テスト種別カタログ

第1.3版

2022年3月31日

変更履歴

※1.1版以前の履歴は、以下を参照してください。

テスト種別&観点カタログ変更履歴_第1.1版以前.pdf

No.	版数	変更日	区分	変更項目の番号・名称	変更内容
1	1.2版	2018/9/27	変更	表紙	ライセンス情報を付与
2	1.3版	2022/3/31	変更	テスト種別 　性能テスト 　ストレステスト 　ボリュームテスト	性能テスト計画ガイドに合わせて、整合性が取れる表現に変更。

【テスト種別】

★：項目の詳細は「補足」シートを参照

テスト種別		インプットとなる主要な成果物（★）	目的	品質指標（★）	観点有無（★）
構文チェック		・コーディング規約	・ソースコードについて、コーディング規約違反、潜在バグ等を発見する。(※1、※2)	指摘数	-
機能テスト	モジュール単位	・システム機能設計書 ・共通コンポーネント設計書	・クラス、関数に実装された各業務ロジックについて、設計との差異を発見する。 ・SQLにより取得した結果について、設計との差異を発見する。	テストケース数 / ソースコード行数 カバレッジ（C0、C1など）	○
	画面・バッチ単位	・システム機能設計書 ・共通コンポーネント設計書	・1画面、1バッチごとの実行結果について、設計との差異を発見する。	テストケース数 / ソースコード行数	
	ユースケース単位	・画面遷移図 ・システム機能設計書	・複数の処理をまとめて1つの業務的な意味を持つまとまりにする場合、その実行結果について、設計との差異を発見する。 ・帳票やインターフェースファイルの出力結果について、設計との差異を発見する。	テストケース数 / ソースコード行数 不具合数 / ソースコード行数	
データ互換性テスト		・システム機能設計書	・下記の違いに伴う動作不良を発見する。 ・データファイルを出力するアプリケーションの種類 ・データファイルを出力するアプリケーションのバージョン ・データファイルを入出力するアプリケーションの設定	不具合数/ソースコード行数 不具合数/FP	△
業務シナリオテスト		・業務フロー ・イベント一覧 ・状態遷移モデル定義 ・業務ルール定義 ・データフロー ・システムフロー ・ネット・ジョブフロー	・業務のシナリオ(※)に沿って、要求仕様や業務の実現性に関する問題点を発見する。 ・業務のシナリオ(※)に基づいた操作手順について問題点を発見する。 ※：アプリケーションを跨ぐ業務や、外部システムと連携する業務も含む。 また、ジョブフローの設定についても検証する。	テストケース数/ソースコード行数 テストケース数/FP 不具合数/ソースコード行数 不具合数/FP	△
構成テスト		・システム機能設計書	・テスト対象のアプリケーションが、同一環境上にある他のハードウェア、ソフトウェアから受ける影響(※)を発見する。 ※：OS、モバイル実機、ブラウザ、プロトコルなどの組み合わせにより発生するレイアウト崩れや動作不良	不具合数/ソースコード行数 不具合数/FP	△
セキュリティテスト		・セキュリティ要件定義	・不当な行為に対する脆弱性を発見する。	不具合数/ソースコード行数 不具合数/FP	○
性能テスト		・性能・拡張性要件定義	・テスト対象システムに平常時やピーク時の負荷を加えた場合のパフォーマンスについて、性能目標値との差分やボトルネックとなる処理を発見する。必要に応じて、SQL、1画面・1バッチなどの単位でも実施する。 ・納品後にリソース増強が計画されている場合、リソース増強後のパフォーマンスについて、想定との差分を発見する。	ターンアラウンドタイム レスポンスタイム スループット CPU使用状況 メモリ使用状況 ディスクI/O ネットワークI/O エラー率	△
ストレステスト		・性能・拡張性要件定義	・以下の場合に発生するシステムの動作不良を発見する。 ・システムが処理できる最大負荷量を目標とし、徐々に負荷を加えた場合 ・システムが処理できる最大の負荷を、短時間で一気に加えた場合 ・システムが処理できる最大負荷量を超えた負荷を加えた場合 ・トランザクションの入り口が複数あるような同時処理が物理的に可能な環境で、複数の処理を同時に実行した場合	不具合数/ソースコード行数 不具合数/FP ターンアラウンドタイム CPU使用状況 メモリ使用状況 ディスクI/O ネットワークI/O	△
ボリュームテスト		・性能・拡張性要件定義	・将来の増分を考慮したデータ量やデータサイズによる負荷が加えられた場合に発生するシステムの動作不良(※)を発見する。 ・将来の増分を考慮したデータ量やデータサイズの負荷により、リソースが不足している場合に発生するシステムの動作不良(※)を発見する。 ※：正常終了しない状態以外に、想定通りのエラーリカバリが行われない状態を含む	不具合数/ソースコード行数 不具合数/FP ターンアラウンドタイム CPU使用状況 メモリ使用状況 ディスクI/O ネットワークI/O	△
ロングランテスト		・性能・拡張性要件定義	・アプリケーションの長期稼働を見据え、一定の負荷状態が長時間続いた状態で発生するメモリリークなどの異常を発見する。	CPU使用状況 メモリ使用状況 ディスクI/O ネットワークI/O	△
障害テスト		・可用性要件定義 ・運用・保守要件定義	・障害発生時の対応について問題点を発見する。	不具合数/ソースコード行数 不具合数/FP 障害からの復旧時間	△
運用シナリオテスト		・運用・保守要件定義 ・バックアップ要件定義 ・監視要件定義 ・セキュリティ要件定義 ・環境運用設計書	・通常時の運用サイクル、監視（ログ）について問題点を発見する。 ・運用や保守の手順について問題点を発見する。	不具合数/ソースコード行数 不具合数/FP 運用時間	△
移行テスト		・移行要件定義 ・データ移行設計書	・移行シナリオ、システムやデータの移行結果、切り戻しの計画、移行所要時間などについて、妥当性、実現性などの問題点を発見する。	不具合数/ソースコード行数 不具合数/FP	-

※1：主要なプログラミング言語だけでなく、データベース言語なども実施対象とする。
例えば、Javaで書かれたソースコードだけでなく、JSP、JavaScript、SQLなどを使用しているコードについても構文チェックを行う。

※2：構文チェックとして、下記のような観点のチェックも実施する。
・コンパイルチェック（ソースコードがコンパイルできるか）
・ソース重複チェック（ソースコードが重複していないか）
・ソース複雑度チェック（サイクロマティック複雑度が適切か）
・非公開API使用チェック（プロジェクトで利用が許可されていないAPIを使用していないか）
・パッケージ依存関係チェック（プロジェクトで許容しないパッケージの依存関係がないか）

【参考情報】

以下のテスト種別は、本書が前提としている要件定義～設計の成果物に含まれない資料をインプットとします。
プロジェクトの状況に応じて、必要があれば追加を検討してください。

テスト種別	インプットとなる主要な成果物（★）	目的	品質指標（★）	観点有無（★）
現新比較テスト	・現行システムの実行結果など	・機能を変えないマイグレーション案件の場合、実行結果について、現行アプリケーションと修正後アプリケーションの差異を発見する。	テストケース数 / ソースコード行数 不具合数 / ソースコード行数	-
ユーザビリティテスト	-	・ユーザーの視点で、操作性やアクセシビリティなどの観点から問題点を発見する。	操作にかかった時間 ユーザーの主観的満足度	△

■「インプットとなる主要な成果物」について

「インプットとなる主要な成果物」には、検証内容の検討で必要になる要件定義～設計の成果物について、主要な例を挙げています。

該当の成果物がないと具体的なテストの実施内容を検討できないため、成果物の入手可否を確認することで、計画したスケジュールのフィジビリティを確認できます。

記載している成果物は、後述の例外を除き、以下のFintanコンテンツで定義されているドキュメントを指します。

・要件定義フレームワーク (2.20)

・Nablarch開発標準

また、例外として、上記には定義されていないがプロジェクトで作成されることが多いと想定した以下のドキュメントも記載しています。

・方式設計書

■「品質指標」について

「品質指標」として、テスト種別ごとに、品質の状態の確認・改善に使用するメトリクスの例を記載しています。

カタログの記載を参考に、プロジェクトで利用する品質指標を選択してください。

選択した品質指標は、以下のポイントを考慮して品質評価に使用します。

・反復開発を行う場合、開発の途中（イテレーション単位、スプリント単位など）で取得した数値の推移を評価する。

・工程の途上で取得した数値をもとに、その時点での品質評価と、それ以降の品質対策の検討を行う。

■「観点有無」の凡例について

「観点有無」として、テスト観点カタログの提供有無を記載しています。

記号の意味は下記の通りです。

○	テスト種別に対応するテスト観点を提供する。 プロジェクトで作成するテストケースに近い形式で記載しており、可変部分（[ ]で囲まれた部分）や、前提条件などを見直すことで使用可能となっている。
△	テスト種別に対応するテスト観点を提供する。 ただし、該当のテストは、テストケースにアプリケーション固有の要素を多く含むことが想定されるため、検証しておきたい観点のみを記載している。 テスト観点を使用する場合は、補足シートの記載も参考にして、プロジェクトでテストケースの形に成形する必要がある。
-	テスト種別に対応するテスト観点を提供しない。 該当のテストは、検証しておきたい観点にアプリケーション固有の要素を多く含むなど、標準化が困難であるため、プロジェクトで観点の検討から実施する必要がある。