若手エンジニア
のための

ソフトウェアテストの勘所

バルテス株式会社

はじめに

ソフトウェアテストは、成果物の品質を対外的に証明するための非常に重要なエビデンスです。そのため、開発者の意図を理解し、品質を担保できるテストを積み重ねていく必要があります。

プロジェクトに配属されたばかりの若手テストエンジニアは、各工程が持つ意味や重要性を理解しないまま、日々の業務に忙殺されることもあるでしょう。また、基本的なテスト手法の知識はあっても、それを適切に扱うための視点や考え方が身についていなければ、スキルを高めていくことは困難です。

先輩社員がつきっきりでこれらを教育できれば良いですが、常に人的リソースが不足しがちなソフトウェア開発の現場において、教育リソースの確保は難しいものです。

そこで本資料では、若手のテストエンジニアが最低限知っておくべきソフトウェアテストの勘所を15ページの資料にまとめました。

この資料を読むことで、若手エンジニアはソフトウェアテストのノウハウを得ることができ、先輩エンジニアは効率的な若手教育を行うことができるでしょう。

新人テストエンジニアの教育や、マネジメント層による新規メンバーへの啓蒙活動の一環として、ぜひご活用ください。

目次

1. ソフトウェアテストとは何か？

ソフトウェアテストは、一言でいうならソフトウェアの品質を担保するための「最後の砦」です。形になったばかりのソフトウェアには、開発者が意図していない不具合がいくつも含まれています。これら大小さまざまな不具合を探し出して修正し、ソフトウェアの最終的な品質を高めるには、以下2つの視点をもった作業が求められます。

「検証（Verification）」と「妥当性評価（Validation）」

検証（Verification）とは、「正しく製品を作っているか」の確認です。

もう少し具体的に言うと、「仕様書に記載されている機能が適切に実装されているか」を確認する視点です。つまり、設計どおりの動作が実現できるかを確認するのです。例えば、会員制Webサイトへのログインページを制作しているとき、仕様書に「ID入力・パスワード入力・ログインボタンをクリックし、会員ページに遷移する」旨が記載されているとしましょう。このとき、この3つを順番に行って、無事会員ページに遷移できるかを確認します。

一方、妥当性評価（Verification）とは、「正しい製品を作っているか」の確認です。言い換えると、「顧客の要求を満たした製品であるか」を確認する視点です。ここでいう顧客要求とは「処理時間（待ち時間）」や「使いやすさ」であることが多いです。例えば、「ログインボタンをクリックしてから会員ページに遷移するまでの時間（秒数）」や「入力項目のわかりやすさ（GUIの使いやすさ）」などが該当するでしょう。

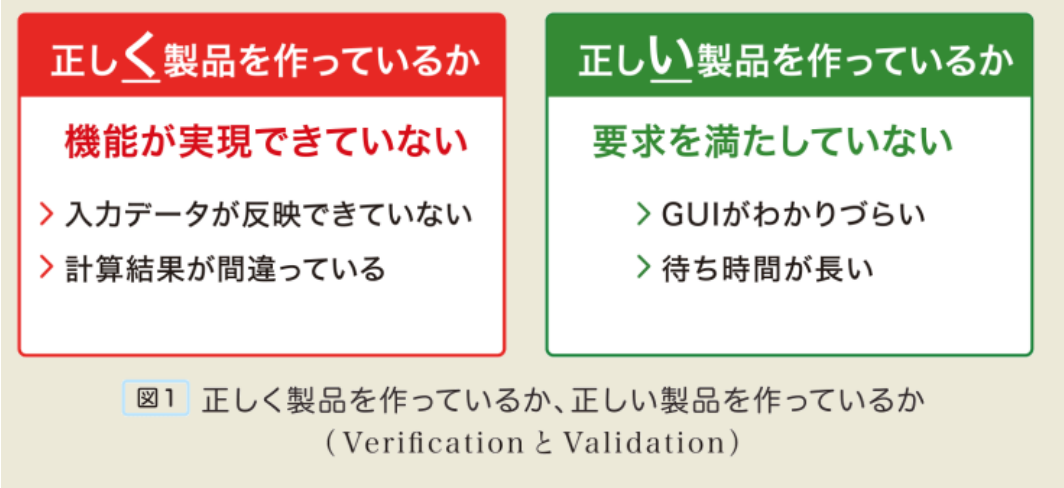

図1 正しく製品を作っているか、正しい製品を作っているか（VerificationとValidation）

1. ソフトウェアテストとは何か？

「当たり前品質」と「魅力的品質」を担保する視点

当たり前品質とは、「顧客の基本要求」を満たす品質です。顧客の基本要求とは「できて当然である一方、できないと不満を感じる」事柄と言い換えられます。スマートフォンであれば、「通話機能」「メール機能」「Web閲覧機能」など、一般的にベーシックな機能だと考えられているものが該当します。

これに対し、魅力的品質とは「顧客の変動要求と潜在要求を満たす品質」です。変動要求は「顧満たすことで顧客の評価が変動する事柄」、潜在要求は「満たすことで顧客の期待を上回る事柄」と言い換えられます。つまり魅力的品質を担保できれば、顧客からの信頼を得られやすくなるということです。

この2つの品質は「狩野モデル」と呼ばれ、さまざま製品の企画・開発に用いられています。

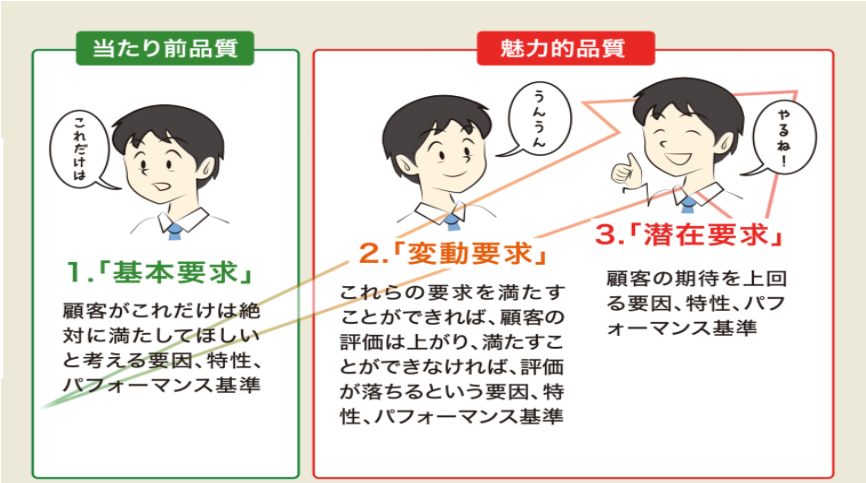

図2 狩野モデル

「検証と妥当性評価」「当たり前品質と魅力的品質」を担保する視点があれ、ソフトウェアテストが持つ「最後の砦」としての役割を果たしやすくなるでしょう。

2.ソフトウェアテスト各工程の勘所

全工程共通の勘所

　では、実際のテストにおける勘所を解説します。まずテストの全工程を把握しておきましょう。左の図は、テストプロジェクトの全体図を表したものです。

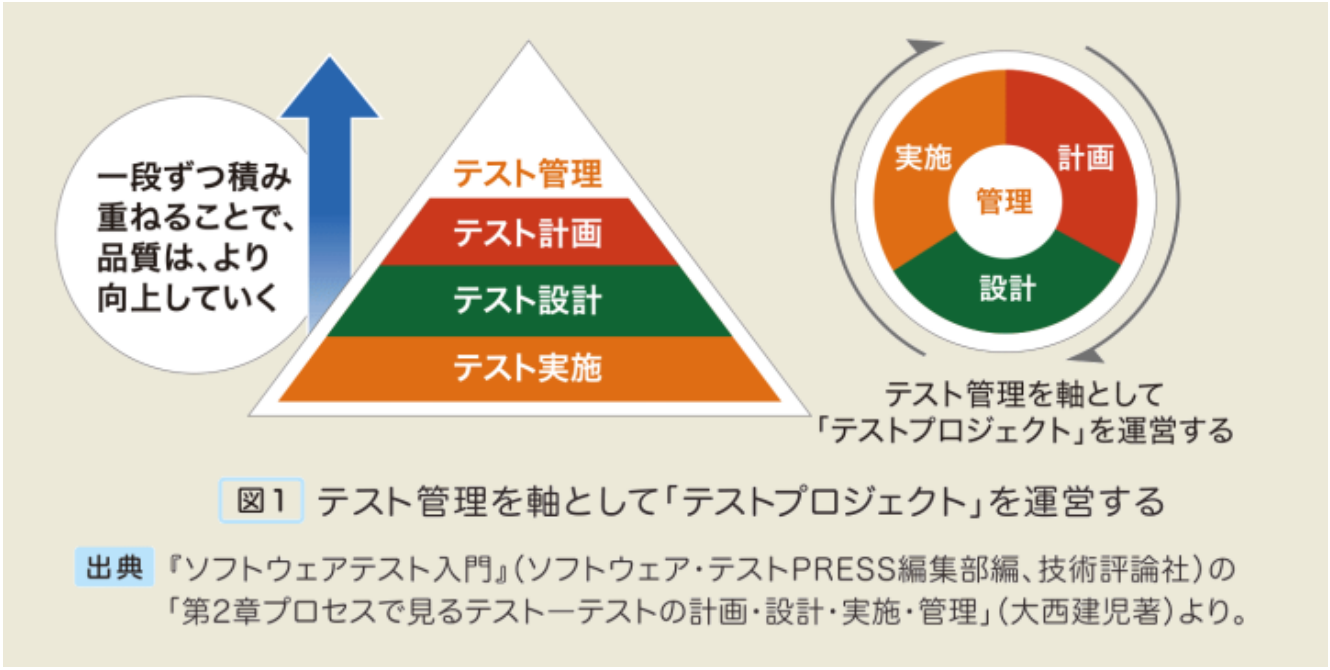

図1 テスト管理を軸として「テストプロジェクト」を運営する

出典 『ソフトウェアテスト入門』(ソフトウェア・テストPRESS編集部編、技術評論社)の「第2章プロセスで見るテスト―テストの計画・設計・実施・管理」(大西建児著)より。

　高品質なテストを効率よく「実施」するためには「テスト設計」が必要です。また、テスト設計を活かすためには「テスト計画」が重要になります。さらに、テスト計画を現実的なものとするためには、綿密なテスト管理が求められます。加えて、右の図のようにテスト管理を中心に置き、計画・設計・実施をサイクルさせるのがプロジェクトと考えてください。

　ではテストプロジェクトの全容を把握したところで、各工程に共通する勘所を整理していきます。

2.ソフトウェアテスト各工程の勘所

1.チームメンバーとの協調性を重視する

これはテストエンジニアというよりも、社会人としての心構えに近いといえるでしょう。テストは単独の作業だけで成立するものではありません。常にチームメンバーと知識やノウハウを交換しつつ、作業を進めていきます。また、スケジュール遅延が起こった場合には、他メンバーへの応援を頼むこともあるでしょう。こういった「相互扶助」の有無が、テストの品質に関わってきます。さらにソフトウェアの仕様でわからないことがある場合には、開発者へ情報提供をお願いすることも珍しくありません。チームメンバーや開発者とは、常日頃から良好な関係を保ち、いつでも応援・アドバイスが求められる環境を築いておきたいところです。

2.情報共有およびユーザー視点を忘れない

テスト計画書はチームメンバーのみならず、開発者とも共有しておくようにしましょう。また、開発計画の共有も要請したいところです。開発者とテストエンジニアが相互に情報共有を行うことで、スケジュール遅延や予期せぬ仕様変更に対応しやすくなるからです。また、実際にソフトウェアを使用するユーザーが、「何を・どの程度」求めていて、「どういった用途に使用するか」は、常に意識するようにしましょう。これらは要件定義書や基本設計書に記されていることが大半です。したがって、開発者との情報共有は、ユーザー視点の把握にも役立つのです。

3.重点的にテストする箇所を把握する

ソフトウェアには「バグの多発地帯」が存在します。ソフトウェアテストの分野では「20%の領域に80%のバグが存在する」という説もあるほどです。バグが集中しやすい箇所としては、「ソフトウェアの継ぎ目」が良く知られています。つまり「機能と機能の境目」や「状態遷移が発生するタイミング」です。こういった箇所は、重点的なテスト項目として認識しておきましょう。

4.準備・プロセス・作業量を意識する

テストの成否は「準備」「プロセス」「作業量」で決すると言っても過言ではありません。準備とは、テストに取り掛かる前に仕様書や設計書を読み込み、テストケースが持つ意味を理解することです。また、プロセスは「テスト設計・テスト計画の質」、「作業量」はテストにかかる人員や時間と言い換えられるでしょう。これら3つを常に意識しながら、適宜PDCAサイクルにかけ、改善していくことが重要です。

2.ソフトウェアテスト各工程の勘所

5.実際のテストスケジュールでイメージをつかもう

下の図は、実際にソフトウェアテストの現場で用いられるスケジュールの一例です。縦軸が人員の数、横軸が時間と考えてください。また、機能の重要度は、A＞B≧Cです。これらを踏まえ、この図のポイントをまとめると次のようになります。

○機能の優先度に応じて長方形の大きさ（=工数）が変わる。

○不具合の修正結果を確認する回帰性テスト（リグレッションテスト）でも優先度は変わらない。したがって、最重要機能Aは、スケジュールが遅延しても一定以上の工数をかけてテストされる。

○修正していない部分の品質を確認するデグレードチェックは、チェック範囲が広くなるため工数が大きくなりがちである。

○最後に「バグの多発地帯」を狙い撃ちするアドホックテスト（ランダムテスト）で機能間のバグをチェックしている。

このように、機能の優先度や作業の範囲・性質によって、適切にリソースを割り当てていくことが重要です。

2.ソフトウェアテスト各工程の勘所

テスト計画段階の勘所

テスト計画は、「仕様調査と抽出」「テスト計画作成」という2つのステップで構成されます。仕様調査と抽出では、仕様書・設計書の読み込みや開発者からの情報提供により、テスト対象となる機能や範囲、優先順位を決定します。

次のテスト計画では、各機能テストに対し、リソース（人員・時間など）を配分します。全ての機能をあらゆる角度から網羅的にテストする「100%のテスト」は、実現不可能です。したがって、効率と品質担保を両立させるような計画が求められます。また、計画を作成する際には、以下のようなポイントにも注意したいところです。

1.ゴールをイメージする

テスト計画を作成したら、「ゴールのイメージ」を明確にしましょう。つまり、「対象範囲」「テスト終了基準」「成果物」の可視化です。これら3つを可視化することで「何がどのようにできていればOKか」をクリアにするのです。

2.優先順位付けと開発者との合意

繰り返すようですが、「100%のテスト」は実現不可能なものです。そのため、開発者と協議しながらテストの優先順位を決定していく必要があります。つまり、「コストとリスク」を天秤にかけながら「どのテストにどれだけのリソースを投入するか」を決定するのです。例えば、顧客の基本要求に紐づく機能では、テストを行わないリスクが非常に高くなります。したがって、基本要求部分のテストには、多くの人員と時間を投入するという方法が考えられるわけです。

2.ソフトウェアテスト各工程の勘所

3.リスクを考慮する

実際のテストでは、テスト範囲の変更や重大不具合の発覚、上流工程の遅延（仕様書・設計書確定の遅延）など、さまざまなリスクファクターが考えられます。これらをあらかじめ考慮し、実際にリスクが顕在化したときの対処法を検討しておきたいところです。例えば、応援を要請できるよう体制を整えたり、休日でもテストが進むような人員配置を検討したり、といった方法が考えられます。

4.開発初期からテストを実施する

「1:10:100の原則」をご存じでしょうか。これは、開発フェーズごとのテスト修正工数に対する格言です。つまり、設計フェーズで見つかった不具合の修正工数を1としたとき、単体テストでは10、リリース後の修正コストは100になる、という考え方です。これは、開発の早い段階からテストを行うことが、いかに重要かを示しています。上流工程にも積極的に関与し、早期からテスト計画を練ることが重要です。

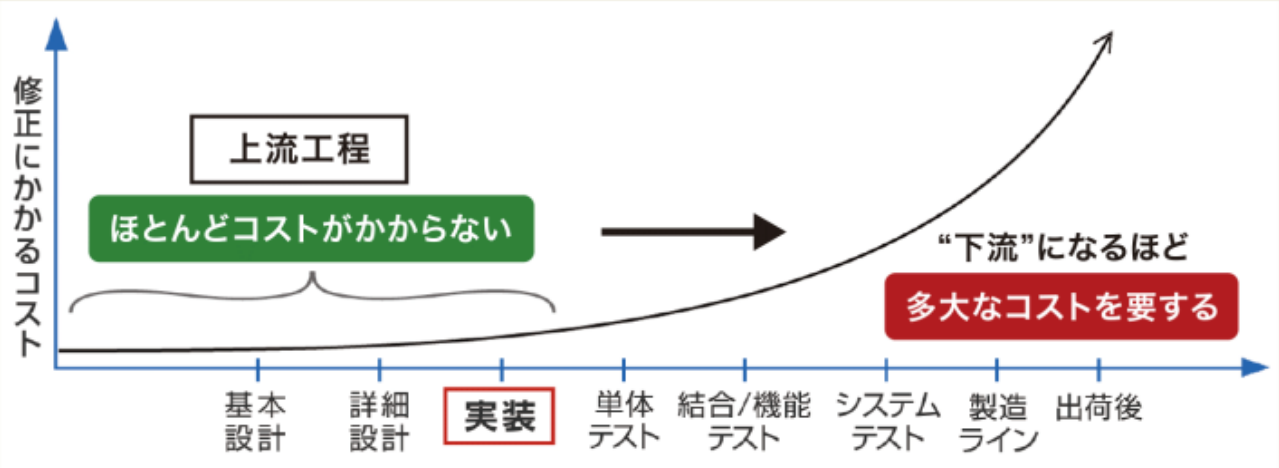

図4 修正にかかるコスト

2.ソフトウェアテスト各工程の勘所

テスト設計段階の勘所

テスト設計では、テスト計画をもとに「どのような観点から」「どのテスト技法を用いて」「どれだけテストするか」を決定します。つまり「テストの設計図を作成する段階」です。具体的には、「テスト設計仕様書」と「テストケース」の作成が含まれます。

1.開発者へのレビューに一覧表を活用する

テスト設計では、テスト計画書とともに、開発者が作成した仕様書・設計書もベースになります。ただし、開発者がテストを考慮して仕様書・設計書を作成しているとは限りません。

仕様書・設計書に加筆する形でテスト設計を行い、一覧表で見やすく整理しておきましょう。開発者がテスト項目を確認しやすくなり、テストの抜けや漏れを未然に防ぐという効果が期待できます。

2.リスクとコストを意識したテスト技法の使い分け

テスト技法は、テストエンジニアにとって心強い味方です。例えば、ブラックボックステストで行われる各テスト技法を駆使すれば、劇的な工数削減効果とテスト品質を両立させることができます。これには、状態遷移図や状態遷移表を使った「状態遷移テスト技法」や、設定値や設定項目を組み合わせる「組み合わせテスト技法」が該当します。

3.テストの目的、背景、理由を把握する

目的、背景、理由を把握していないままにテスト設計を行うと、本来不要であるはずのテストにも手を広げてしまいがちです。過去の不具合やテスト履歴をなどを確認し、不必要なテストを行うことがないように注意しましょう。

2.ソフトウェアテスト各工程の勘所

テスト実施段階の勘所

最後に、テスト実施段階の勘所を紹介します。実際のテスト作業は、労働集約的な作業であるがゆえに、人的コストが高く遅延が発生しやすいフェーズといえます。最終的な納期の遅れを防ぐには、人員のやりくりや関連部署との調節が欠かせないでしょう。また、テスト実施者は、常に以下のような事柄を意識して遅延の防止に努める必要があります。

1.ソフトウェアのバージョン確認

テストは、不具合の発見・開発者による修正とバージョンアップ・再テストの繰り返しです。そのため、常にテスト対象のソフトウェアバージョンを最新に保ち、適切な環境下でのテストを心がけてください。バージョンが異なる環境でのテストは、全て無効になってしまう可能性があるからです。不具合報告書へのバージョン記載も忘れないようにしましょう。

2.仕様書・設計書・テスト対象への理解

不具合を見つけるのがうまいテストエンジニアは、仕様書や設計書を熟読しています。テスト対象の構造・機能に対する理解が深いのです。これは、「ある不具合に関連した不具合」を芋づる式に発見する能力につながるほか、テストケースの不備を検知することにも役立ちます。

2.ソフトウェアテスト各工程の勘所

3.不具合か仕様かの切り分け判断は開発者へ

一見、仕様のように見えても実は不具合だったということは珍しくありません。もちろん、逆のケースも存在します。「検証（Verification）」と「妥当性評価（Validation）」を行っても判断がつかない場合は、すぐに開発者へ判断を仰ぐようにしましょう。

4.ユーザー視点からの「機能改善提案」

すでに紹介したとおり、テストエンジニは「妥当性評価（Validation）」によってユーザー視点でのテストを行います。つまり、ユーザー視点での機能改善提案が可能な職種なのです。こういった改善提案の積み重ねが「魅力的品質」へ、さらには顧客からの信頼獲得につながるのです。

3.おわりに

早期かつ頻繁なテストを

本資料では、ソフトウェアテストの概要や各工程における勘所を若手テストエンジニア向けに紹介してきました。

工程共通の勘所・テスト計画段階の勘所・テスト設計段階の勘所・テスト実施段階の勘所と理解が進んだのではないでしょうか。冒頭でも述べたように、テストは「品質を守る最後の砦」です。

しかし、「最後」という意味をはき違えてはいけません。

品質の高いテスト計画、テスト設計をたて、できるだけ早期から頻繁にテストを行うことで製品全体の品質を高めることが、テストエンジニアに求められる役割といえます。

本資料を参考に、ソフトウェアテストの各工程をもう一度見直し、効率的かつ高品質なテストを目指してみてください。

また、テスト技法の詳細は弊社の別資料でも詳しく解説しています。あわせて読むことで、さらに理解を深めることができるでしょう。ぜひ参考にしてください。

基本の「き」シリーズ１～４｜テスト技法編

VALTES
FOR QUALITY CONFIDENCE

ソフトウェアテストや品質のお悩みや課題に関してお気軽にお問い合わせください

バルテス　検索

バルテス株式会社

03 - 5210 - 2080

valtes-inquiry@valtes.co.jp

平日 10:00 ～ 18:00