

第 30 章 テストの方法

ソフトウェア開発におけるテストの位置づけ

テストは、ソフトウェア開発の後半に位置づけされる作業である。しかしその計画立案と準備は、開発のごく最初の段階から始めなければならない。今有効な「共通フレーム 2013」の 1 つ前の版である共通フレーム 2007 では、ではテストについて以下のような記述があり、この考え方が明確に示されていた¹[IPA07]。

『開発プロセス』の最初のアクティビティである「プロセス開始の準備」には「開発プロセス実施計画の作成」というタスクがあって、ここでこれから行う開発についての全ての活動の計画を立てることが要求されている。当然この計画には、テストの計画も含まれる。あるいはテストの基本的な計画は、全体の開発計画とは別枠で立てる方がよいかもしれない。

共通フレーム 2013 ではテストの種類として、使っている言葉は違うが、単体テストから運用テストまでの通常の 4 段階のテストが定義されている²。しかし仮にこれから開発するソフトウェアが「重要インフラ等システム」と呼ばれるようなソフトウェアで通常以上の高信頼性が要求される場合、この共通フレームの「開発プロセス」を修整してもっとテストの段階を増やさなければならない³。

テストの計画

ISO 9001 では、全ての領域で何か作業を始めるときには、まずしっかりと計画を立て、その計画をチェックし、次に十分な準備を行い、そしてその計画に基づいてその作業を実施し、その結果をチェックして不具合があれば是正し、さらにその結果を基にしてプロセスの改善を図ることを要求している[JIS15b]。これは、ソフトウェアの開発でも変わらない。この作業の中に、当然テストも含まれる。

テストの計画は、2 段階のものが必要である。全体としてのマスター・テスト計画と呼ばれるものと、個別のテストについての詳細な計画である[CRA02]。

マスター・テスト計画ではこのソフトウェアの開発で、全体としてどのようなテストを、どのような順序で、具体的に何を対象に実施するのか、といったテストの全体像と、相互のテストの間の関連、分担などを明確にする。システム・テストの範疇に含まれることになる非機能要件の確認のためのテストも、これに含まれることになる。それを受けて、個別のテストについての計画が策定されることになる。

個別のテストの計画は、そのテストの対象物が作成される時には完了していることが望ましい。つまり要求仕様書を作成している段階では要求仕様書通りにソフトウェアが開発されていることを確認する運用テスト、あるいは受け入れテストについてテスト計画を作成し、設計段階では設計されたとおりのものが実現されていることを確認するシステム・テストの計画を作成する、というように進めるのが好ましい。そうすることで、確認すべき対象物が何で、そ

¹ 共通フレーム 2013 については、第 12 章で記述した。

² 単体テストや運用テストなどのテストの種類については、第 31 章で記述する。

³ 1 つ段階のテストで除去できる欠陥は 30% 見当とされている。4 段階のテストだけで必要な品質を確保できない場合には、テストの段階を増やす必要がある。

れをどのように確認することができるのか、といったことを十分に深く考えることができる。

別のいい方をすると、これはテストの実施とは逆の順序で計画を立てることを意味する。これはさらに、次のような点でも役立つ。つまりテスト計画には、テストの開始基準と終了基準の項目が不可欠である。最後のテストの終了基準は、そのソフトウェアの開発プロジェクトの終了基準と合致している必要があるが、その開始基準はその前のテストの終了基準と合致していなければならない。それが順次前のテストに継続していったら、全体としてのテストの実施に漏れや重複がなくなる、ということになる。

ソフトウェアのテストはいくつもの種類があり、それぞれ確認すべき内容と方法が異なるので、それぞれのテストでこの計画策定から終了までの一連の手順が必要になる。

個別のテストについての計画の中では、次のような事項を明確にする[GRA07]。

- テストの目的／目標
- テストのポリシーと戦略（もし必要なら）
- テストの対象物（テスト・アイテム）
- テストの対象範囲
- テストの方法
- スケジュール
- テスト担当者
- 使用するツール
- 開始基準／終了基準
- ……

テストのポリシーとは、例えば「単体テストでは、LOOP の継続／停止を含む全ての判断について、True と False の両方にそれぞれ最低 1 回は分岐するケースを用意することを通して、結果として全てのステートメントを 1 回以上通過させる」というような、テストについての基本的なルールをいう。また戦略とは、例えば「システム・テストは品質管理担当の管理者の配下の、開発プロジェクトとは独立したテストチームが実施する」というような、全体的、かつ上位からのアプローチを指す。これらはいずれも、ソフトウェアを開発する組織としての基準としてあらかじめ定められているのが望ましく、その場合には原則全ての開発プロジェクトがその基準に従うことになる。仮に組織としての基準がない場合には、この計画の中でそれらを明確にすることが望まれる。あるいは組織としての基準があってもその基準とは異なるものを要求したい場合には、やはりこの計画の中で記載することになる。

テストでは、全てのケースをテストし尽くすということとはできない。つまりテストでは、テストできる範囲や領域、時間に制約があり、その制約の中で最も効率よくテストをすることが望まれる。そのため、テスト計画の中でテストを行う範囲を限定し、明示する必要がある。

計画はその作業が実施されている間、テストの進捗の管理にも使用される。計画はその進捗の管理や、進捗について問題があればそれを発見し、是正できるものであることが必要である。

この計画は、「テスト計画書」として文書化される[IEEE98g]。計画策定の時点を含めて、テスト期間中に作成される文書については後述する。

テストの分析と設計

テストの分析と設計とはテスト計画に基づいて、何を、どのようにテストするのかを明確にしたテスト・シナリオを作り、それをベースにテスト・ケースを設定し、テスト・データを用意することをいう。テストのために特別の環境などが必要であれば、それも合わせてこの段階

で準備する。

テスト計画の中でテストする領域が明示されているので、その領域の中でソフトウェアの品質を短時間に最も高めるための方法を考え、まずそれをテスト・シナリオとして明示する。

このテスト・シナリオを、テスト・ケースにブレークダウンする⁴。ここで明らかにしたテスト・ケースを、さらに具体的なテスト・データに変える。ここで重要なことはテスト・データを準備した時に、ソフトウェアが作成すべき出力を併せて明確にしておくことである。テストを実施している時に正しい出力は何かを考えたり、思い出そうとすると、時間が無駄になるだけでなく、思い違いが生じて、結果として欠陥の発見に支障が出ることもある。

さらにこの後、テスト実施の手順を決める。

このテストの分析と設計の作業の結果、「テスト設計書」、「テスト・ケース」と「テスト手順書」の3つの文書が作成される[IEEE 98g]。さらにテストの実施に備えてテスト対象物の状況を明確にするために、「テスト対象の送付レポート」がプログラムなどのテストの対象物を用意した人から送られてくる。

テストの実施

上で述べたような形でテスト・データを用意したのなら、テストの実施はデータを入力できる状況を作ってデータを入力し、得られた出力を当初想定したものと順次チェックすることで、淡々と行われることになる。

この間に「テスト・ログ」と呼ばれる文書を作成してそこにテストの状況を記録する。さらに出力が作成できない、あるいは当初想定したとおりの出力が得られない場合には、テスト・データを用意したときに思い違いをしていないかをチェックして、それでもやはり実際の出力がおかしい場合には「インシデント・レポート」を書いて欠陥（インシデント）の発見を報告する。ソフトウェアが修正されてその欠陥が除去されたという報告が戻ってきた時には、確認テストを行って本当に欠陥が除去されていることを確認する。

併せて一通りのテストが終わった後で回帰テストを実施して、欠陥を除去した際に、それ以前にOKであることを確認した機能が損なわれていないことを、改めて確認する。

テストの実施中に作成する文書は、「テスト・ログ」と「インシデント・レポート」の2種類である。

テストの終了

用意した全てのテスト・データの確認が終わり、発見した欠陥の除去も全て終わっていることが確認できて、計画時に定めた終了基準を満たしたなら、そのテストを終了することができる。

テストを終了するときには、「テスト報告書」を記述してそのテストを締めくくる。この間に作成される文書は、この「テスト報告書」だけである。

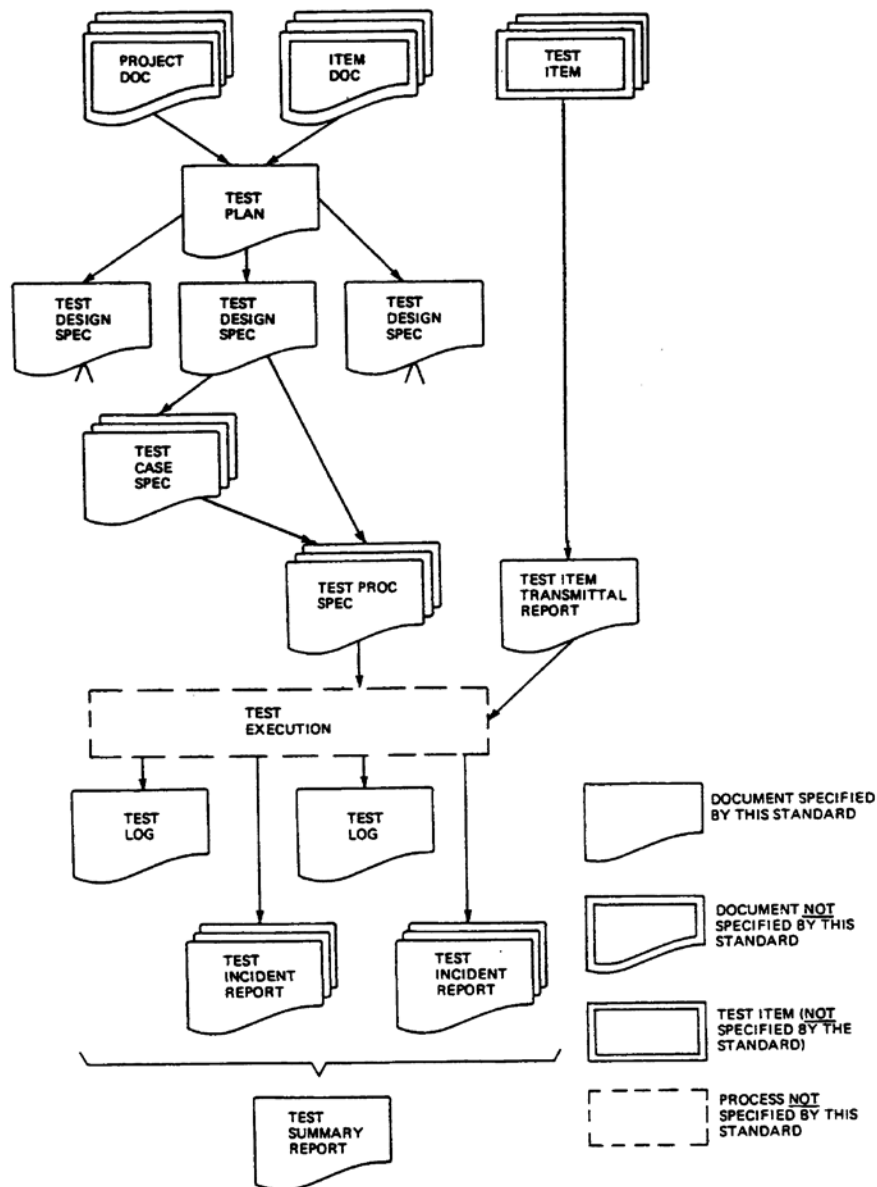
このテストの段階で発見された欠陥を基に、ソフトウェア開発のプロセスを改善することができる。しかしこれはテスト担当者やテストチームの仕事ではなく、ソフトウェア・エンジニアリング・プロセス・グループ（Software Engineering Process Group : SEPG）あたりの仕事と考える方が妥当である。したがってここでは、このテーマについてこれ以上の議論はしない。

⁴ テスト・ケースの設定方法については、次の第 31 章で記載する。

テストで作成する文書類

計画の策定から終了までの全テスト期間中に作成される文書には、以下の 8 種類がある。

- 「テスト計画書 (Test Plan)」
- 「テスト設計書 (Test Design Spec.)」
- 「テスト・ケース (Test Case Spec.)」
- 「テスト手順書 (Test Proc. Spec.)」
- 「テスト対象の送付レポート (Test Item Transmittal Report)」
- 「テスト・ログ (Test Log)」
- 「インシデント・レポート (Test Incident Report)」
- 「テスト報告書 (Test Summary Report)」



図表 30-1 テストで作成される文書と文書間の関係 ([IEEE 98g]より)

これらは、IEEE Std 829 という規格の 1 つで詳細に定義されている[IEEE98g]ので、ここでは文書名の紹介だけにとどめ、詳細な内容は割愛する。これらの文書間の関係は、図表 30-1 に示す⁵。

テストの限界

前述の通りテストでは、ソフトウェアがそのライフ・サイクルの間に処理する可能性がある全てのデータを事前にテストし尽くすということができない。テストのために割くことができる時間は、著しく限られている。これが、テストの最初の限界である。

従ってテストを実施するときは、実際にテストする範囲や領域を最初に明確にしてから始めなければならない。逆のいい方をすると、最初からテストをすることをあきらめなければならない領域が出てくることがある。どの領域をテストし、どの領域をあきらめるのかについては、リスク分析を行って、リスクの小さな領域のテストを割愛するという提案がなされている⁶[GRA07]。

2 つ目の限界は、テストではテスト・データを用意してそれを入力し、作成された出力を的確にチェックして正しいものが作成されていることが確認できた領域では、「欠陥がない」ということができることである。つまりテスト・データを用意しなかった領域、テスト・データを入力したけれど結果を充分には確認しなかった領域などでは、テストが終了してもまだ欠陥が残っている可能性がある。経験の浅いソフトウェア技術者は、よく「テストが通ったから OK です」ということがある。これは、大きな間違いである。

繰り返しになるが、テストが終わっても、まだソフトウェアの中には欠陥が残っている可能性がある。このことを銘記し、ソフトウェア技術者は常に謙虚でなければならない。

ホット・モジュールの取り扱い

ソフトウェアの欠陥の分布はパレートの法則に従って、「80%の欠陥が 20%のモジュールに集中している」とされている。この欠陥が集中しているモジュールを、「ホット・モジュール」と呼ぶ。

レビューやテストでこのホット・モジュールを発見した場合には、もっとレビューやテストに時間をかけてこのモジュールの欠陥を取り除こうと努力するより、そのモジュールを破棄して、そのモジュールの機能とインタフェースを変更しないで再度設計からやり直し、再びホット・モジュールにしないように注意しながら新しいモジュールを作る方が良い、とされている。

上で述べたように、完全なテストを実施することは不可能である。ある特定のモジュールから多くの欠陥が発見されたということは、それでそのモジュールの中の欠陥が全て見つかったと考えるべきではなく、そのモジュールの中にたくさんある欠陥の一部がこれまでに見つかり、もっと多くの欠陥がまだそのモジュールの中に残っている、と考えなければならない。これが上で述べた方法を推奨するに当たっての、根底にある考え方である。

テスト・プロセスの改善

ISO 9001 の考え方に従えば、テストのやり方を改善することでよりの確に、効果的にテ

⁵ IEEE Std 829-1998 は、2017 年（平成 29 年）1 月 25 日現在廃止されてしまった。

⁶ リスク管理（分析）については、第 51 章で述べる。

トを実行することができるようになる[JIS15b]。ISO 9001 は単に PDCA サイクルを回すことでこのプロセスを改善しようとするものであるが、SW-CMM や「開発のための CMMI v3」はもっとソフトウェア開発に関連する作業に焦点を絞ってプロセスを改善しようとするところに特徴がある⁷[CMM10a]。

ここで、この CMM 流のアプローチでテスト・プロセスを改善しようとする考え方がある[KOO99]。CMM とこのテスト・プロセスに焦点を絞っての改善の違いについては、CMM ではテストは単に 1 つ（実際は「開発のための CMMI v3」では、レビューまで含めて「検証」と「妥当性確認」の 2 つ）のプロセス領域にしか過ぎず、したがって CMM だけではテスト・プロセスの改善の議論は充分ではない、との立場を取っている。具体的には、この立場を取っている人たちはテスト・プロセスを全体として 20 個の領域に分け、0 から 13 までの 14 段階にレベルを分けて、順次テストに関わるプロセスの改善を図るというアプローチを取っている[KOO99]。

例えばこの 20 個のプロセスの領域の中に「テスト戦略」というものが定義されており、これには 4 段階のレベルを設けている。この「テスト戦略」のプロセスの目的は、「最も重大な欠陥をできるだけ早期に、できるだけ低コストで発見すること」であるとしている。

そして最初のレベルは「単一高位レベルの戦略」で、1 つのテストだけでこの目的の実現を目指そうとする。次のレベルではこれが「高位レベルテストを組み合わせた戦略」に変わり、3 つ目で「高位レベルテストと低位レベルテストあるいは評価を組み合わせた戦略」になり、そして最後は「全てのテストレベルと評価レベルを組み合わせた戦略」に至る。

つまりこの最初のレベルでは、それぞれのテストで独立して目的を達成しようとするが、レベルが上がるにつれて他のテストと手分けし、連携し、分担し合って欠陥を発見しようとし、最後はレビューまで含めた全てのテスト関係の活動を総合してソフトウェアの品質向上に当たろうとの考え方を取っている。当然といえば当然のことであるが、このアプローチはすばらしい。

前述の「テスト戦略」も含めて、20 個のプロセス領域を図表 30-2 に示す[KOO99]。

図表 30-2 テスト・プロセス改善での全プロセス領域 ([KOO99]より)

No.	プロセス領域	No.	プロセス領域
1	テスト戦略	11	コミットメントと意欲
2	ライフサイクルモデル	12	テスト役割と訓練
3	関与の時点	13	方法論の範囲
4	見積りと計画	14	コミュニケーション
5	テスト仕様化技法	15	報告
6	静的テスト技法	16	欠陥管理
7	尺度	17	テストウェア管理
8	テストツール	18	テストプロセス管理
9	テスト環境	19	評価
10	オフィス環境	20	低位レベルテスト

⁷ ISO 9001 については第 39 章で、CMMI については第 40 章で、それぞれ述べる。

キーワード

重要インフラ等システム、単体テスト、結合テスト、システム・テスト、運用テスト、テストの計画、テストの分析と設計、テスト・シナリオ、テスト・ケース、テスト・データ、テストの手順、テストの実施、インシデント、テストの終了、テストの限界、パレートの法則、ホット・モジュール、テスト・プロセスの改善

略語

SEPG : Software Engineering Process Group

規格

共通フレーム 2013、ISO 9001 : 2015、開発のための CMMI v3

参考文献とリンク先

[CMM10a] CMMI 成果物チーム、「開発のためのCMMI® 1.3 版 CMMI-DEV, V1.23

CMU/SEI-2010-TR-033 ESC-TR-2010-033 より良い成果物のためのプロセス改善」、カーネギー・メロン大学ソフトウェア工学研究所、2010年

この資料は、次の URL からダウンロードできる（確認日：2017年（平成29年）1月25日）。

<http://cmmiinstitute.com/resource/japanese-language-translation-of-cmmi-for-development-v1-3/>

[CRA02] Rick D. Craig, Stefan P. Jaskiel 著、宗雅彦監訳、成田光彰訳、「体系的ソフトウェアテスト入門 アジャイル開発時代のテスト計画、準備、実行、プロセス改善まで」、日経 BP 社、2004年10月25日。

この本の原書は、以下のものである。

Rick D. Craig, Stefan P. Jaskiel, “Systematic Software Testing,” STQE Publishing, 2002.

[GRA07] ドロシー・グラハム他著、秋山浩一他訳、「ISTQB シラバス準拠 ソフトウェアテストの基礎」、センゲージ・ラーニング（株）、2008年。

この本の原書は、以下のものである。

Dorothy Graham, Erik van Veenendaal, Isabel Evans, and Rex Black, “Foundations of Software Testing : ISTQB Certification,” Cengage Learning, 2007.

[IEEE 98g] IEEE-SA Standards Board, “IEEE Standard for Software Test Documentation IEEE Std 829-1998,” The Institute of Electrical and Electronics Engineers, Inc. 1998.

[IPA07] 独立行政法人情報処理推進機構ソフトウェア・エンジニアリング・センター編、「共通フレーム 2007～経営者、業務部門が参画するシステム開発および取引のために～ ソフトウェアライフサイクルプロセス SLCP-JCF2007」、（株）オーム社、2007年。

[JIS15b] 日本工業標準調査会審議、「品質マネジメントシステム—要求事項 JIS Q 9001 : 2015 (ISO 9001 : 2015)」、日本規格協会、平成27年。

[KOO99] Tim Koomen 他著、富野壽監訳、「テスト・プロセス改善 —CMM 流実務モデル—」、構造計画研究所、2002年。

この本の原書は、以下のものである。

Tim Koomen, Martin Pol, “Test Process Improvement,” IQUIP informatica B. V., 1999.

(2008 年 (平成 20 年) 9 月 20 日 初版作成)

(2011 年 (平成 23 年) 1 月 9 日 一部修正)

(2016 年 (平成 28 年) 5 月 18 日 一部修正)

(2017 年 (平成 29 年) 2 月 1 日 一部追加)