なデータから推論ができること。人間が自分で集められるデータには限界があり、推論も簡単にはできません。さらに、プロダクトとしてカットオーバーしたら終わりではなく、使い込むほどに成長する。これによって、利用者も経験したことが無いような最適解の提供が可能になります。これが人工知能の大きな利用価値と言えるでしょう。

これらを製品の仕様として定義し、設計することは可能です。ただ、最適解のバリエーションが学習データに依存する以上、仕様としては存在しても結果がそれに紐付かないということが起こり得ます。

そこで少し見方を変えて、人工知能はどんな「責務」を果たさなければならぬかということを考えてみたいと思います。

なお、ここでいう「責務(果たすべき役割)」は本稿における固有の用語であり、オブジェクト指向設計における「責務」とは異なることをお断りしておきます。

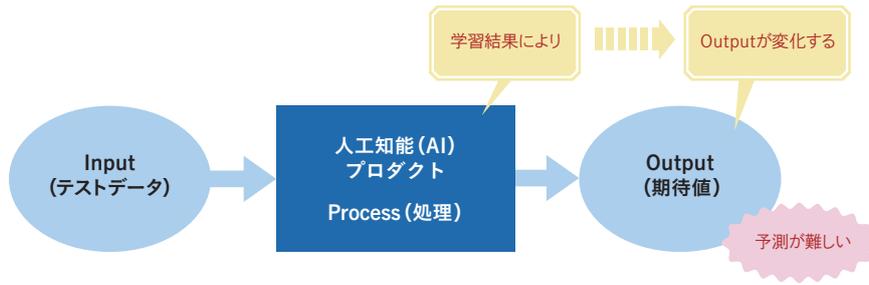
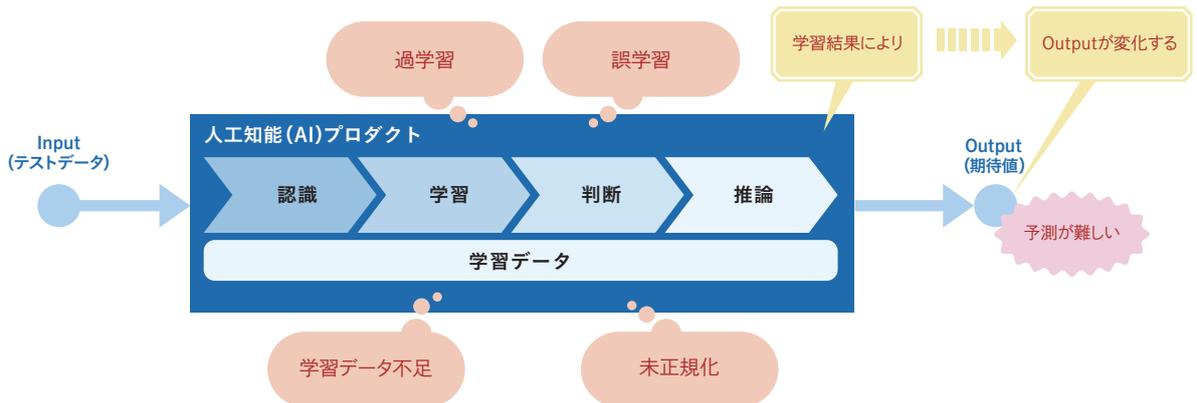


図3 IPO Input Process Output を利用して考える



※期待値の変化については、「メタモルフィックテストング」などもご参照ください。

図4 IPO Input Process Output を利用して考える

Ⅲ 人工知能を搭載した製品の例 AL

身近なものとして、お掃除ロボットを例に取りましょう(図5)。人工知能を搭載していないお掃除ロボットは、決められたアルゴリズムで常に同じコースを掃除します。これに対し、人工知能を搭載した製品は、家の間取りや家具の配置を学習し、ルートを選択しながら自律的に掃除を行うという責務を持ちます。ただし、人工知能を搭載するなら、利用者に時間的制約を与えない、つまり人が監視していない留守の間にも掃除をしてくれる、こういう利用価値も責務として必要でしょう。一方で、留守中に家の中を動き回るとすれば、安全性はどうなのかという課題もあります。故障や障害だけでなく、やや極端ですが火災を引き起こすようなことは無いのか、こういったリスク回避も責務となります。

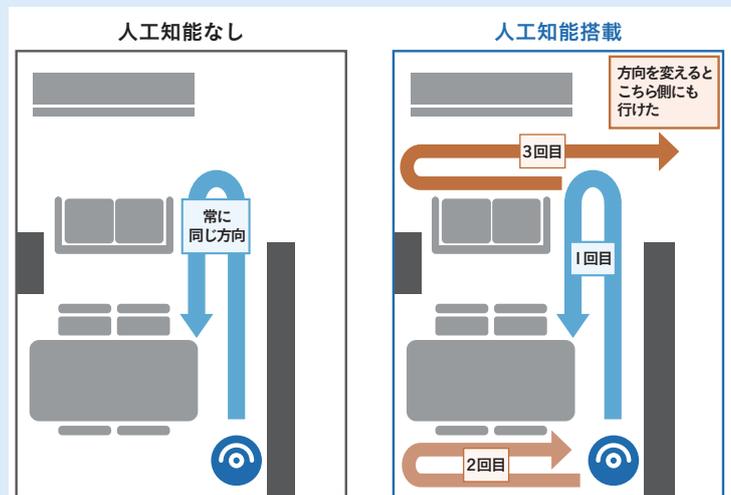


図5 人工知能を利用した製品の例

IV 人工知能の責務を開発する AI

要求元とテスト担当者の間では、テストゴールについて事前の合意が必要ですが、対象が人工知能の場合、仕様書だけのテストベースでテストゴールを設定するには情報が十分とは言えません。そこで提案したいのが、人工知能の責務を開発し、学習と推論のライフサイクルによって結果を繰り返し確認していくという手法です(図6)。

お掃除ロボットなら「位置情報を取得して適切なルートで掃除をする」、これを最初の責務と定義し、テストを設計します。次に学習と推論を実施して適切に実行されるかを確認、これが無事終わったら、例えば「掃除の方法を変える」という次の責務を定義します。一気にゴールを目指すのではなく、順序を踏んだサイクルを繰り返し、その過程でもしギャップが発生したら推論データを検証して何が原因だったのかを分析します。このとき、学習前の状態に戻せるという仕組みは考慮しておく必要があるでしょう。

アルゴリズムで作られたプログラムは決められた通りにしか動きませんが、人工知能の場合は人間と同様に間違える可能性があり、どの程度の正答率なら実用性があるかが解答となります。これを前提に、レジリエンス(回復力・復元力・柔軟性)を考慮しながら徐々に正答率を高めていくというのが、責務を確認するプロセスとなります。

V 開発プロセスと学習プロセスの統合 AI

さらに、この学習プロセスをシステム開発プロセスと統合することも提案したいと思います(図7)。システム開発では、最初に決めた仕様通りに製品が作られているかを確認しながら作業を進めますが、人工知能の場合、たとえ仕様に適合したとしても、学習内容によっては最終的なシステムとしては欠陥品となる可能性があります。一気通貫型の開発で、最終の運用・評価の段階から機械学習や深層学習でAIの利用価値を考えていては、もう手戻りができなくなってしまいます。システムテストの時点から学習プロセスを早めに組み込み、人工知能の責務定義を繰り返し開発して、運用・評価の時点ではベースとなる責務データと学習モデルが完結していることが重要であると考えます。

- 人工知能の責務定義(P)
推論結果の定義、過ち水準の決定
- 責務をテスト設計(D)
学習データの分析、期待値、ゆらぎの測定方法の検討
- 学習 推論の実施(L)
段階的な深層学習の実施と推論結果
- テスト実行と責務のGap分析(A)
責務(定義された推論結果)が果たされていることを繰り返して確認する。

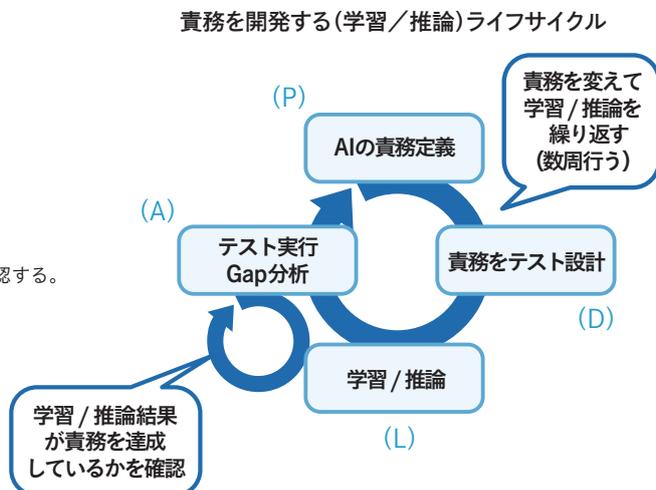


図6 人工知能の責務を用いてテストベースを明らかにする

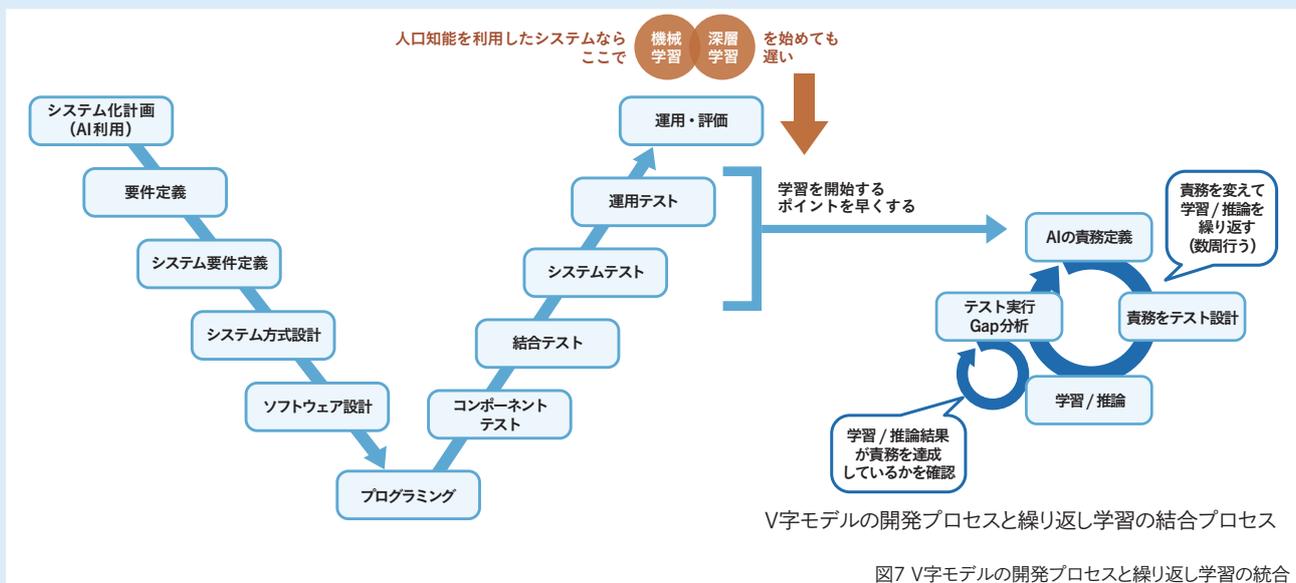


図7 V字モデルの開発プロセスと繰り返し学習の統合

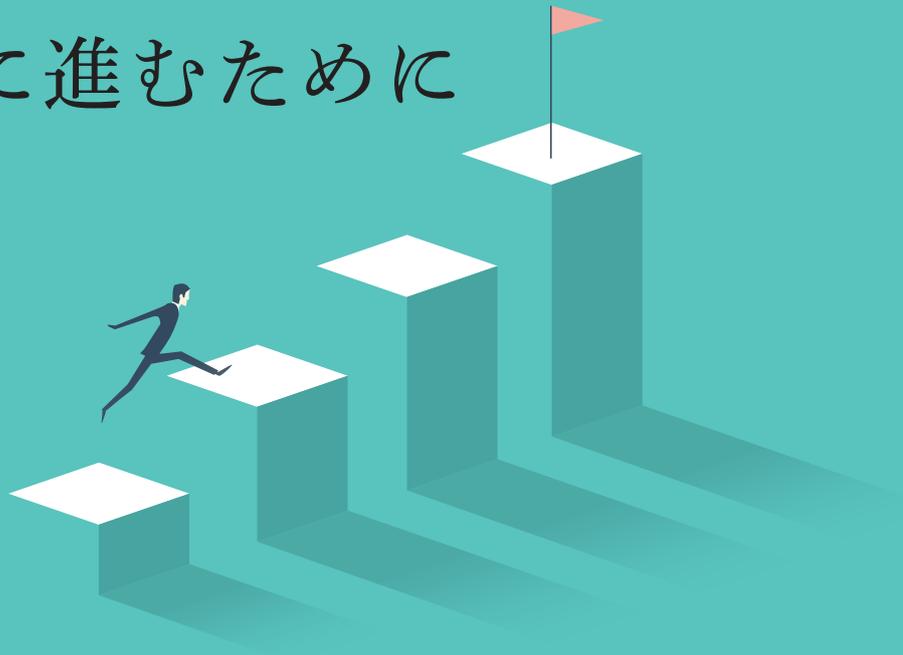
おわりに

従来、プロダクトは利用者と供給者だけの閉鎖的な環境下にありましたが、IoTやビッグデータを利用する人工知能には、見知らぬ他者とのつながりも発生します。これはまさに社会システムと同じです。人工知能のテストはプロダクト単体の完成度だけでなく、社会システムにどのように影響するか、さらにはプロダクトの社会的な「責務」ということまで考慮される必要があるでしょう。

現在、人口知能の技術はまだ発展途上にあります。このような技術を踏まえた社会環境の成長に合わせ、テストモデルも柔軟に変化させていくということが重要かと思います。本講演で紹介したモデルは実際にはまだ考察段階で、実証実験というのも今後必要になると思いますが、これからのテストを考える上でも参考にしていただければ幸いです。

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

テスト自動化現場の 「いま」と、 次のステップに進むために



伊藤 由貴
いとう よしき

株式会社ベリサーブ 研究企画開発部 自動テスト推進課 課長

2012年、東北大学大学院情報科学研究科情報基礎科学専攻修士課程修了。同年、株式会社ベリサーブに入社。テスト自動化ツールの開発を経験し、その後様々な現場でのテスト自動化導入支援を担当。2019年よりテスト自動化エバンジェリストとして社内外でのテスト自動化技術の普及活動を行っている。JaSST Tohoku実行委員。著書『ソフトウェアテスト技法練習帳～知識を経験に変える40問～』

※【用語定義について】本講演において「テスト自動化」とは「テスト実行の自動化」を指し、また「テスト実行の自動化」におけるテストレベルは主に「システムテストのレベル」を想定しています。

はじめに

市場のニーズや取り巻く環境が急激に変化する現在、開発サイクルの高速化と素早い価値提供を促進するテスト自動化は、ますますその重要性を増しています。本講演では、これまでさまざまなお客様に対してテスト自動化の技術支援を行ってきた私自身の経験や公開されている情報を元に、テスト自動化の現状から効果的なテスト自動化を行うために有効な知見や考え方、さらに世の中で行われているテスト自動化の実例などを紹介していきたいと思います。

テスト自動化の「いま」

❶ テスト自動化が「あたりまえ」の時代に

かつてお客様から相談を受けた中で多かったのは「テスト自動化を導入したいが、どうやって始めたらいいかわからない」というものでした。また、頑張って自分たちでテスト自動化を進めたものの「運用できなくなってしまった」という声も少なくありませんでした。

10年ほど前は、テスト自動化の成熟度が十分ではなく、まだテスト自動化に及んでいなかったり、たとえ、テスト自動化を導入したと

しても十分な効果を得られていなかったり、という企業が数多く存在する状況だったように思います。

ところが昨今では、お客様から受ける相談内容も変化してきました。例えば「すでにテスト自動化をしているが、もっと効率的に運用できないか」、あるいは「社内の一部チームでテスト自動化に成功したのでこれを社内で横展開したい」といった、導入後の試行錯誤に関するものが増えているのです。

このような相談の変化からも、テスト自動化はすでに世の中では「あたりまえ」になってきているということを実感します。逆に言えば、テスト自動化はもはや先進的な取り組みではないということです。テスト自動化が先進的だった時代なら、開発のアジリティ(顧客ニーズや新たなビジネス環境に素早く適応しうる柔軟性)を高めることで他社に先んじることができたかもしれませんが、現在では「いかに他社より開発のアジリティを高めるか?」という部分に勝負は移っており、テスト自動化をやらなければ他社に遅れを取ることになりかねません。

テスト自動化は「いま」、このような状況にあります。

効果的なテスト自動化のために

Ⅰ 一般公開されるテスト自動化手法

現在、多くの企業ではテスト自動化導入後の試行錯誤へと課題が移ってきています。同時に、テスト自動化に成功し、またはそこから確実な効果を得られている組織が増えてきたことによって、効率的な運用のための知見や、テスト自動化できる範囲をさらに広げるための知見が公開されることも多くなりました(図1)。

Ⅱ ISTQBテスト自動化エンジニアシラバス

まず、ご紹介したいのが「ISTQBテスト自動化エンジニアシラバス^{*1}」になります。ISTQB(国際ソフトウェアテスト資格認定委員会)というソフトウェアテストの国際認定試験を行っている団体が、そのテスト範囲を定めたものをシラバスとして作成しています。「テスト自動化エンジニア」のシラバスはまだ英語版しかありませんが、

■ 試験・ナレッジ

- ISTQB Certified Tester Advanced Level Syllabus Test Automation Engineer

■ 考え方

- テスト自動化の8原則
- テストピラミッド/アイスクリームコーン

■ 組織毎の知見やツール

- クックパッド社の **SPLIT** (2018)
- Google社の **Test Size** (2010)
- LIFULL社の **Bucky & Gazo san** (2019)

■ AI 自動テストツールの活用

■ SET ロールの設置

図1 テスト自動化の知見の一例

「テスト自動化の準備について」や、「テスト自動化システムのアーキテクチャ」など、テスト自動化をうまく始めて、なるべく失敗を避けて運用するための知見が豊富に掲載されています。

Ⅲ 企業による知見やツールの公開

企業の例でいうと、LIFULL社では、ブラウザのテストを自動化する「Bucky」、画像差分を検出する「Gazo-san」という自社開発した自動テストツールを運用するとともにOSSとして一般公開しています(図2)。LIFULL社ではこれらの内製ツールを公開することでフィードバックを受け、さらに使いやすいツールへと進化させることを目指しています。現在ではこのように自らの知見やツールを公開して、ブラッシュアップするという取り組みをされている企業が増えています。



図2 LIFULL社のBuckyとGazo-san

* 1: <https://www.istqb.org/downloads/category/48-advanced-level-test-automation-engineer-documents.html>

Ⅳ AI自動テストツールの活用

テスト自動化をサポートするツールも充実してきており、特にAIを活用したツールが増えています。例えば、一般的にテスト対象の画面構成が変わった場合には自動テストも人間が修正しなければなりません。AI自動テストツールであれば、テスト対象の変更に対応してAIがテストシナリオ自体を自動で修復してくれます(オートヒーリングまたはセルフヒーリング機能)。保守にかかわる人間の手間を削減すると同時に運用コストの削減も可能になり、今後は日本国内でも導入が増えていくものと思われます(図3)。



図3 AI自動テストツールの例
(左:mabl社「mabl」 右:TRIDENT社「Magic Pod」)

Ⅴ SETロール

また、SET(Software Engineer in Test)ロールを設置している企業も現在増えています。SETロールというのは開発環境やQA環境の整備や運用、テスト自動化を含め、開発やテストの生産性を向上させるためにさまざまな取り組みを担当するロールとなります。ネット上で「SET」を検索すると、さまざまな企業で「SETの募集」や自分たちのSETロールに関する情報が公開されており、人気が高まっています。

このように、現在ではテスト自動化の導入・運用がかなり進み成果を上げている企業・組織が増えており、テスト自動化を成功させるための知見や有用な情報もかなり公開されている状況にあります。テスト自動化において、いまだ「道半ば」にある企業・組織の方は、ぜひこうした知見から自らの参考になるものを最大限に活かして、次のステップに進んでいただきたいと思います。

世の中で行われているテスト自動化

テスト自動化は現在さまざまな領域で行われていますが、実際にはそれぞれの領域ごとの難しさや環境面での違いに直面することになります。ここからはもう少し踏み込んで、それぞれのテスト自動化の課題や具体的な取り組み例を紹介してまいります。

Ⅰ 組込みのテスト自動化

まず一般的に難しいと言われているのが「組込みのテスト自動化」です。ポイントはいくつかありますが、その一つに「テスト対象の操作の難しさ」があります。テスト対象を操作するために何かしら物理的なデバイスの操作が必要になる、あるいは外部からデバイスを操作するためのAPIが存在せず直接操作するしかないという場面にしばしば直面します。

こういった場合にはインターフェースの実装が工夫のポイントになります。外部から自動操作するための専用APIを実装し、コマンドラインやプログラミング言語から扱えるようにする手法がまず考えられます(図4)。

また、組込みのテスト自動化では、組込みの機器自体が気温や光、位置情報などの実環境に沿って動いている場合、テストを行う環境によって期待結果が異なる場合があります。こうした場合、自動で結果を判定することは困難になります。このような問題に対応する方法の一つとして、テスト対象には手を入れず、カメラや光センサーを用いて状態を観測する仕組みを構築する方法が考えられます(図4)。例えば、暗室の中にテスト対象とウェブカメラを入れ、そこで撮影した画像を元に、その後パソコン側でさまざまな画像認識処理を行うことで期待通りの動作がなされているかを判断します。こうした試みもすでにさまざまな企業で行われています。

Ⅱ ウェブアプリケーションのテスト自動化

ウェブアプリケーションのテスト自動化についてはブラウザという土台が共通していることからノウハウの横展開がしやすく、OSSツールやノウハウが非常に充実していることが特徴です。またAIテストツールも非常に多く、テスト自動化の導入や運用がしやすい状況にあります。

ウェブアプリケーションの自動テストにおいては、CIツールやチャットツールとの連携もよく行われています。例えば、CIツール

■I/Fを実装する

- 外部から自動操作をするために専用のAPIを用意し、
コマンドラインや既存のプログラミング言語から扱えるようにする、等

■音声・光・画面などを観測する仕組みを構築する

- テスト対象には手を入れず、カメラや光センサーなどを用いて状態を観測する
 - ◆LEDが機器の状態に応じた点灯すること
 - ◆メーターの表示が想定通りであること
 - ◆音声再生されていること

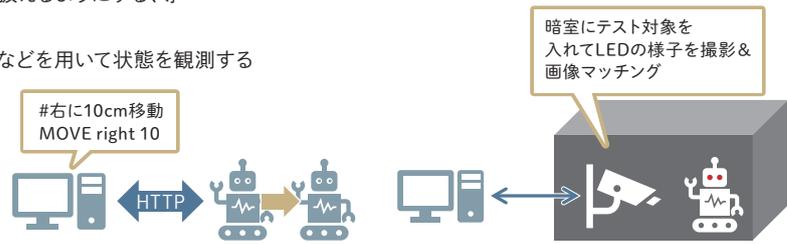


図4 組み込みシステムの自動テストの工夫例

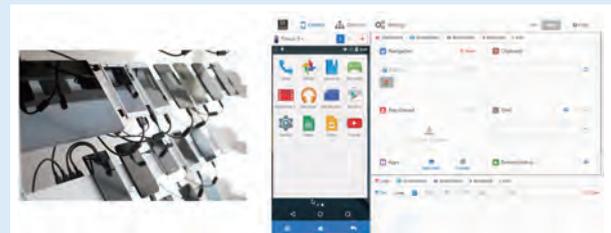
を用いて夜間にテストを実行したり、夜間に実行したテスト結果を Slack や Teams などのチャットツールで通知して、朝出勤してきた開発者がその結果を確認したりするなどの実用例もあります。Slack や Teams などから簡単なコマンドを打つことで、プログラミングに詳しくないデザイナーやテスターの方がテスト実行する「ChatOps」を実現している企業も増えています。

また、ウェブアプリケーションにおいて、さまざまなブラウザで同じように動くことを確認したい場合には、コンテナ技術を使ってクラウド上に複数の実行環境を作り、そこでテストを並列に実行させて結果を集計(実行が終わったら環境ごと削除)するという手法も採用されています。

III モバイルアプリケーションのテスト自動化

モバイルアプリケーションにおいてもテスト自動化をサポートするツールやノウハウは充実しています。Appium や AirTest といった OSS や無償ツール、さらにはクラウドサービスとしての AI のテストツールも登場し効率化に寄与しています。

一方、問題点としては OS と OS バージョン、さらには端末のバリエーションの多さから、膨大な組み合わせパターンが発生し、さらに画面サイズや OS バージョンごとの UI 差異によってテスト自動化に困難を来すことがしばしばあります。この点を解消するモバイル特有の方法としてデバイスファームの活用があります。デバイスファームでは、図5の左側写真のようにサーバーなどに実際の多数のスマートフォンを接続し、右側画面のようなアプリケーションを用意して外からネットワーク経由でアクセスし、実機での自動テストを実行することができます。



左写真: https://github.com/openstf/stf/blob/master/doc/shelf_closeup_790x.jpg
右イメージ: https://github.com/openstf/stf/blob/master/doc/7s_usage.gif

図5 デバイスファームの例

おわりに

本日は、テスト自動化を次のステップに進めるためにというテーマでさまざまな知見や事例をお話しました。テスト自動化を「まだやっていない」という組織では、テスト自動化の専門のロールを設定することが理想です。しかし、現実的には難しいことも多いため、まずは公開されている成功事例を元に、テスト担当者がテスト自動化を始めてみる。また、テスト自動化をすでに「やっている」組織では、一般的なベストプラクティスを活用しつつ、これを自社組織に合った形にテラリング(開発標準の手直し)を行う、小さく始めて少しずつ適用範囲を広げていくという進め方が良いかと思います。テスト自動化のノウハウを一般公開し、そこからフィードバックを得て改善のサイクルに結びつけることも有効です。このような方法により、「やっている」から「効果を得られている」状態にステップアップさせていくことができるのではないかと考えます。

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。

はじめに

社会の急激な変化はニーズの多様化をもたらし、結果としてソフトウェア開発の大規模化、複雑化は加速する一方で。必然的にテスト管理業務は、増大するテストケースや短納期化、開発プロセスの変化にも対応しなければならず、日々テストチームメンバーの頭を悩ませています。お客様からは次のようなご相談を多く頂戴しておりました。

- ・表計算ソフトでの煩わしいマージ作業や集計作業をどうにかしたい
- ・テスト実行結果や不具合の傾向をリアルタイムで確認したい
- ・複数拠点、かつ大規模プロジェクトのテスト資産を一元管理したい

こうした課題をクリアし、テストエンジニアが分析業務に専念できるようにするためのテスト管理ツールが「QualityForward」です。本稿では、本ツールを導入した現場より、その効果について2つの事例をご紹介します。



Case1

15万を超えるテストケースもサクサク集計、スピード感のある品質改善へ

車種や仕向け地の組み合わせにより、管理対象のテストケースが肥大化

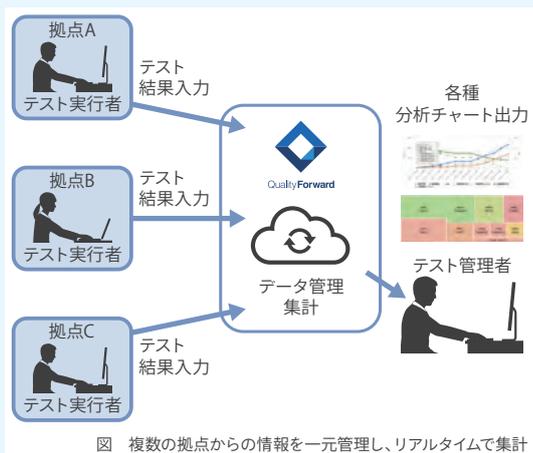
はじめにご紹介するのは車載メーカー様の事例です。現場担当者様のお話によると、次のような課題を抱えていらっしゃいました。「複数の開発拠点でそれぞれテストが行われていて、各拠点から送られてくるメールでの報告を集計してテスト管理を実施しています。また、テストケースは車種と仕向け地の組み合わせ単位で作成する必要がありますが、ケース数がどんどん増えていきます。管理表を開くだけでもかなりの時間がかかってしまいます」

この課題をお伺いした際、テストケースの総数は15万件を超えていたそうです。テストの全体像を把握するため

には、各拠点からのデータを統合する必要がありました。この作業は多くの時間を要し、テスト管理者の負担は非常に大きくなっていました。

複数の拠点からの情報が一元管理され、テスト実行結果が迅速、かつ正確に把握できるように

QualityForwardは開発当初から大規模プロジェクトを想定して設計・製造されており、膨大なテストケースを処理する場合にも優れたパフォーマンスを発揮します。複数拠点からの同時実行が可能のため、導入後はさまざまな分析チャートをリアルタイムで確認できるようになりました(図)。結果として、「無駄なテスト管理工数を大幅に減らし、本来の業務である品質改善の取り組みのスピードが改善されました」との声を頂いています。



Case2

開発チーム、テストチームの連携がスピーディーに。アジャイル開発におけるテスト管理の課題を解消

テスト管理ツールの盲点、テスト実行中のテストケース修正ができずに作業待ちが発生

次にご紹介するのは、ITサービス系企業様の事例です。このお客様のチームはアジャイルでの開発を行っています。アジャイルでは一つの課題に対し、開発、テストに関わる全てのメンバーで取り組むことが理想とされていますが、このチームがそれまで利用していたテスト管理ツールの仕様では、テスターによるテスト実行中はテスト設計メンバーによるテストケース修正ができず、無駄な作業待ちが発生していました。「テスト管理ツールが開発プロセスに合わず、ボトルネックを生んでしまっている」というお客様からのご相談がありQualityForwardの導入をご提案しました。

無駄な待ち時間の解消と共に、BTSツール連携の強化によりアジャイルのプロセスがスムーズに進むように

本ツールは、テストに関わる全てのメンバーが同時に作業を行うことができるよう考慮されたデータ構造となっているため、テスト実行中でもテストケースの追加・修正が可能となりました。

お問い合わせ先 株式会社ベリサーブ

TEL:03-6629-8540(本社代表) 052-325-5010(中部支社) 06-6223-6110(西日本支社) URL:<https://www.veriserve.co.jp/>

また、開発チームが利用しているBTSツールへの自動連携に関して、タイトルだけのチケット自動起票から、テスト手順や条件などの反映もできるようになった結果、チーム間連携のスピードが大幅に改善されました。お客様からは「待ち時間の削減と共に、情報連携の強化により、スピード感や一体感といったアジャイルの強みを活かすことができるようになりました」との評価を頂いています。

おわりに

本稿では、QualityForwardによって現場の課題を解消した2つの事例についてご紹介しました。しかし、これらはツール活用事例のほんの一部に過ぎません。現在(2021年2月15日時点)、製品を実際に操作して現場に適合するかを確認していただける、30日間の完全版無償トライアルキャンペーンを実施していますので、この機会に、ぜひシンプルで快適なテスト管理を体験ください。

QualityForward 開発チームではさまざまなツール活用方法や、APIを利用したサンプルスクリプトなどをブログにてご紹介しています。本稿ではご紹介しきれなかった情報も数多く掲載していますので、ぜひ併せてご覧ください。本稿がテスト管理業務改善のきっかけとなりましたら幸いです。

30日間無料トライアル実施中

QualityForward 製品ページ



お問い合わせ、トライアルのお申し込みはこちら

QualityForward 公式ブログ



最新のアップデート情報、ツール活用のTipsなど随時更新中

品質を創造する企業



Veriserve Navigation 2021年2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

*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。