

Web サービス・モバイルアプリケーション開発における テスト設計を支援するための標準テスト観点の整備

河野 哲也
株式会社 ディー・エヌ・エー
tetsuya.kouno@dena.com

前川 健二
株式会社 ディー・エヌ・エー
kenji.maekawa@dena.com

柏倉 直樹
株式会社 ディー・エヌ・エー
naoki.kashiwagura@dena.com

菊武 祐輔
株式会社 ディー・エヌ・エー
yusuke.kikutake@dena.com

要旨

Web サービス・モバイルアプリケーション開発におけるテスト設計においてテスト観点を抜け落ちなく網羅的に抽出するために、標準テスト観点を整備し、それをテスト設計で活用する研究を報告する。まず、実際のテスト業務に基づきテストエンジニアの特性を整理し、問題を明らかにする。次に、テスト分析・テスト設計の流れを解析し、標準テスト観点の枠組みとして必要な要件を整理し、その要件に従い枠組みを定める。そして、過去に作成したテストスイートを分析することで様々なテスト観点を導出し、枠組みに従いそれらのテスト観点を整理することで標準テスト観点の実装を行う。最後に、実際の Web サービス・モバイルアプリを対象として標準テスト観点の適用を行い、一定の有効性が確認できた。

1. はじめに

我々は、Web サービスやモバイルアプリケーション(以降、モバイルアプリと略す)分野における機能テスト以降のテスト業務全般を担当している。我々の組織におけるテストプロセスは、テスト分析・テスト設計・テスト実装・テスト実施といった一般的なプロセス[1]を採用している。そのようなプロセスにおいて、テスト分析・テスト設計においてテスト観点を抜け落ちなく網羅的に抽出することが課題となっている。

また我々の組織に限らず、テスト分析・テスト設計領域における業界的認知度が高く、例えばテスト設計の質を競うテスト設計コンテスト[2]や先行事例[3][4][5][6]ではテスト観点到フォーカスした議論がなされている。

以上を踏まえ本稿では、テスト分析・テスト設計領域に焦点を当て議論を進める。

現状、実際のソフトウェアテストの現場では、テスト分

析やテスト設計においてテスト観点を網羅的に抽出するために、マインドマップのようなツリー構造による階層化や表構造による一覧化を活用するものの、現実的には経験豊富なテストエンジニアをアサインすることにより解決している。しかしながら、ソフトウェアのビジネス領域の広がりとともに、Web サービス・モバイルアプリ分野の人材の流動性の激しさも相まって、そのような解決では対応できず、経験の浅いテストエンジニアにとってテスト観点を抜け落ちなく網羅的に抽出することが課題となっている。よって、経験の浅いテストエンジニアに対してそれを補うような体系的な仕組み・仕掛けが必要である。

以上を背景として、テスト観点を網羅的に抽出するために枠組みやテンプレートを提案した取組み[7][8]や研究[5][9][10][11][12]がある。これらの取組みや研究は、テスト観点を抽出するための指針やテスト観点を整理するための枠組みであり、多面的なテスト観点を知識として保有するテストエンジニアにとっては有用であるが、経験の浅いテストエンジニアの支援という点で捉えた場合、大きな効果は期待できない。他方、具体的にテスト観点を例示した報告[4][12]もあるが、Web サービス・モバイルアプリ分野に関して言及されていない。

一方、我々の組織ではそのような課題に対して、Web サービス・モバイルアプリ分野で具体的に活用可能なテスト観点を蓄積・整理するような取組みを進めてきた。先行して報告した取組みでは、既存のテストスイートからテスト観点を導出するための手法に言及し、それによって導出されたテスト観点の有効性を確認した[13]。本稿では、この取組みをさらに進め、Web サービス・モバイルアプリ分野における標準的なテスト観点を整理するための枠組みを検討し、標準テスト観点の実装に言及する。

本稿では、第2章で本研究が対象とする問題を示し、研究の目的を述べる。第3章では、テスト分析・テスト設計の流れを詳細に解析することによりテスト観点を整理

するための枠組みを定める。第4章では、前章の枠組みに従いテスト観点の実装に関して述べ、さらにその使用例を概説する。第5章の適用例では、前章の実装によって得られた標準テスト観点の有効性を確認する。

2. 問題と目的

2.1. 本研究の問題

本節では、まずテスト分析・テスト設計の流れを概観することで本研究が対象とする標準テスト観点の支援が必要なテストエンジニアの特性を整理し、本研究の問題を示す。そして、どのような側面のテスト観点の整備が必要なのかを明らかにする。

まず、文献[1]を援用し、実際の現場におけるテスト分析・テスト設計の流れを Process Flow Diagram (PFD) [14]によって表現したものを図1に示す。

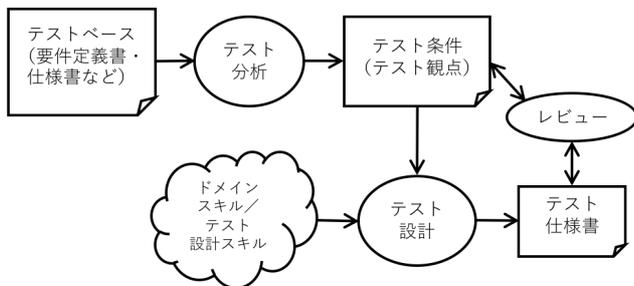


図1 テスト分析・テスト設計の流れ

テスト分析では、要件定義書や仕様書などのテストベースを分析することでテスト条件が得られる。次に、テスト設計では、テスト分析で得たテスト条件をきっかけとして、ドメインスキルを活用しながらテスト条件を補完させつつ、テスト設計スキルをベースとしたテスト技法に従いハイレベルテストケース[1]を作成することでテスト仕様書が得られる。また、テスト条件やテスト仕様書の質を第三者が確認するためのレビューを必要に応じて実施している。

例えば、Webサービスのログイン機能に対する機能仕様書からユーザIDとその有効・無効やパスワードとその有効・無効、ログイン成功・失敗などがテスト条件として得られる。それらに対して、例として機能仕様書に「複数回ログイン失敗することでアカウントロック」や「通信不可状態でログイン」などが記載されていない場合は、それらはドメインスキルを活用しながら補完していくことになる。そして、一例としてユーザIDの有効・無効とパスワードの有効・無効の組み合わせに対してテスト技法を援用しな

がらハイレベルテストケースを作成することになる。

以上で述べた補完されたテスト観点は、Wモデル[15]の議論でも見られるように開発成果物のレビューで指摘することも可能である。しかし、本稿では開発成果物の質向上につなげるというレビューの立場ではなく、テスト分析・テスト設計の結果をテスト実装・テスト実行につなげるというテストの立場をとる。

ここで、ドメインスキルは、テストエンジニアが保有するドメイン知識に照らし合わせながら、必要なテスト条件を抽出するためのスキルとする。テスト設計スキルは、テストエンジニアが保有するテスト技法に関する知識に照らし合わせながら、テスト条件からハイレベルテストケースを設計するためのスキルとする。

また、テスト分析においても一般的にドメインスキルやテスト設計スキルが必要となるが、本稿では議論を簡単にするために、テスト分析ではテストベースに記述された情報のみからテスト条件が得られるものとする。加えて、先に述べた流れでは文献[1]に従いテスト条件という用語を使用した。現状のテストの現場では、テスト条件に対応する用語としてテスト観点が広く一般に認知されているため、辞書的な定義よりも実用性を優先し本稿ではテスト観点という用語で統一する。

以上の流れにおいて、テストエンジニアの技術として極めて重要なのが、ドメインスキルとテスト設計スキルである。そして、ある一定のテスト設計スキルがあれば、テストベースからテスト観点を網羅的に抽出することはできるが、テストベースの情報は常に不足しているという経験則[16]も報告されているようにドメイン知識に関する情報は開発成果物に記載されない傾向が多いため、一定のテスト設計スキルがあったとしてもドメインスキルが不足している場合、当該ドメインに関するテスト観点が網羅的に抽出されない。つまり、ドメインスキルとテスト設計スキルという2つの側面から捉え、当該ドメインにおいてテスト観点を抜け落ちなく網羅的に抽出するという点にフォーカスすると、ドメインスキルが重要な位置づけとなる。

一方、Webサービス・モバイルアプリ分野は比較的新しいドメインであるため、テスト設計スキルにおいて習熟しているものの、その分野のドメインスキルにおいては未習熟であるテストエンジニアがアサインされることが我々の組織では多い。

以上で述べたテストエンジニアの特性を表1に整理する。表1が示すように本研究では、テスト設計スキルは習熟しているものの、Webサービス・モバイルアプリ分野のドメインスキルが未習熟なテストエンジニアがテスト設計を行うという制約に対して、現状、そのようなドメインス

キルの不足を補うような仕組み・仕掛けが体系立って整備されていないことが問題である。

表 1 本研究の対象とするテストエンジニアの特性

		ドメインスキル	
		未習熟	習熟
テスト設計スキル	未習熟	そもそもテスト技術不足であるため 対象外	当該分野では稀な特性であるため 対象外
	習熟	多くのメンバの特性であるため 本研究の対象	中長期でアサインできない特性であるため 対象外

よって、そのような問題を解決するために、Web サービス・モバイルアプリ分野というドメインスキルの側面におけるテスト観点を標準化し、体系立って利用できるような支援が必要となる。

2.2. 関連研究

テスト観点、もしくはテスト観点と明示しないもののそれに関連する研究は2つに大別される。

1 つ目は、テスト観点を抽出・整理するための手法や考え方、枠組みを提案した研究である。まずツリー構造で多面的にテスト観点を抽出・整理する研究がある[5][7][10]。次にマトリクス構造で多面的にテスト観点を抽出・整理する研究[3][8][11]がある。これらの研究は、ドメインスキルが十分に習熟しているテストエンジニアにとっては有用であるが、そもそもこのようなスキルが未習熟であると具体的なテスト観点そのものを得ることが難しい。また、要求仕様書の機能や制約などの要素に対して逸脱の視点でテスト観点を抽出する研究[9]がある。この研究はドメインスキルが未習熟なテストエンジニアにとって、逸脱の視点でテスト観点を広く抽出することを支援する研究であるものの、逸脱というレアケースを想定したテスト観点到重点がおかれる。

2 つ目は、テスト観点そのものを提示した取組みや研究である。まず外部機能仕様書の曖昧さに着目して、具体的なテスト観点を提案した研究[12]がある。次に、品質特性や経験則を活用してテスト観点テンプレートを提案し機能テストレベルにおける具体的なテスト観点を例示した研究[4]がある。加えて、既出の不具合を分析し複数のテスト観点を例示した研究[3][6]もある。これらの研究では、部分的に参考になる観点はあっても Web サービス・モバイルアプリ分野に言及されていないため、その分野のテスト観点を体系的に得ることが難しい。

一方、Web サービスにおけるテストに関して論述した文献[17]がある。これらは一部テスト観点として活用可能なトピックもあるが、Web サービス分野のテストの考え方や手法を体系的に述べたものである。

2.3. 本研究の目的

本研究の目的は、Web サービス・モバイルアプリ分野における標準的なテスト観点を整備し、それを活用することにより2.1で述べたような特性を持つテストエンジニアがテスト設計の際に、テスト観点が抜け落ちなく網羅的に抽出できるようにすることである。ここで、整備された標準的なテスト観点群を標準テスト観点と呼ぶ。

この目的のために、具体的には、テスト分析・テスト設計の流れを詳細に解析する事でどのような枠組みであればテストエンジニアが有効に活用できるかという点を考察し、標準的なテスト観点を整理するための枠組みを定める。そして、過去に作成したテスト仕様書やテストケースから標準的なテスト観点を導出し、枠組みに従い整理することで標準テスト観点の実装を行う。ここでテスト仕様書やテストケースを総称してテストスイートと呼ぶ。図2に本研究の全体像を示す。図2は本稿の章・節の構成に対応させているため、以降、適宜参照されたい。

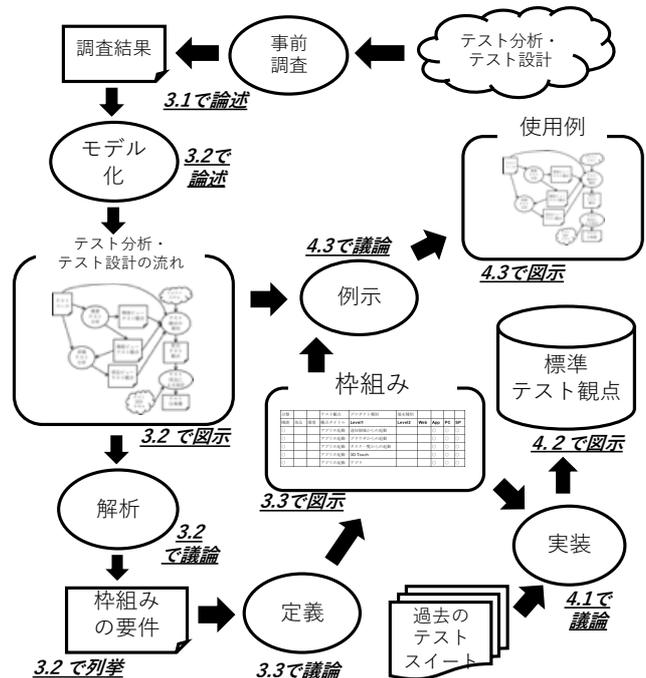


図 2 本研究の全体像

なお、本稿では標準テスト観点の一部を例示するが、

その全貌を示すことは紙面の都合上難しい。とはいえ、本稿の解析や実装は、本研究の対象とする同様の問題を保有する組織やエンジニアにとって参考になるものと考えられる。なお、本研究の標準テスト観点のオープン化は今後の課題と考えている。

3. 解析

本章では、テスト観点をどのような枠組みで整理すればテストエンジニアが有効に標準テスト観点を活用できるのかという点でテスト分析・テスト設計の流れを解析し、それに基づき枠組みの要件を整理したのち枠組みを定める。そのために、まず事前調査として経験豊富なテストエンジニアがどのようにテスト分析・テスト設計を行っているのかをヒアリングし、その流れを明らかにする。

3.1. 事前調査

本節では、解析のための事前調査として経験豊富なテストエンジニアにテスト分析・テスト設計の流れに関してヒアリングを行う。

まずテスト業務経験 10～20 年のテストエンジニア 5 名を対象とし、テスト分析・テスト設計の流れに関してヒアリングを実施した。ヒアリングでは 5 名の集合形式の 1 時間程度のミーティングを実施し、2.1 節で示したテスト分析・テスト設計の流れをベースに更に詳細な流れに関して意見交換を行った。

このヒアリングにより、テストエンジニアによっては多少異なる点はあるものの、概ね以下に示すような流れであることが結果として得られた。

- テスト分析では大きく機能と環境の 2 つのビューによって分析されている
- 機能ビューによって抽出された各機能に対して機能を構成する部品のビューによって更に詳細にテスト分析が実施されている
- テスト設計ではテスト分析において機能ビュー・環境ビュー・部品ビューで抽出されたテスト観点をきっかけとして、経験として蓄積したドメインスキルによってテスト観点を補完しつつテスト設計スキルを活用しながらテスト仕様書が作成される

3.2. 調査に基づく解析

本節では 3.1 節のヒアリング結果に基づきテスト分析・テスト設計の流れをモデルで表現し、そのモデルに基づき枠組みの要件を整理する。

まず、3.1 節のヒアリング結果を PFD により表現すると

図 3 のようになる。以下で、流れの詳細を述べる。

はじめに、概要テスト分析では、テストベースに対して 2 つのビューでテスト分析を行い、環境ビューのテスト観点と機能ビューのテスト観点が得られる。次に、詳細テスト分析では、テストベースと機能ビューテスト観点に対して機能を構成する部品のビューでテスト分析を行い、部品ビューのテスト観点が得られる。なお、概要テスト分析と詳細テスト分析の 2 つを総称してテスト分析と呼ぶ。

ここで部品ビューに関して更に詳細な検討を進めた。部品ビューでは特定の機能に関するテスト観点ではなく複数の機能を横断するような処理やデータ、構造に関するテスト観点である。例えば、メールアドレス入力というテスト観点はユーザ登録機能やログイン機能、問い合わせ機能など複数の機能を横断するテスト観点である。このように複数の機能横断に該当し機能を構成する要素・構造に対してのビューを部品ビューとする。

そして、テスト観定の補完では、環境ビュー・機能ビュー・部品ビューのそれぞれのテスト観点に対して、全てのビューによるテスト観点を統合的に照らし合わせて、ドメインスキルを活用することによりテスト観定の抜け落ちを補完することで統合テスト観点が得られる。また、ハイレベルテストケースの設計を鑑みて、その設計作業を行いやすくするためにテスト観点を整理する作業もこの作業に含まれる。

最後に、テスト技法による設計では、統合テスト観点に対してテスト設計スキルを活用することで、ハイレベルテストケースが記述されたテスト仕様書が得られる。

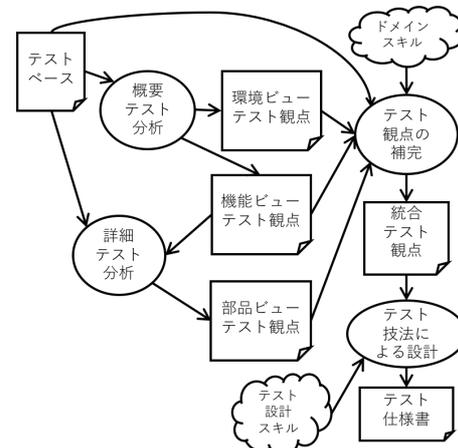


図 3 テスト分析・テスト設計の詳細な流れ

具体的には、ある Web サービスがテスト対象だった場合に、テストベースとしてその機能仕様書が与えられたとすると、概要テスト分析では一例として新規ユーザ登録

やログインといった機能の観点機能が機能ビューテスト観点に該当し、当該サービスがサポートするOS 種別やブラウザ種別といった環境の観点が環境ビューテスト観点に該当する。次に、詳細テスト分析では、ログインといった機能ビューテスト観点に対して一例としてメールアドレス入力やパスワード入力といった機能を構成する部品の観点が部品ビューのテスト観点に該当する。

また、概要テスト分析と詳細テスト分析では、一例として示したこれらテスト観点に対して、機能仕様書には、より詳細な仕様、例えばログイン機能の有効・無効な条件やOS 種別・ブラウザ種別の具体的なOS 名やブラウザ名及びそのバージョンなどが記載されているため、それらをテスト観点として抽出していく。

そして、テスト観点の補完では、機能ビューテスト観点のログイン機能に対しては、複数端末や複数ブラウザで多重ログインといった仕様は機能仕様書に記載されることは多くないためそれらがドメインスキルを活用することにより、補完されることとなる。一方、環境ビューテスト観点としては、通信状態に関する仕様は機能仕様書に記載されることは多くないためそれがドメインスキルを活用することにより補完され、さらに通信状態という観点に対して弱電波や「通信不可からの復帰」といった詳細な観点が更に補完される。さらに、部品ビューテスト観点のパスワード入力に対しては、「入力不可文字のコピーアンドペースト(以降コピペに略す)」といったテスト観点がドメインスキルを活用することにより補完される。これら補完されたテスト観点を含めて統合的にまとめると統合テスト観点が得られる。

以上のテスト観点の補完のイメージを文献[5]に従い表現した。それを図 4 に示す。この図に示すように、「入力不可文字のコピペ」というテスト観点は、以降のテスト技法による設計のためにテスト観点を詳細に整理する。

最後に、テスト技法による設計では、一例として統合された部品ビューテスト観点のパスワード入力に対してテストスキルを活用することで、デジジョンテーブルといったテスト技法を援用し図 5 に示すようなハイレベルテストケースが得られる。

以上テスト分析・テスト設計の詳細な流れに基づくと、テスト観点の補完が本研究のスコープとなり、そこにおいてドメインスキルが不足するテストエンジニアに対してテスト観点が抜け落ちなく網羅的に抽出できるようにすることである。よって、ドメインスキルの不足するテストエンジニアでも、図 4 に示すような統合テスト観点が得られるようにする必要があるのである。そのために、先に述べたテスト観点の補完の詳細に基づく枠組みとして以下に示すような

要件が必要となる。

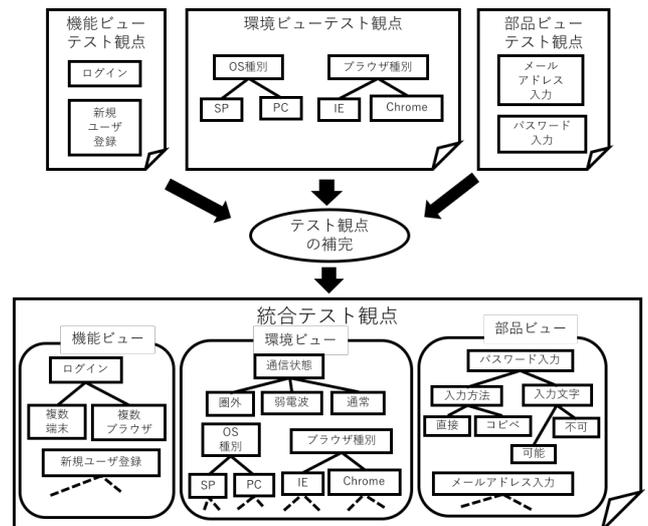


図 4 テスト観点の補完のイメージ図

		1	2	3	4
原因	入力方法	直接	Y	Y	
		コピーアンドペースト			Y
	入力文字	可能	Y		Y
		不可		Y	
結果	入力ログインできること	Y		Y	
	入力できないこと		Y		Y

図 5 ハイレベルテストケースの一例

要件1: 標準的なテスト観点群の中から機能・環境・部品のビューでフィルターができること。例えば、機能ビューテスト観点の補完を行っている際に、そのビューのみのテスト観点到フォーカスが当たるようにする必要がある。

要件2: 各ビューのトップのテスト観点が一覧から選択できること。例えば図 4 の例であれば、ログイン・新規ユーザ登録・アドレス入力・パスワード入力・通信状態・OS 種別・ブラウザ種別がそれに対応する。

要件3: 標準テスト観点は階層的に表現されていること。例えば、ログインの配下には複数端末、複数ブラウザなど必要に応じ複数の階層により表現されることによってテスト観点的意図がわかりやすくなるようにする必要がある。

3.3. 枠組み

3.2 節で定めた要件3を満たすためには、標準的なテスト観点を整理するための構造はツリー構造になるが要件1と要件2を満たすためにはスプレッドシートのソート機能を使用したほうが簡便である。また、テスト領域ではスプレッドシートは慣れ親しんだツール[18]であるための、

その導入の容易性を踏まえスプレッドシートで要件を満たすための枠組みを定めることとする。

ビュー			テスト観点			
機能	部品	環境	観点タイトル	Level1	Level2	Level3
	○		ファイルのアップロード	画像ファイル	ファイル種別	JPG
	○		ファイルのアップロード	画像ファイル	ファイル種別	PNG
	○		ファイルのアップロード	画像ファイル	ファイル種別	GIF

図 6 テスト観点を整理するための枠組み

以上まとめると図 6 のような枠組みとなる。枠組みを理解しやすくするために、テスト観点をいくつか例示する。

この枠組では、まず左側に位置するビューの列でテスト観点に該当するそれぞれのビューを○と記載し、フィルター機能を利用することで、それに応じたテスト観点をビューを表示することができる。よって、テスト観定の補完の際に、テストエンジニアがそれぞれのビューでのテスト観定の抜け落ちを確認することができる。

次に、図 4 のそれぞれのツリーの頂点のテスト観点を総称して観点タイトルと呼び、図 6 に示す観点タイトルの列で一覧化される。そして、その列から詳細が必要なテスト観点を選択できるようになる。この選択する際も、フィルター機能を使用すれば、簡便に一覧表示ができる。

加えて、観点タイトルの配下のテスト観点は、Level1 - Level2 - Level3 といった形式で表し、ツリー構造と比較すると全体の見通しの悪さの弱点はあるものの、階層的表現は実現されている。それにより、各階層に位置するテスト観定の意図が理解することができる。ここで図 6 では、Level1 から Level3 の階層で示しているが、第 4 章で述べる実装において、テスト観点を導出し整理した結果、全てのテスト観点が Level3 までで表現することができたからである。将来的にはテスト観点を拡充することが考えられるため、更に階層が深くなるのが想定される。

4. 実装

4.1. 実装方法

第 3 章で定めた枠組みに従い、標準的なテスト観定の実装を行う。なお、本節の議論は先行研究[13]で詳述しているため、本節では実装方法の概要に留める。

まず、筆者らがテスト業務を担当する当社のプロダクト、

例えば、オートモーティブ関連・ヘルスケア関連のプロダクトに対して過去 2 年間で作成したテストスイートを対象としそれらを分析することでテスト観点を導出する。

なお各プロダクト独自のテスト観点は横断的な汎用性がないため標準テスト観点としては対象外とした。加えて、テストベースから抽出可能なテスト観点とドメインスキルによって補完されるテスト観点を切り分けながら導出を行っていくことに注意されたい。例えば図 4 のハイレベルテストケースであれば、入力方法-直接と入力文字-可能文字はテストベースから抽出される観点であるが、入力方法-コピーアンドペーストや入力文字-不可文字はドメインスキルによって補完が必要なテスト観点となる。

導出したテスト観点を第 3 章で定めた枠組みに従いテスト観定の一般化・抽象化や詳細化を行いツリー構造で表現し、テスト観定の整理を行う。

以上のテスト観点を 3.1 節で述べた経験豊富なテストエンジニア 5 名によってレビューを行い、標準的なテスト観点としての十分性やドメインスキルという視点での有効性の確認を行う。

4.2. 実装結果

4.1 節の方法に従い実装を行った結果、観点タイトルが 90 種類、テスト観定総数は約 700 観点となった。この観点導出から整理までに 340 時間を要した。

標準テスト観定の抜粋を付録に示す。標準テスト観点はスプレッドシートで実装されているため、そのソート機能を使用しながら、テスト観定の補完の際の関心に合わせて、必要なテスト観点を検索していくことになる。

4.3. 使用例

本節では、図 1・図 3 のそれぞれのテスト分析・テスト設計の流れに基づき、標準テスト観定の使用例を示す。

まず、図 1 で示したように実際の現場ではレビューを実施しているため、図 3 にそれを組み込んだ結果を図 7 に示す。この図に示すように、標準テスト観定の使用例は大きく 2 つ挙げられる。

使用例 1: テスト観定の補完で活用することである。これはテスト分析・テスト設計を行っているテストエンジニア自身がセルフレビューとして活用することを意味する。まず、テスト対象に該当するプロダクト種別・端末種別を選択する。そして、テストベースに基づき機能ビュー・環境ビュー・部品ビューのそれぞれのビューに照らし合わせながら観点タイトルのレベルで抜け落ちが無いかを確認していく。続いて、階層構造でテスト観点を捉えた際に、その詳細のテスト観定に抜け落ちがないかを確認する。

使用例2: 統合テスト観点のレビューで活用することである。これはテスト分析・テスト設計を行っていないテストエンジニア、もしくはテストマネージャが第三者レビューとして活用することを意味する。活用方法としては、使用例1と同様である。

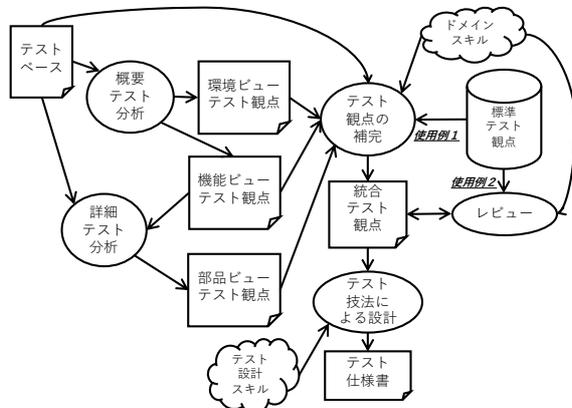


図 7 標準テスト観点の使用例を含むテスト分析・テスト設計の流れ

5. 適用例

本章では、まず概要で適用の方針やテスト対象について述べ、次に適用方法を示し、続いて適用結果を示し本研究の有効性を確認する。最後に本稿の考察を行う。

5.1. 概要

本研究の目的は、Web サービス・モバイルアプリ分野におけるドメインスキルが不足するテストエンジニアに対してテスト観点の抜け落ちなく網羅的に抽出できるようにするための何らかの支援を用意することである。

そのため本適用では、実際の Web サービスのテスト分析・テスト設計(適用1)とモバイルアプリのテスト分析・テスト設計(適用2)を対象として、標準テスト観点を活用することで、どの程度テスト観点の補完ができたのかを有効性として確認する。本適用では、4.3節で示した2つの使用例を用い、テスト観点の補完を実施する。

適用の概要を表 2に整理する。適用1では、ある Web サービスの機能追加の開発においてアクセス制御のためのユーザ管理機能をテスト対象として選定し、表に示すテスト業務経験のテストエンジニア・活用シーンによってテスト業務が遂行されるようにした。適用2では、あるモバイルアプリの新規開発においてインストール・ログイン・アップデートの機能をテスト対象として選定し、表に示す

テスト業務経験のテストエンジニア・活用シーンによってテスト業務が遂行されるようにした。

ここで適用 1 の担当テストエンジニアを TE1 と呼び、適用2の担当テストエンジニアをTE2と呼ぶ。また、TE1・TE2ともに Web サービス・モバイルアプリ分野の経験が豊富とはいえず、ドメインスキルが習熟しているわけではない。よって、TE1・TE2ともに本研究が対象とするドメインスキルが不足するテストエンジニアに該当する。

表 2 適用の概要

		適用1	適用2
テスト対象		Web	モバイルアプリ
開発形態		機能追加	新規
対象機能		ユーザ管理	インストール ログイン アップデート
担当テスト エンジニア の業務経験	テスト全般	3年	11.5年
	Webサービス ・モバイルアプリ	3年	0.5年
標準テスト観点の活用シーン		使用例 1	使用例 2

5.2. 適用方法

まず、テスト対象機能の機能仕様書をベースにテスト分析を行う。

次に、適用1ではテスト分析結果に対して TE1 自身が標準テスト観点を活用しテスト観点の補完を行う。適用2ではテスト分析結果に対して第三者のテストエンジニアが標準テスト観点を活用しながらレビューを行い、テスト観点の抜け落ちを確認することでテスト観点の補完を行う。ここで、第三者のテストエンジニアは標準テスト観点の使用方法を熟知しているものとし、またレビューの際には、有効性の確認の目的を踏まえ標準テスト観点のみを利用しテスト観点を補完する。

そして、それぞれの適用において、標準テスト観点を活用することで、どの程度、テスト観点が補完できたかを適用結果として整理し、本研究の有効性を確認する。

5.3. 適用結果

5.2 節の適用方法に従い、2 種類の適用を実施した。適用 1 のテスト観点の補完では 1 時間を要し、適用 2 のレビューでは 1 時間を要した。表 3に適用結果を示す。

適用1では、TE1 がテスト分析を行い 86 のテスト観点が抽出され、TE1 が標準テスト観点を活用し 6 のテスト観点が補完された。また、TE1 およびテスト対象の開発エンジニアが補完したテスト観点の必要性を確認しその結

果, それらのテスト観点全てがテストすべきテスト観点として採用した. 補完したテスト観点の一例として, 「入力欄に制御文字を含む文字列の入力」や「同一名称のファイルの再アップロード」, 「ファイルサイズ0 KB のファイルをアップロード」などが挙げられる. なお参考として, この補完したテスト観点によって不具合は検出されなかった.

表 3 適用結果

	適用1	適用2
テスト分析で抽出したテスト観点の数	86	36
標準テスト観点活用で補完したテスト観点の数	6	43
上記のうちテストすべきテスト観点として採用した数	6	21

適用 2 では, TE2 がテスト分析を行い, 36 のテスト観点が抽出され, 第三者のテストエンジニアが標準テスト観点を活用し 43 のテスト観点を補完した. また, TE2 とテスト対象の開発エンジニアが補完したテスト観点の必要性を確認し, 内部仕様上の品質リスクとテスト工数を踏まえて, 21 がテストすべきテスト観点として採用した. 採用しなかったテスト観点の一例として「端末の日時変更」や「端末の言語設定変更」, 「退会済みのアカウントでログイン」などが挙げられる. また採用されたテスト観点として「ストレージ容量不足でインストール」や「通信不可状態からの回復」などが挙げられる.

加えて参考として, 適用 2 では, 採用したテスト観点に基づき作成したテストケースから 3 件の不具合が検出された. これらの不具合は, 環境ビューテスト観点到該当する弱電波・通信不可状態でアプリ起動やログインを行うと発生するものであり, このようなテスト観点を補完していなければ見逃した可能性が非常に高い.

適用 1・適用 2 で補完されたテスト観点の数に大きな差が見られたが, TE1 と TE2 のスキルによる違いではなく, テスト対象の機能仕様書の記載粒度が適用 1 では詳細に記載されており, 適用 2 では概要レベルでしか記載されていなかったことが主要な原因である. そのため, 適用 1 ではテスト観点を補完する前のテスト分析で詳細にテスト観点が抽出でき, 適用 2 ではテスト分析でテスト観点がだまかにしか抽出できなかった.

以上の結果より, 適用 1・適用 2 共に標準テスト観点を活用することでそれぞれ 6 と 21 のテスト観点がテストすべきテスト観点として採用されたことが分かる. すなわち, これらのテスト観点は, テスト分析においてテスト観点が抜け落ちていたことを意味し, それらが標準テスト観点を活

用することで補完でき, テスト観点の抜け落ちに一定の効果があることが分かる. よって, Web サービス・モバイルアプリ分野におけるドメインスキルが不足するテストエンジニアに対してテスト観点の抜け落ちなく網羅的に抽出できるようにするという目的に対して一定の有効性が確認できた.

5.4. 考察

表 3 の評価結果が示しているように, 補完したテスト観点を含めて統合テスト観点の段階で品質リスクやテスト実行の実現可能性および工数に配慮し, テスト観点の選別が必要なことが分かる. テスト観点の選別では, テスト観点の抽出段階でテストの必要性の有無を判断するのではなく, 多面的に抽出された多数のテスト観点に対して必要の有無の判断, もしくは優先順位の設定を行うことが重要となる. 本稿で整備した標準テスト観点は, その点において一定の効果が期待できる. また, このようなテストの優先順位付けに関する研究としてはリスクベーステスト(例えば[19][20]など)が挙げられる.

とはいえ, 本稿で述べた標準テスト観点は, 過去に作成したテストスイートから導出されたテスト観点であるため, それらに該当しているテスト観点に限られる. そのため, 十分性という点では, 限定的にならざるを得ない. 今後は, 市場不具合に対して分析を行い, テスト観点を拡充していくことを考えている.

本稿では, テストエンジニアの経験を踏まえたテスト分析・テスト設計の流れに基づき, テスト観点を整理するための枠組みの要件の一つとして, 機能ビュー・環境ビュー・部品ビューという構造を提示した. この構造の視点に関する類似の研究として光永らの研究[3]がある. 光永らは機能・環境・処理の 3 つの分類を提案している. これらの違いは, テスト対象のプロダクトや担当しているテストレベルの違いに大きく依存しているものと推察される.

また, 本稿で述べた機能ビューと部品ビューは相対的なビューであり, 細かく詳細を検討すると部品ビューにも機能に相当するテスト観点, 例えば検索やアプリ連携, 地図表示などがある. これらの分類に関しては, 形式的に一意に定めるのではなく, 対象分野のスコープ内(本研究では Web サービス・モバイルアプリ)における実用性を勘案して, 分類していくことが重要であると考えられる.

本稿の 4.3 節では, 標準テスト観点を活用するために, テスト観点の補完と統合テスト観点のレビューについて使用例を示した. 他の使用例として, 開発成果物のレビューにおいて標準テスト観点を活用することが考えられる. この使用例は今後の課題として検討を進めたい.

6. おわりに

テスト分析・テスト設計においてテスト観点を抜け落ちなく網羅的に抽出することが課題となっている。我々の組織でも同様の課題を抱え、特に Web サービス・モバイルアプリ分野において、ドメインスキルの不足するテストエンジニアを対象としてテスト観点を網羅的に抽出するために、標準テスト観点を整備し、それを活用することでこれらの課題解決を行うことを本稿で論じてきた。

本稿では、Web サービス・モバイルアプリ分野のドメインスキルが不足するテストエンジニアがテスト設計を行うという制約において、そのようなドメインスキルの不足を補うような仕組み・仕掛けが体系立って整備されていないという問題を取り上げた。その問題を解決するために、テストエンジニア5名にヒアリングを行いテスト分析・テスト設計の流れをモデル化した。そして、そのモデルを解析し、テスト観点を整理するための枠組みの要件を明らかにした。続いて、その要件に基づき、枠組みを定め、枠組に従い標準テスト観点の実装を行った。標準テスト観点の実装では、過去に作成したテストスイートを分析し、多様なテスト観点を導出し、枠組みに従いテスト観点を整理した。最後に、本研究の目的に基づき標準テスト観定の適用を行い、本適用の範囲に限られるが、標準テスト観点について一定の有効性を確認することができた。

今後の課題としては、標準テスト観定のテスト観点そのものの精査・拡充や標準テスト観定の運用方法の構築、教育を含めた現場導入の検討などが挙げられる。

参考文献

- [1] “テスト技術者資格制度 Advanced Level シラバス 日本語版 テストアナリスト Version2012.J01”, http://jstqb.jp/dl/JSTQB-Syllabus.Advanced_TA_Version2012.J01.pdf, (参照 2019-3-15).
- [2] ASTER: テスト設計コンテスト, <http://aster.or.jp/business/contest.html>.
- [3] 光永 他, “故障事例によるテスト観点知識ベースの構築とテスト設計への適用”, ソフトウェア品質シンポジウム 2012 (2012).
- [4] 稲葉 他, “テスト設計テンプレートを使用したテストケースの充実”, 第 28 回ソフトウェア品質研究会 (2013).
- [5] 西, “テスト観点に基づくテスト開発方法論 VSTeP の概要”, <http://qualab.jp/materials/VSTeP.130510.color.pdf> (参照 2019-3-15).
- [6] 吉川, “NGT の記法を応用した不具合分析からのテストの補強”, ソフトウェアテストシンポジウム 2018 東京 (2018).
- [7] 池田 他, “マインドマップから始めるソフトウェアテスト”, 技術評論社 (2007).
- [8] 湯本, “テストアーキテクチャの具体例” (<http://www.jasst.jp/symposium/jasst12tokyo/pdf/A2-6.pdf>), ソフトウェアテストシンポジウム 2012 東京 (2012).
- [9] 大林 他, “逸脱分析を用いた要求仕様書からのテスト項目抽出手法”, 情報処理学会論文誌, Vol.57, No.4, pp. 1262-1273 (2016).
- [10] Grochtmann, M., et al. “Classification Trees for Partition Testing”, *Software Testing, Verification & Reliability* (1993).
- [11] 吉岡 他, “見通しのよいテストの段階的詳細化の手法”, ソフトウェアテストシンポジウム 2013 東京 (2013).
- [12] 河野 他, “ソフトウェア開発における外部機能の仕様記述の曖昧さに着目したテスト項目設計法”, 品質, Vol.41, No.2, pp. 143-151 (2011).
- [13] 柏倉 他, “リバーズモデリングを用いたテスト観点標準化の取組み”, ソフトウェアテストシンポジウム 2019 東京 (2019).
- [14] 清水, “「派生開発」を成功させるプロセス改善の技術と極意”, 技術評論社 (2007).
- [15] 鈴木, “W モデルは何か”, ソフトウェアテストシンポジウム 2012 東京 (2012).
- [16] 井芹, “品質・仲間・技術と向き合ってテスト設計技術の力を引き出す”, ソフトウェアテストシンポジウム 2018 九州 (2018).
- [17] Nguyen, H. Q., “Testing Applications on the Web: Test Planning for Mobile and Internet-Based Systems, 2/Edition”, Wiley (2003). (松村 (監訳), “インターネットアプリケーションのためのソフトウェアテスト”, ソフトバンク パブリッシング (2003).)
- [18] The State of Testing Report 2018 (テストの現状調査レポート), https://qablog.practitest.com/wp-content/uploads/2018/10/State_of_Testing_2018_Japanese.pdf (参照 2019-3-15).
- [19] 平山 他, “機能モジュールに対する優先度に基づいた選択的ソフトウェアテスト手法の提案”, 電子情報通信学会技術研究報告, ソフトウェアサイエンス, 01, [98], SS2001-6, 1-8 (2001).
- [20] 石田 他, “Risk based Testing の実践方法と適用事例からの考察－短期間で要求品質を確保する－”, ソフトウェアテストシンポジウム 2008 東京.

付録 標準テスト観点の抜粋

ビュー			観点タイトル	テスト観点		
機能	部品	環境		Level1	Level2	Level3
○			アプリの起動	通知領域からの起動		
○			アプリの起動	ブラウザからの起動		
○			アプリの起動	3D Touch クイックアクション(iOS)		
○			アプリの起動	アプリショートカット(Android)		
○			アプリの起動	複数再起動		
○			ログイン	複数ブラウザからの多重ログイン		
○			ログイン	複数端末からの多重ログイン	OS別アプリの組み合わせ	Android-Android
○			ログイン	複数端末からの多重ログイン	OS別アプリの組み合わせ	Android-iOS
○			ログイン	複数端末からの多重ログイン	OS別アプリの組み合わせ	iOS-iOS
○			ログイン	複数アカウントの切り替え		
○			ログイン	既存アカウントへのログイン		
○			ログイン	連続認証失敗	失敗回数許容上限の超過	
○			ログイン	ログイン状態の維持		
○			ログイン	キーチェーンアクセス		
○			ログイン	仮登録アカウントでのログイン		
○			ログイン	自動ログイン		
○			ログイン	端末起動直後のログイン		
	○		文字列の入力	文字数	最小文字数未済	
	○		文字列の入力	文字数	最小文字数/最大文字数	
	○		文字列の入力	文字数	最大文字数超過	
	○		文字列の入力	文字数	未入力	
	○		文字列の入力	文字種	半角数字	
	○		文字列の入力	文字種	半角大小英字	
	○		文字列の入力	文字種	半角カタカナ	
	○		文字列の入力	文字種	半角記号	
	○		文字列の入力	文字種	半角スペース	
	○		文字列の入力	文字種	全角数字	
	○		文字列の入力	文字種	全角大小英字	
	○		文字列の入力	文字種	全角カタカナ	
	○		文字列の入力	文字種	全角ひらがな	
	○		文字列の入力	文字種	全角漢字	
	○		文字列の入力	文字種	全角記号	
	○		文字列の入力	文字種	全角スペース	
	○		文字列の入力	文字種	絵文字	
	○		文字列の入力	文字種	機種依存文字	
	○		文字列の入力	文字種	制御文字	
	○		メールアドレスの入力	@マークを含まない		
	○		メールアドレスの入力	@マークの前に文字列がない		
	○		メールアドレスの入力	@マークの後に文字列がない		
	○		メールアドレスの入力	複数@マークを含む		
	○		メールアドレスの入力	カンマを含む		
	○		メールアドレスの入力	コロンの含む		
	○		メールアドレスの入力	セミコロンを含む		
	○		メールアドレスの入力	ドメインが存在しない		
	○		メールアドレスの入力	アカウントが存在しない		
	○		パスワードの入力	入力不可文字のコピーペースト		
	○		パスワードの入力	推測されやすい文字列の制限		
	○		パスワードの入力	登録確認入力エリアとの不一致		
	○		パスワードの入力	強度ポリシー違反		
	○	○	OS種別	PC	Windows	サポート対象バージョン
	○	○	OS種別	PC	Windows	サポート対象外バージョン
	○	○	OS種別	PC	macOS	サポート対象バージョン
	○	○	OS種別	PC	macOS	サポート対象外バージョン
	○	○	OS種別	SmartPhone	iOS	最新バージョン
	○	○	OS種別	SmartPhone	iOS	旧バージョン
	○	○	OS種別	SmartPhone	Android	最新バージョン
	○	○	OS種別	SmartPhone	Android	旧バージョン
	○	○	ブラウザ種別	Chrome	最新バージョン	
	○	○	ブラウザ種別	Chrome	旧バージョン	
	○	○	ブラウザ種別	Firefox	最新バージョン	
	○	○	ブラウザ種別	Firefox	旧バージョン	
	○	○	ブラウザ種別	Safari	最新バージョン	
	○	○	ブラウザ種別	Safari	旧バージョン	
	○	○	ブラウザ種別	Edge		
	○	○	ブラウザ種別	InternetExplorer		11
	○	○	ブラウザ種別	InternetExplorer		10
	○	○	ブラウザ種別	InternetExplorer		9
	○	○	ブラウザ種別	InternetExplorer		8
	○	○	ブラウザ種別	フィーチャーフォン専用ブラウザ		
	○	○	端末の通信状態	モバイル通信	通信規格	3G
	○	○	端末の通信状態	モバイル通信	通信規格	4G
	○	○	端末の通信状態	モバイル通信	通信不可	圏外
	○	○	端末の通信状態	モバイル通信	通信不可	機内モード
	○	○	端末の通信状態	モバイル通信	通信不可	SIMカードなし
	○	○	端末の通信状態	モバイル通信	弱電波	
	○	○	端末の通信状態	モバイル通信	ローミング	
	○	○	端末の通信状態	モバイル通信	ハンドオーバー	