

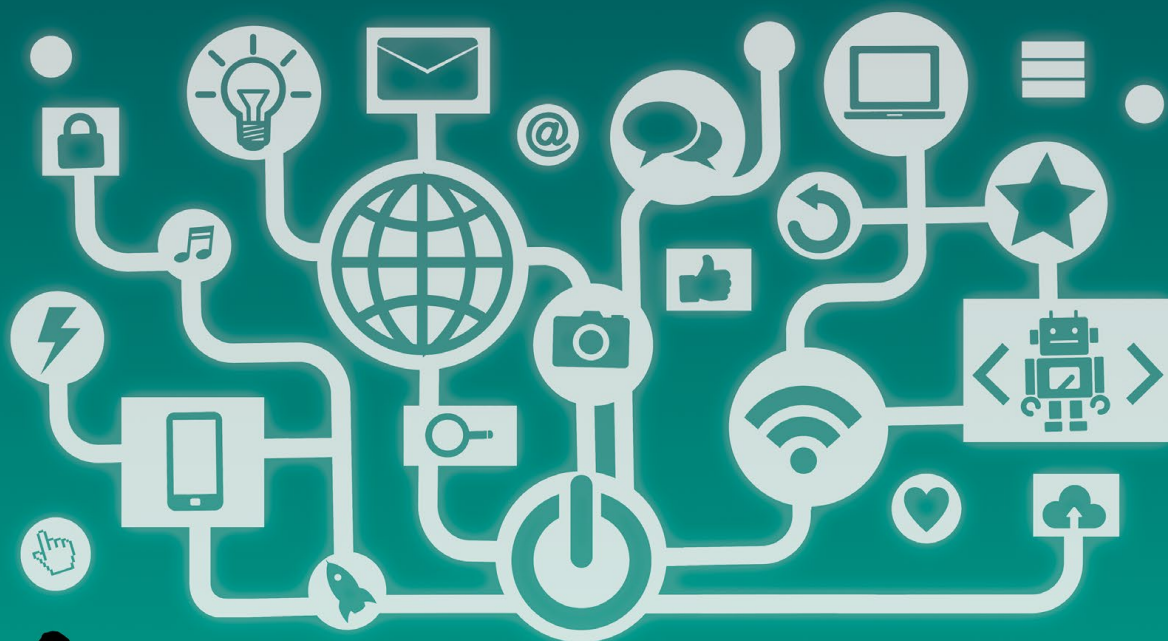
ソリューション技術通信 ベリサーブナビゲーションでは、さまざまな分野で検証に携わるベリサーブが「今とこれから」を皆様にお届けします。

VERISERVE NAVIGATION

2019
January
Vol.16

ソフトウェアと品質を考える

ベリサーブ アカデミック イニシアティブ 2018 特集



UI自動テストツールとAI

～AIを使った自動テストの「今」と「未来」～

伊藤 望 氏

自律運営要員・チーム・組織を 創るためのアプローチ

～テストプロセス改善の取り組みを事例に

安達 賢二 氏



伊藤 望氏
いとう のぞみ

日本Seleniumユーザー
コミュニティ代表
株式会社TRIDENT
代表取締役社長

大手パッケージベンダーにて社内自動テストツールを開発した後、2012年独立し、テスト自動化を支援する株式会社TRIDENTを設立。現在は、機械学習を活用した自動テストサービス「Magic Pod」の開発・運営に取り組む。また、日本Seleniumユーザーコミュニティの運営や、講演・執筆活動を通じ、テスト自動化の普及にも努めている。
著作に「システムテスト自動化標準ガイド」「Selenium実践入門」。

UI自動テストツールとAI

～AIを使った自動テストの「今」と「未来」～

はじめに

今世の中のさまざまな産業が、AI(人工知能)によって大きく変わりつつあり、その波は自動テストツールの世界にも押し寄せて来ています。本稿では、筆者が「ベリサーブアカデミックイニシアティブ 2018」で行った講演内容を基に、この新しいAIの技術が自動テストツールの分野にどのような影響を与えるかについて紹介します。

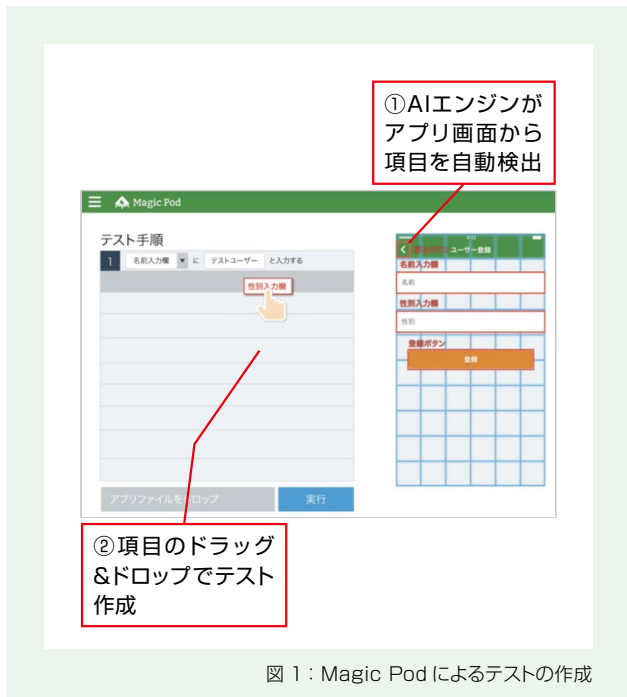
1. AI自動テストサービス「Magic Pod」

最初に、AI技術をUI自動テストに適用した具体的な事例として、筆者も開発に関わっている「Magic Pod」を取り上げ、その仕組みについて解説します。

Magic Podとは

Magic Pod (<https://magic-pod.com>)はディープラーニングなどの各種AI技術を利用した商用自動テストサービスです。現在はiOS/Androidのモバイルネイティブアプリテストに対応しています。Magic Podの特徴は、AIエンジンを用いて、アプリ画面の画像から画面項目を「名前入力欄」「登録ボタン」などの分かりやすい日本語名称で検出できることです。

具体的には、Magic Podの画面上にアプリ画像をアップロードすると、そこからAIエンジンによって画面項目が検出されます(図1の①)。後は、ユーザーが検出された画面項目をテストケース作成エリアにドラッグ&ドロップしていくことで、テスト手順を簡単に作成することができます(図1の②)。なお、<https://www.youtube.com/watch?v=f6SO9pZANas>ではこの一連の流れを動画で分かりやすく解説しています。



Magic Podの仕組み

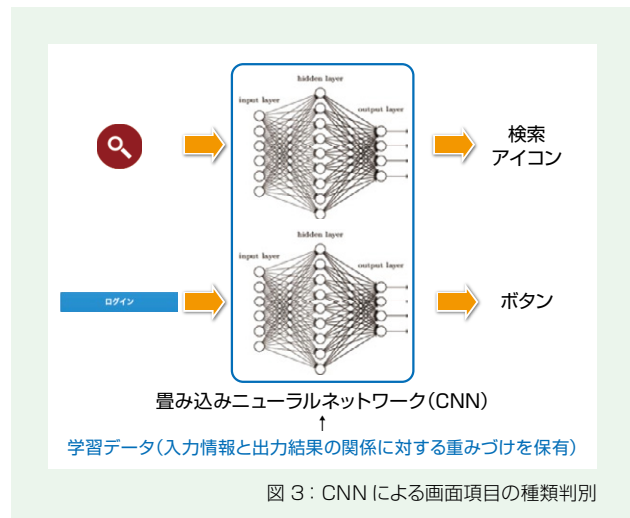
テストを作成する具体的な仕組みは実際にはこうです。Magic Podでアプリ画像をアップロードする際には「Magic Pod Desktop」という専用のアプリを利用しますが、これを使うと実際には「アプリ画面の画像」と「画面の内部システム情報」の2つがMagic Podのサイト上にアップロードされ(図2の①)、さらAIエンジンがこの画像を解析して画面項目を検出します(図2の②)。解析が終わればユーザーは画像から



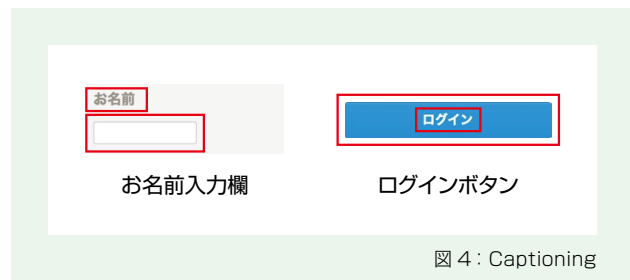
画面項目を選べるようになり、「ボタン」なら「タップ」、「入力欄」なら「入力」のように、選んだ項目に応じたコマンドが生成されます(図2の③)。実はこの時、コマンドの種類と、画面項目の位置に対応する内部システム情報を基に、テストスクリプトが内部では自動生成されています(図2の④)。このテストスクリプトは、モバイル向け自動テストツールAppium(<https://appium.io>)のスクリプトであり、テスト実行時にはAppiumでこのスクリプトを実行します(図2の⑤)。

Magic Podの画像解析AIエンジンの仕組み

続いて、アプリ画像から画面項目を検出するAIエンジンの仕組みについて説明します。エンジンの核となる技術は大きく2つあります。1つは、画像の見た目から「ボタン」「入力欄」など画面項目の種類を判別する技術です。この部分には「畳み込みニューラルネットワーク」(Convolutional Neural Network、CNN)と呼ばれるディープラーニングの技術を利用しており、読み込んだ画像の見た目と学習済みデータ(一例として、数多くのボタンの画像を「ボタン」として学習させたもの)を基に、「ボタン」「入力欄」などの画面項目の種類、さらには「検索アイコン」「設定アイコン」などのアイコンの種類を判別することができます。(図3)



もう1つは、図4のように、テキストと画面項目の位置関係を基に「お名前入力欄」「ログインボタン」のようなラベル付けを行う技術です(Magic Podの内部ではこれを「Captioning」と呼んでいます)。この部分はディープラーニングではない別の機械学習技術を利用しています。



この2つを組み合わせた解析エンジンの全体像は以下のようになります。

- 色や模様を基準に、画面の領域を分割 (図5)
- 分割した各領域の画面項目は何かを、CNNを使って判別
- OCR(光学文字認識)技術を使って、画像からテキストを検出
- Captioningを使って、画面項目とテキストを結合
(例)名前+入力欄、登録+ボタン

AI技術の進歩による機能の向上

AIの技術は日々進歩しています。続いて、この進歩を取り入れてMagic PodのAIエンジンをさらに改善しようとしている試みを紹介します。まず、現在のMagic PodのAIエンジンの問題点として、画像解析の最初に画面の領域分割を実施する点があります。この領域の分割は画面項目の種類などを考慮せず色や模様を基準に行われるため、例えばグラデーションのかかった画面では、図6のようにうまく分割を行うことができません。また、この領域分割の処理が低速であることも問題です。

そこで現在Magic Podでは、単純なCNNに代わり、ここ数年でメジャーになってきたSingle Shot MultiBox Detector(SSD)と呼ばれる新しいディープラーニングの手法を用いて、この問題の改善に取り組んでいます。単純なCNNでは、画像に「何が」写っているかしか判別できませんが、SSDを使えば、図7のように、画像の「どの位置に」「何が」写っているかを検出できます。SSDをうまく活用すれば、図8のように、複雑な画面からもうまく画面項目の位置、さらに種類も同時に検出できます。

加えて、低速な領域分割の処理を廃止することもできます。SSDによる検出処理も通常のPCでは時間がかかりますが、ディープラーニングの処理によく使われるGPU(Graphics Processing Unit)^{*1}を搭載したマシンであれば、この処理は画像1枚あたり0.2~0.3秒程度で行えます。

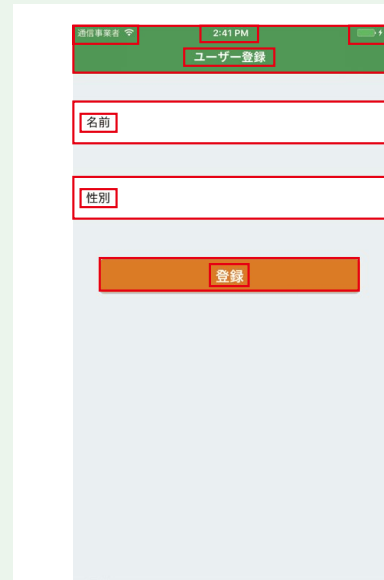


図5：画面領域の分割

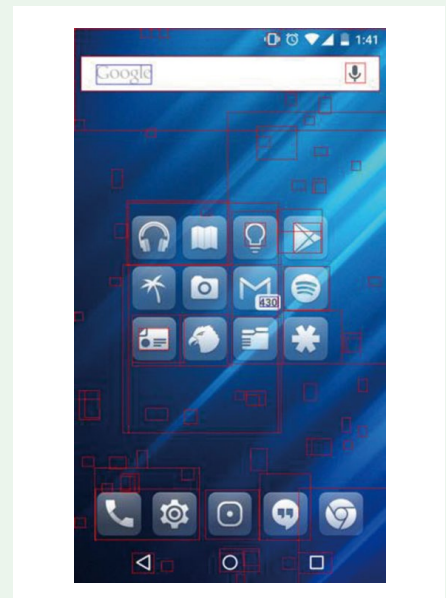


図6：グラデーションのかかった画面の領域分割結果

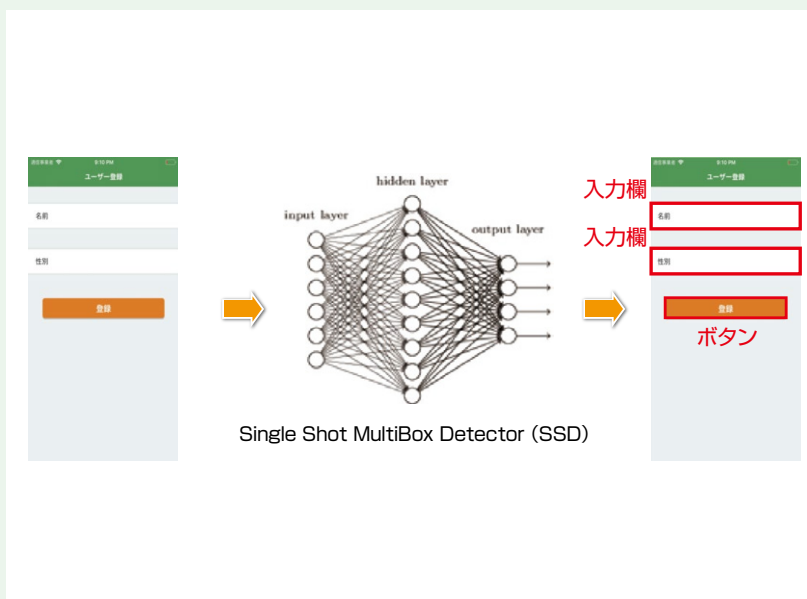


図7：SSDによる画面項目の位置・種類の検出



図8：SSDによる画面項目の検出結果

将来的には、テスト実行時にSSDによる画像認識をリアルタイムで行うことで、Appiumaでは扱いが難しいUnity*2などのフレームワークを使ったアプリケーションや、組み込み製品などの自動化もMagic Podで対応できるようになるでしょう。

その他のAI自動テストツール・サービス

Magic Pod以外にも、海外ではAIを使ったさまざまな自動テストツール・サービスが登場しています。図9では、どの部分にAI技術を使っているかで、代表的なツール・サービスを分類しました。筆者の所感としては、「スクリプト自動生成」や「バグ自動修正」については、単純なものはAIで自動化できると思いますが人間のよう複雑なタスクを行うのはかなりハードルが高いと感じています。一方、「スクリプト自動修復」「画面要素検出」「モンキーテスト」については、AI技術による大きな改善が十分に可能だと思います。ここではそれぞれについて詳しくは取り上げないので、詳細は各ツール・サービスのWebサイトをご確認ください。

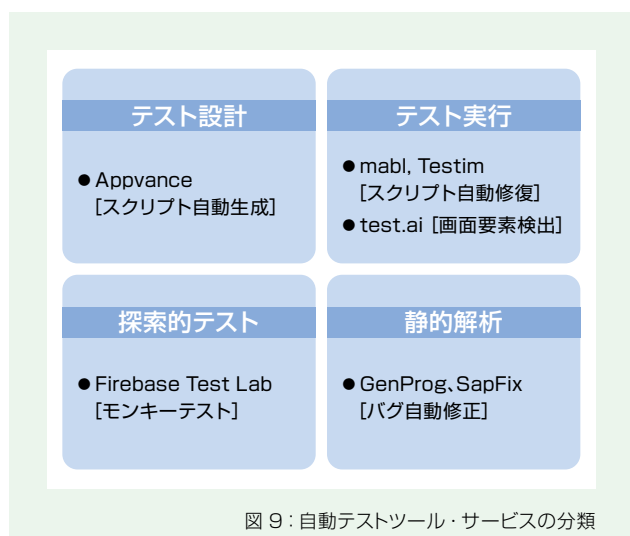


図9：自動テストツール・サービスの分類

参考サイト

- "Magic Pod". <https://magic-pod.com/>
- "Magic Podコンセプト動画". <https://www.youtube.com/watch?v=f6SO9pZANas>
- "Firebase Test Lab". <https://firebase.google.com/docs/test-lab/>
- "Appvance.ai". <https://www.appvance.ai/>
- "mabl". <https://www.mabl.com/>
- "Testim". <https://www.testim.io/>
- "test.ai". <https://test.ai/>
- "GenProg". <https://squareslab.github.io/genprog-code/>
- "SapFix". <https://code.fb.com/developer-tools/finding-and-fixing-software-bugs-automatically-with-sapfix-and-sapienz/>

*1: GPU (Graphics Processing Unit): 3Dグラフィックスなどの画像処理に特化した演算装置。定形かつ大量の計算処理を並列的に行うことが可能。

*2: Unity: Unity Technologies社(アメリカ)が開発した世界シェア1位のゲーム開発用ツール

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

AI自動テストの未来

これまで見てきたようなAI自動テストツールがどんどん進歩すれば、いずれAIがテストを全て自動化し、品質保証担当者の仕事は無くなってしまおうのでしょうか? 筆者は、当面はそうはならないと考えています。例えば、以下に述べるようなスキルをAIで実現することは、結局のところAIでソフトウェアを設計・開発するのと同じくらい難しく、現在のAIのレベルから考えるとまだまだ困難であると感じます。

- ユーザーや業務を理解し、良いテストを設計するスキル
- 仕様バグや業務ロジックの不備を見つけるスキル

AIを使った自動テストツールは今後もどんどん進化していきますが、それらが人間を全て置き換えるのではなく、人間がテストを設計し、さまざまなAI自動テストツールでそれらを効率よく実行する、という形が、最も現実的な未来だと筆者は考えています。

おわりに

人間の仕事がAIに全て奪われることはないとはいえ、単純な作業は今後確実にAIで代替されていきます。日々の業務においても、AIに代替されにくいテスト設計などのスキルを磨いていくことが、これからの時代に重要になるのではないかと思います。



安達 賢二氏
あだち けんじ

株式会社HBA
共創推進グループ
エグゼクティブエキスパート

HBA入社以来、システム保守・運用・開発業務、プロジェクトマネージャなどを経験後、部門→全社品質保証担当、全社品質・セキュリティ・環境管理統括責任者、全社生産革新活動技術コーディネータなどを担当。

2012年社内イントレプレナー第一号事業者として品質向上支援事業を立ち上げ。

現在、自律運営チーム構築・変革メソッドSaPIDをベースに、関係者と一緒に価値あるコトを創る共創ファシリテータとして活動中。

自律運営要員・チーム・組織を創るためのアプローチ

～テストプロセス改善の取り組みを事例に

はじめに

このたびご縁があり、アカデミックイニシアティブ2018で講演させていただきました。本稿ではその講演内容を抜粋してご紹介させていただきます。

講演の中でご紹介した事例は、以下の問題意識から当初ソフトウェアテストシンポジウム2018東京で発表したものです。

実務の現場が組織的なプロセスモデルの導入やAgile、DevOps、テスト自動化など、毎年変化する施策に混乱し、疲弊しているケースが多いのではないか。

一つの例として、プロセスモデル活用が普及しているように見えて、形式対応や、やらせる／やらされる図になっているなど、使いこなせていないケースが多いのではないか。

このような問題意識の解消を目指し、「テストプロセスモデル適用」に対するアプローチを試みました。本稿では、第1章で「テストプロセス改善の取り組み事例」を、第2章で「事例で伝えたかったこと」をお伝えしたいと思います。

1. テストプロセス改善の取り組み事例

1-1 きっかけ

この事例は、自らのチームにTPI NEXTを活用して現場改善を進めようとしたテストチームのリーダー中山さん(仮名)が自らアセスメントを実施したところから始まりました。

1-2 TPI NEXTとは?

TPI NEXTとは、オランダのSogeti社が開発したテストプロセス改善モデルです。SR:利害関係者との関係、TM:テスト管理、TP:テスト業務の専門性の3グループに16のキーエリア、146のプラクティス(実践事項)があり、それぞれに初期・コントロール・効率化・最適化の4レベルが定義されています。当モデルをベースにアセスメント(評価)を実施。プロセスの成熟度や能力を判定し、改善目標やビジネスゴール達成に寄与するキーエリアを中心に改善を進めることができます。(図1.1)

1-3 セルフアセスメントの実施とその結果

現状のテスト運営に問題意識を持っていた中山さんは、何かヒントになることはないかとTPI NEXTの日本語書籍を手に取り、Webサイトに掲載されているアセスメントシートにてセルフアセスメントを実施してみたところ、どうしてよいのか迷う状態となりました。

懸念事項(1):アセスメント結果が正しいのか確信が持てない。
わかるようなわからない質問が多く、実態を反映した結果になっているのが不安。

懸念事項(2):多くのNGがあり、どこから手を付けたらよいかわからない。
TPI NEXTが推奨しているクラスタA*¹から順に行うのがよいのか?
中山さんが普段から気にしているテスト環境の改善から着手したらよいのか?

この状態は、プロセスモデルへの専門性を備えたアセッサーなどの有識者が実施しないセルフアセスメントにおいてよく起き得ることで。地域のテストコミュニティの会合で一緒になった中山さんからこの状況をうかがい、一緒に状況打開に向けて取り組んでみよう、ということになりました。

1-4 プロセスモデルを活用した改善におけるリスクと打開策

プロセスモデルを使う／使わないにかかわらず、現在の状況を的確に把握することなくして有効な改善はできません。今回は日本語書籍を一読した状態の中山さんが1人でアセスメントを実施しています。よって、以下のリスクが発生しやすくなります。

リスク(1):アセスメント結果が実態を反映していない可能性がある。
リスク(2):チームメンバーにとって改善が他人事になりやすい。

この2つのリスクを軽減、解消するために、メンバーの声を直接収集して現状を把握すること、できるだけチームに余計な負荷をかけないようにするために、毎週チームで実施している「ふりかえり」の結果を活用することを提案しました。

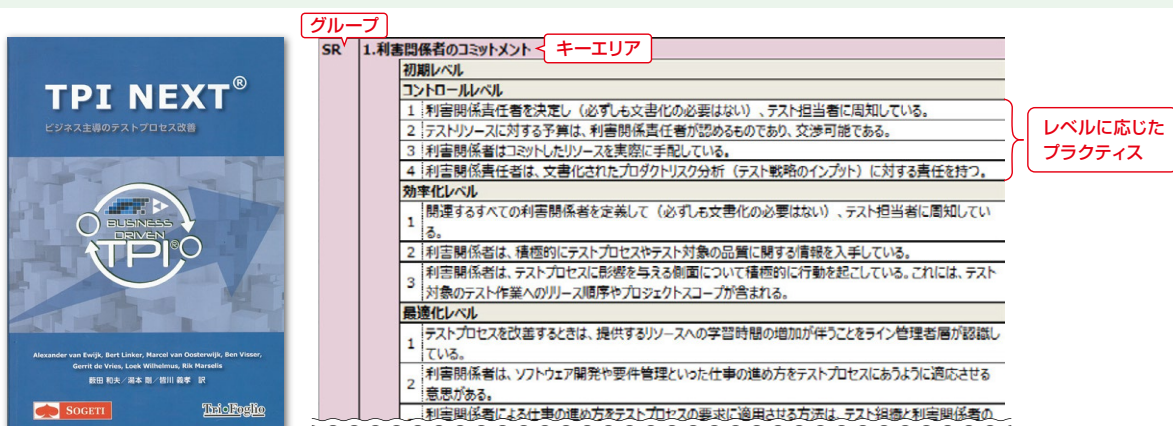


図1.1 TPI NEXT 日本語書籍とチェック項目の一部

*1: 小さな改善ステップを段階的に踏むために、1つの改善ステップを構成する複数のキーエリアのチェックポイントをグループ化したもの。各クラスタはアルファベット1文字で識別され、Aから順に改善の優先順位を意図している。

1-5 メンバーの問題意識の収集と分類

まずは直近数週間分のふりかえり結果のテキスト情報を列挙し、分類して、プロセスモデルのどの領域に問題意識があるのかを確認してみました。(図1.2、図1.3)

その結果、チームメンバーは「テストウェア管理」と「テストケース設計」に多くの問題意識を持っていることがわかりました。もし、これらが解決できればメンバーの困り事の多くが解消され、仕事がしやすくなるはずですが。

1-6 状況を掘り下げるヒアリングの実施

次に、現状をより詳細に把握するための追加ヒアリングを行います。メンバーが発した現状の問題点や困り事の表現(「ふりかえり」における表現)は、感情に引っ張られて歪曲していたり、具体的なイメージが湧かない内容だったり、特定のメンバーにしかわからない表現が混在しています。作業の状態や発生している事象が、できるだけリアルに関係者全員で把握できるレベルになるまで質問を繰り返して認識を共有します。(図1.4)

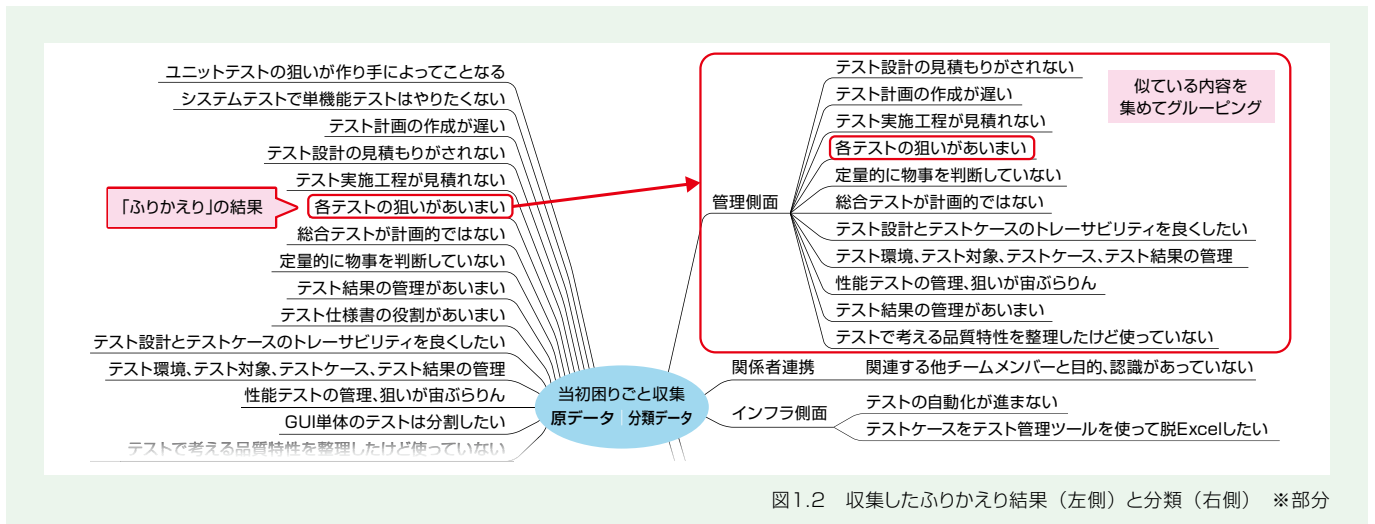


図1.2 収集したふりかえり結果（左側）と分類（右側） ※部分

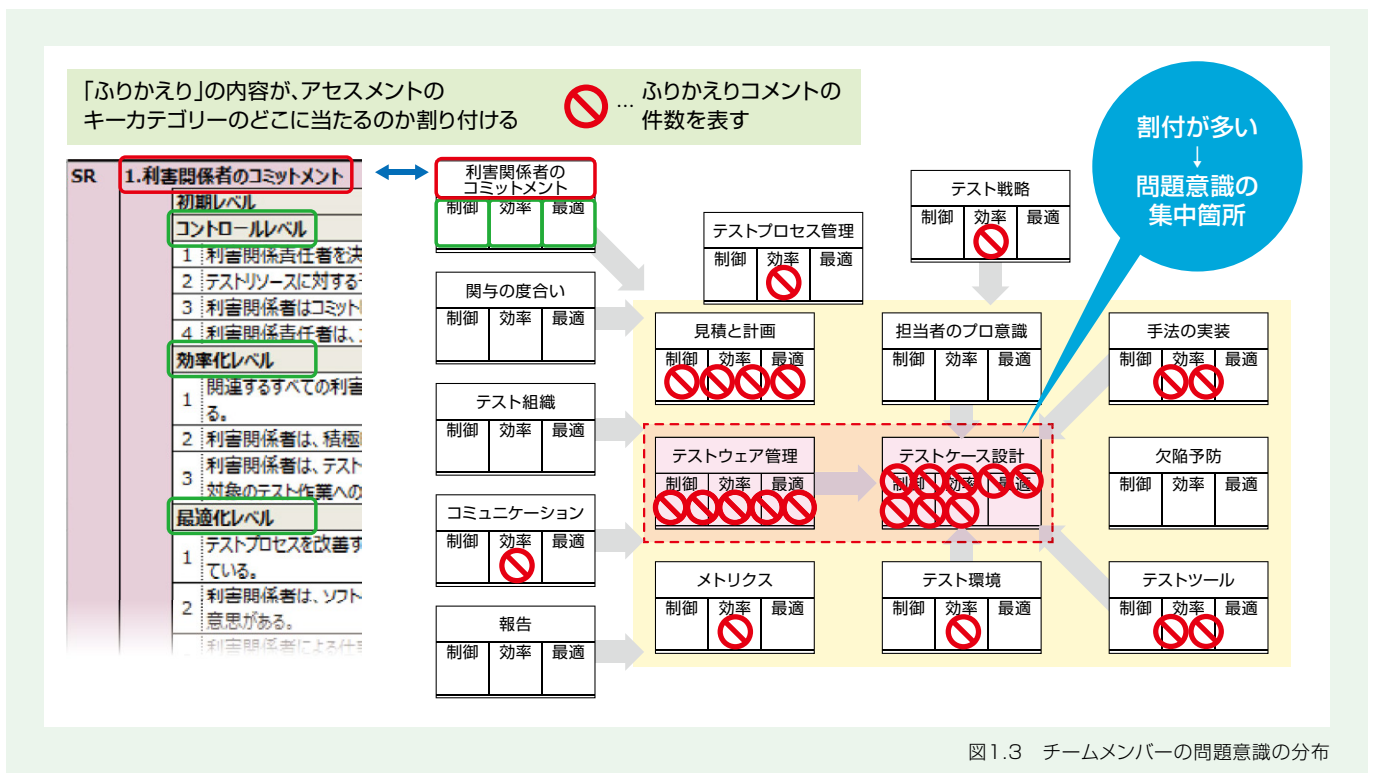


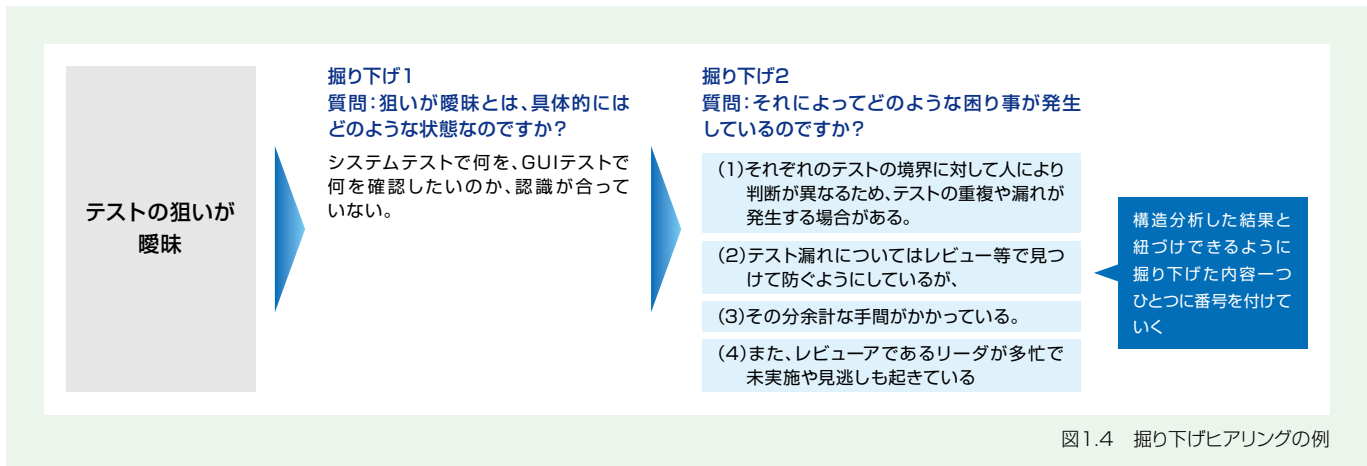
図1.3 チームメンバーの問題意識の分布

1-7 ヒアリング結果の構造分析

掘り下げヒアリングの結果を、「ふりかえり」の際と同様に分類・整理して、それらを構造的に分析します。

構造的に分析するとは、それぞれの要素の因果関係を

分析し、関係性のある要素を矢印でつないでいくことを指します。さらに、構造図の要素の意味をまとめ、それぞれに名前付けします。(図1.5)これらは、テストチームが実施している1サイクル(テスト作業の開始~終了+その結果まで)の問題構造の詳細と概要を表しています。ここまで来たところで

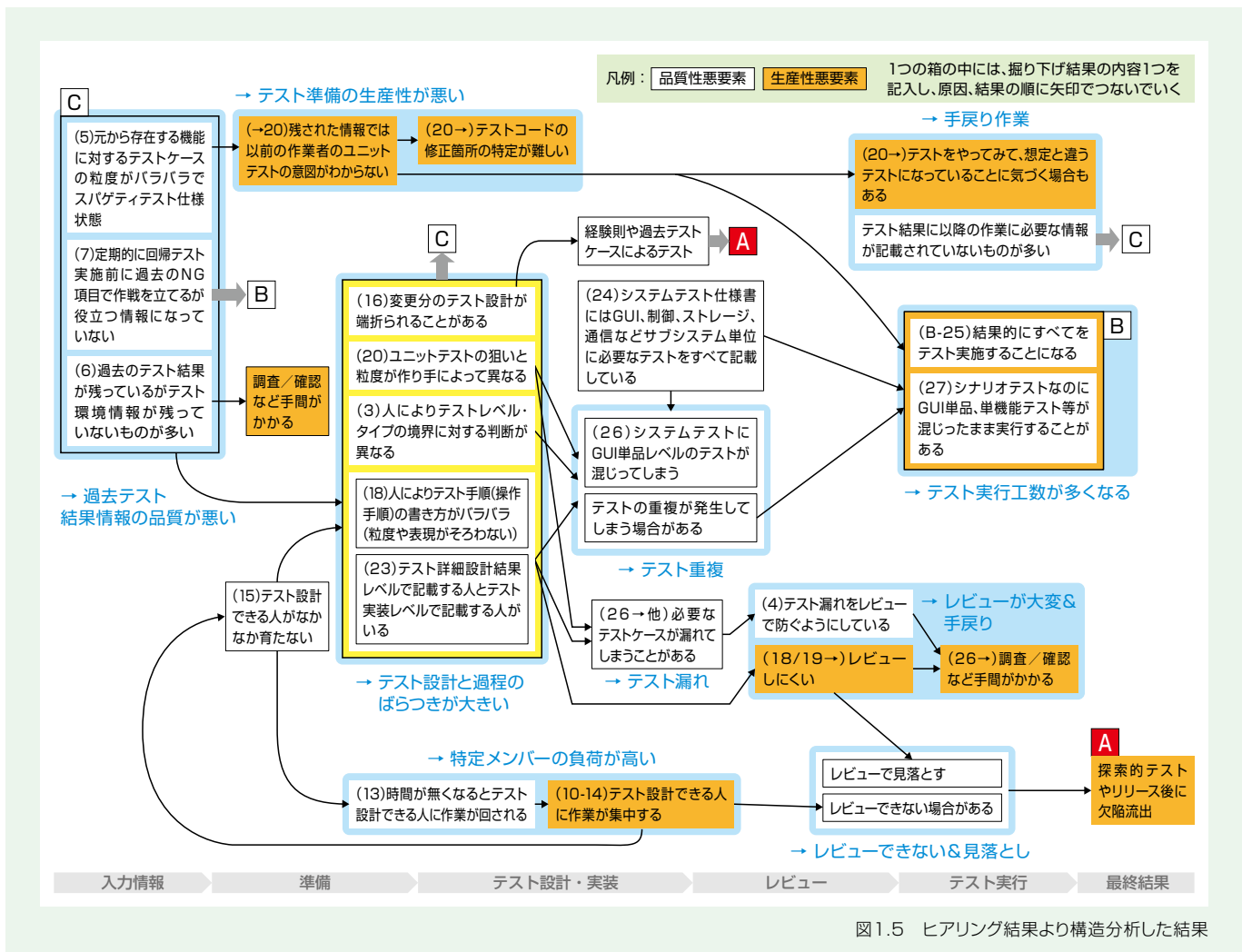


中山さんは当初のセルフアセスメント結果に誤りがあることに自ら気付いて修正を行っています。

1-8 改善対象の明確化

問題構造が明確になったところで、獲得したい結果(改善効果)とその実現のためにどこに改善施策を打つのがよいか、つまり改善対象の検討と特定を行います。検討の結果図 1.6 のように案1、案2の2つの候補が上がりましたが、今回は「テスト漏れによる欠陥流出を防ぐ」ために「テスト設計の方法を見直す」案1を優先で対応することにしました。

改善対象が決まれば、次は具体的な改善施策とその結果獲得する効果を明確にします。改善施策を検討する上で、最後に活用するのはTPI NEXTプロセスモデルに定義されたプラクティスです。テストプロセスモデルは、効果と効率の両面を兼ね備えたテストを実現するためにすべきこと(プラクティス)の集合体ですから、特定した改善対象に適用できるプラクティスを特定できればそれが「改善施策要件」になり得ます。今回のケースでは、主にキーエリア「テストケース設計」に記載されているプラクティスを改善施策要件の抽出に利用しました。



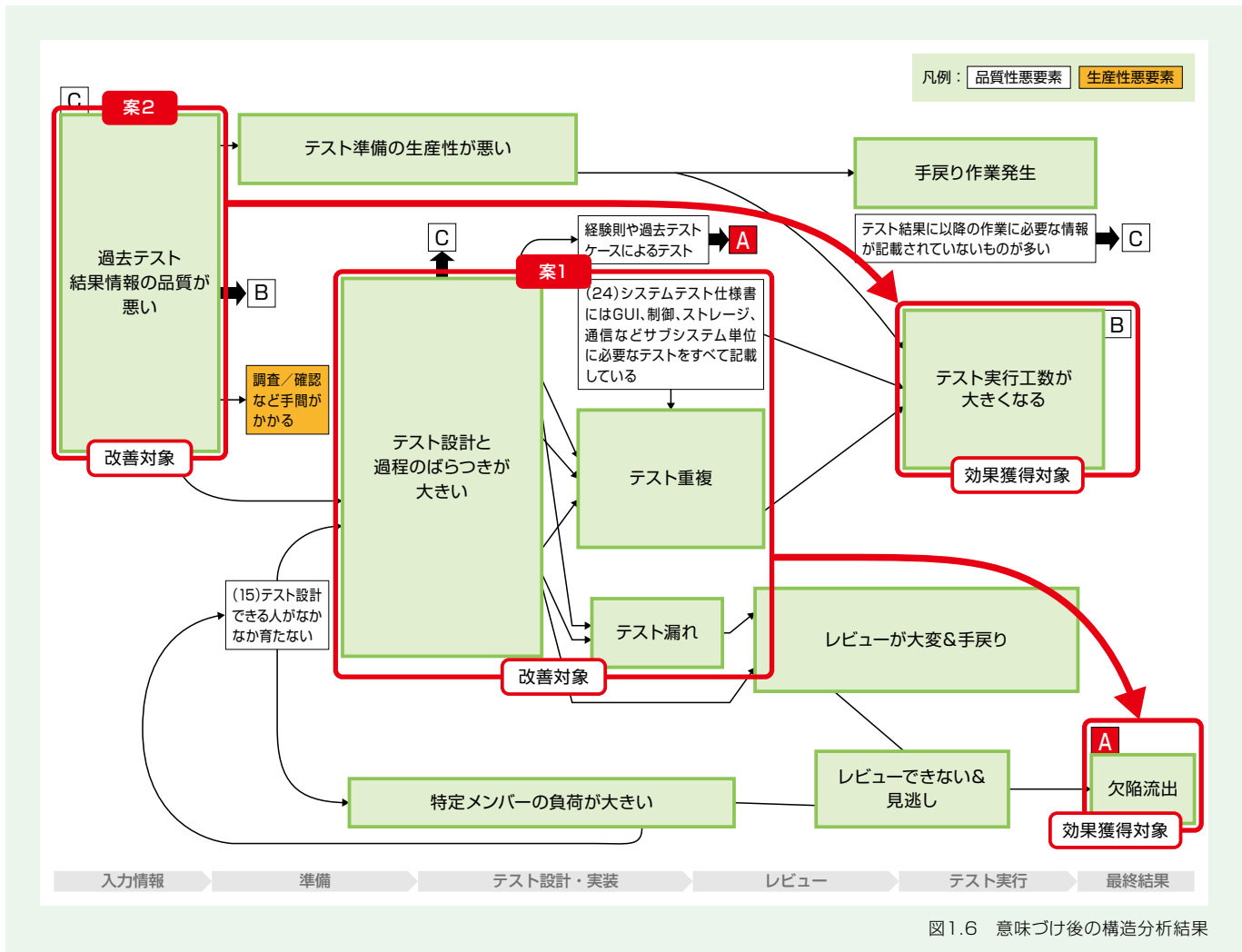


図1.6 意味づけ後の構造分析結果

改善施策要件が抽出できれば、それを参考に具体的な改善施策を検討することができるようになります。こうして、中山さんは納得してプロセスモデルを活用し、現場の改善に効果的な施策を打つことができるようになりました。

2. 事例で伝えなかったこと

今回の事例はプロセスモデル適用でしたが、新たなものを導入・適用することは、それにより実務の効果や効率が現在より良くなることを目指す「実務の改善」となります。それなのに、これらの取り組みにより余計な作業が増えて非効率になったり、やる気をなくしたり、疲弊するような結果になることは避けなければなりません。さて、これらの組織的な施策がどうして正反対の結果になることが多いのでしょうか？

2-1 組織的施策がうまくいかない要因と当事例における解決策

組織的施策がうまくいかない場合、大事な原理・原則に反していることが多いと感じています。それは「人は、他人から

言われたことでは変わらない」ということです。

人は、本心からそれが必要と思わない限り、自らを変えようとはしません。

組織的施策はたいていの場合、指示・命令を中心とするトップダウンで進められます。結果を出すためのスピードを求めれば求めるほど、短絡的に指示・命令で人を動かすことが多くなると思いますが、指示・命令された側は反抗するか、良くて表面的にやったことにするだけで、実態としては内面を変えていないため、時間が経つと元に戻ってしまいます。

1. 内発的動機づけを失わせる。
2. かえって成果が上がらなくなる。
3. 創造性をむしばむ。
4. 好ましい言動への意欲を失わせる。
5. ごまかしや近道、倫理に反する行為を助長する。
6. 依存性がある。
7. 短絡的思考を助長する。

図2.1 アメとムチの致命的な7つの欠陥
書籍「モチベーション3.0」ダニエル・ピンクより

指示・命令はアメとムチで表現されるように「外的動機付け」に分類され、現在ではさまざまな負の効果が指摘されています。(図2.1)

今回発表した事例では、指示・命令ではなく、自ら感じている問題点や困り事から改善すべきこと、改善策を自ら決め、行動する(私はその場づくりと後方支援だけを行う)アプローチを採用することでこの原則に対応しています。

2-2 プロセスモデル導入・適用がうまくいかない要因と当事例における解決策

プロセスモデル導入・適用がうまくいかないケースでは、「自ら手を染めたことは自分事になりやすい」に反している場合が多い、と感じています。

プロセスモデルによるアセスメントは、通常第三者、第三者などの有識者やリーダーが実施します。そのため、他者が出した結果をベースに改善を自分たちで行うように仕向けられるように受け取ってしまう可能性が高くなります。

また、「感情は論理に勝る」というように、いくら適切な論理を並べても、感情には勝てません。

プロセスモデルによる改善では、論理的なものづくりや管理の視点から現状評価(アセスメント)を行い、その結果をベースに論理的な対策を立案します。

毎日さまざまな問題に苦しめられている実務の現場に、論理の集合体であるプロセスモデルで現状を把握して説明しても、心に響かないことが多くなってしまいます。問題に苦しめられている度合いが大きいほど、負の感情が前に出てしまい、心に余裕が無くなり、論理を受け止める余裕が無くなってしまいます。

今回発表の事例では、メンバーそれぞれの負の感情につながる問題点や困り事を自ら記載し、その情報をベースに現状を把握して感情論の解決をチーム(自分事)として取り組める支援を行っています。

多くのメンバーが関心を寄せる問題の発生頻度を軽減し、メンバーが心の余裕を持てる状態になってから、組織やチーム、個人のミッションや目標達成のためにプロセスモデルを本格的に活用していくのが段階的にゴールへ向かう方法の一つになると思います。

2-3 最終目標は自律運営

事例で示したアプローチが最終的に目指している目標は自律運営の実現です。

自律とは、自ら考え、行動し、振り返りながら、周囲と協調しつつ目的をより良く達成して自ら成長することを意味します。周囲と協調し、三方よし(売り手、買い手、世間よし/現場、顧客、会社よし)を実現するために取り組むため、ワガママ三昧、周囲に迷惑を撒き散らしてでも自分のやりたいようにやる自分勝手とは異なります。また、自律とは何でも自分でできることを意味しません。自分の得意なところ、不得意なところ、至らないところをも把握して、意図的に他社・他者に依存(相互依存)しながら一緒に価値のある目的を達成するように行動します。

自律した方(やチーム、組織)は、たとえ困難に出くわしても自らの思考と行動、周囲への働きかけと協調で解決していきます。周囲への貢献により自信を深め、仕事がどんどん楽しくなる。人間関係も良好になっていき、さらに大きな価値を実現していきます。

自律に向かう第一歩は、自ら考えてやってみる、その結果を振り返ってより良いやり方や結果を引き寄せて「自分でもできる!」と実感する領域を拓けながら徐々に自信をつけてもらうことです。これは目先の行動や成果を強制する指示・命令の繰り返しでは成し遂げられないことです。いくら急かしても思考は突然速くなることはありません。要員やチームが持つ現状の思考力を徐々に高めていくように、根気強く支援する必要があります。今回の事例はその第一歩となるものです。

経営やマネジメントの方たちには部下の皆さんの育成方法として、リーダーの方たちにはチームが一体となる改善への取り組み方として、要員の方たちには自らの能力向上への取り組み方としてのヒントになれば幸いです。

参考文献

- TPI NEXT®ビジネス主導のテストプロセス改善. 藪田和夫,湯本剛,皆川義孝 訳. 株式会社トリフォリオ, 2015, 318p.
- “SaPID3.0”. <https://www.software-quasol.com/sapid3-0/>
(SaPID=Systems analysis/Systems approach based Process Improvement method)
- ソフトウェアテストシンポジウム(JaSST)2018東京投稿事例:
TPI Nextを活用した チームメンバーの問題意識から始めるテストプロセス改善【導入時:改善計画立案編】HBA安達賢二

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



要求仕様の曖昧さを理解し、ソフトウェアテストを組み立てる

株式会社ベリサーブ 東日本オートモーティブ事業部 第一ビジネスユニット 課長 東 弘之

はじめに

自然言語は思っているより曖昧です。当社ソフトウェアテストチームは、自然言語で記述された要求仕様書をテスト活動の入力情報として用いる機会が多くあり、多くの要求仕様曖昧さと向き合ってきました。

本稿では、日本品質管理学会誌に投稿いたしました自著論文を一部引用させていただきながら、要求仕様の曖昧さや、それら曖昧さへの向き合い方をご紹介します。

1. 要件定義にまつわる課題

JUAS^{*1}の調査によると、プロジェクトの工期遅延の理由のうち55%が要件定義の問題であるといわれています。その要因には、大きく2つが考えられます。

1. ユーザーからベンダーへ要件が正しく伝わらないこと。
2. リスクの認識・合意が不十分であること。

近年、短納期化やコストダウンを達成するための開発手法の一つとして、ソフトウェア開発の外部委託化が進み、開発工程ごとに、複数の組織を跨いで開発することが増えています。その結果、組織間に要求の認識齟齬が発生、受入試験までその齟齬に気づかず問題発覚が遅延するなどの問題が生まれる

^{*1}: JUAS(Japan Users Association of Information Systems):一般社団法人 日本情報システム・ユーザー協会

ようになりました。このような問題へ対応するために、要求仕様書を非曖昧に作成し、要件を正しく伝える、要件を正しく理解していただくことが重要な課題となっています。

自然言語には曖昧さに気付くための言語学的な技術に限界があり、曖昧さを完全に排除することは困難です。しかし、自然言語の特徴を理解し、適切な対応をすることにより、曖昧さを減らす・曖昧さによる影響を下げることは十分可能です。

2. 要求仕様の曖昧さを知る

要求仕様書を非曖昧に作成するためには、まず「曖昧な要求仕様」とは何かを知る必要があります。要求仕様の曖昧さについてはさまざまな定義が存在しますが、本稿では Daniel M.BerryとErik Kamsties による、書籍「ソフトウェア要求の全体像」(原題:Perspectives on Software Requirements) 2章における定義と分類(図1)をご紹介します。

曖昧な要求仕様(Kamstiesによる定義)

読み手の背景知識があるにも関わらず複数の解釈があるなら、曖昧な要求である。

本定義における「背景知識」とは、要求を解釈するために必要な知識のことを指し、要求文書、適用分野、システム分野、開発分野の4つからなります。これら背景知識が読み手に不足していた場合、要求文書が全うでも複数の解釈が生まれて

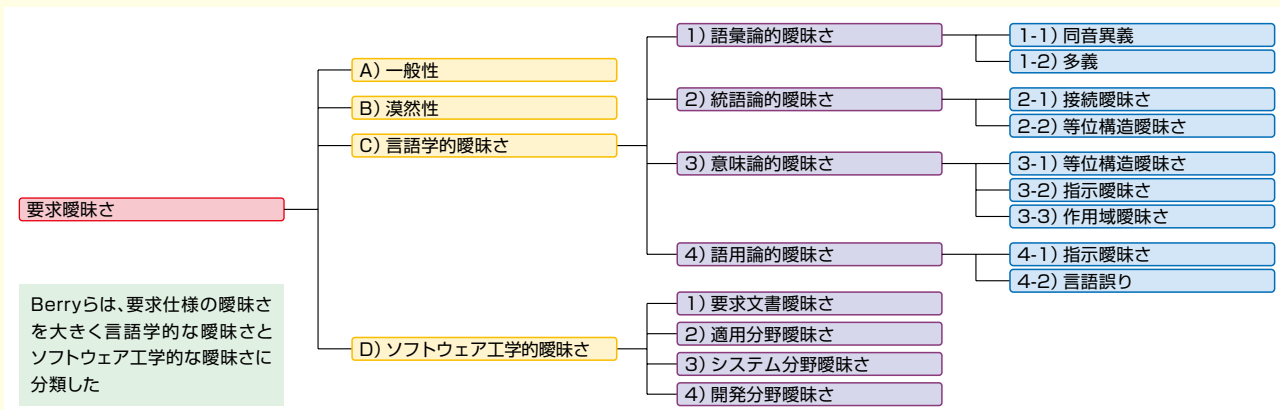


図1 Berryらによる要求曖昧さの分類

しまいます。書き手は、読み手の背景知識レベルの想定と対処を考慮に入れる必要があり、読み手は要求文章を正しく解釈するために必要な「背景知識」が何かを知り、理解する必要があります。

ここで、書籍「ソフトウェア要求の全体像」より曖昧さの例を紹介します。

自然言語の曖昧さの例

①The police shot the rioters with guns.

(警官は暴徒たちを銃で撃った。)

①の例は、言語学的な曖昧さを含んでいます。人によっては、「銃で武装した暴徒たちを、警官は撃った」というように、この文章を捉えます。これは、“with guns”が”shot”と”rioters”のどちらの修飾語句にもなり得るからです。これは2-1)接続曖昧さの例となります。

②Sue is visiting her cousin.

(スーは彼女のいとこを訪ねている。)

③He is tall.(彼は背が高い。)

②、③の文章は共に不確定性と言われる曖昧さを含む文章で、②は一般性、③は漠然性に分類されます。②の文章のcousin(いとこ)は、性別の判断はできないので一見曖昧だとも言えます。しかし読み手、書き手にとって性別情報が不要であるならば、満足な意思疎通を行えます。一方③のtall(高い)は、文化などにより異なります。プロバスケットボール選手にとって、身長180cmは高いとは言えません。漠然性は読み手によって解釈が異なりやすく、境界値ケースで多く現れます。

④Shut off the pumps if the water level remains above 100 meters for more than 4 seconds. (4秒より多く水位100メートルを上回ったなら、ポンプを止める。)

④の例は、ソフトウェア工学的曖昧さの例で、読み手の適用分野知識の有無により複数の解釈が生まれます。下記のように色々な解釈が可能ですが、多くの工学領域での標準的な解釈は、解釈3となります。タンク内の水が波打たない場合は全て同じ結果となり、実装が簡単な解釈4でも良いですが、波打った場合は危険な水位になる可能性があります。

解釈1：平均水位が100メートルを上回る

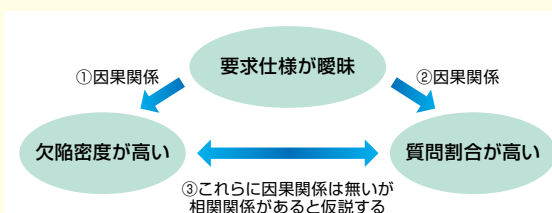
解釈2：中央値水位が100メートルを上回る

解釈3：二乗平均平方根水位が100メートルを上回る

解釈4：最小水位が100メートルを上回る

3. 要求仕様の曖昧さ情報を活用する

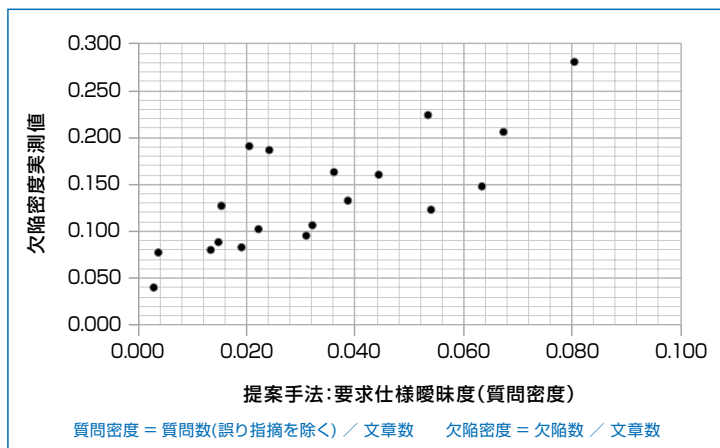
自然言語の特徴から、自然言語で書かれた要求仕様の曖昧さを完全に排除することは不可能です。実際には、要求仕様曖昧さを排除することがコストや期間的に困難で、曖昧なまま進めるプロジェクトもあります。そこで、要求仕様の曖昧さ度合の情報をソフトウェアテスト工程に生かすことはできないかと考え、次のような仮説を立て検証を行いました。日本品質管理学会誌に投稿した自著論文では、この仮説について考察を行っています。



仮説 要求仕様が曖昧な領域ほど、欠陥密度が高い

- ①要求仕様が曖昧だと欠陥密度が高くなる。(因果関係)
- ②要求仕様が曖昧だと、テスト設計時の質問割合が高くなる。(因果関係)
- ①と②の因果関係から③の相関関係が仮定される。
- ③質問割合の高い機能領域は、欠陥密度が高い。(相関関係)

図2 要求仕様の曖昧さが与える影響



- 検証対象のシステムを30の機能領域に分け、各領域の質問密度と欠陥密度を計測した。(外れ値、てこ比の大きなデータは除外)
- これら密度間の相関係数は0.76となり、相関があることを確認できた。

欠陥密度測定手法	各測定値と実測値の相関係数 (括弧内は有意確率)	
	ピアソン積率相関係数	スピアマン順位相関係数
提案手法	0.76(P<0.001)	0.75(P<0.001)
Kaner法	-0.18(P=0.348)	-0.10(P=0.621)
鎌田法	0.28(P=0.151)	0.37(P=0.054)
経験法	0.10(P=0.615)	0.07(P=0.725)

- 提案手法以外にも、仕様書改編数や序章の充実度、事前テスト結果から推測する手法を試し、比較を行った。
- 提案手法のみ相関が有意であると評価でき、相関も提案手法が最も高いことを示した。

図3 要求仕様曖昧度と欠陥密度の分布

この仮定が正しいかを、とあるプロジェクトで検証を行いました。その結果、曖昧さ度合いと欠陥密度の間に相関があること示す結果を得ることができました。(図3)

この関係性を用いることで、リスクベースドテストのインプットとしてより効果的な指標が得られます。すなわち、テスト設計時の質問内容や質問領域を調査し、欠陥の多い領域を推定、テスト組織やテストの厚みを決めるときの参考にすることができます。ソフトウェアテストの見積・計画は、困難なことが多いのですが、要求仕様の曖昧さは、外部委託開発形態の

プロジェクトで確認することのできる、数少ない欠陥予測の情報として活用が期待できます。

4. 要求仕様の曖昧さを正す

曖昧さを活用する一方、曖昧さを減らす取り組みも、もちろん行っています(むしろ、こちらが王道です)。プロジェクトを成功に導くためには、上流成果物の品質を高めることが不可欠であり、近年、お客様(委託元企業)からのニーズも、テスト設計工程から、要求仕様の作成工程、プロセス作りと、より上流工程からの品質づくりへと変化しています。図4は当社の品質向上のソリューション例となりますが、①、②の部分における取り組みが曖昧さ削減へ効果的な活動となります。

おわりに

本稿では自然言語を用いた要求仕様の曖昧さ、および当社で行っている曖昧さへの向き合い方を2つご紹介しました。何より重要なのは、自然言語は曖昧であることを知っていることです。曖昧さにはどのような種類があるか、要求仕様書を読むために(書くために)必要な背景知識は何か、などを知っておくことで、より正しく要求を理解できるようになると思います。本稿が、皆様の品質向上活動の一助になれば幸いです。

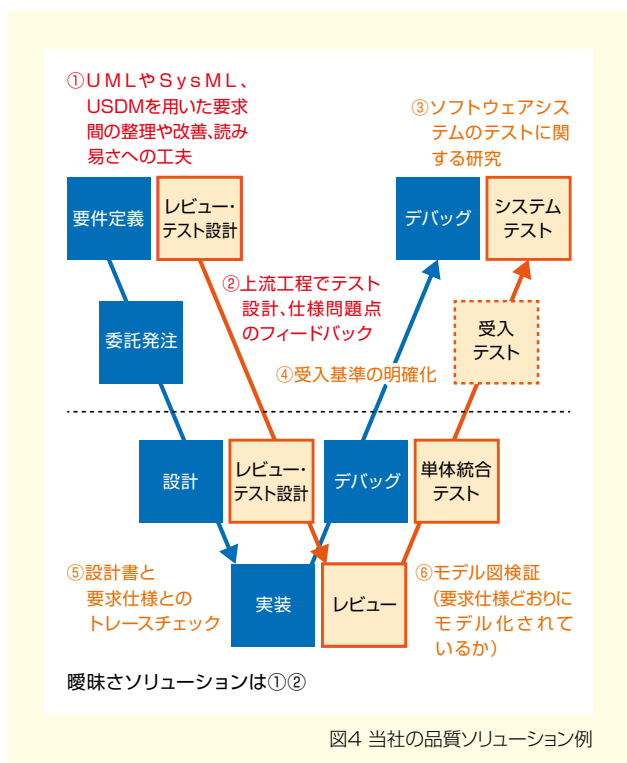


図4 当社の品質ソリューション例



非機能テストにおけるベリサーブの取組みと、IoTのセキュリティ要求への対応

株式会社ベリサーブ ソリューション事業部 事業部長 桑野 修

はじめに

当社は、検証専門企業として2001年に設立した当初から、専門性を追求する分野の一つとして「非機能要件」の検証に携わってまいりました。

昨今のIoT関連製品の普及に伴い、「性能」や「セキュリティ」をはじめとした「非機能要件」に関する検証のご相談が増加の一途をたどっております。

そこで、本稿では「非機能要件」中でも特に重要視されている「セキュリティ」を中心に、当社での取組みをご紹介します。

1. 非機能要件について

1.1 非機能要件とは

「非機能要件」という言葉については、耳慣れない方もいらっしゃるかもしれません。ソフトウェア製品・システムには、満たすべき振る舞いが「要求仕様書」などで明示的に定義される「機能要件」と、システム全体が満たすべき特性として期待されるが、必ずしも明記されていない「非機能要件」の2つが存在します。本来、非機能要件の実現には「システムアーキテクチャ設計」などの「最上流工程」での考慮が必要

となるにも関わらず、暗黙の要件として見落とされてしまうことが少なくありません。実際に非機能要件に関する開発の手戻りは非常に大きく、ユーザーと開発者間でのトラブルを引き起こす要因の一つになっています。

1.2 非機能要件の重要性の高まり

ネットワークの発達によりシステムの規模やユーザー数の増大、多種多様なデバイス同士がつながることで、想定外の使われ方が出てくることから、非機能要件をきちんと考慮した開発がよりその重要性を増しています。2008年ごろからユーザー側、ベンダー側双方の立場を考慮した、「非機能要件要求仕様定義ガイドラン(日本情報システムユーザー協会)」、「非機能要求グレード(2010年からはIPAより発行)」といった非機能要件についての要求ガイドラインが発表されるようになりました。品質特性においても、ISO/IEC 9126-1から、ISO/IEC 25010への改定時、「機能性」の副特性として定義されていた「セキュリティ」が、副特性から主特性として格上げされたことから、その重要性が伺えます。このような流れからソフトウェアの開発に携わる企業は、ガイドラインや標準化をもとに、品質面でのリスクを十分に考慮する必要に迫られています。

*1: ISO/IEC 9126-1:ソフトウェア製品の品質—第1部:品質モデル

*2: ISO/IEC 25010: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

2.非機能要件に対するアプローチ

当社では、非機能要件の品質に対する関わり方として、主に「性能」「セキュリティ」の2つの側面よりアプローチを行っています(図1)。非機能要件については、テスト工程から対策を行うのではなく、要求・要件整理やレビューなどの活動を通して、開発の上流工程からテストエンジニアがプロジェクトに関わるケースも増えてきました。テストプロセスを開発プロセスと並走させることにより、早い段階で課題を検出することが可能で、プロジェクトの効果的なリスク回避につながっています。

3.セキュリティ要件に対する取り組み

図2は、IT、IoTを取り巻く代表的な脅威と、それに伴うセキュリティ基準の開始時期を表しています。インターネットの普及をきっかけに、2000年頃より情報セキュリティの基準が設けられるようになりました。しかし、IoT機器の普及により、機器同士やIoT機器とWebサービスの接続が行われるようになってきたことで、新たなセキュリティリスクが急増しています。ITサービス・製品の開発に携わる事業者様、とりわけ標準化のスピードが速い自動車・車載分野を中心に、新たなセキュリティ基準の策定および適用が急速に進んでおり、多くの事業者様がその対応に苦慮されております。

そこで、当社でもセキュリティ基準の準拠のご支援を行っている、代表的なセキュリティ基準や業界の動向についてご紹介いたします。

① PCI DSS

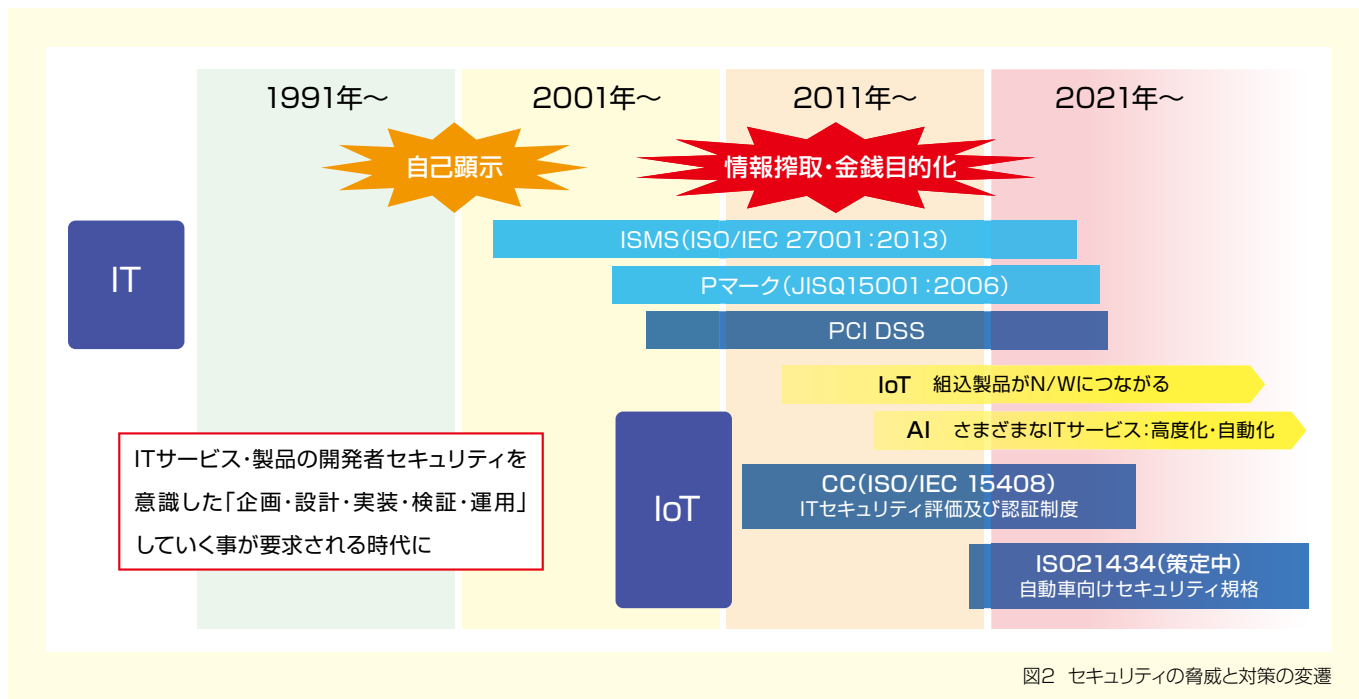
PCI DSSとは、Payment Card Industry Data Security Standard の略で、VISA、MasterCardなどをはじめとするペイメントカードブランド5社が展開しているセキュリティプログラムの拠り所となる、国際的なデータセキュリティ基準です。2006年9月に公開後、改定が繰り返され2018年12月時点では2018年5月に改訂された3.2.1版が最新版となっています。この基準にはセキュリティ要件と、具体的なテスト方法が記載されているため、カード業界だけでなく、機密性の高いデータを保持するあらゆるシステムのセキュリティに関する要件抽出に役立ちます。また、2020年の東京オリンピック開催に向けた、クレジットカードのセキュリティ強化を目的として、すべてのカード事業者についてはPCI DSS準拠が義務化されるようになりました。

② CC認証・JISEC

CC(コモンクライテリア)認証とは、IT製品に対するセキュリティ評価の国際標準規格を指します。CCは1999年にISO標準(ISO/SEC 15408)に制定されており、車載などの業種別セキュリティガイドラインでも広く引用されています。日本ではJISEC(Japan Information Technology Security Evaluation and Certification Scheme)と呼ばれる、IT関連製品のセキュリティ機能の適切性・確実性を、第三者(評価機関)がCCに基づいて評価し、その結果を認証機関が認証する制度があります。この制度は主に政府調達に利用されています。JISECよりCC認証を受けた場合、その認証はCCRA(CC国際認証アレンジメント)加盟国17カ国と、11カ国受入国からも相互認証を受けることが可能です。

対象	開発工程での取り組み	テストプロセスでの取り組み	主なテストの種類
性能	<ul style="list-style-type: none"> ●キャパシティ予測 アクセスログ・リソース分析 ビジネス要求からのリソース予測 	<ul style="list-style-type: none"> ●テストアウトソース (テストプロセス全般) ●テストPMO テスト推進・方式設計 テスト実行支援 	<ul style="list-style-type: none"> ●性能テスト ●負荷テスト ●限界負荷テスト ●累積稼働テスト
セキュリティ	<ul style="list-style-type: none"> ●要求分析支援 脅威分析・対策検討 ●レビュー支援 要件策定支援 要件・設計レビュー ●ソースコード検証 セキュアコーディング規約検証 脆弱性診断 	<ul style="list-style-type: none"> ●テストアウトソース (テストプロセス全般) 	<ul style="list-style-type: none"> ●セキュリティ性能適合性テスト ●脆弱性テスト ●Fuzzing

図1 当社における非機能要件に対するアプローチ



③自動車・車載分野における動向

組み込みソフトウェア製品領域の中で、セキュリティ対策の具体化が進んでいる分野として「自動車・車載」の分野が挙げられます。2020年以降を目途に、自動走行やコネクテッド化も視野に入れたセキュリティ対策に関する国際標準ISO21434の策定が予定されており、OEM(完成車メーカー)を中心に、CCなどを活用したセキュリティ要件の明示化が自主的に進められています。このような動向に伴い、Tier1(一次請サプライヤ)をはじめとした各種サプライヤについても、セキュリティ要件に対する対応の検討・実施についての説明責任が伴うようになりました。これらの対応については、お客様からのご相談も多く、検証の立場よりご支援を行っている分野となります。

④セキュリティ運用についての課題

セキュリティ要件の新たな動向として、SIRT(Security Incident Response Team)組織の立ち上げを課題とされているお客様が増えています。背景としては、プロダクトの開発にOSSを利用するケースなどが増えてきており、自社製品の中に組み込まれたOSSの脆弱性がリリース後の運用時に見つかるケースなどが増えてきたためです。情報システムに対する対応組織であるC(Computer)SIRTの普及はここ5~10年で進んできましたが、IoT機器の普及に伴い、組み込み製品に対する対応組織P(Product)SIRTの立ち上げを急がれるお客様が増えています。組織の立ち上げに関するお客様の課題は、概ね次のようなものに分類されます。

- a. PSIRT 組織構築・ガイドライン策定
- b. 付帯業務の効率化
- c. 脆弱性管理運用

当社では、これらの課題に対し、製品を構成するソフトウェアのBOM(部品表)の整備や、ナレッジの集約などのご提案を通して、SIRT組織の立ち上げや、運用効率化のためのシステム整備および運用のご支援を行っております。

おわりに

IoT技術の進歩と普及により、非機能要件の中でも特にセキュリティのリスクは飛躍的に高まっています。そのセキュリティリスクを回避するために、さまざまな分野においていろいろな規格や基準が設けられ、それが常に改変されているのが現状です。当社は、それらのトレンドを常に取り込みながら、適切なサービスやソリューションを考えて、お客様のご要望にお応えして参ります。

AI利用テストツールの最新動向

AIはテスターを超えるのか

はじめに

第三次AIブームと言われる中、ソフトウェア開発の世界でもAI製品の開発方法(SE4AI: Software Engineering for AI)とソフトウェア開発へのAIの利用(AI4SE: AI for Software Engineering)の両面からの議論が盛んになっています。実践面の事例ではAI4SEに関するものが先行しているようで、ここ1~2年インターネットニュースやSNS、あるいはカンファレンスでAI4SEのツールの記事や講演を目にすることが多くなりました。

本稿では、主にインターネットで収集した情報に基づいて、AI4SEの概況と我々に関係の深いソフトウェアテストへのAIの利用の典型例としてAI利用テストツールの最新動向を紹介します。



辰巳 敬三氏
たつみ けいぞう

1976年、富士通株式会社に入社。ソフトウェア製品検査部門でメインフレームOSの検査、品質保証を担当。その後、UNIXやPCのソフトウェア検査に従事。1999年に検査の現場は離れたがソフトウェアの品質、テストの技術の探求をライフワークとして研鑽を続けている。

2009年から2016年にNPO法人高度情報通信人材育成支援センター(CeFIL)に出向し産学連携のIT人材育成を担当。

1. ソフトウェア工学関連の会議や活動の最近の動き

従来から開催されているICSE(International Conference on Software Engineering)などのソフトウェア工学系国際会議やISSTA(International Symposium on Software Testing and Analysis)などのソフトウェアテスト系国際会議ではAI関連の論文発表が増えてきています。さらに2017年頃からはAIとソフトウェア工学を主題とする新たな研究系の国際会議やワークショップの開催、実務家のコミュニティの設立、標準化などの動きが始まりました。筆者が認識している新たな会議や活動を表-1にまとめましたので状況を見ましょう。

海外の研究系の会合では、RAISEが比較的早く第三次AIブームが始まった頃の2012年に開始されました。ICSE併設のワークショップで、AIとソフトウェア工学の関係を双方向から研究者と実務家で議論することを目的としています。SEMLAはAIの中でも機械学習に焦点をあて、機械学習

アプリケーションへのソフトウェア工学の適用を主題としてモントリオール理工科大学の研究者が立ち上げた研究会です。2018年6月に第一回がカナダで開催されました。WAISEはコンピュータの安全性、信頼性、セキュリティに関する国際会議SAFECOMPに新たに併設されたAIの安全工学ワークショップです。2018年9月にスウェーデンで第一回が開催されました。iMLSEはアジア太平洋ソフトウェア工学国際会議APSEC 2018併設の機械学習工学に関する第一回国際ワークショップです。前述のSEMLAと後述の機械学習工学研究会(MLSE)の共催で12月に奈良で開催されました。

ソフトウェアテストの研究系の会合としては、2019年4月にIEEEの第一回AIテスト国際会議AITestが開催されます。また、ソフトウェアテスト国際会議ICST 2019でも新たに併設で機械学習のテストとテストへの適用の双方向を扱うワークショップMLSTが2019年4月に開催、さらにICSE 2019でもディープラーニングのテストとテストへの適用の双方向を扱うワークショップDeepTestが2019年5月に開催されます。

日本国内の動きとしては機械学習工学研究会が日本ソフトウェア科学会の研究会として2018年度より活動を開始し、2018年5月にキックオフシンポジウムが開催されました[1]。また、AI製品の品質保証に関するAIプロダクト品質保証コンソーシアム(QA4AI)が2018年3月に発足しています。

実務家のコミュニティの活動もいくつか見受けられます。AISTAはAI利用テストツール会社test.ai(旧称Appdiff)の創立者のJason Arbon氏とUltimate Software社のTariq King氏が2017年に立ち上げた米国の組織で、テストと品質工学に関するAIの調査やコミュニティでの情報共有の促進を図っています。AIFORSE CommunityはウクライナのValentin Grigoryevskiy氏らが2017年に設立した組織で、AIを使ったソフトウェア開発企業や利用者の協業プラットフォームになることを目指して、AI適用に関するカンファレンスやWebinarの開催、情報収集や公開を行っています。

標準化の活動も始まっています。国際標準化機構ISOでは2017年に情報技術専門委員会(JTC1)の配下にAIの委員会(SC 42,人工知能)を設け、ISO/IEC 22989(AIの概念と用語)とISO/IEC 23053(機械学習を使ったAIのフレームワーク)の策定を開始しました。

AI製品の品質についても先述のQA4AIの発足が示すように課題は多く高い関心もたれています。オランダのSogeti社はAIの時代におけるテストや品質に関する書籍"Testing in the digital age: AI makes the difference"を2018年6月に発刊しました[2]。この書籍では、ISO 25010の製品品質特性に、AIやロボティクスで使うことを想定した品質特性/副特性を追加することを提案しています。

分野	名称	発足	概要
研究	海外	RAISE (International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering) http://promisedata.org/raise/2018/	2012年2月 AIとソフトウェア工学の双方向の関係を扱うICSE併設のワークショップとして2012年に開始。RAISE'18で6回目を開催。
	SEMLA(Software Engineering for ML Applications) http://semmla.polytml.ca/	2018年6月	機械学習アプリケーションへのソフトウェア工学の適用を扱う研究会。第一回をカナダで開催。
	WAISE(International Workshop on Artificial Intelligence Safety Engineering) https://www.waise2018.com/	2018年9月	AI適用システムの安全工学のワークショップ。SAFECOMP(安全/信頼性/セキュリティ工学国際会議)2018に併設で第一回を開催。
	iMLSE(International Workshop on Machine Learning Systems Engineering) http://research.nii.ac.jp/iMLSE/	2018年12月	APSEC 2018併設の機械学習工学に関する第一回国際ワークショップ。SEMLAと機械学習工学研究会(MLSE)の共催。奈良で開催。
	AITest(International Conference on Artificial Intelligence Testing) http://www.ieeeaitests.com/	2019年4月	AIアプリケーションのテスト、テストへのAIの適用、AIアプリケーションのデータ品質を扱うIEEEの第一回AIテスト国際会議。
	MLST(International Workshop on Machine Learning and Software Testing) https://mlst2019.github.io/	2019年4月	機械学習とソフトウェアテストの双方向を扱うワークショップ。ICST 2019併設で第一回を開催。
	DeepTest (International Workshop on Testing for Deep Learning and Deep Learning for Testing) https://deeptestconf.github.io/index.html	2019年5月	ディープラーニング(深層学習)とソフトウェアテストの双方向を扱うワークショップ。ICSE 2019併設で第一回を開催。
日本	AIプロダクト品質保証コンソーシアム(QA4AI: Consortium of Quality Assurance for AI-based products and services) http://www.qa4ai.jp/	2018年3月	AI製品の品質保証に関する調査・体系化、適用支援・応用、研究開発の推進とAI製品品質の適切な理解の啓発活動を展開。
	機械学習工学研究会 (MLSE: Machine Learning Systems Engineering) https://sites.google.com/view/sig-mlse	2018年4月	機械学習システムの開発・運用の生産性や品質の向上を追求する研究会。2018年度より日本ソフトウェア科学会の配下で発足。
コミュニティ	AISTA (Artificial Intelligence for Software Testing Association) https://www.aistesting.org/	2017年	Jason Arbon氏とTariq King氏が立ち上げた米国の組織。テストと品質工学に関するAIの調査、コミュニティの協力や情報共有を促進。
	AIFORSE Community (Artificial Intelligence for Software Engineering) https://aiforse.org/	2017年	ウクライナのValentin Grigoryevskiy氏らが設立した組織。AI適用に関するカンファレンスやWebinarの開催、情報収集/公開を実施。
標準化	ISO/IEC JTC1/SC42 Artificial intelligence https://www.iso.org/committee/6794475.html	2017年	ISO 22989 (AIの概念と用語)、ISO 23053(機械学習を使ったAIのフレームワーク)の策定開始。
品質の定義	Testing in the Digital Age - AI makes a difference https://www.sogeti.com/explore/books/testing-in-the-digital-age/	2018年6月	オランダのSogeti社の著者が出版した書籍。AI製品の品質についてISO 25010の品質特性を拡張したモデルを提示している。

表 -1. AIに関する最近のソフトウェア工学関連の会議や活動

2. ソフトウェア開発を支援するAIツール

前章で紹介したAIFORSE Communityでは定期的にAI4SEツールの提供状況を調査して"Industry Landscape"と呼ぶAI4SEツール提供状況表を公開しています[3]。

このLandscapeは横軸をソフトウェアライフサイクル、縦軸をツールの成熟度(提供状況)にしてツールが分類、集計されているので、ライフサイクルのフェーズごとにどの程度のAI4SEツールが提供されているのかが分かります。ツールの成熟度はそのツールのWebサイトで、顧客、デモ、価格の3つの情報のうち2つ以上あれば提供可能(B2B Ready、B2C Ready)な状態になっていると判定されています。Webサイトに記載された情報に基づく判断なので厳密なものではないですが大まかな状況は把握できていると思います。

表-2は本稿執筆時の最新のLandscape(2018年8月版)です。掲載にあたってはAIFORSE CommunityのGrigoryevskiy氏の許可を得るとともに、項目名の翻訳と注釈をつけました。集計結果に基づいてAIFORSE Communityは次のようにコメントしています。

- AI4SEツールの数は前回調査(2017年10月)の29から今回は43に増加し10ヶ月で48%の増加。
- フェーズ別に見ると、実装(コード構築/構成管理)と品質管理/テストが多い。
- 成熟度で見ると、品質管理/テストの方が実装よりも実用段階のツールが多い。
- 前回調査時点から対象フェーズを変更したツールが5件あり、ベンダーはまだまだ手さぐり状態。

まとめると、AIを利用する動きはソフトウェアライフサイクルの各フェーズで始まっており、実装やテストという比較的作業が具体的なフェーズのツールは実用段階に入ってきたと言えます。

3. AIを利用したテストツール

テストに関するBlog、TestTalksという対談、Automation Guildというオンラインカンファレンスの開催などで著名な米国のJoe Colantonio氏は2017年11月のBlogでAIと機械学習によりテスト自動化に第三の波が来ていると書いています[4]。

- 第一の波：WinRunner、Silk Test、QTPなどのベンダーツール
- 第二の波：Seleniumの登場
- 第三の波：AIと機械学習を利用したツールにより次の段階へ

確かに前章で紹介したLandscapeの状況を見てもこの第三の波が感じられます。それでは、この波を具体的に見てみましょう。

表-3はLandscapeで調査された品質管理/テストツールに筆者が調査したAI利用テストツールを加えて一覧化したものです。盛り上がっている状況を把握するために、ツールの機能概要に加えてツール開発企業の設立情報も調査して記載しました。

まず、表-3の18社の設立年を見ると、2010年代の企業が13社、内8社は2015年以降の設立なので、AI利用テストツールを提供しているのは設立数年のスタートアップ企業が多いことが分かります。スタートアップのデータベースCrunchBaseの情報からはこれらの企業へのベンチャー投資が盛んな様子がうかがえます。また、2018年の半ばにはアジャイルテストで著名なLisa Crispin氏がMabl社に、AIテストに関して数多く講演しているAngie Jones氏がApplitools社に移籍するなど、本格的なビジネス展開に向けて人材の移動も始まっています。一方、2010年以前に設立され第二の波をリードしている中堅スタートアップ企業も既存ツールへのAIの追加を始めていますが新興スタート

INDUSTRY LANDSCAPE ARTIFICIAL INTELLIGENCE for SOFTWARE ENGINEERING (ver.2018-08-12)

ツール成熟度(*1)	プロジェクト管理	要求	設計	コード構築/構成管理	品質管理/テスト	保守	合計
提供可能 B2B Ready	0	1	0	4	9	2	14
提供可能 B2C Ready	0	0	3	3	0	0	3
社内ツール Internal Development	0	0	1	1	0	0	1
説明のみ Landing Page	2	3	8	12	8	3	25
合計	2	4	12	20	17	5	60\43

(*1) B2B Ready: Webページに顧客、デモ、価格のうち2つ以上の情報あり
B2C Ready: B2B Readyと同じ。SEツールではなく最終顧客用製品

社内ツール: 大規模ソフトウェアエンジニアリング企業の社内開発ツール
説明のみ: ツール説明Webページはあるが顧客、デモ、価格情報なし

表-2. AI4SE ツールの提供状況

アップに比べて少し出遅れているようです。

次に、テストに関する作業のどこにAIが利用されているのかを見ます。次ページの表-4は各社のWebサイトに記載されている情報からAIを利用している機能を筆者が推定して分類したものです。実際にツールを使用して調査した訳ではないので情報が正確でない可能性があることにご注意ください。アプリケーション画面を画像解析AIで処理してテスト実行、ビジュアルの検証、テストの自動修復(画面要素の変更を認識して修正)を行うツールが半数の9件あります。現在のAI利用テストツールの中心を占めていると言えます。

また、AIを使ってテストの自動生成をしているツールが7件あります。中でもFunctionizeは人間が記述したテストプランを自然言語処理(NLP:Natural Language Processing)とAIを使って解析してテストケースを自動生成する機能を提供しており、かなり先進的なツールとなっているようです。他には、コード解析や性能テスト/テスト結果分析を適用分野とするツールが提供されています。

テスト自動化の第三の波は、単なるテスト実行フェーズの自動化にとどまらずテスト活動のすべての段階に押し寄せてきています。

No.	ツール名	機能概要	会社名	国	設立年	ファンド状況 最新ファンドタイプ (*1)
1	Acellere	ソフトウェア静的解析ツールGAMMAを開発・提供。内蔵するAIを利用して影響範囲を検証しながら効率よくリファクタリング作業可能。	Acellere http://www.acellere.com/	ドイツ	2009年	Early Stage Venture Series A
2	Apimetrics	エンドツーエンドの性能監視ソリューションを提供。機械学習とAIにより、見逃した問題、懸念事項、APIとの関連を特定。	Apimetrics https://apimetrics.io/	米国	2014年	Seed Seed
3	AppAchhi	機能テスト、性能テストの実行プラットフォーム。AIテストボットが実行状況、スクリーンショット、ログからクラッシュを識別。	AppAchhi https://appachhi.com/	インド	2015年	— —
4	Appltools Eyes	アプリケーションのUIレイアウト、コンテンツ、外観を自動的に検証するクラウドベースのテストツール。	Appltools https://appltools.com/	イスラエル	2013年	Late Stage Venture Series C
5	Appvance IQ	AI駆動の統合テスト自動化システム。AI Scriptingによりユーザのビヘイビアに基づくテストケースを生成する。	Appvance https://appvance.ai/	米国	2012年	Early Stage Venture Series A
6	Autonomiq	テストケース作成から影響分析まで全テストライフサイクルを自動化する自律型プラットフォーム。ディープラーニングとAIにより変更を検出してテスト資産を自動修復。	Autonomiq https://autonomiq.io/	米国	2018年	投資ファンドなし —
7	DeepCode	機械学習によりOSSのコードから得た知識を元にコードレビューとQA監査を行う。スイス連邦工科大学チューリッヒ校(ETH)のスピンオフ企業。	DeepCode https://www.deepcode.ai/	スイス	2016年	Seed Seed
8	Diffblue Cover	AIにより単体テストの自動作成、リグレッションスイートの改善、未テスト部分を検出し単体テストの自動作成を行う。オックスフォード大学のスピンアウトベンチャー。	DiffBlue https://www.diffblue.com/	英国	2016年	Early Stage Venture Series A
9	Eggplant AI	グラフィカルツールで記述したアプリケーションのモデルに基づいて、機械学習を使ったEggplant AI Serverによりカバレッジ分析やバグ検索を行ってテストケースを作成。	Eggplant https://eggplant.io/	英国 米国	2008年	— —
10	Functionize	テストプランを自然言語処理(NLP)とAIを使って解析してテストケースを自動生成し、数千のテストを数分で実行するクラウドテストプラットフォーム。	Functionize https://www.functionize.com/	米国	2015年	Seed Seed
11	Mabl	機械学習を使った自動テストサービス。外観の異常検出、リグレッションテスト、テストの自動修復などを自動化。	Mabl https://www.mabl.com/	米国	2017年	Early Stage Venture Series B
12	Magic Pod	モバイルアプリケーションの自動テストのスクリプト作成を支援。ディープラーニングなどの技術を活用して画面から項目を自動検出してテストを作成。	TRIDENT https://www.trident-qa.com/	日本	2012年	— —
13	Perfecto	Web、モバイル、IoTアプリの継続的テストやデリバリーのためのクラウドプラットフォーム。最近のWebinarで機械学習やAIによるリポーティング機能を紹介。	Perfecto https://www.perfecto.io/	イスラエル 米国	2006年	M&A Series E
14	ReTest	GUIリグレッションテストツール。AIを使ったモンキーテストの機能をもつ。	ReTest https://retest.de/	ドイツ	2016年	— —
15	SOAtest Smart API Test Generator	APIテスト自動化ツール。AIで、APIコールの識別とデータの関係性を分析し、さまざまなユースケースのAPIテストシナリオを自動生成する機能を追加。	Parasoft https://www.parasoft.com/	米国	1987年	— —
16	test.ai	AIボットによりリグレッションテストを実行するモバイルテスト自動化サービス。アプリの画面と要素を識別しユーザのシナリオを実行。	test.ai http://test.ai/	米国	2015年	Early Stage Venture Series A
17	TestComplete	GUIアプリケーションのテスト自動化ツール。従来手法とAIによるビジュアル認識を組み合わせたハイブリッドエンジンですべてのUIを検出してテスト。	SmartBear Software https://smartbear.com/	米国	2009年	— —
18	Testim.io	機械学習を使用して、自動化テストの編集、実行、保守をスピードアップ。機械学習によりテストスイートの安定性が向上。	Testim https://www.testim.io/	米国	2014年	Early Stage Venture Series A

(*1)スタートアップのデータベースCrunchBaseの2018年11月時点の情報

表-3 AI 利用テストツール

No.	ツール名	AI利用機能のテスト適用分野						
		静的解析／ コード解析	テスト 自動生成	テスト実行	ビジュアルの 検証	テスト 自動修復	テスト実行 状況監視	性能／テスト 結果分析
1	Acellere	○						
2	Apimetrics							○
3	AppAchi						○	
4	Applitoools Eyes				○			
5	Appvance IQ		○	○				
6	Autonomiq				○			
7	DeepCode	○						
8	Diffblue Cover		○					
9	Eggplant AI		○					
10	Functionize		○					
11	Mabl			○	○	○		
12	Magic Pod		○		○			
13	Perfecto							○
14	ReTest			○				
15	SOAtest		○					
16	test.ai		○	○	○			
17	TestComplete			○	○			
18	Testim.io			○		○		

表 -4. AI利用機能のテスト適用分野

4. AIはテスターを超えるのか

test.aiのJason Arbon氏はAIのチュートリアル講師を務めた際、AIには難しいと思われる定性的な品質評価の能力を人間のテスターと比較する実験を行ったそうです。参加していた70人の優秀なテスト技術者にログイン画面を見せ、「このページが信頼できるものかどうか、どのように評価しますか?」という質問をしました。結果は、70人が5分かけて出した回答が2つ、一方訓練されたAIは3つを回答したそうです。AIが70人のテスターを打ち負かした(AI Beat 70 Testers) [5]と騒ぐほどではないかもしれませんが、AIに厳しい問題も訓練データによっては人間を超える可能性を示したと言えそうです。AIがすべてにわたって人間のテスターを凌駕するとは思えませんが、現時点でもある種のテストではAIの方が確実に上回るものがあるようです。前章で紹介したAI利用テストツールの中心を占める画像解析AIを使用したものがそれに当たるのではないのでしょうか。果たしてAI利用テストツールはどこまで進化していくのでしょうか。

今、自動車の世界では自動運転車の開発競争が激しくなっていますが、開発にあたっては自動運転レベルの分類と定義が行われ、業界及び社会での共通認識となっています [6]。2017年10月、11月にApplitoools社のGil Tayar氏、test.ai社のJason Arbon氏が相次いで自動運転レベルにな

ぞらえたテスト自律性レベルをBlogで提示しました [7] [8]。また、Appvance.ai社のKevin Surace氏も2018年10月にBlogで提示しています [9]。なお、単なる自動化ではないので"automation"ではなく"autonomy"や"autonomous" (自律) という用語が使われています。表-5に自動運転レベルと各氏のテスト自律性レベルをまとめました。各氏の定義は少し異なるところがありますので、比較してみると考え方の違いが分かり興味深いです。

レベル5になると人間のテスターに取って代わることになります。まだまだ先のことだと思っていたら、先日(2018年11月) Functionizeが最新機能はGil Tayar氏の定義のレベル5にほぼ到達しているとBlogで表明したり [10]、Appvanceも彼らの定義のレベル5に到達したとアナウンスするなど俄かに現実味を帯びてきました。ただ、FunctionizeのPaul Clauson氏はTayar氏の定義モデルは少し狭く、考慮されていない要素があると指摘していますので、テスト自律性レベル自体の定義の妥当性はまだ議論の余地がありそうです。

いずれにしろ、テスト自律性レベルを定義して、その目標に向かってAI利用テストツールの開発競争が進めばそう遠くない日にAIがテスターを超えるかもしれません。テスターのみなさん、油断は禁物ですよ!

レベル	運転自動化レベル[6]		テスト自律性レベル		
	名称	概略 (各種情報から筆者が要約)	Gil Tayar [7] (Appltools)	Jason Arbon [8] (test.ai)	Kevin Surace [9] (Appvance.ai)
0	運転自動化なし	ドライバーがすべてを操作 Hands On Eyes On	人間はテストを書き、ベースラインの変更を検証。 AIは何もしない。	【手動テスト(無支援)】 人間が物理的に実行する手作業による探索的およびリグレッションテスト	自動化なし
1	運転支援	システムがステアリング操作、加減速のどちらかをサポート Hands On Eyes On	人間はテストを書き、ベースラインの変更を検証。 AIはテストアサーションの記述とページの見映えチェックを支援。	【スクリプトによる自動化(Hands On)】 テストを繰り返し実行でき、アプリケーションの変更に対する自己修復性をもつこともある手作りのテスト自動化スクリプト	スクリプト作成/コーディング
2	部分運転自動化	システムがステアリング操作、加減速のどちらかもサポート Hands Temp Off Eyes Temp Off	人間はテストを書き、ベースラインの変更を検証。 AIは変更をグループ化して変更の検証を支援。	【探索的ボット(Hands Off)】 自動化された準知的探索と人間の介入なしの性能/安定性の測定	コードレスのキャプチャ/プレイバック
3	条件付運転自動化	特定の場所でシステムが全てを操作、緊急時はドライバーが操作 Hands Off Eyes Off	人間はテストを書く。 AIはベースラインの変更を検証。	【人間指示リグレッション(Eyes Off)】 人間はテストケースの高位の意図を記述、自動マシンはアプリケーションに対してリグレッションテストケースの目的をどのように実行するかを自律的に決定	機械学習-人間が作成したスクリプトとモンキーボットの自己修復
4	高度運転自動化	特定の場所でシステムが全てを操作 Hands Off Mind Off	人間はテストを書くのを支援。 AIは人間の指導のもとテストを書く。	【生成的(Mind Off)】 マシンは、探索的テストとリグレッションテストのほとんどを網羅するテストを生成して実行	機械学習 - ほぼ全自動 - スマートなスクリプトの自動生成
5	完全運転自動化	場所の限定なくシステムが全てを操作 Hands Off Driver Off	人間は何もしない。 AIは人間の指導なしにテストを書く。	【完全自律(人間のテスターは任意)】 人間の支援なしで、マシンはアプリケーションの評価と、いつ、どこで、どのようにテストを実行するかを決定ができ、テスト結果に基づいてリリース判定できるように人間(またはマシン)に結果を要約	機械学習 - 完全自動化 - 検証済スマートスクリプトの自動生成 (mind-off)

表 -5. テスト自律性レベル

おわりに

現在のAIブームの盛り上がりを見るとこれからもいろいろなAI利用テストツールやサービスが登場してきそうです。ツールの

導入を検討する際は実際に使用した上での評価が欠かせません。日本でもTRIDENT社のAI自動テストサービスMagic Podが提供されており身近でAI利用テストツールに触れられますし、海外のAI利用テストツールの試行評価結果[12]も報告され始めていますので参考にされるとよいでしょう。

参考文献

- [1] 鶴林 尚靖, 鷲崎 弘直, 鄭 顕志, 機会学習工学研究会キックオフシンポジウム参加報告, コンピュータソフトウェア, 35巻3号, J-STAGE, 2018
https://www.jstage.jst.go.jp/article/jssst/35/3/35_3_96/_article/-char/ja
- [2] Testing in the digital age: AI makes the difference, Sogeti, 2018 <https://www.sogeti.com/explore/books/testing-in-the-digital-age/>
- [3] Valentin Grigoryevskiy, AI for Software Engineering - Industry Landscape (12/Aug/2018)
<https://medium.com/ai-for-software-engineering/ai-for-software-engineering-industry-landscape-12-aug-2018-e8e028628663>
- [4] Joe Colantonio, 8 Innovative AI Test Automation Tools for the Future : The Third Wave, Nov 7, 2017
<https://www.joecolantonio.com/7-innovative-ai-test-automation-tools-future-third-wave/>
- [5] Jason Arbon, AI Beat 70 Testers - and You!, Nov 1, 2018 <https://medium.com/testdotai/ai-beat-70-testers-and-you-c4d14e8f8007>
- [6] 自動車技術会, JASO TP 18004:2018 自動車用運転自動化システムのレベル分類及び定義 <http://www.jsae.or.jp/08std/>
- [7] Gil Tayar, Not Only Cars: The Six Levels of Autonomous Testing, Oct 30, 2017
<https://medium.com/@giltayar/not-only-cars-the-six-levels-of-autonomous-testing-a22611ab1d7f>
- [8] Jason Arbon, Test Autonomy Levels, Nov 21, 2017 <https://blog.appdiff.com/test-autonomy-levels-7de7967d030e>
- [9] Kevin Surace, 5 Levels of AI Testing Autonomy, Oct 24, 2018 <https://www.appvance.ai/5-levels-ai-testing-autonomy>
- [10] Paul Clauson, Levels of Automation in Testing, Applying this model to Functionize, Nov 5, 2018
<https://www.functionize.com/blog/levels-of-automation-in-testing/>
- [11] Appvance Announces Level 5 Autonomy for Software QA, Nov 15, 2018
<https://www.appvance.com/appvance-announces-level-5-autonomy-software-qa>
- [12] 板垣 真太郎, AI for Testing (AIを活用したテスト)のスタートアップ情報共有, システムテスト自動化カンファレンス2018
<https://www.slideshare.net/ssuser5c492c/ai-for-software-testing>

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

デジタルイノベーションが もたらす創造的破壊とは

～ソフトウェア大革命に備えよう～



大原 茂之氏
おおはら しげゆき

現在(株)オプテック代表取締役
会長、工学博士
東海大学名誉教授、九州工業
大学客員教授、一般社団法人
スキルマネジメント協会理事長



はじめに

前編では、新製品や新サービスをマーケットへ提供する側と、それら新製品などを利用する側を一体化したイノベーション・イノベータモデルを示し、イノベーションの拡大について述べました。その上で、未だにアナログ的なソフトウェア開発をデジタル化できなければ、日本の多くの産業が世界の中で遅れをとることを指摘しました。特にAIの導入はソフトウェア開発をベンダー側からドメイン側に大きく移行させるなどソフトウェアの産業構造を大きく変革していきます。

そして、ソフトウェア開発以上にデータとその履歴の収集と活用が大きな価値を生み出します。過去のデータが価値を持つため、機能を開発する技術力が如何に高度であっても勝負できないのです。データの収集が遅くなればなるほど産業競争力は弱体化していくことを認識する必要があります。この後編では、IoT、AI、ビッグデータによるデジタルイノベーションの実際の事例として歯科用電子カルテにおけるAIについてご紹介します。

1. IoTとデジタルイノベーション

1-1. リアル空間、IoT空間、サイバー空間

ここではこれらの空間の一般論は避けて、AIを活用する歯科用電子カルテシステムの観点から俯瞰した場合(図1)について述べます。

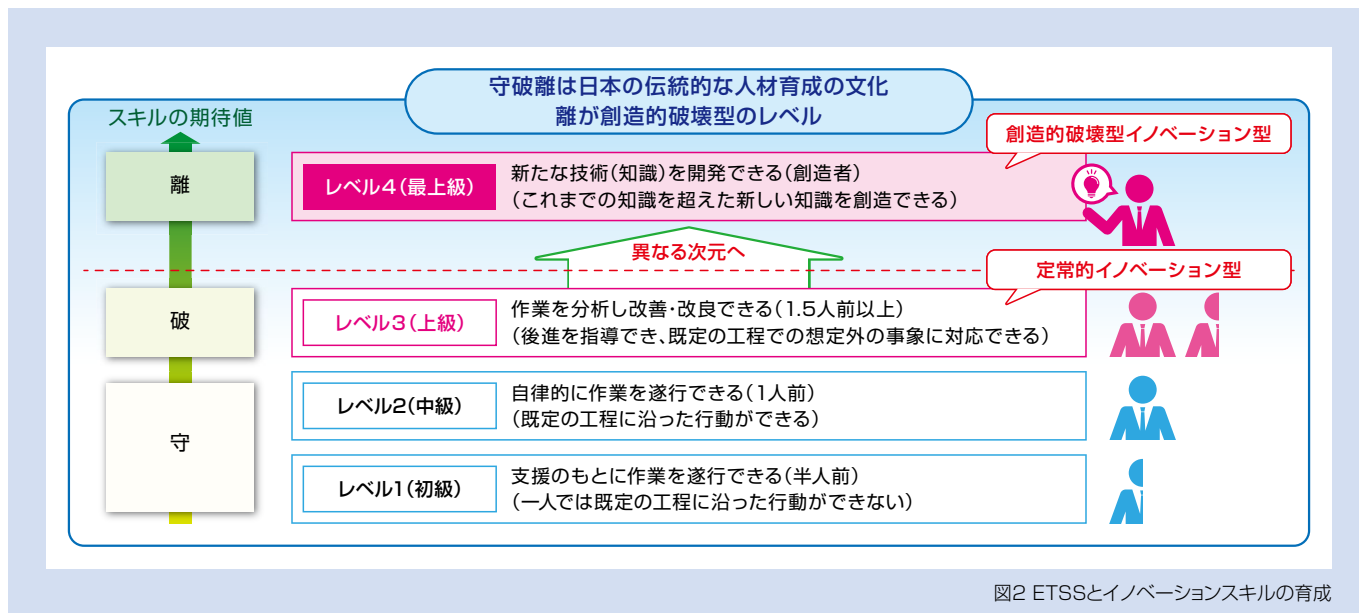
リアル空間では歯科医師が口腔内の治療について電子カルテシステムに患者の所見や治療状況を入力します。さらに、大学での教育活動、スタディー

グループの研究活動、本院・分院間の連携や、患者が治療の予約をするなど、物理的な作業やデータを扱います。

IoT空間はリアル空間との間でカルテ情報などをやりとりします。電子機器などの場合はセンサがデータを取得してIoT空間側へ送り、IoT空間側からモータなどのアクチュエータ(駆動装置)をコントロールするデータが送られてきます。機器等をコントロールする機能が機器を離れてIoT空間側へ移行していくのです。いわば、歯科医師がセンサとアクチュエータの役割を果たし、電子カルテシステムと一体になってエッジと



図1 IoT空間、サイバー空間、リアル空間



なっていると解釈できます。

サイバー空間はIoT空間の一部として位置付けられます。AIやAIの学習データはここに位置付けると分かりやすい構造になります。AIを活用する電子カルテシステムの一部機能はクラウド側に存在することになり、データの収集と一括管理に寄与することになります。このことがシステムの運用強化やコストの削減につながるようになります。

1-2.開発、製造および運用コストの観点

リアル空間だけで機能する製品に対し、一部もしくは大半の機能をIoT空間に移行した製品では、その開発、製造および運用コストは大きく異なってきます。

製造コストにおいては、リアル空間における部品点数の削減、製造工程の簡略化、品質管理の簡素化などコスト削減が可能となります。

運用コストにおいては、IoT空間で提供される機能やストレージといったリソースは、多くのリアル空間のユーザが共有するため、可動率が高くなりユーザ当たりのコストは低下します。

サイバー空間にあるAIは多くのユーザが使うことから当然安価なものになってきます。デジタル化された空間の結びつきは、多くのユーザが保有することなくシェアできるため、1ユーザ当たりのコストを低減するサービスにつながるのです。

2. イノベーションを創造する人材育成と環境

2-1.イノベーションを起こす骨太の人材育成

欧米の人材に教えを受けながら近代化に向けて踏み出した明治時代、その中にフリードリッヒ・ワグネルというドイツ人の技術者がいました。ワグネルは東京工業大学の創設に深く関わったイノベーターでした。ワグネルは西洋の油絵などに傾注し日本の伝統文化を軽視する当時の日本の姿を猛批判しています。

「日本固有の画術を捨てるとは何事か。外国の一芸を真似する流れを止めないと、日本の諸製品の質は低下し、製品は似て非なるものとなり、外国に容易に製法を真似されよう」。

ワグネルは画の上手下手ではなく、日本の画に代表される日本固有の文化がものづくりとその品位向上に大きく貢献していることを説いたのです。ワグネルに学ぶならば、単に「IoTだ」「AIだ」「ビッグデータだ」と無批判にまねるのではなく、日本の文化の観点から人材育成とイノベーションをセットにして新しい時代に臨むべきだと考えるべきでしょう。

2-2.日本の伝統「守破離」に学ぼう

図2はIPAで開発した組込みスキル標準(ETSS)のスキル評価基準を示した

ものです。この評価基準は日本の伝統である守破離をベースにしています。守破離は武道、茶道などさまざまな流派(グループ)に伝わっている考え方です。守の段階は、技の形を覚え、それらの形を使いこなせるように指導します。破の段階は、形の存在を意識することなく身体が勝手に動くまで指導します。守破離の凄いところは離にあります。離はその流派を超えた高い次元の流派を創始できるように指導するのです。すなわち流派の指導者のミッションは、その流派を創造的に超えていくイノベーターの人材育成にあるのです。こうした考えを現代風にアレンジしたものがETSSです。

2-3.組織改革を阻害するマニュアル

イノベーションあるいは組織改革を起こすには組織的な環境が重要なファクタとなります。どのような環境がマイナスとして機能するかについては、米国戦略諜報局(OSS)が敵国をおとしめるように作成した諜報員用のサボタージュマニュアルが衝撃的です。

基本的な考え方は、誰も否定できない理屈や規則を作って無駄な時間を増やすことや、リソースのライフサイクルを縮めることなどです。アレンジした例を以下に示します。

- ①重要な現場の業務があっても、あらかじめ設定した会議やイベントには必ず出席させ、組織的活動を守ることの重要性を認識させる。
- ②例えば前回の会議で決まったことでも、再度検討を促しあるいは質問を繰り返して結論の質を高めていく。
- ③拙速に物事を決めると先々問題が発生することを理解させ、注意深く時間を掛けて判断する習慣を付けさせる。

こうしたサボタージュは、参加者のモチベーションを徐々に削り、イノベーションを阻害していくことにもなります。

3. 歯科診療とAIによるデジタルイノベーション

3-1. AIを活用する場面

ここからは、どのような問題解決にAIを活用しイノベーションに結び付けていくかについて、歯科用電子カルテシステムへの応用事例に基づいてご説明していきましょう。

① 組み合わせ爆発を狙え

組み合わせ爆発^{*1}を起こしている問題領域のプログラムを開発するには、解を導くアルゴリズムが発見されている必要があります。しかし、アルゴリズムが無くても人間が経験知(暗黙知)で解決している

場合は、それらの経験知を丸め込んだ解決事例をAIに学習させることで、人間と同等の解決レベルを目指すことができます。具体的には図3の口腔内の状態で組合せ爆発が起こります。

この図は親知らずを含む大人の歯牙の場所(部位)を示したもので、上下左右で合計32カ所になります。ここで、各部位ごとに正常と虫歯(5段階)の計6通りの状態のいずれかになっていると考えます。すると、口腔内の状態の場合の数は $6^{32}(7.95866111 \times 10^{24})$ 通りになります。実際には一つの部位だけで100通り以上の状態があり、手を付けようがない組合せ爆発になります。さらに治療順序に対する患者からの要望もあり、治療に関しては次の②で述べるように歯科医師の経験知がものをいうことになります。

② 経験知を狙え

組み合わせ爆発ではなくても、感覚、直感あるいはコミュニケーションのニュアンスなど、データにできない対象はプログラム開発に向いていません。こうした場合は、質問と回答の事例を集めてAIに学習させることで、AIがあなたも経験知を持っているかのように機能させることができます。

3-2. マイクロソフトのAzure AI

ここでは、組み合わせ爆発や経験知に囲まれた歯科治療に関わる内容を弊社の

歯科用電子カルテに記録する際に、適切な事例を参照できるようにしたAzure AI(マイクロソフトが提供するAIプラットフォーム)の概要を図4に示します。

大づかみには、Azure AIはURLの集合として認識することもできます。

3-3. 歯科の電子カルテ作成支援用Azure AI

図5に電子カルテに記載する際に関連する臨床データを参照できるように支援するAzure AIの全体の関係を示します。所見から病名の事例を探索するAI、病名に対する治療の事例を探索するAI、口腔内全体の治療部位の治療計画事例を探索するAIまでの流れに対し、逆の方向で探索するAIもあります。これらのAzure AIにはそれぞれクレンジングした学習データを用意しています。

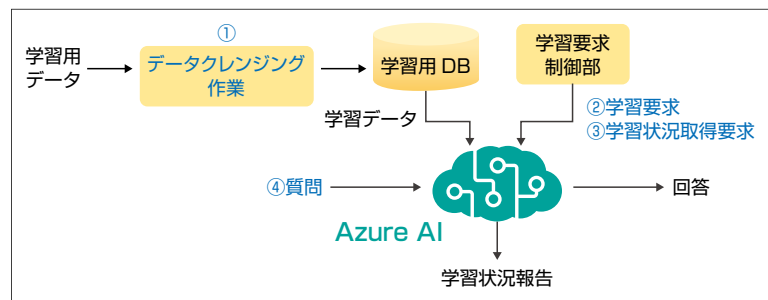
組み合わせ爆発の一方で、歯科医師の経験知がものをいう処理であるため、基本的にプログラミングで解決する対象にはなり得ず、ソフト開発ベンダーが活躍する場面は限定的になります。ちなみに、このAIは機械学習のタイプであり、AIとしては基本的なアーキテクチャになります。

3-4. 実行例

図6.1と図6.2は病名から治療の流れの臨床事例を探索するAIを、弊社の



図3 口腔内の部位



- ①図に示すように、学習させるカルテデータをクレンジングし、学習用DBに登録していきます。クレンジングとしてはカルテデータの訂正、個人情報が含まれないようにするなどを行います。
- ②学習用DBができあがった段階で学習要求を発行します。ここでは約20万人分のカルテデータを学習させています。
- ③学習状況取得要求を発行することでAIの学習状況を把握できます。
- ④学習完了後、質問に対して、確信度を付けた回答を得ることができるようになります。

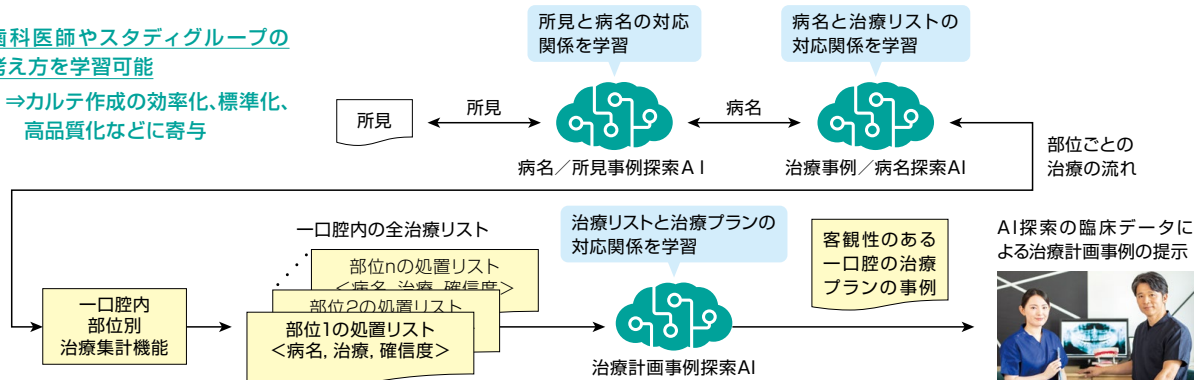
図4 Microsoft Azure AI の概要

*1: 問題の規模に対して解の数が指数関数的に急激に大きくなっていくために、有限時間で解を発見することが困難になることをいう。(Wikipediaより要約)

歯科医師側と患者側の要望に応えるAIソリューション

歯科医師やスタディグループの考え方を学習可能

⇒カルテ作成の効率化、標準化、高品質化などに寄与



*本研究開発は平成28年度補正革新的ものづくり・商業・サービス開発支援補助金 第四次産業革命型による成果です。

図5 電子カルテ作成支援用AI

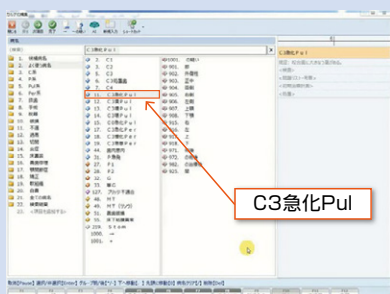


図 6-1 病名からの「治療事例探索 AI」

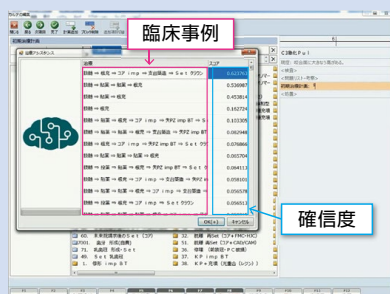


図 6-2 病名からの「治療事例探索 AI」の結果

電子カルテシステムOpt.oneと連動させた結果です。

図6.1は実行前の画面で、病名を選択している画面です。ここではC3急性Pulという病名を選択しています。

図6.2はAzure AIに上記病名を投げた後で返ってきた確信度付きの臨床事例です。

これらのAIは次のような活用シーンに投入していきます。

- ①都道府県別探索支援
- ②スタディーグループ別探索支援
- ③歯科大学向け教育支援ツール
- ④本院一分院間標準化支援
- ⑤代診向け標準化
- ⑥歯科クリニック向け探索支援
- ⑦弊社ユーザ以外でもクラウドから使用可能にする支援

このように歯科医療機関やさまざまな組織の事情に合わせてチューニングできることもAIの強みになります。

おわりに

本稿ではデジタルイノベーションの具体的な事例として歯科医療分野におけるAI開発と活用をご紹介しました。プログラミング技術が大きな曲がり角を迎えていると言えます。これまでプログラミングの対象から外れていた領域が急速に拡大していきます。この加速を持続させようとする、データ収集が壁になってくる可能性があり、この壁をどのように乗り越えるかが大きな課題となります。

Azure AIを用いて本稿のシステムを開発する上で、日本マイクロソフト、クレスコの2社様には多大なご協力をいただきました。この場をお借りして御礼申し上げます。またこのような執筆の機会をいただいたベリサーブ様にも感謝申し上げます。

参考文献

- [1] エベレット・ロジャーズ イノベーションの普及 翔泳社、2007
- [2] 鎌田富久 テクノロジー・スタートアップが未来を創る 東京大学出版会、2017
- [3] クレイトン・クリステンセン イノベーションのジレンマ 翔泳社、2001
- [4] ジェフリー・ムーア キャズム2 翔泳社、2014
- [5] ヨーゼフ・シュンペーター 経済発展の理論(上・下) 岩波文庫、1977
- [6] ヨーゼフ・シュンペーター 資本主義、社会主義、民主主義(I-II) 日経BP社、2016
- [7] 瀧梯三 日本近代美術事件史 東方出版、1993
- [8] 米倉誠一郎 イノベーターたちの日本史 東洋経済新聞社、2017
- [9] IPA/SEC 組込スキル標準 ETSS概説書[2008年版] 翔泳社、2008
- [10] 米国戦略諜報局(OSS) サポーターズマニュアル★諜報活動が照らす組織経営の本質、北大路書房、2015

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



VERISERVE NAVIGATION (ベリサーブナビゲーション) 2019年1月号

編集・発行

株式会社ベリサーブ
東京都新宿区西新宿6-24-1 西新宿三井ビル14F

お問い合わせ

マーケティング部：03-6302-0707
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。