

PART 1

ソフトウェアテストの
概要



テストはなぜ必要なのか 不幸な“手戻り”を呼ぶテストの不備

1-1 はじめに

近年、ITが社会のすみずみに浸透したおかげで、システム障害やソフトウェアの不具合がもたらす経済的／社会的影響は計り知れないものになっています。そのような背景の下、バグの混入したソフトウェアが出荷されるのを食い止める“最後の砦”として、「テスト」の重要性が急速にクローズアップされています。ソフトウェアテストとはどんなことを踏まえてどんなことをすればよいのでしょうか。

パート1の目的は、

「テストに興味は持っているのだけれど、勉強する時間がとれない」

「普段、テストを我流でやっているけれど、基本をきちんと押えておきたい」

などと日頃感じておられる方々に、ソフトウェアテストの基本を知っていただくことです。大まかにテストの全体像が理解できるよう、以下の3つの事柄を説明します。

- テストフェーズの基本的な流れ
- テスト手法や技法の基礎知識や分類の仕方
- テストのプロを目指すために必要なスキル

テストに対して基本的なスキルをすでに習得している方なら、パート1はざっと目を通してもらえれば十分でしょう。これから本格的にテストについて学ぼうという方は、ひと通りパート1を読まれたうえで、ほかのパートで解説さ

れている手法や技法の詳細について、興味のあるところから読まれることをお勧めします。

1-2 ちゃんとテストしていますか？

これからソフトウェア開発エンジニア（以下エンジニア）になろうとしている方、すでにエンジニアとして日々開発にいそんでいる方など、現在の立場や職務内容によって、テストに対するスタンスはさまざまなことでしょう。本書を読んでいるということは、少なくともテストに注目していると思いますが、ここで自問してみてください。

「ソフトウェアテスト（以下テスト）にきちんと取り組んでいるか？
手を抜いているか？」

どうでしょう。

「それなりにがんばっている」

「工程が厳しい中でも、できるだけテストを実施するようにしている」

「動くものを作るのに精一杯で、正直テストにまで手が回っていない」

いろんな声が聞こえてきそうですが、実際のところ最後の例が一番多い現実であると、たくさんのエンジニアの方とお話を通じて筆者は感じています。

プロですから簡単には「いい加減」にやっているとは認めたくないかもしれませんが、ただ本音を言うと、プログラミングにかけるエネルギーと同じように、テスト技法を駆使し、信頼性の高いテストケースを設計し、テスト実施も効率的に実施していると言い切れますかと聞かれれば、「そこまでは……」となるのかもしれませんが。

技法云々ではなく、そもそもテストなどというものはコーディングに付随しているものでデバッグの延長線に行うこと……という程度に捉えられていることすらあります。「まずは動くコードを書くのが先決」という考えだと、「動くことを確認すること」＝「テストすること」というような誤った理解をしていても不思議ではありません。

テストを軽視したり後回しにしてしまうと、結局、作ったソフトウェアの不具合に常に悩まされることになるでしょう。不具合が多くて予定通りに進まないプロジェクトの状況を見ては、大きなため息をつくプロジェクトマネージ

ヤが、あなたの周りにいませんか？ 出荷後の不具合対応に辟易^{へきえき}とするエンジニアの姿（あなた自身かもしれません）を見ることも珍しくはないでしょう。

1-3 テストは開発工数を増やすだけなのか？

残念なことに、テストを真面目にすると工数が増えるだけでタイトな開発期間から十分な工数をテストのために捻出することは難しいのだと、多くの管理層に思われているフシがあります。本当に、テストを真面目にやると工数は増えるのでしょうか？ テストを実施することは、手間ばかりかかるものなのでしょうか？

いいえ、違います。テストを効果的にかつ効率よく実施する術を身につけられれば、開発はスムーズになりますし、不具合に悩まされることもなくなってきます。何より、開発工数が最適化されていきますので、組織的な費用対効果も上がります。それだけではなく、結果としてエンジニア個人にとっても楽ができるようになるのです。

これから、悩めるエンジニアやデスマーチプロジェクトに関わる方、テストに興味を持ち始めた方など広い層に対して、テストの基本的な考え方や取り組むためのノウハウを紹介していきます。テストにより開発を「楽」にできる知恵があることと、工夫できることを全体を通して理解していただければと思います。

1-4 テストの印象は？

みなさんは普段、ソフトウェアテストや、テストを実施しているエンジニアに対して、どのような印象を持っていますか？ コンピュータ関連の学科を専攻している学生さんに、「これからIT関連の職に就くとして“テスト担当者”という職種に抱くイメージを一言で表現してみたい」と聞いてみたところ、次のような回答がありました^{注1}。

大変そう、きりがない、心身ともに疲れそう、めんどくさそう、頭使いそう、

注1) ソフトウェアテストシンポジウム (JaSST'05) 「クイズテスト関係100人に聞きました (アンケートまとめ)」を参照。本データはJaSSTのサイトである「<http://www.jasst.jp/archives/jasst05e.html>」からダウンロード可能。

難しそう、嫌われそう、いじめられそう、重要な仕事、やりがいがありそう、理想的、地味、地道、縁の下の力持ち、人気なさそう、細かい、厳格、油断しない、びりびりしてそう、華やか、スター、人気者 [回答の多かった順]

どれもこれも、テストに対する印象として、私たちエンジニアが抱えているものとそんなにはかけ離れてないでしょう。試しに学生さんの印象を、ネガティブな言葉でつないでみましょう。

「テストは大変で、面倒くさくて、地味だから組織内で嫌われている」

まったくもって損な役回りにしか見えてきませんが、^{あた}中らずといえども遠からずでしょう。では、ポジティブな言葉でつないでみると、どうなるでしょう。

「頭を使う重要な仕事で、ときには縁の下の力持的な存在でもある。また細かなところまで気の配れる組織内での人気者がやる仕事」

どうでしょう。まったく違った印象になることに驚きますよね。どうせテストを実施するなら、後者を目指すべきなのはいうまでもありません。組織の中で人気者だということは、技術的にも頼れる存在ですから、いっばしのスキルを持っているエンジニアだということでしょう。

1-5 テスト担当の立場から

開発組織の中で、テスト担当者と呼ばれているエンジニアの方々へ、開発者に遵守してほしいことや直してほしいこと、スキルを磨いてほしいことを聞いてみると、次のような答えが返ってきました^{注1}。

スケジュール、コミュニケーション、仕様の理解、単体テスト、ロジックの組み立て方、ドキュメンテーション、プロジェクトのルール遵守、変更通知、スキル [回答の多かった順]

上位3つからは、工程のしわ寄せがテストフェーズにくることや、テスト項目を考えるための前提となる、仕様がきちんと伝わってこないことに関する不満が見えてきます。本来プログラムを行うものとしてきちんと実施すべき事柄についても、テスト担当者は不満を持っているようです。単体テストがきちん

とされていないコードだと、ロジックの組み立て方がなっていないでしょうし、まともにドキュメンテーションもされていないということでしょう。

開発者は、どうしても実装技術に目が行きがちです。しかし、いくらテクニカルに満足がいくコーディングができたとしても、動かなかったり、使い物にならなかったり、どうなって動いているのかわからないような代物では、テストで満足な結果が得られないということを、アンケートの回答は示唆しています。

ここまでは、エンジニアや開発者といった「個」の視点からテストに対して持つ印象などを述べました。次に、私たちは何のためにテストを行っているのか、「組織」的な視点から考えます。

1-6 テストで防ぎたいこと

コードやドキュメント、データなどあらゆる場所に潜むバグを見つけることは、テストの大きな役目の1つです。テストで見つけきれないと、バグを作りこんだプロセスにまで遡^{さかのぼ}って修正を行わなければなりません。このことを「手戻り」と言いますが、実際どういったことが原因でソフトウェアの手戻りが発生しているのでしょうか。

経済産業省が組み込み分野にて調査した資料^{注2}によると、手戻り原因のトップは「要求仕様の不備 29.9%」、次に「ソフトウェア設計上の不具合 18.6%」、続いて「仕様書の不備 17.4%」となっています（図1-1）。

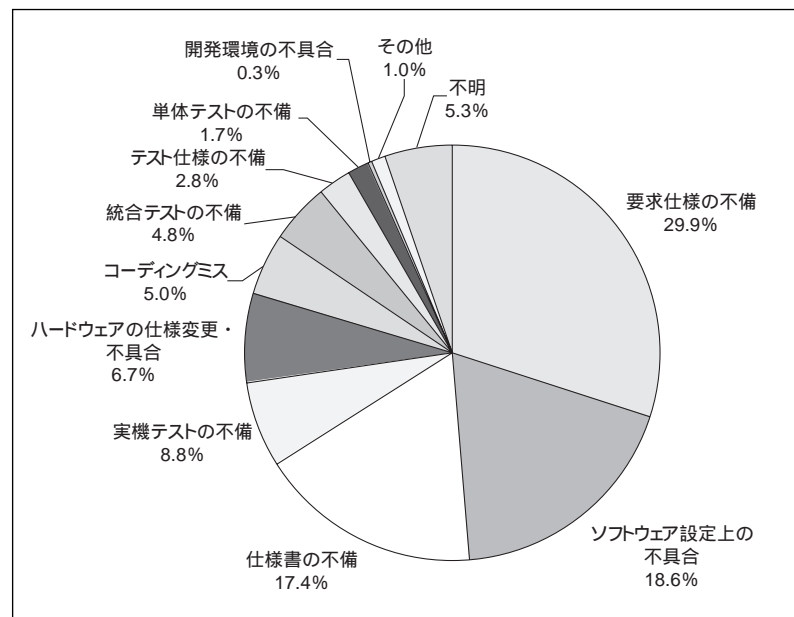
1-6-1 開発での手戻りの影響

手戻りが発生すると、その影響はソフトウェア開発に広く波及します。それだけではなく、開発工数や予算、エンジニアのモチベーションに対しても悪影響を及ぼします。マイナスの影響がないように開発組織は、できる限り効果的に手戻りを防ぐ努力を行います。

その手段として、プログラムやシステムを動作させて確認するテスト（以下、動的テスト）を行います。コードレビューやドキュメントのレビューやウォークスルーなど、机上での論理的な確認やエンジニア同士の相互チェックによって、プログラムやシステムを動作させずに確認するテスト（以下、静的テ

注2) 経済産業省 商務情報政策局「2004年版 組込みソフトウェア産業実態調査 報告書」より。この傾向は、本書をまとめている2008年現在も変わっていないようです。

●図1-1 手戻りの原因（経済産業省調べ）



スト）も行います。このように、動的／静的の両方のアプローチを用いることで手戻りを防がなければならないのです。

動的／静的どちらのテストにも関与していないエンジニアは皆無ですから、私たちは何らかの形で、テストに関与しているといえるでしょう。

1-6-2 テストの不備による影響

案外見落とされがちなのが、テストの不備による手戻りです。

同じく経済産業省の調査によると、テスト関係の不備（実機テスト、テスト仕様、単体／結合テスト）は、合わせると15.9%となるそうです。この数字は意外だったでしょうか、それとも実感があるでしょうか。テストをやみくもに実施したり、十分なスキルなしにテスト設計を行うと、手戻りを防ぐどころか、その原因にすらなりかねないことがおわかりになるでしょう。

「テストは単にプログラムやシステムを動かして、仕様合っているかどうか確認するだけだ」。エンジニア個人や組織がこの程度の認識しかなかったらどうなるでしょう。バグによる問題だけではなく、テストそのものの不備によっても、開発プロセス全体で手戻りに悩まされるはずですよ。