

アジャイルソフトウェア開発におけるテストの課題およびその解決アプローチ

河野 哲也^{†1}

概要: アジャイルソフトウェア開発は、ソフトウェア開発の主流な開発形態の一つとなり、その流れは一層増加する傾向にある。本稿では、アジャイルソフトウェア開発宣言における「包括的なドキュメントよりも動くソフトウェアを,」「計画に従うことよりも変化への対応を,」を取り上げテストにおける課題を考察し、Web サービスの開発を事例として、実際の課題とその解決アプローチを概説する。

キーワード: 品質保証, リーンスタートアップ, テストプロセス, 探索的テスト

1. はじめに

現在、アジャイルソフトウェア開発は、ソフトウェア開発の主流な開発形態の一つとなり、その流れは一層増加する傾向にある。特に Web サービスの開発では顕著でありユーザーにサービスやプロダクトを柔軟に提供できるため、アジャイルソフトウェア開発との相性が良い。

一方、弊社では、大小さまざまな種類のサービス・プロダクトの新規開発が頻発しており、今後もその数は増加する傾向にある。それらの開発の多くはアジャイルソフトウェア開発宣言に見られるいくつかのプラクティスの導入が進められており、アジャイルソフトウェア開発の形態を採用したものが多くを占めている。

以上を背景として、それらサービス・プロダクトの品質保証（以降 QA と略す）を行うためのテスト活動全般に話題を移すと、いくつかの課題が挙げられる。ここでテスト活動全般をテストという総称で呼ぶ。例えば、テストは基本的にテスト計画、テスト分析・設計、テスト実装・実行 [1] といった流れであるが、それらのテストプロセスの多くは開発ドキュメントが拠り所になることが多く、例えばテスト設計の多くは仕様をベースに行われることが一般的である。しかし、アジャイルソフトウェア開発宣言の「包括的なドキュメントよりも動くソフトウェアを,」をプラクティスとして導入している場合には、テストに必要十分な開発ドキュメントが用意されることは少ない。

そこで本稿では、まずアジャイルソフトウェア開発宣言における「包括的なドキュメントよりも動くソフトウェアを,」「計画に従うことよりも変化への対応を,」を取り上げ文献[2]の解説を参照しつつ、テストにおける課題を考察する。次に、筆者が担当した新規の Web サービスの開発を事例として、実際の課題とその解決アプローチを概説し議論のたたき台として提出する。

2. 課題の考察

2.1 「包括的なドキュメントよりも動くソフトウェアを,」に対する考察

「進捗も品質も現物で」[2]の項目には、次のような記述がある。

「設計書の完成度など、動くソフトウェア以外のもので測る進捗では、実際に動かしてみるまでは問題に気づけないことがあるため、どうしても表面上の進捗状況把握に留まってしまいます。」

この記述に基づくと、開発ドキュメントの作成は進捗として重きを置かれられないためテスト設計を行うための十分な開発ドキュメントが用意されることが難しく、ソフトウェアの仕様と期待される結果も不明確な状態でテスト設計を行う必要がある。よって、以下の2点の課題に整理できる。

- 1) テスト設計するための十分な仕様がない
- 2) 期待される結果が明確にできない

2.2 「計画に従うことよりも変化への対応を,」に対する考察

アジャイル宣言の背後にある原則の一部として次のような記述がある。

「要求の変更はたとえ開発の後期であっても歓迎します」

この記述に基づくと、確定的に要求を定めることができず、それに紐づく仕様や実装も開発の終盤における変更を許容することを示唆している。そのため、テストにおいては工程の終盤においてもテストケースが確定できないため、テストケースを柔軟に変更できるようにする必要がある。つまり、課題は以下に整理される。

- 3) テストケースが確定できない

3. 事例

本章ではまず事例のコンテキストを開発及び QA の両側面から共有し課題を整理する。次に課題解決のアプローチについて述べる。

^{†1} 株式会社 ディー・エヌ・エー
DeNA Co., Ltd.

3.1 コンテキストと課題

ここで取り上げる事例における開発は BtoC の Web サービスをリーンスタートアップアプローチで開発する形態であり、以下に述べるような特徴がある。

D1) イテレーションを回しながら機能の作りこみを行いビジネスオーナーやプロダクトオーナーが実際のサービスに触りながら実装が終わった機能の確認を行うようなアジャイル開発に近いプロセスである

D2) D1 の確認の過程で仕様やデザインが変更になることが多く、事前に正確に仕様を定義することが難しい

D3) 少人数のチーム編成でドキュメンテーションを通して開発を進めるのではなく多くがコミュニケーションを通して開発を進める傾向が強い

D4) D3 を背景として、要求仕様書や機能仕様書など、開発ドキュメントは必要最低限しか作成されていない

次に筆者が所属する QA 部門の従来のテストプロセスは、テスト計画・テスト設計・テストケース設計・テスト実行・テスト結果報告、といったオーソドックスなプロセスであり、基本的には開発ドキュメントが整備されていることを前提とし、テスト設計では一般的に用いられている大中小項目[3]でテスト観点を整理して、その観点に対して詳細なテストの手順や期待結果をテストケース設計で作成するという流れで進めている。また、テストプロセスは分業されており、テスト設計者とテスト実行者は明確に区別されて運用されている。

以上のコンテキストに基づく第 2 章で述べた課題と同様の課題があり、それに対する解決が必要となった。そのため、本事例では次節に示すようなアプローチを採用した。

3.2 解決アプローチ

本節では、第 2 章で述べた課題にして、本事例で採用したアプローチについて述べる。なお、本事例の開発は現在進行中であるため、その定量的な評価や妥当性の確認などは行っていない。

1) テスト設計するための十分な仕様がない

D1 を進めるうえで、ビジネスオーナーやプロダクトオーナーが確認するために、最新の実装がプロダクション環境に反映されて、QA にリリースされる環境は QA 環境として明確に区別されるような開発スタイルを取っていた。よって、開発ドキュメントを補足的に確認するためにデベロップメント環境を最新の仕様として捉え、それをベースにテスト設計を行うようなアプローチを採用した。D2 で述べたようにデベロップメント環境は頻繁に変更が反映されるため、開発者と密接にコミュニケーションを取りながら仕様を把握するように工夫した。

2) 期待される結果が明確にできない

確定的な仕様および明確に定義できる期待結果を基準と

して詳細に設計したテストケースに基づくスクリプトベースのテストと区別して、ユーザ要求やビジネス要求に照らし合わせてそれをベースにテストチャータを用いる探索的にテストを実施するようなアプローチを採用した。例えば、テスト目的として「ユーザが迷いなくエントリーできることを確認する」を設定し、テストチャータとして「ユーザのペルソナ」を設定するような探索的テストを実施した。

前者のスクリプトベースのテストは Verification 視点のテストであり、後者の探索的テストは Validation 視点のテストとして区別するとわかりやすい。両者を区別しつつ、バランスよく割り振ることが本事例ではテストングの重要な活動の一つであった。

3) テストケースが確定できない

本事例では、詳細なテストケース設計を戦略的に省略し概要的なテストケースに留めることで変更柔軟に対応できるようにした。また、詳細なテストケースを設計するための大きな目的の一つは、テスト設計の意図をテスト実行者に正確に伝えるためである。よって、テスト設計者がテスト実行を行うようなアプローチを採用し先に述べた分業を行うことはせずに、テスト設計の意図を漏らさずテスト実行ができるようにした。

以上で述べたアプローチは本事例で採用した非常に限られた範囲におけるアプローチであり、今後の議論の小さなたたき台になればと考えている。

4. おわりに

本稿では、アジャイルソフトウェア開発宣言における「包括的なドキュメントよりも動くソフトウェアを」、「計画に従うことよりも変化への対応を」を取り上げテストングにおける課題を考察した。次に Web サービスの開発を事例として、実際の課題とその解決アプローチを概説した。

参考文献

- [1] “テスト技術者資格制度 Foundation Level シラバス 日本語版”, http://jstqb.jp/dl/JSTQB-SyllabusFoundation_Version2011.J02.pdf, (参照 2018-8-03).
- [2] “アジャイルソフトウェア開発宣言の読みとき方”, <https://www.ipa.go.jp/files/000065601.pdf>, (参照 2018-8-03).
- [3] 吉岡 他, “見通しのよいテストの段階的詳細化の手法”, JaSST 2013 Tokyo