

第4章 テスト

4.1 テストの種類

単体テストが完了したモジュールを結合し、テストを行います。テストには以下の種類があります。

種類	概要
結合テスト	単体テストが完了したモジュールを結合し1つのプログラムとして、機能テスト、性能テスト、ストレステストを実施する
システムテスト	実際の業務処理手順に従って処理を実行し、機能要件を満たしているか、またシステム全体の整合性や操作性、業務処理との接続性等を確認する

結合テスト、システムテストで行う内容には、以下の種類があります。

種類	概要
機能テスト	機能間のインタフェースや基本設計書に規定されている機能が正しく動作するかを確認する 結合テストおよびシステムテスト内で実施する
性能テスト	プログラムの性能（応答速度、リソースの消費状況など）を測り、想定する性能を満たしていることを確認する 結合テストもしくはシステムテスト内で実施する
ストレステスト (負荷テスト)	プログラムに高い負荷を与え、性能低下、機能障害、データ破壊など想定外の状況が発生しないこと（もしくは想定外の状況が発生する条件）を確認する 結合テストもしくはシステムテスト内で実施する
妥当性 確認テスト	プログラムもしくはシステムが、利用者が意図している動作（要求仕様で規定している動作）を満たしているかを確認する システムテスト内で実施する

4.2 結合テスト

結合テストは単体テストが完了したモジュールを結合し1つのプログラムとして、モジュール間のインタフェースや基本設計書に規定された機能が正しく動作することを確認するテストです。

4.2.1 結合テスト計画の立案

結合テストのテスト方針やテスト環境、使用ツール、テストデータ、スケジュール、体制、留意事項などを計画して結合テスト計画書にまとめます。テスト方針については通常行う機能テストの他に、性能テスト、ストレステスト（負荷テスト）の実施有無を検討します。これらはシステムテストで行うこともあり、どの工程で実施するのかを検討する必要があります。

また、結合テストに必要となるリソース（ハードウェア、ソフトウェア）の利用スケジュールも検討が必要です。大規模な開発の場合、複数のサブシステムやプログラムについて同時にテストを行うケースがあり、その場合はどの時期にどのプログラム、サブシステムがリソースを使用するのか計画する必要があります。

4.2.2 結合テストの品質管理

(1) 品質目標値の設定

テストを実施するにあたり、最初に潜在する不具合の予測値を設定します。この予測値は、テスト対象のソースコード Step数やFP数から算出されます。具体的には、過去の類似システムや経験値から導き出されたStep当たりの不具合発生件数を使用し算出します。

(2) 実績の管理

発見した不具合数をプログラム単位で管理します。もし、実績値が目標値を大幅に下回る、もしくは上回る場合、または不具合数累計の推移に収束傾向が見られない場合は、その原因を分析し目標値やテスト方法の見直し、追加テストなどを行います。

(3) 品質改善

テスト結果を評価し、次工程で注意すべき点など申し送り事項を整理します。

(4) 回帰テスト

単体テストと同様に回帰テストの実施範囲を検討します。

FP (Function Point)

開発工数の見積もり手法のひとつであるファンクションポイント法 (Function Point Method) により求められたソフトウェア規模を計算するための点数。

4.2.3 結合テストの実施手順

(1) テスト観点およびテスト内容、データの作成

テストを行う観点、テストの内容、およびテストで使用するデータを作成します。単体テストと同様に、テスト観点、テストの内容、およびテストで使用するデータについては、必ずレビューを行います。

(2) テスト環境、ツールの準備

テストに必要な環境を準備します。テスト対象プログラムが動作するために必要となるプログラム、ミドルウェアなども合わせて準備が必要です。

(3) テストの実施および結果の確認

テストスケジュールに沿ってテストを実施します。テスト結果が期待する結果と異なる場合、その原因を調査しプログラムに不具合がある場合は修正後、再テストを行います。

また、テスト実行時の実行結果をエビデンスとして保存します。

4.3 システムテスト

4.3.1 システムテストとは何か

実際の業務処理手順に従って処理を実行し、要件定義時の機能要件を満たしているか、またシステム全体の整合性や操作性、業務処理との接続性等を確認する工程です。

システムが複数のサブシステムで構成される場合は、そのサブシステムやハードウェア、通信環境、DBなどのミドルウェアをすべて組み合わせて実施します。

単体テスト・結合テストを終了した後、必ず実施します。

(1) システムテスト計画の立案

テスト作業の方針や作業方法、資源等を「システムテスト計画書」としてまとめ、作業全体の目標設定と統合を行います。計画は事前に策定し、顧客責任者の承認を得るとともに、併せて顧客関係者と開発担当者に周知する必要があります。

システムテスト計画書の記載項目の例

記載する項目
<ul style="list-style-type: none">・テストの目的・テスト項目の設定方針・テスト基準の設定・テスト範囲、テスト形態、テスト手順の概案・テスト実施体制（組織、役割分担）・検証の方針・検証責任者、承認者と手続き・検証結果の保存期間・管理方法（進捗、ドキュメント、ライブラリ、マシン）・不良摘出時の対処方法・ハードウェア、ソフトウェアの構成、手配方法・テストID、パスワードの準備配布・テストデータの準備方法・ツールの要否と準備方法・実施工数の概算の見積り・実施期間、テストスケジュール

(2) システムテストの品質管理

(a) 品質目標値の設定

テストを実施するにあたり、最初に潜在する不良の予測値を設定します。この予測値は、過去の類似システムや経験値から導き出し、テスト作業で抽出すべき目標値とします。

(b) 管理曲線の設定

一般的な経験では、テスト項目数に対する検出不良数の累積数はゴンペルツ曲線を描くと言われています。その曲線を踏まえた目標値に対応する下限、上限の2つの管理曲線を設定します。

(c) 管理曲線による実績の管理

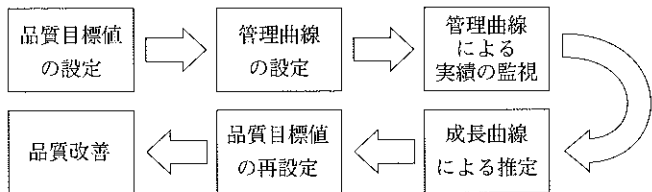
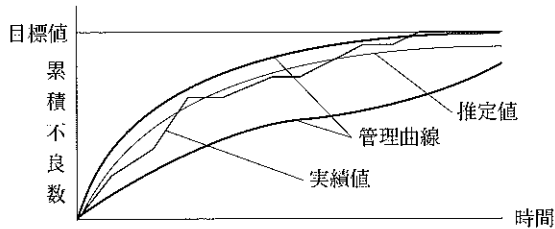
発見した不良数を折れ線グラフで書き込んでいきます。もし、実績値の折れ線グラフが管理曲線の上限、下限を超えた場合は管理曲線の設定、目標値、テスト方法に問題があるとみなし、問題点を追求し再設定します。

(d) 成長曲線モデルによる推定

信頼度成長曲線は最初は増加曲線となり、終盤は曲線が緩やかになります。この特性を実績の管理曲線にあてはめて残存不良の予測とテスト終了の予測を行います。

(e) 品質改善

テスト結果を評価し、次システム開発時の品質向上のフィードバックデータとします。



第4章 テスト

(3) システムテストの実施手順

システムテストを実施するための作業手順として次の作業項目があります。

作業項目	作業の内容
テスト要件の検討	システム仕様、コンピュータ処理、業務手順を確認し、テスト対象を確定する
テスト項目と手順の決定	機能面、例外処理、性能、信頼性と障害対策等の確認要件を設定し、テスト項目及びチェック内容を試験仕様書に作成する 試験仕様書は、何を入力し(入力条件)、どのような結果を期待するか(検証方法)を明確にする
実施体制の決定	テストの実施体制及び分担、要員を選出し作業内容を徹底する(SE、PG、顧客、管理者、承認者)
テスト環境の準備	ハードウェア(サーバ、ネットワーク)、データベースやソフトウェアの準備及び環境の設定
ツールの準備	ツールの開発あるいは購入
テストデータの準備	マスタファイルの登録やデータベースの登録や入力データを作成する データベース登録プログラムの利用と移行プログラムを開発する
テスト期間中のID、パスワード	情報セキュリティ(個人情報)に十分な配慮と保護、配布と回収
テスト日程の策定	部署、システムの関連が複雑なときは詳細の日程に展開する日別、時間別の小日程が望ましい
テスト実施および検証結果の保存	テスト結果(リスト、ハードコピー)は、原資料としてプロジェクト完了まで保存する
テスト結果報告書の作成	不具合の影響範囲、対策内容を明確化する 品質評価を客観的に判断する 品質管理資料の添付が望ましい ・不良発生、試験項目消化曲線 ・プログラム不良、修正報告書 ・プログラム別品質評価表等
テストの完了	システムテストの可否判定

システムテストのチェックポイント

- テストデータの作成およびシステムテストは、テスト計画に基づいて行っているか
- システム要求事項を網羅したテストケースを設定しているか
- 開発当事者以外の公正かつ客観的立場の者が参画しているか
- テスト結果を記録および保管しているか
- テスト結果は開発、運用、保守および顧客責任者が承認しているか
- 個人情報などの機密情報に十分な配慮と保護が実施されているか
- テスト用のID、パスワードは終了後回収されているか