

ベリサーブナビゲーション

Veriserve Navigation

Veriserve
Navigation

【巻頭特集】

トラブル・事故の防止と 日常管理の基本

中央大学 理工学部 ビジネスデータサイエンス学科 教授
中條 武志氏

【特別インタビュー】

ISTQB会長 Klaudia Dussa-Zieger氏に聞く： ソフトウェアテストの過去、現在、 そしてAIがもたらす未来

ISTQB会長
Klaudia Dussa-Zieger氏



私どもベリサーブは、ソフトウェアの第三者検証機関のパイオニアとして40年以上にわたり、お客様が届ける提供価値の品質向上に寄与してまいりました。昨今、技術革新が進み社会が大きく変化する中で、ソフトウェアが必要とされる領域はさらに広がっています。同時に、企業にとって、ソフトウェア開発のスピードと品質の両立を実現することは、ますます重要な課題となっています。

本誌「ベリサーブナビゲーション」は、ソフトウェア品質保証に関わる業界の活性化、日本のソフトウェア品質向上への貢献を目指して、2014年からベリサーブが発行してきた季刊誌です。これまで、数多くの有識者の皆さまに、記事を執筆いただいてまいりました。本号では、内容とデザインを刷新し、ソフトウェアに携わる多くの皆さまに向けて、時代の変化と共に高度化するソフトウェア品質を維持・追求するための情報や知見をお届けします。

ベリサーブは、パーパスとして「品質を創造する力でイノベーションを加速し、未来に続く幸せを実現する。」を掲げています。これまで培ってきたソフトウェア品質創造の実績と信頼を糧にして、お客様へのサービスをより進化させていくため、これからもお客様と未来を共有し、共に考え、走り続けます。引き続き一層のご支援を賜りますよう、よろしくお願い申し上げます。



新堀 義之

代表取締役社長 新堀 義之

Veriserve Navigation

ベリサーブナビゲーション

2025 March Vol.25

Contents

4 巻頭特集

トラブル・事故の防止と 日常管理の基本

中央大学 理工学部 ビジネスデータサイエンス学科 教授
中條 武志氏

14 特別インタビュー

ISTQB会長 Klaudia Dussa-Zieger氏に聞く： ソフトウェアテストの過去、現在、 そしてAIがもたらす未来

ISTQB会長
Klaudia Dussa-Zieger氏

24 特集

「SBOM.JP」を目指す “ソフトウェアサプライチェーン管理”

株式会社ベリサーブ サイバーセキュリティ事業部 脆弱性マネジメント課
藤原 洋平

32 特集

ベリサーブが考える MBTの世界

株式会社ベリサーブ 研究開発部 技術流通課
工藤 邦博



今回の「ベリサーブナビゲーション」の表紙には、新たなステージへの意気込みを込めて「新」の文字を選びました。デザインと内容を進化させ、ソフトウェア品質保証の未来を切り開く新たな視点と知見をお届けします。この新たな一歩が、ソフトウェア業界のさらなる活性化と日本のソフトウェア品質の向上に貢献することを願っています。

トラブル・事故の防止と 日常管理の基本

プロセスにおける変化を見つけるための管理図と、
管理図を有効に働かせるための標準化が一体となって
出来上がってきたのが日常管理である。

人の不適切な行動によるトラブル、事故、不祥事を
防ぐためには、日常管理の徹底が欠かせない。

中央大学 理工学部
ビジネスデータ
サイエンス学科

教授
中條 武志氏
Takeshi Nakajo

プロフィール

1986年東京大学大学院工学系研究科博士課程修了。工学博士。1987年東京大学工学部助手、1991年中央大学管理工学科（現在のビジネスデータサイエンス学科）専任講師を経て、1996年より同教授。開発・製造・サービスにおける人の不適切な行動の防止、TQMおよび小集団改善活動、潜在ニーズの把握に関する研究に従事。日本品質管理学会会員、品質マネジメントシステム国際規格委員会委員、デミング賞審査委員会委員。

01 はじめに

最近、さまざまな分野においてリコール、事故、不祥事の報道を聞くことが多くなった。他方、組織における品質マネジメントの状況について見ると、経営環境の変化を追いかけることに忙しく、顧客・社会のニーズと組織の持つシーズ（技術・リソースなど）を結び付けて価値を作り出すことを基本とし、ニーズやシーズの変化に対応できる、さらには変化をチャンスと捉えて仕事のやり方を変えていくことのできる組織能力を作る大切さが忘れられている様子が見られる。

本稿では、トラブル・不祥事・事故の原因を掘り下げると、ルールを決めて守る、異常（通常と異なる事象）を見つけて職場の全員で共有し、その原因を追究して再発防止を図るという日常管理の基本が徹底できていない場合が多いことに着目し、トラブル・不祥事・事故の防止と日常管理の二つを取り上げ、両者を関連付けて解説する。

02 最近のトラブル、 事故、不祥事

技術が未熟な時には、仕事に関するノウハウ、すなわちプロセスと結果の間の因果関係、プロセスに対して取ることが望ましい対策などに関する十分な知見がないために、問題（結果が狙いから外れる事象）が発生す

る。そこで、データを集めて解析を行うことで、今まで分かっていなかった原因に気付く。原因が分かればその原因が起こらないような、または起こっても大丈夫なような工夫をプロセスに組み込む。このようなことを繰り返した結果、プロセスに関するノウハウの蓄積・活用が進み、それに伴って問題の発生率は下がる。

理論的には、ノウハウの増加に伴って、問題の発生頻度は0に近づいていくはずだが、そうはならず、あるところで下げ止まる。これは、考慮すべきノウハウが増えるにつれて、全てのノウハウを完璧に守って仕事をするのが難しくなるからである。

担当者のちょっとした不適切な行動（既知のノウハウからの逸脱）がさまざまなところで見られるようになり、この状態を放置すると、一つ一つは大したことのない行動が重なって重大なトラブル・事故・不祥事が起こる。これがさまざまな職場・組織で起こっているのが最近の状況である。

03 人の不適切な行動 の4つのタイプ

人の不適切な行動は、発生メカニズムから見ると、表1（表中の局所要因、組織要因については次項04参照）に示す4つのタイプに分けられる。

例えば、必要な工数やプロセスのばらつきを考えないで顧客の注文を引き受ける、障害物があってよく見えないのに目視で点検するなどは、「人間の能力範囲を無視した行動」である。また、資格が必要な仕事を資格のない人が行う、標準が改訂されたことを知らないまま仕事を続けるなどは、「知識・スキル不足の行動」である。さらに、急いでいたため決められた通路を通らずに近道をする、従来と同じだろうと仕様の詳細を確認しないなどは、「意図的な不遵守」である。また、うっかり書類を持ってくるのを忘れる、似た機器を取り違えるなどは、「意図しないエラー」である。

表1 人の不適切な行動のタイプと関連する局所要因・組織要因

人の不適切な行動	局所要因	組織要因
人間の能力範囲を無視した行動	科学的な考え方の不足 ノウハウの不足	●製品・サービス、機器・設備、工程などを設計・計画する人に「仕事を設計する」という考え方がない ●人間の能力範囲に関するノウハウを共有できていない
知識・スキル不足の行動	仕事に必要な知識・スキル > 担当者の知識・スキル	●仕事に必要な知識・スキルが曖昧 ●担当者の知識・スキルが不明確 ●必要な教育・訓練が不足している ●知識・スキルを考慮した仕事の割り当てをしていない
意図的な不遵守	ルールを守る手間・悪影響 > ルールを守る効用 （事実＋認識の偏り）	●ルールの効用や悪影響・手間を考慮していない ●認識の偏りを防ぐ取り組みをしていない ●権限を委譲する場合、問題が発生した場合のことを考えていない
意図しないエラー	注意力の低下× エラーしやすい仕事のやり方	●エラー防止のために仕事のやり方を工夫・改善することが必要という認識が薄い ●仕事のやり方を工夫・改善する方法を教えていない ●仕事に潜むエラーしやすさに気づけていない

04 人の不適切な行動の要因と対策

人の不適切な行動の4つのタイプは、関係する局所要因や組織要因が異なる[1]。

ここで言う局所要因とは、「人の行動に直接影響を与える条件」であり、1) 仕事を行う手順や仕事に関わる資料・情報、2) 仕事で用いる設備・機器・材料、3) 仕事を行う際の環境、4) 本人の注意力・意識・知識・スキル、5) 周りの人の行動などが含まれる（英文の頭文字SHELLを用いて説明されることが多い）。

他方、組織要因は、「局所要因を適切な状態に保つためのマネジメントの状態」であり、a) 経営層によるリーダーシップの発揮、b) 方針管理、日常管理、小集団改善活動、人材育成、新製品・新サービス開発管理、プロセス保証などの仕組みや実践状況、c) こ

れらの基礎となる組織文化などが含まれる。

例えば、意図しないエラーは、担当者の注意力の低下とエラーしやすい仕事のやり方が重なって発生する。ただし、人間である以上、注意力が低下するのはやむを得ないため、エラープルーフ化、すなわち仕事を構成している人以外の要素（手順・資料・情報、設備・機器・材料、環境など）を工夫・改善することで、エラーを起こしにくい、起こしても大きなトラブルや事故にならないようにすることが大切である[2]。ところが、エラー防止のために仕事のやり方を工夫・改善することが必要という認識が薄かったり、仕事のやり方を工夫・改善する方法を教えていなかったり、仕事に潜むエラーしやすさに気付いていなかったりするために、エラープルーフ化が十分行われず、意図しないエラーがさまざまなところで発生してい

る場合が多い。

また、意図的な不遵守は、ルールを守る効用がルールを守る手間・悪影響に比べて小さい、または実際には効用の方が大きいのにそう認識できていないことで発生する。このため、効用を大きくし、手間・悪影響を小さくすると共に、効用や手間・悪影響に関する認識の偏りをなくすることが大切である。ところが、効用や手間・悪影響を評価していなかったり、意識の偏りを防ぐ取り組みをしてなかったりするために、効用と手間・悪影響の不均衡が生じ、意図的な不遵守が発生している場合が多い。

05 日常管理とは

日常管理の原点は、W・A・シューハートによって提案された「管理図」である（図1参照）。管理図では、まず、プロセスの出来栄を評価するための特性（尺度）を決める。次に、この特性の分布を調べ、その結果に基づいて中心線および管理限界を引いたグラフを用意する。その上で、適切な頻度でデータを取り、点をプロットしていく。点が管理限界の外に飛び出したり、点の並び方に癖が見られたりした場合には、プロセスの条件が変化したと考え、応急処置を取るとともに原因を追究し、当該条件の変化が二度と起こらないようにしたり、より良い結果が得られるように条件の設定を変えたりする。このようなことを繰り返

返すことで、安定したプロセスを実現する方法が管理図である。

他方、管理図によってプロセスの条件の変化を見つけるためには、プロセスの出来栄を評価するための特性が一定の分布に従うことが必要になる。このため、結果に大きな影響を与える5M1E（Man、Machine、Material、Method、Measurement、Environment）に関する取り決めを定め、それに従って業務を行う。これによってプロセスにおける条件のばらつきを押さえ込み、結果が一定の分布に従うようにする。このような取り組みは「標準化」と呼ばれる。

さらに、プロセスが変化したということは、標準化がうまくいっていないということなので、異常を発見した後の原因追究や再発防止においては、標準化に関する取り組みのどこが悪かったのかを分析し、取り組みの改善を図る。

このように、プロセスにおける変化を見つけるための管理図と、管理図を有効に働かせるための標準化が一体となって出来上がってきたのが日常管理である。日常管理というと、それぞれの職場や部門が担当している仕事そのものと考えて人が少ないが、日常管理は、仕事に関して既に分かっているノウハウをベースにしながらその不足点を見つけてノウハウの確実なレベルアップを図ることで、より効果的・効率的に仕事を行えるようにする活動である。

06 SDCAサイクル

日常管理を進める場合に基本になる考え方がSDCAサイクルである（図2参照）。これは、標準化（Standardize）、実施（Do）、チェック（Check）、処置（Act）のサイクルを確実かつ継

図1 管理図の概念（出典：JSQC-Std 32-001:2013 日常管理の指針[3]、p.8、図4）

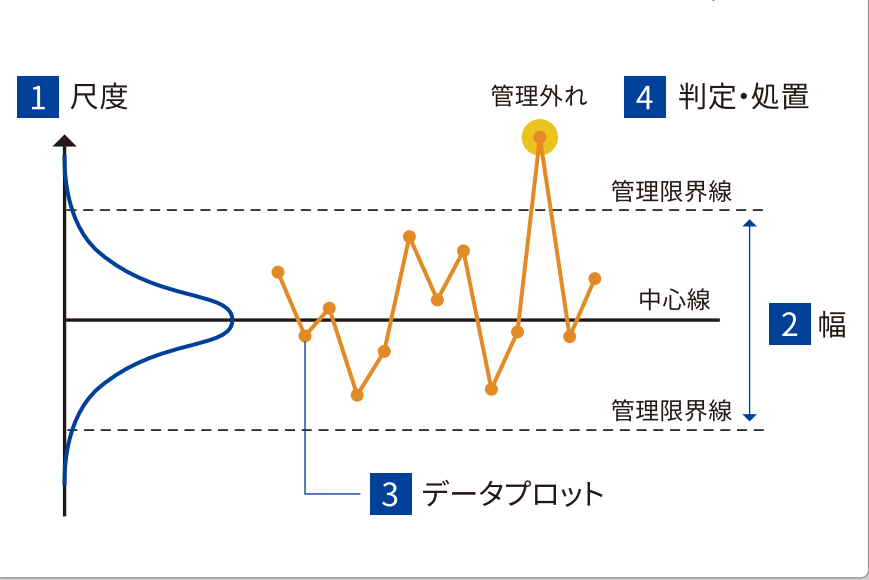
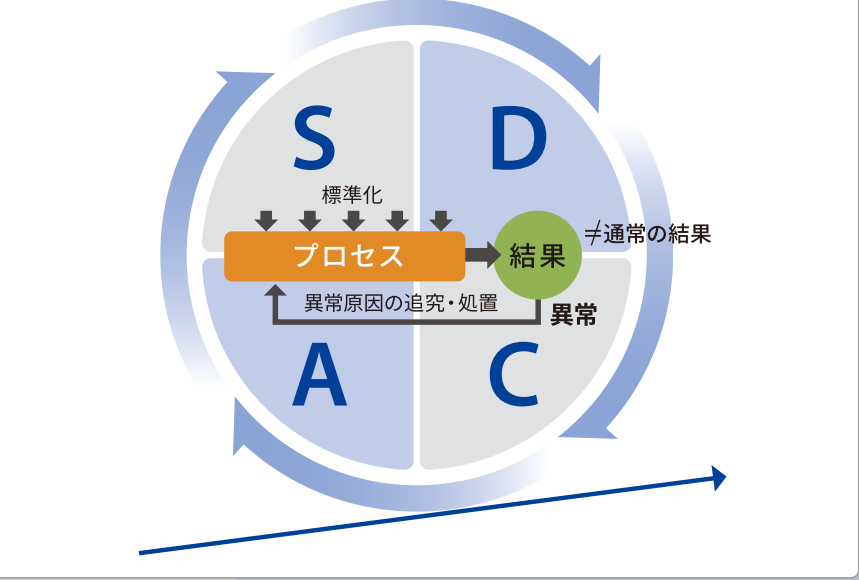


図2 SDCAサイクル



続的に回すことによって安定した結果が確実に得られるようなプロセスを作り上げるという考え方である。

a.標準化 (Standardize) :一定の結果が得られるようにするには、結果に影響を与えるプロセスの条件が一定に保たれるように、これらに関する取り決め(標準)を決めて守られるようにしなければならない。ただし、取り決めを決めるに先だって、仕事の目的、結果に対する要求事項、結果を生み出すプロセス、プロセスと結果の関係に関するノウハウなどを明確にしておくことが必要になる。さらに、必要な教育・訓練を行ったり、守れる工夫をしたりすることも必要になる。

b.実施 (Do) :取り決め通りプロセスを実施する。取り決め通り行えているかどうかを確認し、必要な場合には、教育・訓練や守れる工夫を補強する。

c.チェック (Check) :上記のような努力をしても、決めた内容が不十分、または決めた通りに実施されない場合も多い。従って、異常、すなわちいつもと違う結果に素早く気付き、その原因、すなわち取り決めの不十分さや取り決めを守る仕組みの弱さを見つける。

d.処置 (Act) :チェックの結果を踏まえて、取り決めの内容やそれが確実に守られるようにする仕組みをより良いものにする。

なお、マネジメントを行う方法を包括的に表したものにPDCAサイクル

があるが、SDCAサイクルは、PDCAサイクルの中の計画(Plan)において、目標を現状またはその延長線上に設定すると共に、現状の業務のやり方を組織の取り決め(標準)として定めて活用することで、維持向上(ばらつきの少ない安定した結果が得られるように、日常の仕事のばらつきの中でより良い方法を選択することでさらなる向上を図る活動)を分かりやすく示したものと言える。

07 日常管理のための8ステップ

図3は、SDCAサイクルに沿って日常管理の進め方を示したものである。

ステップ 1 職場の使命・役割の明確化と業務の分析

日常管理を効果的かつ効率的に実践するためには、まず職場の「使命・役割」を明確にする必要がある。ここで言う使命・役割とは、組織が経営目標を達成するに当たって必要となる役目・任務を分解し、各々の職場またはその構成員に割り当てたものである。職場の使命・役割は、一般に、「誰に対して何を提供するのか」という形で表現できる。例えば、販売職場では、顧客の話をよく聞いてニーズを満たす製品・サービスを提供することが使命・役割になるし、開発職場では、顧客のニーズを満たす製品・サービスの設計図を製造職場や営業職場

に提供することが使命・役割になる。

次に、職場の使命・役割を「業務」に展開し、実行可能なレベルまで具体化する。ここで言う業務とは、行う必要のある活動・行為である。業務を具体化することで、各人が持っている、業務に関するノウハウ(経験、知恵、工夫など)を集約し、活用するための枠組みが得られる。業務は、一般に、名詞と動詞を組み合わせることで表現できるため、日常行っていることをこの形に表し、1次機能、2次機能などの階層にまとめることで、その内容を明確にすることができる。

ステップ 2 業務を行うプロセスの明確化

職場で行っている各業務については、それを行う手順を「プロセスフロー」として明確にする必要がある。ここで言うプロセスフローとは、複数のプロセスをつなげて、狙いとする価値を提供できるようにしたものである。一つのプロセスのアウトプットが複数のプロセスのインプットになる場合もあれば、複数のプロセスのアウトプットが一つのプロセスのインプットになる場合もある。

プロセスフローを明確にしたら、次に、これに含まれる一つ一つのプロセスについて、

- a.アウトプット
- b.インプット
- c.アウトプットを生み出すために行うべき活動

- d.担当者
- e.使用する他の資源(設備・機器、エネルギー、ネットワークなど)を特定する。

その上で、

- f.アウトプットに対する要求事項(判定条件)
- g.アウトプットが要求事項(判定条件)を満たすためにインプット、活動、担当者および他の資源が満たすべき条件

を明確にする。このうち、gは「良品条件」と呼ばれ、これを明確にするにはプロセスの因果関係についての知見が必要となる。

職場で行っている各業務についてのプロセスフローとそれに含まれる

各プロセスのa～gをまとめた表は、当該の業務についてのこれまで蓄積されたノウハウを整理したものであり、これが標準のベースとなる。

ステップ 3 標準の制定・改訂・見直し

インプット、活動、担当者および他の資源に関する良品条件やその実現方法、アウトプットに関する判定条件やその確認方法については、文書やマニュアルなどにまとめておく必要がある。「標準」とは要求事項をできるだけ効果的・効率的に実現するための方法を定めたものであり、担当者が代わった場合でも同じ業務が行われ、同じ結果が得られることを確実にする上で役立つ。

図3 日常管理のための8ステップ



ステップ
4 必要な知識・スキルの確保

標準で定めた通り業務が行われるためには、業務を担当する人が標準通り行うのに必要な知識を身に付けておく必要がある。このためには、教える必要のあることを明確にした上で、新人や応援者を含め、どのような時（採用時、職場配属時、問題発生時、標準の改訂時、管理・監督職への昇格時など）にどのような内容をどのような方法で教育するか計画し、必要な人に必要な教育が抜けなく行われる仕組みを確立する必要がある。

また、業務の中には、難しいところやカン・コツの部分など、熟練を要し、習得するのに時間がかかるものもある。これらに対しては、業務ごとに習得すべきスキルおよびそのレベル、それらを定量的に評価する方法を明確にしておく。その上で、業務を担当する一人一人のスキルレベルを評価し、不足している部分について

は、目標を設定した上で計画的に訓練を行い、習熟度が基準を満たした人に限定して業務に従事させる必要がある。一人一人の習熟レベルを明確にする上では、スキルマップなどの手法を活用するのがよい。

ステップ
5 意図的な不遵守と意図しないエラーの防止

人は「標準を守ることによる効用」と「標準を守るための手間・悪影響」をはかりにかけ、前者よりも後者が大きいと標準を守らない（図4）。ただし、標準を守るか守らないかの判断をする人が、標準を守ることによる効用や標準を守る手間・悪影響を必ずしも正しく認識できていないこともある。従って、標準の意図的な不遵守を防ぐためには、なぜそうしなければならないのか、また、守らなかった時の影響について正しく理解できるよう、標準を守らなかったために発生した

トラブルの事例などを用いて教育すると効果的である。また、担当を決めてモニタリングやパトロールなどを行い、標準が守られていない場合には指摘・指導を行い、守られていない状態が放置されないようにするのがよい。さらには、標準を自分で作るだけの能力を身に付けさせ、その作成・改訂に参加してもらうとよい。これらに加え、アンケートなどを用いて日常的に作業担当者からの声を集めて、認識の偏りが生じていないか検討することも大切である。

他方、うっかり忘れる、取り違えるなどの意図しないエラーは、あらゆる人があらゆる作業で起こす可能性があり、注意力、教育・訓練・意識付け、多重チェックなどで完全に防ぐことは難しい。このため、仮想体験などを用いて注意力に頼った取り組みの限界を知り、意図しないエラーを防ぐためにはエラープルーフ化（間違いにくく、間違えても直ぐに分かるようにする工夫）が必要なことを理解する必要がある。

意図しないエラーの発生率は低いいため、発生したエラーだけを対策しても十分な効果が得られない。FMEA(Failure Mode and Effects Analysis、失敗モード影響解析)などのリスクアセスメント手法、ヒヤリハットやインシデントなどの事例を活用し、各人が行っている作業において、まだ起こっていないが起こりそうなエラーを系統的に洗い出し、事前に対策を取っておくのがよい。

また、エラープルーフ化の基本的な考え方（排除、代替化、容易化、異常検出、影響緩和）を、パートナーなどを含め業務を担当する全員が学ぶと共に、過去に行った有効な対策を整理した対策発想チェックリストなどを共有して活用することも大切である。

ステップ
6 管理項目・管理水準の設定と異常の見える化

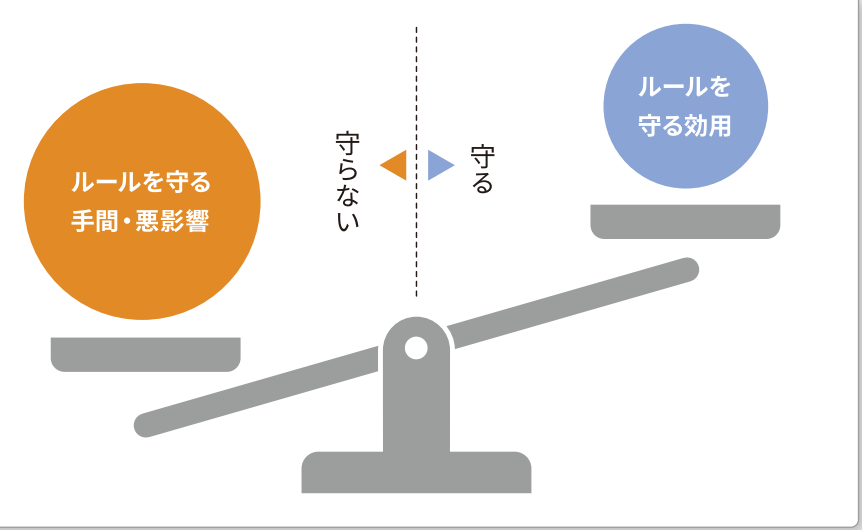
異常はプロセスの時間的な変化なので、いくら要求事項（判定条件）を満たしているかをチェックしても分からない。異常に気付くには、プロセスの状態を表す尺度を選んでその推移をグラフ化することが必要である。この際に用いる尺度が管理項目である。管理項目というとチェック項目と誤解し、「〇〇が正しく行われているか」、「□□の状態になっているか」などをイメージする人がいるが、これでは管理図や管理グラフを書くことはできない。「異常を見つけるためには

推移グラフが必要であり、グラフを書くための尺度が管理項目である」ということを理解してほしい。

管理項目は網羅的に考える必要はない。候補となるものの中から、後工程・顧客にとって重要で、容易に測れ、しかも当該のプロセスの状態をよく反映するものを少数選ぶのが原則である。

管理項目を用いて異常が起きていることを検出するには、管理水準（中心値および管理限界）を設定し、これと得られたデータとを見比べることが役立つ。管理水準を決める際には、現行のプロセスに関するデータを一定期間集め、明らかに異常と思われるデータを取り除いた上で、データの平均値または中央値（メディアン）が中心値になるよう、大半のデータが管理限界の内側に含まれるように管理水準を決めるのが一つの方法である。なお、管理限界は、良い悪いを判断しているわけではないので、望ましくない側だけでなく、望ましい側にも設定する。

図4 意図的な不遵守の発生メカニズム



ステップ 7 異常の検出と共有、応急処置

管理図や管理グラフにおいて、
1) 点が管理限界の外に飛び出る、
2) 中心線の片側に連続して点が出
現する、3) 上昇・下降が続く、4) 管理
限界線への接近、5) 中心化傾向、6)
周期的な変動などが見られた場合に
は、異常が発生したと判定する。

異常を検出した場合には、直ちに
その事実を関係者で共有し、どのよう
なプロセスの条件の変化が異常を引き
起こしたのか特定する必要がある。
このため、毎日決まった時間(朝礼

時、終礼時など)に定例のミーティン
グを開き、管理図や管理グラフの前
に集まって異常の発生の有無を報告
し合い、異常の発生時期や形(単発
型、継続型、傾向型、周期型など)を
手がかりに、プロセスの状況(担当者
の交替、設備・機器の故障など)と照
らし合わせるのが効果的である。

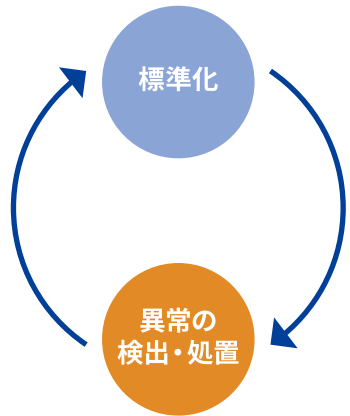
ステップ 8 異常の原因追究・再発防止

異常が発生した場合には、その原
因を追究し、同じまたは類似のプロ
セスの条件の変化が二度と起こらな

いようにする対策を取る必要がある。
再発防止に効果的であることが確認
できた対策は、標準書の改訂、教育・
訓練の見直しなどを行い、継続され
るようにする。

異常を引き起こしているマネジメ
ントのまずさ(組織要因)を追究する
際には、標準化と対応付けて検討す
ることが有効である(図5参照)。この
フローに従って職場で発生している
さまざまな異常を分類することでどの
区分が多いかを明らかにし、その上
で当該の区分に焦点を絞って組織要
因をさらに掘り下げることで職場とし
ての取り組みの弱い部分を明らかに
できる。

図6 日常管理を支える二つの柱

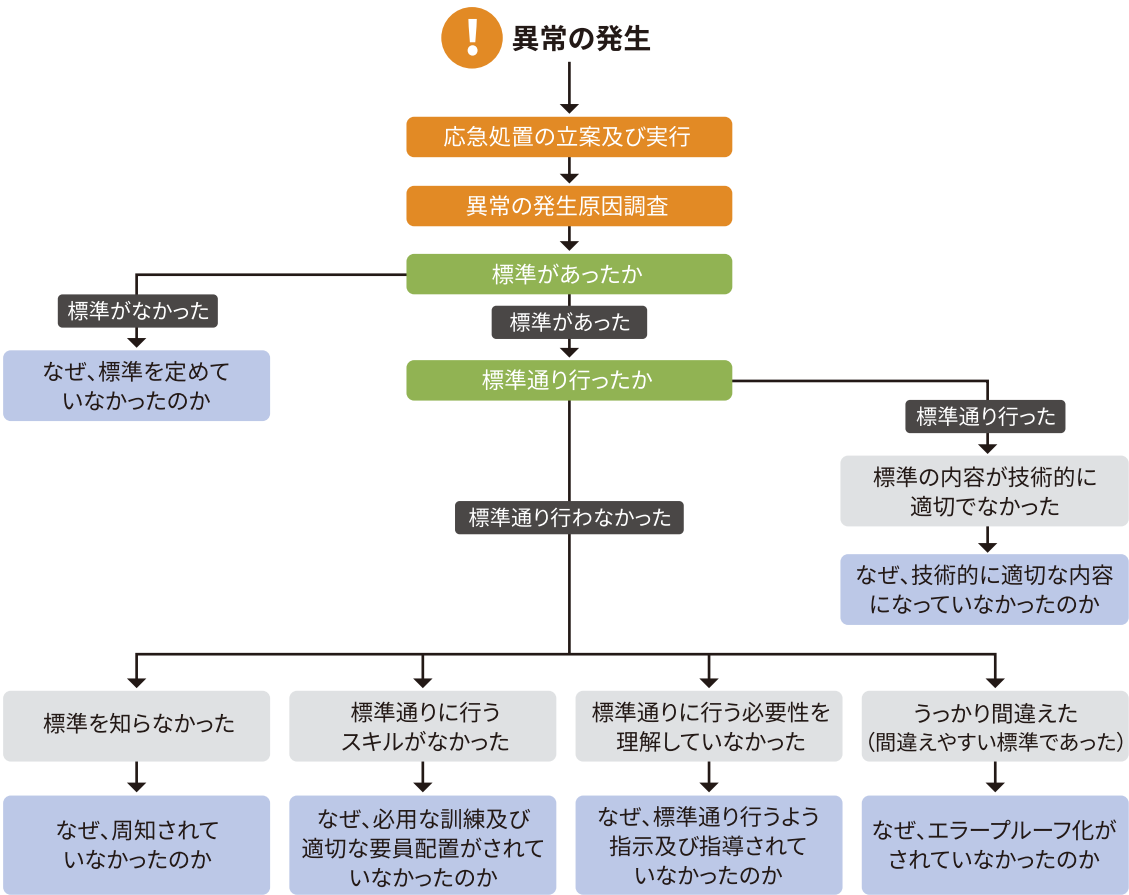


と難しいということである。ただし、
人間は自分で経験しないと納得し
ない。そのため、方針管理や小集団
改善活動などを含めたTQM(Total
Quality Management)を組織的に
推進し、このような行動様式に沿って
行動し成功した体験をした人を増や
していくことが必要である。

参考文献

- [1] 中條武志(2021):「日常管理の基本」、日科技
連出版社。
- [2] 中條武志(2010):「人に起因する事故・トラブル
の未然防止とRCA」、日本規格協会。
- [3] JSQC-Std 32-001: 2013、日常管理の指針。

図5 異常の原因を追究する場合の視点(出典:JSQC-Std 32-001:2013、日常管理の指針[3]、p.24、図13)



08 おわりに

日常管理はやろうと言えばすぐに
できるものではない。また、一度でき
たと思っても、手を抜くとすぐに形骸
化する。実践に当たって大切な点は、
標準化と異常の検出・処置は、日常管
理を動かす両輪であるということだ
る(図6参照)。片方だけ頑張っても
良い成果は得られない。両者を密接
に関連付けて活用できるかどうか
が、人に起因するトラブル・事故・不
祥事を防げるかどうかの分かれ目だ
る。

もう一つ忘れてならないのは、日常
管理に取り組む場合、職場で働く一
人一人に顧客指向、プロセス重視・
PDCAサイクル、全員参加・人間性尊
重などの行動様式が浸透していない

ISTQB会長
Klaudia Dussa-Zieger氏

プロフィール

ソフトウェアテストとソフトウェア品質に20年以上携わり、ISTQB認定トレーナーとして、数多くの専門家を育成。国際的な標準化活動にも積極的に参加し、ソフトウェア開発の未来を牽引する。著書「Software Engineering nach Automotive SPICE」は、欧州の自動車業界における実践的なガイドブックとしてベストセラーになっている。

ソフトウェアテスト業界の重鎮、ISTQB会長 Klaudia Dussa-Zieger氏インタビュー

2024年10月にインドネシアバリ島で開催された「ISTQB General Assembly Meeting」の際にインタビューを行った

ISTQB会長 Klaudia Dussa-Zieger氏に聞く：

ソフトウェアテストの過去、現在、そしてAIがもたらす未来

ソフトウェアテスト業界に20年以上貢献し、現在はISTQB(International Software Testing Qualifications Board:国際ソフトウェアテスト資格認定委員会)の会長を務めるKlaudia Dussa-Zieger氏にインタビューを行いました。今回は、同氏のこれまでのキャリア、ISTQBとの深い関わり、ドイツおよびその他の地域における品質保証コミュニティでの活動、そしてソフトウェアテスト業界の最新トレンドであるアジャイル開発におけるテストの重要性やAIへのアプローチについて詳しく伺いました。

聞き手:大西 建児

JSTQB技術委員会 副委員長/日本代表

株式会社ベリサーブ 広報・マーケティング部 テスティング・エバンジェリスト

キャリアの始まりとimbusとの出会い

—まず、これまでのキャリアと主な業績についてお聞かせください。

私はドイツのエアランゲン大学(Friedrich-Alexander-Universität Erlangen-Nürnberg)でコンピューターサイエンスを学びました。当時はテストや品質保証は、まだ確立された分野ではなく、学生時代にテストの世界に触れる機会もほとんどありませんでした。大学卒業後はアメリカで博士号を取得し、帰国後に就職したのが品質保証会社であるimbus社です。

私がimbus社を選んだ理由は二つで、テストというテーマに興味を持ったこと、そしてもう一つは、当時幼い子供が二人いて、保育園の近くで働きたかったという家庭の事情もありました。

1998年にimbus社に入社しました。まだISTQB設立以前の話です。当時上司だったTilo Linz(imbus社のCEO)とテストについて取り組み始めました。シラバスを開発したわけではありませんが、CTFL(Certified Tester Foundation Level)の最初のトレーニング教材を開発し、多くのクラスやトレーニングを実施しました。これが私のテストとの出会いです。

入社後最初の1カ月は、皆が手動テストを行い、実際に手を動かすことでテストを理解するようにしていました。その後、当時すでに存在していたテスト自動化に取り組みました。Mercury InteractiveのWinRunnerやXRunnerといった、今はもう存在しないツールを使っていました。テスト自動化からテスト管理に移り、テスト管理を通して、テスト担当者や他のコンピューターサイエンティストを教育することの重要性をますます感じるようになりました。人に教えることへの興味と愛情から、教材を作成するだけでなく、大学でもテストを教えるようになりました。

その後、ドイツのソフトウェア品質協会であるASQF (The Arbeitskreis für Software-Qualität und Fortbildung) の活動にも参加し、ソフトウェアテストグループを率いるようになりました。25年以上前を振り返ると、当時は大学でテストについて教えられることはほとんどなく、書籍も少なく、情報を得るのが非常に困難だったのです。

このような状況下で私はテストの世界に入りましたが、それ以来、本当にエキサイティングな経験をしています。自動車、医療、銀行、保険など、さまざまな分野でテストに携わってきました。私は非常に好奇心旺盛な人間なので、いろいろな業界に携われるテストという仕事は私に本当にぴったりだと言えます。



著書

「Software Engineering nach Automotive SPICE」

Automotive SPICEと自動車業界への影響

—著書である『Software Engineering nach Automotive SPICE (以下、Automotive SPICE)』は、ドイツの多くのエンジニアに読まれ、自動車業界におけるソフトウェア開発の品質をさらに向上させるためのガイドとなっています。Automotive SPICEが生まれた背景と、自動車業界で広く採用されている理由について教えてください。

ドイツの自動車産業は非常に強力です。BMW、フォルクスワーゲン、ダイムラー、ポルシェ、アウディなど、多くの自動車メーカーがあります。そして当時、彼らは搭載されるソフトウェアの品質に課題を抱えていました。モーター制御や電子パーキングブレーキなどを制御するソフトウェアです。

25年以上前を振り返ると、ナビゲーションシステムやインフォテインメントシステムが登場する以前から、自動車の機能を制御する最初のソフトウェアが登場し始めていましたが、その品質に問題があったため、自動車分野におけるソフトウェア開発プロセスの成熟度モデルを開発することになりました。

この開発は、ドイツ自動車工業会 (VDA: Verband der Automobilindustrie) が主導するドイツのOEM



(相手先ブランド製造) によるインシアチブでした。彼らは当時ISO規格だったSPICEモデル (ISO/IEC 15504) に着目し、そこからAutomotive SPICEを開発しました。ソフトウェア開発者やソフトウェアサプライヤーは、要件定義からアーキテクチャ、詳細設計、ソフトウェアの実装、そしてテストに至るまでのソフトウェア開発プロセスを、特定のレベルまで満たすことが義務付けられ、評価を受ける必要がありました。

私は約12年間、自動車分野においてさまざまなソフトウェア開発に携わる中で、Automotive SPICEのアセッサー (評価者) として、プロビジュアル

(見習い)、コンピテント (有資格者)、プリンシパル (主任) とステップアップしていきました。そして、培った情報や知識を共有するために、書籍を執筆することになりました。

Automotive SPICEが出版された7、8年後に機能安全規格であるISO 26262の最初のバージョンが出て、安全性の側面がより具体的に規定されるようになりました。そして現在では、サイバーセキュリティ、ISO 21434への対応が非常に重要になっています。Automotive SPICEの最新バージョンでは、機械学習プロセスも含まれ、今後はAIも追加されるでしょう。運転支援の増加、自動運転または支

援付き自動運転の増加に伴い、世界は変化しています。そのため、評価するプロセスを強化し、改善していく必要があります。

ISTQBとの関わりと今後のビジョン

一次にISTQBとの関わりについて伺います。ご自身もISTQB設立当初から認定資格を取得し、15年以上も上級レベルのトレーナーを務めていらっしゃいますが、ISTQBの認定はソフトウェアテスト業界にどのような影響を与えているとお考えですか？

ISTQBが設立された当時、大学でもテストについて学ぶ機会はなく、ソフトウェアテスターという職業を選び、キャリアを積んでいけるようにカリキュラムを確立することは非常に重要でした。

私が活動を始めた頃は、CTFLが必須条件として多くの求人情報に記載されていました。Advanced Level、Foundation Level、またはCertified Tester Automotive Software Tester (CT-AuT)の資格を持っていれば、コンサルタントとしてその分野の専門性を証明することもできるでしょう。

ISTQBが設立されて22年がたち、ISTQBの認定資格が広く認知普及されていると実感しています。求人情報の要件として掲げられているだけでなく、政府など公的機関のRFP (Request for Proposal)でも常にCTFLまたは同等の資格、セキュリティなど、トピックに応じて記載されています。このようなことから、ISTQBやその認定資格は非常に重要な役割を担っていると考えます。

一特にドイツではISTQBの制度が普及しているといわれていますが、ISTQBの認定資格を持っていないことが、就職において不利になることもあるのでしょうか？

はい。認定資格があれば採用されやすいと言えるでしょう。そして、もし資格がなければ実務でより多くを証

明しなければなりません。

ただし、認定資格があることは運転免許証のようなもので、始めに過ぎません。認定資格に匹敵する実力があることを実務で証明する必要があります。

一ISTQB会長として、現在取り組まれていること、今後のビジョンについてお聞かせください。

会長職の期間中に達成したいのは、ISTQBがソフトウェア品質保証とテスト業界において、依然として非常に魅力的で重要な枠組みであり続けるようにすることです。業界の動向を常に把握し、業界を本当に動かすトピックについて議論していく必要があると考えています。

例えば、AI、サイバーセキュリティ、テスト自動化などは重要なトピックスです。私たちは技術の発展に後れを取らず、皆さんがこれらの分野で資格を取得できるようにする必要があります。

加えて、取り組みたいことが二つあります。一つ目は、ISTQBの存続と重要性を維持していくことです。ISTQBは設立から22年がたちますが、この先私たちの理念を引き継ぐ新しい人材を確保し、若い世代が私たちの熱意を受け継いで発展させてくれることが重要です。

二つ目は、私が個人として取り組みたい課題として、若い女性がITやテスト分野に進むことを奨励したい

と思っています。ISTQBには女性向けのプログラム(WIT: Women In Testing)もあります。

ヨーロッパでは、業種や業界にもありますが、テスト分野における女性の割合はおそらく全体の30%から40%くらいです。男性と同数ではありませんが、女性の数は増えているという印象です。

ただ、自動車分野に限って言えばその割合は小さく男性中心で、テストだけではなくソフトウェア開発に携わる女性も非常に少ない状況なのです。

一方で、医療分野、銀行や保険分野などでは女性の割合が一定数を占めているので、業種によって状況が異なっていることも事実です。

一次は視点を変え、新しいトレンド、例えば「アジャイル開発におけるテストの重要性と課題について」どのようにお考えですか？

アジャイル開発は、アジャイルマニフェストが作成された頃から登場したのですが、私が大学で学んだのはそれ以前のことでした。アジャイル開発はますます普及しており、特に保険、銀行、eコマースなど、ソフトウェアのみを扱う分野で非常に多く、そしてうまく活用されています。

一方、ハードウェアコンポーネントを伴う分野、例えばマザーボードやロボットなどを開発する場合は、アジャイル方式で進めるのは非常に難しいと思います。なぜなら、プロトタイプを

簡単に破棄するわけにはいかず、改良を重ねていくため、ハードウェア開発はV字モデルのような手法を踏襲し、ハイブリッドモデルと呼ばれるものになることが多いからです。自動車分野では、企業によってはAサンプル、Bサンプル、Cサンプルと呼ばれるトリプルVモデルがあります。

※自動車向け機能安全標準であるISO26262に記載されているV字モデル

ラピッドプロトタイピングなどの手法もあり、ソフトウェア開発においてはアジャイル開発がますます重要視されるでしょう。また、ハードウェアやシステム全体の開発にもアジャイルを取り入れ始めている企業もありますが、多くの課題があるのも事実です。

ソフトウェア開発単体であれば



問題ありませんが、システム全体を開発している場合、システムテストはソフトウェアだけを対象とするものではなく、デバイス全体を対象とします。

例えば、自動車分野で言えば、特定のHILS (Hardware-in-the-Loop Simulation: ハードウェアインザループシミュレーション) システムが納品されるまで8〜9カ月待たなければならない場合、4週間ごとにイテレーションを行うことは現実的ではありません。

現在、自動車業界では、MILS (Model-in-the-Loop Simulation) などが活用され、特に車載ソフトウェア開発の分野では、すでにアジャイル開発の手法が導入され、システムモデルも、シミュレーションや

デジタルツインの活用が進んでいます。ソフトウェアテストにおいても方向性としては同様に進んでいます。が、まだまだ課題が多い状況だと考えています。

一課題も多い状況であるとのことですが、アジャイル開発においてテストエンジニアはどのような役割を果たすべきだとお考えですか？

テストは常に必要です。私たちは品質の門番であり、全てが正常に機能していることを保証する役割を担っています。特に自動車や他のセーフティ・クリティカルなシステムでは、システムが正常に動作するかどうか、人命に関わるため、その重要性は計

り知れません。eコマースシステムが機能しなければ、顧客や収益を失う可能性があります。自動車、宇宙船、医療機器が正しく機能しなければ、人命が失われる可能性もあり、それは許されることはありません。

AIがソフトウェアテストに与える影響

一AIについてお話が出ましたが、AIがソフトウェアテスト与える影響についてどのようにお考えですか？

AIはソフトウェアテストの方法を大きく変えるでしょう。AI自体は決して新しいものではなく、70〜80年前

から存在していますが、安価な処理能力の向上、ニューラルネットワークのネットワークアーキテクチャの革新、ネットワークをトレーニングするためのデジタル電子データの可用性によって、ますます注目を集めています。

特にChatGPTに代表される生成AIの最新の発展により、テスト仕様の作成支援、オートコンプリートや修正によるテストケースのスクリプト作成支援、テストデータ生成の支援、テストレポート作成の支援など、テストにおける活用が大きく増加しています。

AIをソフトウェアテストに活用することで、効率を大幅に向上させることができます。しかし、生成AIは「ハルシネーション (AIが事実に基づかないもっともらしい嘘を生成してしまう現象のこと)」を起こす可能性があるため、100%信用することはできませんし、常に人間の目で確認する必要があります。

要件、テストケース、最終コードのレビューが必須であるセーフティ・クリティカルな領域では、AIは非常に有効に機能します。いずれにせよ人間による適切なレビューは必須ですが、生成AIツールのサ

ポートによって作業が20〜30%速くなり、大きな差が生まれます。

コンサルティングにおけるAIの活用事例

一コンサルティングでもChatGPTなどのAIを使用しているのですか？

はい、1〜2年前にChatGPTを使用しました。現在進行中の医療機器メーカーとのプロジェクトでは、次世代医療機器の開発を計画していて、最大4〜5,000件のテストケースが想定されています。医療機器はFDA (米国食品医薬品局) の規制対象で、長年製品を販売しているので、これまでの経験から「テスト仕様の作成には、最初の記述から承認まで5日間かかる」という実績データを持っています。

そこで、生成AIを活用することを検討しましたが、機密性の高い情報をChatGPTに入力することは許されていませんので、オープンソースの大規模言語モデル (LLM) を採用し、独自に学習させました。RAG (Retrieval Augmented Generation: 検索拡張生成) のアプローチを採用し、彼らのテスト仕様、設計仕様、アーキテクチャ仕様、要件を使用して、テスト仕様の最初のバージョンを作成しました。

実験段階のツールではありますが、実際にテスターが使用しています。



従来のプロセスでは一つのテストケースに5日間かかっていたところを、20%短縮できれば、かなりの工数が削減できます。ビジネスとして大きなメリットがあると判断され、十分な資金を投入されることになりました。

また、開発期間の大幅短縮だけでなく、機密情報保護の観点からオープンソースのLLMをカスタマイズして使用する点にも注目すべきでしょう。

AI導入によるソフトウェアテストの仕事の変化

—AIなどの新しい技術や手法の導入により、ソフトウェアテストの仕事はどのように変化していくのでしょうか？

さらに多くのソフトウェアテスターがAIを試し、積極的に用いることで、支援ツールとして使用するようになるでしょう。例えば、Robot Frameworkなどのツールを使用してコーディングを行う際に、Copilotなどを活用する例が増えています。

テスト環境の分野では、Infrastructure as Code (IaC) を使用してテストスクリプトを開発する際に、ChatGPTで最初のスクリプトを作成し、修正・実行することで時間を節約する人もいます。

しかし、AIを使用する際にはハルシネーションに注意する必要があります。

ます。生成AIのような新しい技術を使用する場合は、その限界を知る必要があります。何ができて何ができないのか、ハルシネーションを起こす可能性があること、適切な出力はトレーニングデータの質に依存することなどを理解する必要があります。そのため、これまでにはなかったトレーニングを通じた品質保証の取り組みを強化する必要があります。また、AIは確率的なシステムであり、コーディングルールを論理的に理解しているわけではないということも理解する必要があります。

自動化の課題と今後のソフトウェアテストエンジニアの役割

—一次に自動化についてお聞かせください。特にCI/CDやDevOpsにおける課題は何ですか？

CI/CDやDevOpsを導入するには、テスト対象の把握、スクリプトの用意など、自動的にトリガーできるような十分に確立されたプロセスが必要です。パイプライン全体の構築には労力と作業が必要であり、関係者は最初から計画し、サポートする必要があります。しかし、自動実行、特にソフトウェアコードをチェックインする前に行う自動実行は非常に役立ち、プロセスを大幅にスピードアップできます。

一方で、CI/CDが普及しても、ソフ

トウェアテストエンジニアの仕事は変わらないでしょう。テスト自動化が登場した20～25年前にも同じ議論がありましたが、人々はシステム全体を構築し、何をしているのかを理解する人材、スクリプトを作成する人材、CI/CDパイプラインやテスト自動化スクリプトを設定し、システム全体を維持する人材を必要とします。

CI/CDのテスト自動化は、手動テストから自動テスト、開発パイプラインでのテストへと移行するでしょうが、仕事はたくさんあります。AIについても全く同じことが言えます。効率は向上しますが、生成AIが生成するコードが優れているかどうかを判断するためには、人間の知性が必要です。

活動領域をモデルやデータの初期品質保証に拡大する必要もあります。手作業でのレポート作成からAIがサポートする体制へと変わりますので、仕事の種類は変わりますが、まだ仕事は十分過ぎるほどあるのです。

若い世代へのメッセージ

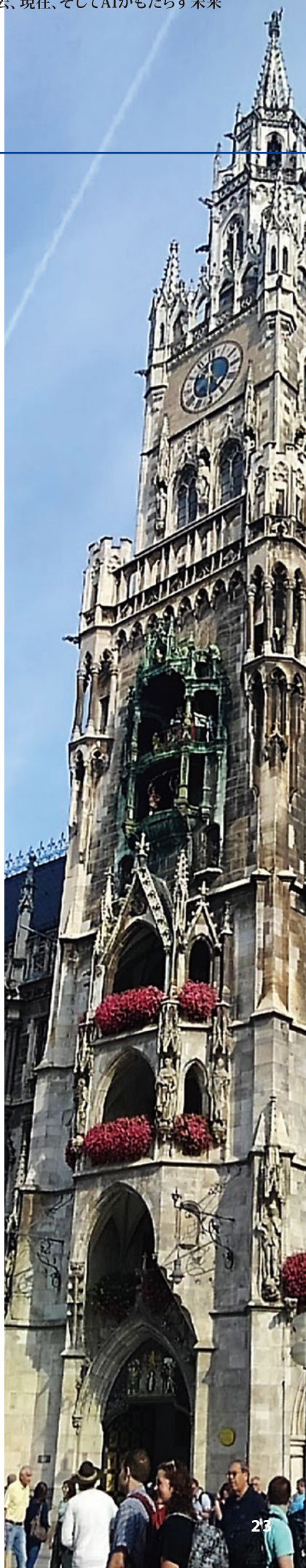
—最後に、ソフトウェアテスト業界で働く若い世代へのメッセージをお願いします。

好奇心を持ち続け、オープンな姿勢で新しい技術を試し、開発を追い続けてください。しかし、AIが生成するものが全て高品質であるとは限ら



ないため、常に警戒心を怠らないようにしてください。私たちは技術変革の時代を生きており、非常にエキサイティングな時代であり、多くの優秀な人材が活躍できる場がたくさんあります。積極的に前へ進んでほしいと思います。

—素晴らしいメッセージをありがとうございました。



「SBOM.JP」が目指す “ソフトウェア サプライチェーン管理”

「自社製品の脆弱性管理をする際に、どうもSBOMというものが有効らしい」
「サイバーセキュリティに関する法規対応のためにSBOMを作成しなければならないらしい」
最近、こんな会話を聞くことが増えてきました。

SBOMとは何なのか、SBOMを導入すると何がうれしいのか—。

本記事では、SBOMの定義から、SBOM対応が求められる理由、
SBOM導入がもたらすメリットについて言及し、
実例としてベリサーブが開発した
ソフトウェアサプライチェーン管理パッケージ
「SBOM.JP」をご紹介します。



藤原 洋平 Yohei Fujiwara

株式会社ベリサーブ サイバーセキュリティ事業部 脆弱性マネジメント課

2010年からベンチャー系開発企業で組み込み系ブラウザエンジンの品質保証業務に従事。2016年、株式会社ベリサーブ入社。OSSのリスク管理サービスの他、セキュリティに関するコンサルティングや脅威分析、脆弱性診断に携わる。2021年ごろからは、自動車のサイバーセキュリティに軸を置き活動中。

01 SBOMを取り巻く 環境

01-1 SBOMの定義

「SBOMとは何か」—
この問いに教科書通りに答えるならば、次の通りとなります。

SBOM-Software Bill Of Materials
とは、ソフトウェアコンポーネントや
それらの依存関係の情報も含めた、
機械処理可能なインベントリ

これは、米国商務省電気通信情報局 (NTIA) が発行するSBOM at a Glanceというドキュメントでの定義です。ただこの言葉だけでは「SBOMとは何か」を想像することは難しいかもしれませんから、もう少し噛み砕いて説明していきます。

上記の定義ではまず「ソフトウェアコンポーネント」という言葉が出てき

ます。このコンポーネントとは何かというと、あるソフトウェアを構成する「部品」の一つと捉えてください。では次に「SBOMがあると何がうれしいのか」について説明します。

SBOMを活用するメリットは複数あります。「ソフトウェアに含まれるライセンスを管理しやすくするため」「ソフトウェアに影響する脆弱性をいち早く特定し対応するため」「構成管理を行う際の補助として、ソフトウェアの不具合管理を行うため」などです。特に、ライセンス管理と脆弱性管理に有用といわれており、本記事ではこの二つをもう少し掘り下げます。

一つ目のライセンス管理についてです。SBOMとはソフトウェアの部品の一覧であると述べました。その部品ごとの情報を記載する際、部品にライセンスが付随していればライセンス情報もSBOMに併せて記載します。ライセンスとは、商用ソフトウェアの使用許諾に関するライセンス、OSS (Open Source Software) のOSSライセンス、特許権などの知的財産に関するライセンスなどが挙げられます。昨今のソフトウェアは大規模化、複雑

化の一途をたどっており、第三者が開発したソフトウェアの部品を自社のソフトウェアに組み込むことは一般的です。そうした第三者による著作物の利用には通常ライセンスが付随しており、そのライセンスに従って利用しなければコンプライアンス違反となってしまう可能性があります。SBOMがあれば、そのソフトウェアに含まれているライセンスを一目で確認でき、対応漏れなどに気付きやすくなるメリットがあります。

二つ目に挙げられるメリットは脆弱性管理です。特にOSSを利用してある場合に、より効果を発揮します。脆弱性については、開発者自身も気付かないうちにソフトウェアのソースコードに欠陥が埋め込まれ、ある日突然顕在化するパターンが多くあります。公知された脆弱性を放置していると悪意のある攻撃者がその脆弱性を悪用し、製品やサービスに攻撃をしてくる可能性があるため、脆弱性が公知され次第いち早く対応する必要があります。ここでも効力を発揮するのがSBOMです。ライセンス管理と同じように、ソフトウェアに含まれ



ている部品が一覧化されているため、そのソフトウェアに影響のある脆弱性の有無の特定にかかるコストを低減させることができます。

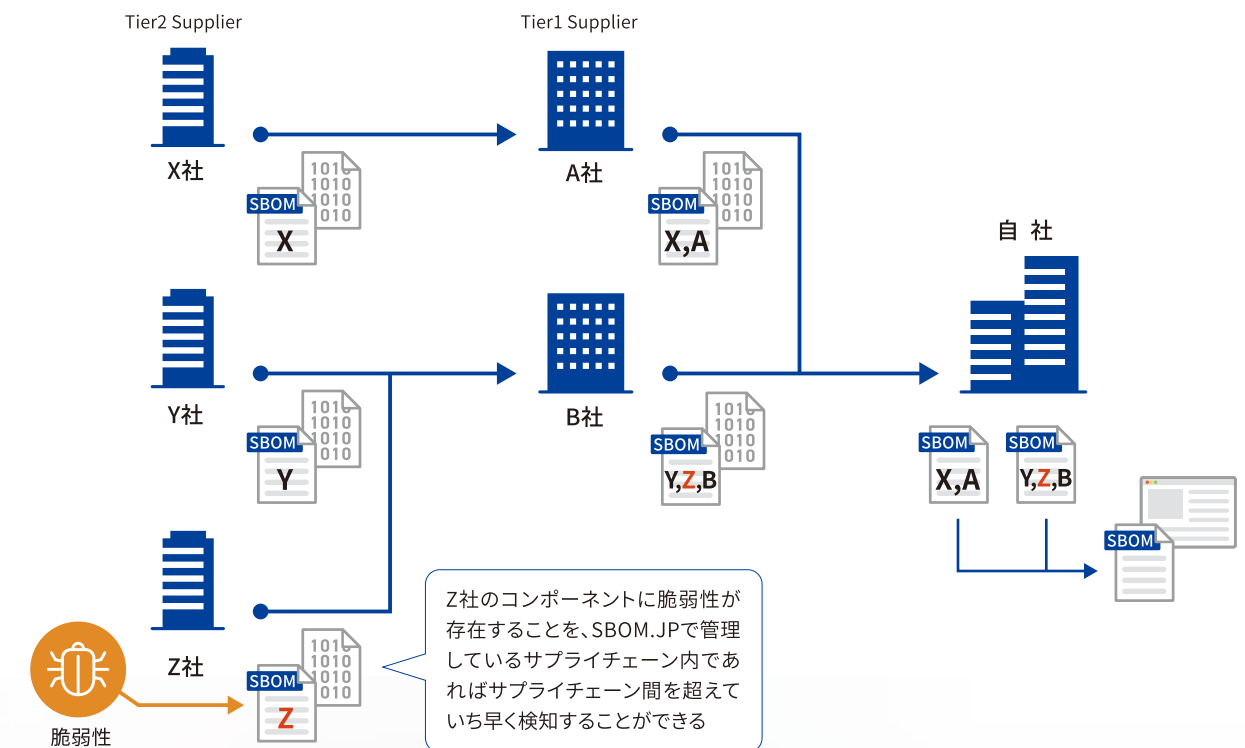
他にも重要なメリットがあります。それは、ソフトウェアサプライチェーン全体でライセンス管理や脆弱性管理の質の向上やコストの削減が期待できることです。自動車を例にします。自動車は多くの部品で構成されます。タイヤを作る会社があり、エンジンを
作る会社があり、エンジンを
作るためのネジを作る会社があり、そして、それらの部品を組み立てる会社があります。これら階層構造のことをサプライチェーンと呼びますが、ソフトウェアでも同様の考え方が成り立ちます。ソフトウェア部品を開発し供給する会社と、それを受け取り自社ソフトウェアに組み込んで開発する会社が存在します。会社間の契約によりますが、ソフトウェアをソースコードではなくバイナリ形式で受け渡することもよくあります。バイナリ形式の場合、ソフトウェアを受け取る側はそのソフトウェアにどのような部品が使われて

いるのかを知るすべがありません。このため、ある脆弱性が発見された時、その脆弱性が受け取ったソフトウェアに影響するかどうかを知るには、そのソフトウェアを開発した会社に問い合わせなくてはなりません。こういった時にSBOMがあれば、また、SBOMをバイナリファイルと一緒に受け渡しできれば、受け取ったソフトウェアにどのような部品が使われているのかを把握できます。つまりSBOMを活用して、サプライチェーン全体でライセンス管理や脆弱性管理が可能になるのです(図表1)。

01-2 SBOMが広く知れ渡った瞬間～今日までの変遷

SBOMと言う言葉が世間一般に広く浸透したきっかけは、2021年に米国で発令された「国家のサイバーセキュリティ改善に関する大統領令」でしょう。当時米国ではSolarWinds社の独自ネットワーク監視ソフトウェアがハッキングされ情報が流出したり、大手石油パイプラインColonial

図表1 サプライチェーン全体でのSBOM活用



Pipeline社がランサムウェア攻撃を受けたりと、サイバー攻撃による被害が相次いで発生していました。そういった背景もあり、大統領令が発令されたといわれています。同大統領令にはサイバーセキュリティに関するいくつかの要求が記されていますが、そのうちの一つに「連邦政府が調達するソフトウェアは、そのソフトウェアの構成要素に関する詳細の開示が必要」との要件が含まれていました。ここでSBOMという言葉が使われ、浸透していくきっかけとなりました。

大統領令をきっかけに米国では NTIAやCISA、NISTというような政府機関から、SBOMに関するガイドラインやガイダンスが次々と発行される

こととなりました。そしてSBOMという概念はやがて、サイバーセキュリティの法規にも取り込まれていくこととなります。筆者が、一番インパクトが大きいと感じているものは欧州で施行されるサイバーレジリエンス法です。サイバーレジリエンス法は「デジタル要素を含む全ての製品」が対象であり、製造者に対してサイバーセキュリティの対応を課す法案です。日本の製造業にも影響があり、欧州に輸出するデジタル要素を含む製品はサイバーレジリエンス法の要求を満たす必要があります。違反した場合は罰則金を支払う要件も含まれており、厳しいものとなっています。この法が求める要求の一つに、SBOMの作成が

含まれています。また脆弱性が発見されたら速やかに当局へ報告し対応する義務が規定されています。

日本国内でもSBOM対応を必須とする流れは広まりつつあります。医療業界では薬機法が改正され、2024年4月からサイバーセキュリティ対応が必須となりました。このサイバーセキュリティ対応の一つとして、SBOMの作成と提供が義務付けられるようになりました。自動車業界では、国連の法規であるUN-R155やUN-R156の施行がすでに始まっています。UN-R155/UN-R156ではSBOMの作成そのものは要求に含まれていませんが、サプライチェーン全体で脆弱性管理に取り組まなくてはならない

図表2 SBOMに関連する法規や各国の状況(主要なものを掲載)

法規 / 標準 など	国/地域 <業界>	内 容
EO 14028	米 国	国家のサイバーセキュリティ向上に関する大統領令。連邦政府が調達するソフトウェアのSBOM提供に関する要求あり
OMB覚書M-22-18 OMB覚書M-23-16	米 国	連邦政府が調達するソフトウェアのベンダーがSSDF実装の自己適合証明を取得する要求あり
Cyber Resilience Act	欧州	欧州で販売されるデジタル製品の製造業者に対するサイバーセキュリティ要求。SBOMの作成、更新、提供の要求あり
ISM-1730	豪	SBOMの作成、提供の要求あり
医療機器のサイバーセキュリティ導入に関する手引書	日 本 <医療>	医療機器の製造業者は、薬事承認を得るためにサイバーセキュリティの対応状況の報告が義務化。SBOMに関する具体的な記載あり
UN-R155 UN-R156	国 連 <自動車業界>	自動車産業におけるサイバーセキュリティマネジメントシステムおよびソフトウェアアップデートに関する国連法規。SBOMに関する直接の要求はないが、サプライチェーン全体での脆弱性管理について要求あり

旨が規定されており、自動車産業の各社は実質的にSBOM作成対応が求められています(図表2)。

現在の状況と今後の流れを鑑みるとSBOMの作成や活用の流れは今後加速し、あらゆるところで求められることが想定されるのではないかと考えています。

01-3 SBOM導入がもたらす利益と投資対効果

SBOMを活用すること自体のメリットは先に説明しました。ここでは、これからSBOM対応を考えている方を対

象に、ソフトウェア開発においてSBOMがどのような利益をもたらすのか、また投資対効果などについてお伝えします。

SBOMが適切に作成され管理・運用されている現場では、間違いなくライセンス管理や脆弱性管理コストが部分的に下がります。ポイントは、部分的に下がるのであって、SBOMがあればライセンス管理や脆弱性管理の手間が全てなくなるわけではないということです。SBOMはソフトウェア部品の把握や脆弱性の特定のためのインプットとして役立ちます。ただし、脆弱性が見つかったなら、そこから先の

対応は別の仕組みや取り組みが必要になります。

他にも、SBOMを新しく導入するには、導入に関わるイニシャルコストと運用に関わるランニングコストが発生することを認識しなければなりません。導入においてはプロセスの策定や、会社によっては組織体制の構築が必要となるでしょう。当然ながらSBOMを作成・運用するためのツールの導入も必須となります。大規模なソフトウェアには大量の部品が使われていることがあり、手動でのSBOMの作成や運用は現実的ではないからです。コスト削減の観点からも、ツールを導入し自動化して運用することがNTIAのガイドラインなどでも推奨されています。

ツールには有償ツール・無償ツール、SBOMの作成に特化したツール、管理に特化したツールなど、用途に応じたツールが存在します。本記事では各種ツールのご紹介は割愛しますが、自社の開発環境や既存のプロセスに即したツールを選択することが大切です。有償のツールを利用する場合はライセンスのツール費用に

加え、ツール利用を前提としたプロセスの構築や自動化、利用者の教育といったコストが発生することに注意してください。

02 SBOM.JPが目指すソフトウェアサプライチェーン管理

02-1 SBOM.JP開発のきっかけ

ベリサーブは、SBOMの管理・運用に特化したツール「SBOM.JP」を自社で開発しました。これまでベリサーブが培ってきた、ソフトウェア管理や脆弱性管理のノウハウが、この「SBOM.JP」の設計思想に継承されています。

ベリサーブでは2008年頃からソフトウェア部品の管理とライセンス・脆弱性管理のソリューションをお客様へ提供してきました。当時はまだSBOMという言葉はありませんでしたが、活動してきた内容はSBOMを活用

したリスク管理そのものです。ただコンサルティングをするだけでなく、実際に現場で手を動かして実務をこなしてきた経験があり、数にすると年間300以上のプロダクトに対してSBOM作成に取り組んだ実績があります。

02-2 SBOM.JPによる課題解決

「SBOM.JP」以外にも複数のSBOM生成ツールやSBOM管理ツールが各社からリリースされています。それぞれのツールにはそれぞれの特徴があり、自社の開発環境やプロセスに合ったツールを選択されることでしょう。本稿ではこういったツールの実例として「SBOM.JP」の特徴を紹介します。

第一の特徴として、ソフトウェアサプライチェーン全体を意識した設計思想が挙げられます。SBOMはソフトウェアサプライチェーンの階層構造全体でもライセンス管理や脆弱性管理の役に立つことを説明しました。「SBOM.JP」は階層構造をそのまま取り込み、管理できるという特徴があります。この特徴により、例えば脆弱

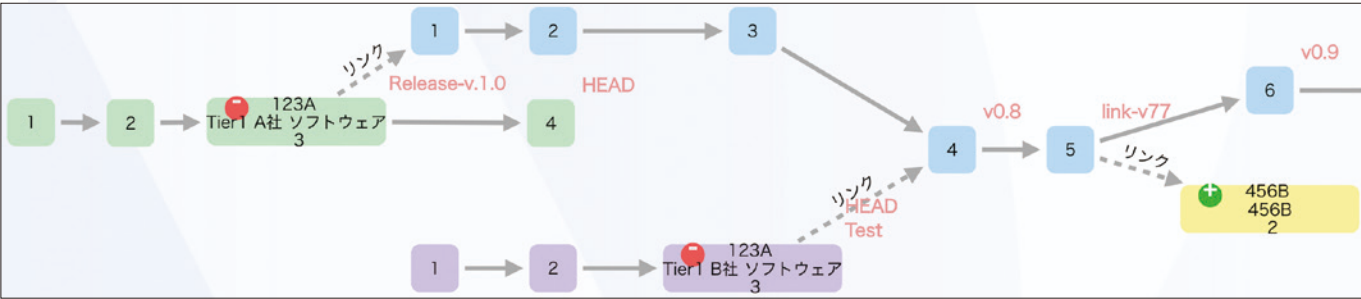
性が発見された時、その脆弱性がどのサプライヤーのどのSBOMに影響するものなのかを瞬時に把握できます(図表3)。現状、このような機能を持つSBOM管理ツールは少ない状況です。「SBOM.JP」が本機能を搭載した理由は「製品に影響のある脆弱性を迅速に特定する」というSBOM導入のメリットを最大化できるからです。

第二の特徴は、脆弱性管理のあらゆるユースケースに基づいた機能を実装していることです。一言で脆弱性管理と言っても、対象がWebシステムなのか組み込み系なのか、BtoBなのかBtoCなのかにより管理の仕方は異なります。例えば、ECサイトなどのWebシステムの場合、脆弱性が発見される都度、対応(修正するしないの判断を含む)を行い、常に最新版のバージョン管理をすることがあります。一方、組み込み機器、自動車や医療機器では話が少し変わってきます。脆弱性が発見され、修正対応したとしても、直ちに修正したソフトウェアをユーザーが使用できるわけではありません。対象によっては修正したソフトウェアをアップデートする形で配信することがあるでしょう。それが強

制的にアップデートされるのであればよいのですが、エンドユーザーがアップデートの実施可否を決める場合、市場にはアップデート前と後のソフトウェアが混在する可能性が出てきます。そのような場合、脆弱性管理の観点では最新版のソフトウェアバージョンだけを見ているだけでは十分でなく、アップデート前、つまり脆弱性対応前のバージョンのソフトウェアの状態も管理しなければなりません。「SBOM.JP」ではそのような状況にも対応できます(図表4)。また脆弱性だけでなく、SBOM自体のバージョン管理や派生管理ができるのも特徴として挙げられます。

最後に、「SBOM.JP」は全ての機能をRest APIの形で提供しているため、自社の既存インフラに組み込みやすいという特徴があります。SBOM導入には先に述べたようなメリットが存在しますが、導入コスト、運用コストがかかることは事実です。全機能のAPI提供により現場での導入コスト削減に寄与し、自動化にも取り組みやすくなることが期待できます。

図表3 サプライチェーンを意識したSBOM管理。A社とB社のSBOMをマージして、SBOMファイルが更新されている様子



02-3 SBOM.JPが目指す未来

SBOMを活用したリスク管理は、まだ過渡期、発展途上です。これから各社がSBOM作成や管理に取り組むことで業界全体の成熟度が上がり、より良いソフトウェア開発の土壌ができると考えています。ただし、SBOMを導入し活用することはゴールでなく、ライセンス管理や脆弱性管理をより

良く行うための手段の一つに過ぎません。法規対応が求められるケースも今後増えることが想定されますが、SBOMを導入する目的をしっかりと定め、適切なプロセスを構築することが肝要です。

SBOM.JPは、ソフトウェアに関わる全ての方の課題解決に寄与するツールとなるよう、これからも機能開発を続けていきます。単純にSBOMファイ

ルを保存して管理するだけのツールとしてだけではなく、SBOMを活用したその先にあるコンプライアンス対応やセキュリティ対応を見据えた機能を今後も追加し、ゆくゆくはソフトウェアサプライチェーン全体でリスク管理ができるプラットフォームを標榜し、展開していきます。

図表4 脆弱性管理ユースケースに基づいた管理機能。赤枠は同一のコンポーネント・バージョンに対する同じ脆弱性を示しているが、任意の時点でのSBOMに対する脆弱性管理ができる

プロジェクト	サブプロジェクト	タグ	コンポーネント	脆弱性ID	CVSSv3	検知日時
123A	Tier1 A社 ソフトウェア	Release-v.1.0	Jena 3.12.0	CVE-2021-39239	7.5	2024/7/12
123A	Tier1 A社 ソフトウェア	Release-v.1.0	Jena 3.12.0	CVE-2023-22665	5.4	2024/7/12
123A	Tier1 A社 ソフトウェア	Release-v.1.0	Jena 3.12.0	CVE-2023-32200	8.8	2024/7/12
123A	Tier1 A社 ソフトウェア	HEAD	Jena 3.12.0	CVE-2021-39239	7.5	2024/7/12
123A	Tier1 A社 ソフトウェア	HEAD	Jena 3.12.0	CVE-2023-22665	5.4	2024/7/12

ベリサーブが考える MBTの世界



工藤 邦博 Kunihiro Kudo

株式会社ベリサーブ 研究開発部 技術流通課

株式会社CSK(現SCSK株式会社)に入社し、証券システムの開発に従事。システム検証部門に異動後は、RDBMSアプリケーション開発ツールおよびコンピュータセキュリティソフトウェア関連のテスト業務に携わる。2001年、株式会社ベリサーブの設立を契機に、同社へ転籍。その後、コンシューマ向け組込み機器のソフトウェアテストを中心とするテスト業務を担当。2023年より現職の研究開発部に所属し、MBTツールの開発に取り組む。

MBTとは何か

MBTとはModel-Based Testingの略で、モデルを使用してテストケースを生成し、モデルに基づく、またはモデルを活用するテストを実施する考え方のことです。MBTが活用できれば、理論上、今よりもテストケースの設計品質と作業効率を高めることができるはずですが、ベリサーブが第三者検証で携わっているプロジェクトでは、まだMBTは十分に活用されていません。

これまでベリサーブでは、長年にわたりテストの知見を持つテスト設計者がオーダーメイドでテストケースを設計してきました。そうした背景があるため、モデルの使用と言っても、テスト設計の一部として境界値分析やデシジョンテーブルを活用したり、開発で使用しているシミュレータ環境を応用したテストを実行したりする程度にとどまっています。この現状を改革するためにMBTを研究した結果、これまで蓄積してきたテストのノウハウを集約・共有するためには、MBTのツール化が必要であるという結論に至りました。

本稿では、ベリサーブが自社開発するMBTツールを例にして、MBTの活用について説明します。MBTの世界について、ご理解いただければと

思います。このMBTツールは、境界値分析やデシジョンテーブルなどを使うだけの“ライト”なMBTではなく、テストベースである機能仕様からテストの目的に合った状態遷移図やアクティビティ図を作成し、そのモデルからテストケースを自動で生成するスタンダードなMBTと呼べるようなツールを目指しています。これまでのテスト設計のスタイルを進化させるべく、開発を続けています。

筆者がMBTを知るきっかけとなったのは、JaSST'07 Tokyo(ソフトウェアテストシンポジウム2007東京)での西 康晴氏の「モデルベースドテスト入門」の講演でした。今見直しても本質を的確に表現されていますので、発表資料の中から「モデルベースドテストの将来(まだ先の話)」を引用します。

- テスト設計モデルからテスト項目を自動生成できるようになる**
- 自動生成されたテスト項目のカバレッジが保証できる**
- テスト項目から自動テストスクリプトに自動変換できるようになる**
- 期待結果と実際のテスト結果との比較・判定を自動でできるようになる**
- MDD(モデル駆動開発)と連携することにより、コードが無くても(分析・設計モデルを用いて)モデルベースドテストが可能になる**

出典:ソフトウェアテストシンポジウム2007東京
西 康晴氏「モデルベースドテスト入門」

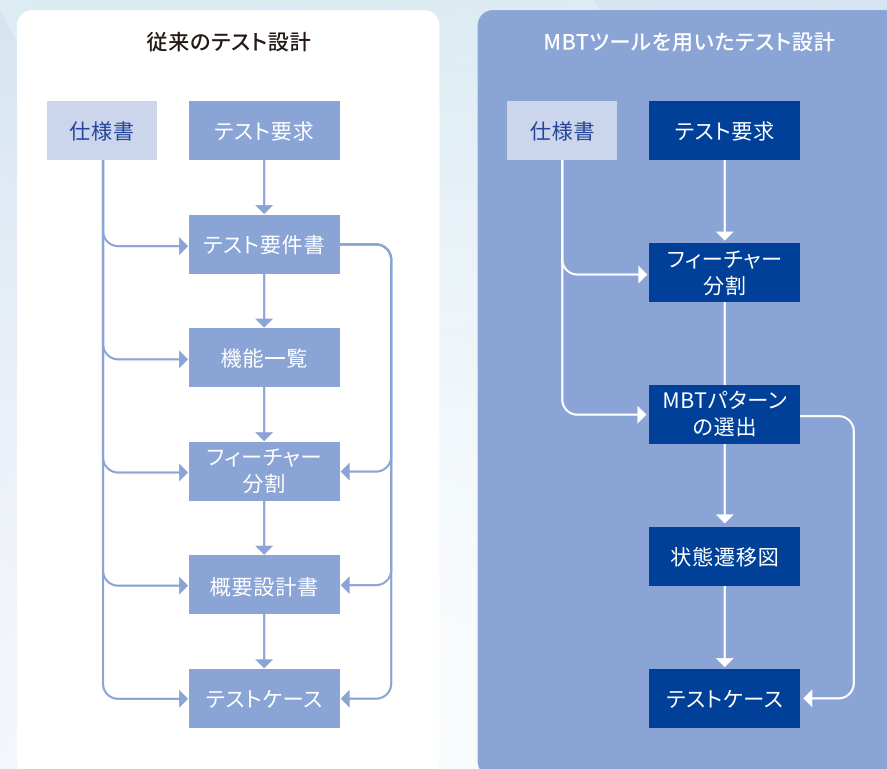
これらは、ベリサーブが実現しようとしているMBTツールの将来像と同じです。

MBTが現場の テスト設計を どう変えるのか

モデルを使用しない従来のテスト設計では、いくつかの段階を経てテスト設計を行うのが一般的です。これは、段階ごとに実施されるデザインレビューを重視しているためと言ってよいでしょう。段階を設けずにテストケースを作成するならば、求められている要求から外れたテストケースが作られるリスクが高くなります。そのため、いくつかの品質ゲートを設け、それぞれで成果物をレビューし、内容を修正しながら完成度を上げていきます。作業プロセスの中で、どこに品質ゲートを設けるかは、プロジェクトによって異なります。例えば、過去に筆者が経験した組み込み機器のテストの現場では、テスト要件、機能一覧、フィーチャー分割、概要設計書、テストケースの5段階でデザインレビューを実施していました。

(1)テスト要求を基にテスト要件書をまとめる

図表1 MBTツールを導入する前後のテスト設計の比較



(2) テスト要件書と仕様書を基に、機能一覧を作成する

(3) テスト要件書と機能一覧を基に、いくつかのテストしやすいフィーチャーに分割する

(4) テスト要件書と仕様書と機能一覧を基に、フィーチャーごとに概要設計書を作成する

(5) 仕様書と概要設計書を基に、テストケースを作成する

テストに関する成果物は、表計算ツールであるスプレッドシートを用いることが多いと思います。しかし、表

計算スプレッドシートでは前段で作成した成果物との間で、トレーサビリティを取ることが難しいのが実情です。そのため、成果物間でテストの内容が誤って書き変わったり、一部が抜け落ちたりしても、気付けないことがあります。

また、テストに関する成果物を、全て同じ設計者が一貫して担当することもまれであるため、自然言語特有の言い回しや日本語特有の主語の省略などに起因する誤解釈も発生しやすくなります。加えて、テスト設計中の仕様変更もたびたび発生するならば、ドキュメントの変更管理にも気を使わなければなりません。

ベリサーブのMBTツールでテストケースを生成する場合、作業手順は次のように変わります。

- (1) テスト要求に合うように、対象の仕様書を基に、いくつかのフィーチャーに分割する
- (2) テスト要求に合うように、MBTパターンを選出する
- (3) テスト要求に合うように、対象の仕様書を使用して、フィーチャーごとに状態遷移図を作成する
- (4) MBTパターンとモデルを使用して、テストケースを生成する

MBTパターンとは、どのような種類のテストを行うかという、いわゆる「テスト方針」とお考えください。

手動による従来のテスト設計と

MBTツールを使用してのテスト設計の大きな違いの一つに、成果物間の関連の少なさが挙げられます(図表1)。

MBTツールを導入すると、扱う成果物の構成がシンプルになることと、成果物に対してスプレッドシートの登場回数を減らすことで、前述のトレーサビリティも確保しやすくなります。また、従来の手法で取り上げた品質を下げる要因も、MBTで削減されと考えています。例えば、状態遷移図のようなモデルを使うことで、自然言語特有の言い回しの問題や日本語特有の主語の省略の問題が発生しにくくなるためです。

MBTの活用には舵を切るということは、テスト要求に合うモデルを描くことと、ツールが提供するMBTパターンでテスト要求がカバーできるかが主要な課題となります。この2点がクリアできれば、最終工程であるテストケースの作成については、ボタンを

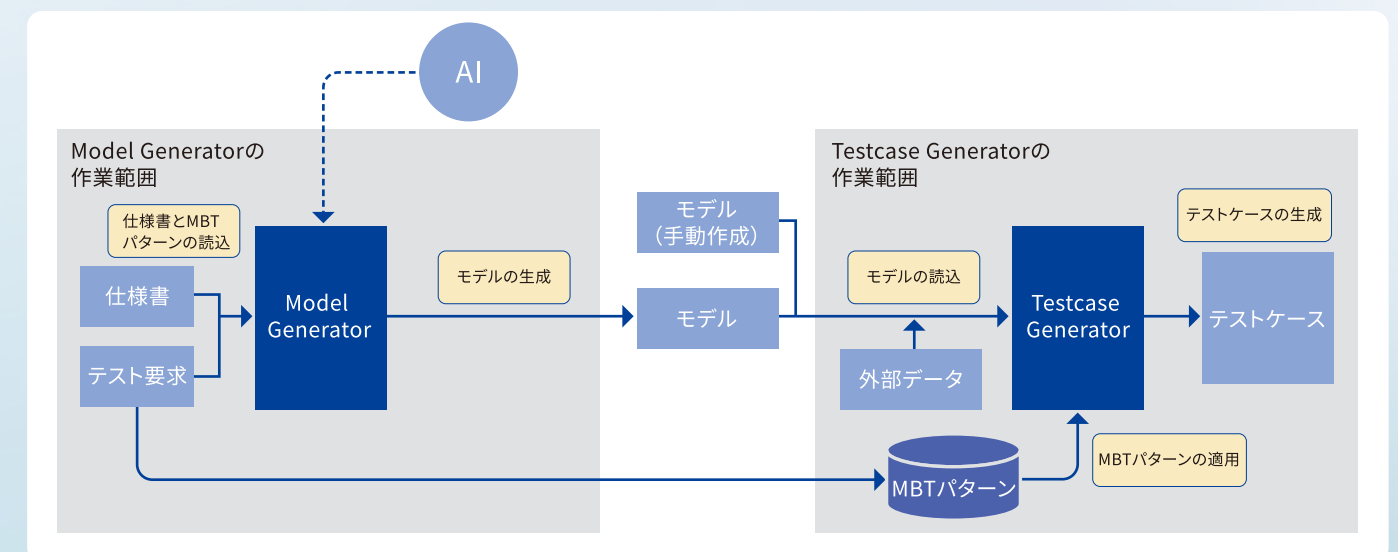
押すだけで自動生成できます。もし仕様変更が発生した場合は、状態遷移図を編集するだけで差分のテストケースを得られるようになります。

言い換えると、テスト要求を反映した状態遷移図さえ作成できれば、テストケースが自動生成できるようになります。突発的に仕様変更などが発生しても、変更内容を状態遷移図に反映すれば差分のテストケースが自動生成されるので、テスト設計の工期を短縮できるようになるわけです。

MBTツールの概要

ベリサーブが開発しているMBTツールは図表2のように2つのユニットで構成されています。

図表2 ベリサーブのMBTツールの構成



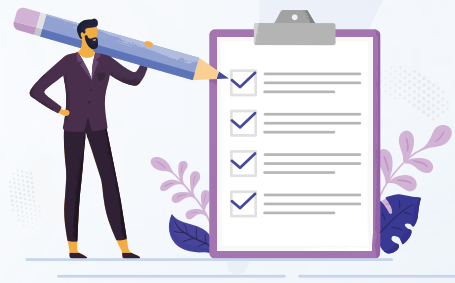
前段(図表2の左側)のユニットでは、「テスト要求に合うモデルを描く」という課題を解決するため、機能仕様書などのテストベースとテスト要求を基に、AI技術を活用して、基本的なモデルを生成します。このユニットの搭載により、これまでは経験や知識がないと難しかった状態遷移図の作成をサポートします。具体的には、機能仕様書から、機能テストに合う状態遷移図をPlantUML形式で生成します。現在、生成するモデルの精度を高めるべく、開発を進めています。

後段(図表2の右側)のユニットでは、モデルと指定したMBTパターンから、テストケースを生成します。モデルとしては、PlantUMLとEnterprise Architect™で作られた状態遷移図とアクティビティ図に対応しています。また、機能テスト向けのMBTパターンを標準で装備しています。

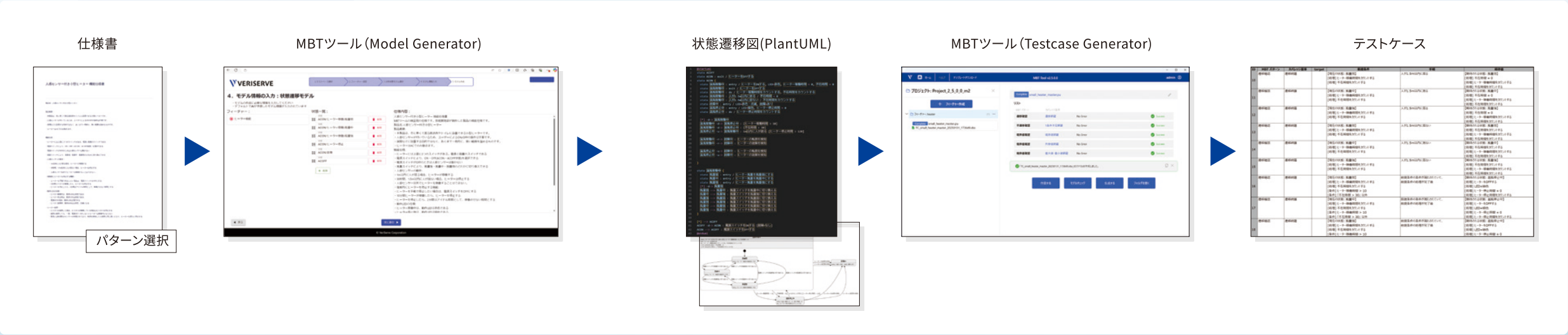
図表3にMBTツールを使用したテストケース生成イメージを示します。

図表4は、状態遷移図向けに標準装備するMBTパターンの一覧です。

機能テストの範囲は、プロジェクトによって異なるため、「ツールが提供するMBTパターンでテスト要求がカバーできるか」という課題があります。標準装備していないMBTパターンが必要になった場合は、追加で開発し、必要なテストがいつでも生成できるサービスを提供できるよう検討しています。



図表3 MBTツールでのテストケース生成イメージ



図表4 状態遷移図向けに標準装備するMBTパターン

MBT パターン		説明
遷移確認	遷移網羅	遷移網羅とは、状態間のエッジを基に、遷移することを確認するテストケースを生成する
	経由型遷移網羅	遷移先状態が経由状態として定義されている場合、一連の経由状態から抜けるまでのテストケースを生成する
不遷移確認	1条件不足網羅	1条件不足網羅とは、状態間のエッジ線に付与されたガード条件を基に、ガード条件が不成立の場合に、遷移しないことを確認するテストケースを生成する
境界値確認	境界値網羅	境界値網羅とは、状態間のエッジ線に付与されたガード条件を基に、ガード条件に変数と数値が比較演算されている場合、その比較演算の前後の境界値で確認するテストケースを生成する
	列挙値網羅	列挙値網羅とは、状態間のエッジ線に付与されたガード条件を基に、ガード条件に列挙値が指定されている場合、列挙値全てのパターンを確認するテストケースを生成する
	最大値・最小値網羅	最大値・最小値網羅とは、状態間のエッジ線に付与されたガード条件を基に、ガード条件に変数と数値が比較演算されている場合、その変数の最小値・最大値で確認するテストケースを生成する
履歴	履歴	履歴を考慮した遷移網羅のテストケースを生成する
並行状態	並列処理の中の組み合わせ	複数の領域を持つ並行状態で、ある領域内での遷移確認の際、他の領域の状態がどのような状態でも、遷移することを確認するテストケースを生成する
	外部への遷移のテスト	領域と外部との遷移を確認するテストケースと並行状態の遷移を確認するテストケースを生成する
	領域の活性・不活性の組み合わせの履歴	並列処理の領域の活性・不活性の組み合わせを、履歴と組み合わせて遷移を確認するテストケースを生成する
イベント確認	イベント競合	複数のイベントが同時に発生した場合のテストケースを生成する
	無効イベント	記載のないトリガーが発生した場合、無効であることを確認するテストケースを生成する

来たるべき MBTの世界

ベリサーブが考えるMBTツールの未来像は、機能仕様書の作成が不要になる世界です。状態遷移図のようなモデル自身が機能仕様書の代わりを務めることになります。モデル上で機能の振る舞いを追加・変更・削除し、デザインレビューを実施し、不備を修正する…つまり、モデルで語り合う世界の到来を期待しています。

しかし、ドキュメントもなく、モデルだけになった世界で、「全ての関係者が、直接このモデルを読み、理解することができるのか？」という疑問が出てくると思います。その解決策は、モデルを読みたい形式に翻訳(解釈)する、複数の「ビュー」を用意することです。

例えば、従来の機能仕様書の形式で読みたい人向けに、仕様書の形式で読める「ビュー」を提供します。複数の用途別の「ビュー」を活用することで、全ての関係者がモデルを理解し、要求通りに設計できているかなど、さまざまな観点で調べることができると考えています。生成されるテストケースも、実は「ビュー」の一つです。

「ビュー」を使用して、状態遷移図に不備が見つかったとしても、修正すればいつでもテストケースを自動生成しますので、期間・コストが圧縮できることは、言うまでもありません。

モデルで語り合う世界は、最終的に仕様変更による機能への影響範囲が可視化できるため、開発工程とテスト工程の工期の短縮に寄与できると考えています。

MBTツールが近い将来、テスト設計の標準になれるようにこれからも開発を進めてまいります。

TOPICS

2024年 IT業界の主なトピック

1月	CES 2024は昨年比15%増の床面積で4,000社以上が参加 JAXAの無人探査機「SLIM」が月面着陸に成功
2月	経済産業省が国内の生成AI開発力強化プロジェクト「GENIAC (Generative AI Accelerator Challenge)」を開始 グーグルがAIサービスブランドを「Gemini」に統合
3月	古川聡宇宙飛行士搭乗の米国クルードラゴン宇宙船運用7号機の帰還成功
4月	マイクロソフトが日本のAIインフラ強化に向けて4,400億円の投資を発表 OpenAIがアジア初の拠点となる東京オフィスを開設 エヌビディアが次世代GPU「Blackwell」を発表 総務省と経済産業省がAIにかかわる企業が守るべき項目を盛り込んだ「AI事業者ガイドライン」第1.0版を公開
5月	スパコン「富岳」で開発のAIが完成 報道陣に公開
6月	スマホソフトウェア競争促進法が成立
7月	経済産業省と独立行政法人情報処理推進機構 (IPA) が「デジタルスキル標準 (DSS)」を改訂
8月	経済産業省が「ソフトウェア管理に向けたSBOM (Software Bill of Materials) の導入に関する手引ver2.0」を公表 NTTの次世代ネットワーク基盤構想「IOWN (Innovative Optical and Wireless Network)」が日台間3,000kmで17msecの超低遅延通信を実現
9月	Sakana AIに3メガバンクや国内大手ITなどが出資 調達累計300億円規模に 「デジタル給与」が国内で初めてスタート
10月	CEATEC 2024は808社が参加、「JAPAN MOBILITY SHOW BIZWEEK」併催
11月	Uber Eatsがロボット配達を大阪で開始 GitHubがOSSのセキュリティ向上を支援する「GitHub Secure Open Source Fund」を創設
12月	ChatGPTがアップルの生成AI「Apple Intelligence」との連携を開始

Powered by **VERISERVE**

イノベーションを加速させる
知恵と品質技術にアクセスする
テクノロジーライフメディア



HQW!
Hello, Quality World!

