

# ProjectAIにおけるテスト戦略

## Test Strategy for ProjectAI

平松 敦郎

**要約** 長期に渡り複数のテスト工程を経てITシステムを作り上げる大規模プロジェクトは早い時期からテスト戦略を定め、各工程がどのようなテストを担い、最終目標に到達するかを可視化することが重要である。ステークホルダー、プロジェクト管理者、テスト担当者、障害対応者と、テスト工程に携わるプロジェクトメンバー全員が共通認識を持ち、テストに臨むことでプロジェクトの無駄を排し、IT技術者が真に重要と考える作業に集中することができる。本稿では、事例に取り上げるプロジェクトが、テスト戦略書をどのように作成、利用したかを明確にし、テスト戦略がプロジェクトを成功に導いた一要因となったことを明らかにする。

**Abstract** In large scale IT system development that requires multiple test activities over long period of time, it is crucial to create the test strategies to visualize what and why each test activity must do to reach the successful goal. Only by sharing these strategies across all members of the project; from stakeholders, project managements, quality assurance staff, testing engineers to programmers, will enable the project to avoid the loss which occurs whenever there's a need to fill the gap between engineers and management-staff/stakeholders. This report describes the details of test strategies as one of the factor in making this project a success.

### 1. はじめに

メインフレーム上で30年以上の長期に渡り、改良を重ねながら稼働してきた基幹業務システムのオープンシステム化を、大きなシステム環境の転換を図りながらも業務に影響を与えることなく実現することが、ProjectAI<sup>\*1</sup>に課せられた課題であった。2008年4月当時、同等の業務が完全にオープンシステム上で構築された事例が存在しない状況であり、想定される開発規模が大規模かつミッションクリティカルでありながらプロジェクトの開始に踏み切ることができたのは、完全にオープンな環境用に再設計された業務パッケージ“AirCore<sup>\*2</sup>”の開発が終盤を迎えていたことに加え、性能的にも本プロジェクトの顧客が求める規模の処理が可能である見通しが立っていたことが大きい。実際、本プロジェクトはこの業務パッケージを国内仕様向けにカスタマイズすることで発足した。

まず小人数によるプロトタイプ開発(STEP0開発)として、カスタマイズ前に業務パッケージの性能評価とカスタマイズ方式の試行を、6ヶ月掛けて実施した。その結果を踏まえ、実開発の中核となる要員を増員し、開発プロセスの試行(STEP1開発)として要員の教育と開発方式の確立を15ヶ月掛けて実施した。この間に開発生産性算出基準値の定義と検証を行い、実開発(STEP2開発)の計画を作成した。計画は開発規模から、長期に渡るものとなったため、アプリケーションの開発を複数のグループに分割し開発時期をずらす方式を採った。この分割

したグループで実際に開発した機能の数は、開発途中で仕様変更等で廃止したものも含めると、オンライン機能（CUI機能、GUI機能、他シス\*<sup>3</sup>機能）が1,831、バッチ機能が318となった。STEP2開発の最初の開発グループは、主要業務機能の要件定義から単体テストまでを対象に、STEP0とSTEP1開発を経験した要員を中心に発足した。この開発に着手すると同時に次の開発工程の検討において、単体テスト以降のテスト工程の可視化が求められ、プロジェクト全体を見通したテストの考え方や進め方を「テスト戦略書」として取りまとめることとなった。本稿ではProjectAIにおけるテスト戦略書の内容と、この内容を基に実施した個々のテスト工程で効果があった点について述べる。

## 2. ProjectAIの開発計画全体像

日本ユニシスではエンジニアリングプロセスの標準としてISEP\*<sup>4</sup>を制定しており、開発を要件定義、論理設計、物理設計、プログラム開発、結合テスト、システムテスト、移行・導入の七つのフェーズに分割している。テスト戦略書はこのISEPにおける「論理設計フェーズ」で作成され、後続フェーズの開始までに「テスト」の全体像を明らかにし、プロジェクトの進むべき道を指し示すドキュメントとして定義されている。本プロジェクトでもこの考えを踏襲し、テスト戦略書を作成した。

本稿が対象とするSTEP2開発は、RUP\*<sup>5</sup>（ラショナル統一プロセス）のイテラティブ開発に基づき、マスタプランとして要件定義から機能テストまでをGROUP1、GROUP2、GROUP3と3回のイテレーション開発で、まずプロジェクト発足時（2008年4月）の現行機能と同等レベルまでアプリケーションを実装し、次に現行改修取込としてプロジェクト発足時以降に改修された現行機能の改修内容の反映を3回に分けて実施することとした。最後に実際に完成したシステムを用いて、現行と比較し差分の取り込みを実施し、確実に置き換えるべき現行システムと同等の機能を実現した。アプリケーションのイテラティブ開発と並行して、アプリケーションの稼働環境を整備する基盤開発も行い、システムテストを開始する前までに基盤としてのインフラシステムテストを終えた状態まで作り上げた。

テスト戦略書は本格的なテストが開始される「推敲フェーズ」の開始時点において作成し、以降のテスト工程の方針を定義した。但し、長期に渡るプロジェクトであることから、初版では機能テストを主体に記載し、システムテストに関する記述は次の「作成フェーズ」にて第2版として改訂した（図1）。

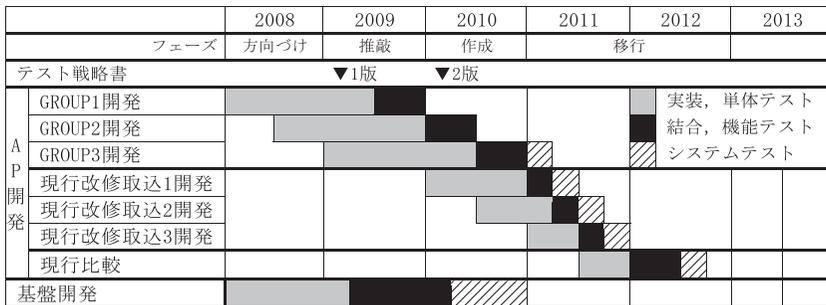


図1 本プロジェクトにおけるテスト戦略書作成時期と主なテストの実施時期

なお、図1のアプリケーション(AP)開発と基盤開発以外にも、アプリケーションのフレームワーク開発、端末・外部システムとの間の「ゲートウェイ開発」、外部システムとの接続性を確認する「他システム接続テスト」、移行ツールの開発と、移行手順そのものを定義する「移行開発」がプロジェクトの一部として行われたが、これらはアプリケーションとの直接的な関係が薄く独立性が高いため、個別に検討する必要があると判断し、それぞれが開発計画書を作成することとした。テスト戦略書にはそれらの概要およびアプリケーションのテストと関わる部分のみ記載した。

### 3. プロジェクトにおけるテスト戦略書の位置づけ

大規模プロジェクトのテスト工程は、長期に渡り、細分化され、広範囲かつ、多くの要員の参加を必要とする個々独立したテスト工程の集合体として実施される。各要員が参加するテスト工程がプロジェクト全体のどの位置にあるのか、前後のテスト工程との関係はどうなっているのか、当該テスト工程は何に注視し実施すればよいのか、何を成果とするのか、何ができれば当該テスト工程は成功したと評価されるのかなど、巨大プロジェクト故に意識することが困難な「目標」としてのプロジェクトの完成像と、これから開始される「現実」であるテスト工程との間の溝を埋め、日々の努力の積み重ねが確実にプロジェクトの目的完遂に至ることを示す「道標」として作成したのが「テスト戦略書」である。

テスト戦略書はプロジェクトのステークホルダーをはじめとして、管理・推進、開発メンバー各々が理解する必要のある文章と位置づけている。そのために、全員が同じ方向を向きつつも、各テスト工程固有の状況に注力できるよう、記載内容について以下の基準を設けた。

- ・技術的背景を持たない人でも理解できること
- ・個々のテスト工程の前後関係を明確にし、漏れなく、重複なく、必要な作業を表すこと
- ・方針に関してはこの文書内で記載し、各計画書ではこの文書の方針を前提にできること
- ・方針を定めた理由の技術的な背景と、それによって到達される目標が定められること

プロジェクト内ではテスト工程を管理・推進する「テスト統制チーム」が組織され、アプリケーションの機能テスト工程の統制と、インフラやゲートウェイ開発などアプリケーション以外のテスト計画書のレビュー及びその状況を把握し報告する役割を担った。このテスト統制チームのメンバーがテスト戦略書を作成し、同時にテスト方式及び運用方式の定義とその内容を記した「テスト計画書」の作成と、テスト実施の推進役も担った。

### 4. テスト戦略書とテスト計画書の関係

プロジェクト発足時に作られた開発計画書に含まれ、実装(コーディング)と同時に実施される単体テストと異なり、独立した工程として管理される機能テスト工程については「テスト計画書」を作成した。テスト計画書は該当するテスト工程のテストの準備～実施～結果報告までの範囲を計画化したものである。図2の通りプロジェクト計画書で定義したテスト工程を具体化する計画書である各種管理計画書も考慮して作成した。その記載内容はテスト戦略書に示す指標に従った。なお、開発プロセスの一部であるテストガイドラインは、開発プロセスと一緒に準備された技術的なテストのガイドラインとして実際のテストを行う場面で活用されるものである。

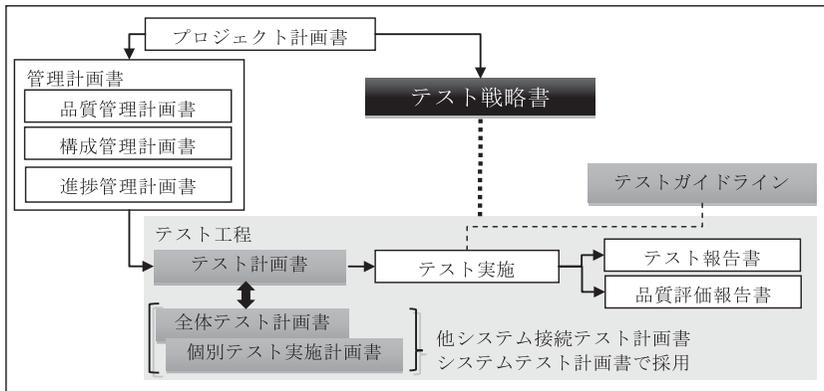


図2 テスト戦略書と各種計画書類の関係

テスト工程には、図2に記載されている通り、単独のテスト計画書を作る場合と、全体テスト計画書と個別テスト実施計画書に分割して作成する場合が存在した。本プロジェクトでは「他システム接続テスト」工程が、約60の外部システムとの接続テストを個別のスケジュールや関係者にて実施するため、共通部分を「他システム接続テスト計画書」（全体テスト計画書）として作成し、個別に異なるスケジュールや具体的なテスト内容部分をテスト対象の外部システム別に「接続テスト実施計画書」として作成した。同様に、「システムテスト」工程も独立性のある10個のテーマで並行してテストが進行するため、共通部分の「システムテスト計画書」と、テーマ別の「テスト実施計画書」を作成した。

## 5. アプリケーションテストの組み立て

プロジェクト計画書に定義されているテストの主要な部分を占めるアプリケーションに関するそれぞれのテストの工程で、具体的に何をするかを割り当てるテストの組み立て作業が、テスト戦略作成の第一歩である。本プロジェクトでは組み立てるべきテスト工程を動的テストだけに留めず、静的テストも含めた。

### 5.1 静的テスト

静的テストとしては、コードレビューの他に、コーディング規約に準拠していることをCheckStyle, FindBugs, JTest<sup>\*6</sup>の各ツールを利用し検証するようテスト戦略書で定めた。これは開発方法論で既に定義されているプロセスをテストの観点から整理し直したに過ぎないが、開発プロセス内でも品質を確保する作業をしていることをステークホルダーに示す狙いがある。特にコーディング規約には保守性（読みやすさ、記述の統一性）に限らず、性能面（処理時間が掛かる特定のSQL文の利用を禁止するなど）、安定性（例外処理の扱い）、運用性（ログ出力の統一）など開発者にとって常識的なことでも、ステークホルダーにも理解できるようにテストの観点別に整理することが重要である。

### 5.2 動的テスト

本プロジェクトでのオンライン機能の動的テストの組み立ては図3の通りである。図の左半分がテストの種類であり、右半分がそれによってテストされるテスト対象になる。両方ともテ

ストが進む（図の下の方向に進む）とテストの種類が実業務に近づき、テスト範囲もシステム全体へと拡大することを示す。テストの種類はUT（単体テスト：メソッド）、UT（単体テスト：サービス）、IT（結合テスト）、FT（機能テスト）、ST（システムテスト）と推移している。

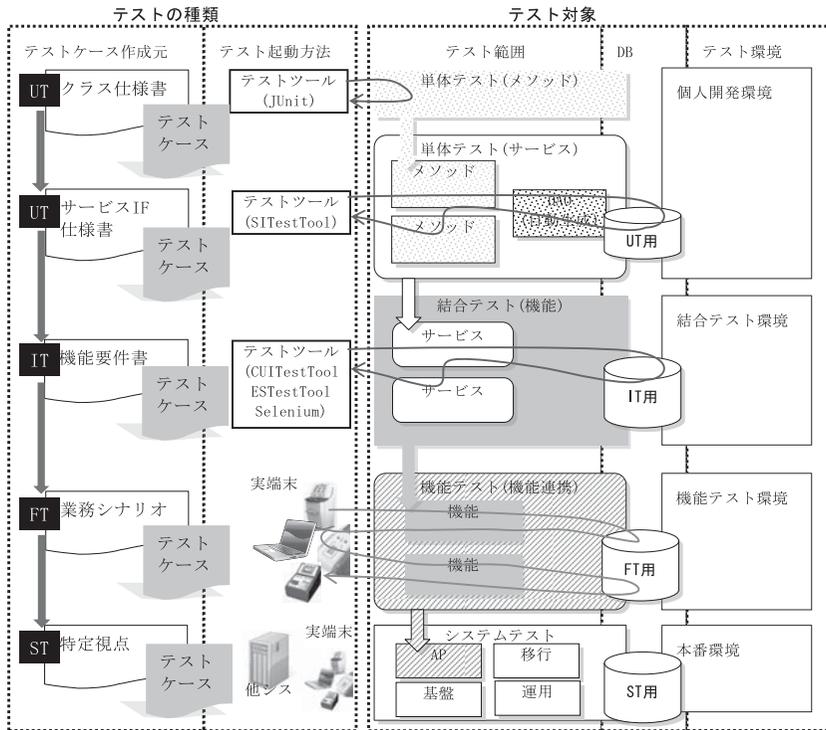


図3 テスト種類の進展とテスト対象の拡大

図3のテストの種類における「テストケース作成元」はテストケースを作成する基とした資料であり、「テスト起動方法」はテストの実施方法を示している。実施方法に記載されているテストツールのうち、SITestTool<sup>\*7</sup>、CUITestTool<sup>\*8</sup>、ESTestTool<sup>\*9</sup>は本プロジェクト専用のツールである。テスト戦略書にはテストケース作成元の資料としてテストが検証すべき仕様を明示し、その資料からの抽出条件についての指標を示した。但し、そこからテストケースを抽出する具体的な手段と、抽出した後の網羅性の検証方法は個々のテスト計画書に委ねた。テストの実施方法を戦略として明確化し、必要な作業をテスト計画作成時に具体化することで、ブレのない計画作成が可能となった。

テスト対象の「テスト範囲」はテストの対象とするアプリケーションのレイヤを指しており、「DB」が利用するテストDBの種類、そして「テスト環境」がそのテストを実施するHW構成/規模になる。テスト範囲の拡大を図式化すると図4となる。以下(1)から(4)で解説する。



図4 テスト範囲の拡大

- (1) 単体テスト (UT) を「クラス仕様書」を基にした「メソッド単位」及び「サービス IF 仕様書」を基にメソッドの集合体である「サービス単位」で実施。
- (2) 結合テスト (IT) を「機能要件書」を基にサービスの集合体である「機能単位」で実施。
- (3) 機能テスト (FT) を「業務シナリオ」を基に機能の集合体である「業務単位」で実施。
- (4) システムテスト (ST) を日次、月次などのサイクルで業務の流れを想定した視点、繁忙期の視点、突発的な事象が発生した場合の視点など、日々のシステムの使われ方に着眼した特定の視点を基に業務 (AP) とそれを支える基盤と運用、そして移行を含めた「システム全体」で実施。

テスト範囲の拡大に合わせ、図5に示すようにテスト環境及びテスト用 DB も拡大させ、システムテストでは本番稼働と同じ環境と DB にてテストを実施した。以下(1)から(4)で解説する。



図5 テスト環境と DB の拡大

- (1) UT用は開発者同士が独立した環境でテスト可能とするため、個々人の PC でデータも各テスト専用の手作りデータで実施した。
- (2) IT用は個々人が作成したアプリケーションの機能部品を共通の構成管理に登録し、一括ビルドするのと同時に、一つのアプリケーションとして専用の結合テスト環境でテストを実施した。但し、開発は業務機能群別のチーム編成としたことからチーム別に複数のテスト環境を割り当て、リリース運用をライブラリアン作業とすることで、環境の統制を行った。テストデータもテスト用に共通の標準 DB を提供し、テストで利用されるデータの内容の統制を計った。
- (3) FT用の移行データは移行プログラムとアプリケーションのかみ合わせの検証も兼ねているため、想定されるバリエーションの予約データを事前に仕込み、移行後のデータを用いたテストを実施した。
- (4) ST用では、稼働後一定期間経過しないとデータが蓄積されない履歴情報など、移行はされないが一定量のデータの存在が望ましいものは、ST用に事前に重石データを用意することで、性能的な評価にも利用できる DB を用いた。

機能テスト環境とシステムテスト環境はアプリケーションから見た差はなく、違いはハードウェア構成や規模の差と、他シスや特殊端末など、特定の環境にしか存在しない機器類の有無であった。

### 5.3 テストの種類

開発したシステムを構成する各要素のテストは、その目的に応じて多数の種類があり、テスト戦略書ではそれをテストタイプと呼んだ。本プロジェクトが対象としたテストタイプは表1

の通りである。テスト戦略書ではどのテスト工程でどのテストタイプを実施するかを定義し、特に同じテストタイプを複数のテスト工程で実施する場合は、その違いを明記することで漏れと重複の防止に配慮した。

表1 テストタイプ

テストタイプ	テストの目的
機能性テスト	ソフトウェアが備えるべき機能の確認
リグレッションテスト	ソフトウェアの改修行為が未改修部分へ影響を及ぼしていないことを確認
サイクルテスト	日時の経過を基にしたソフトウェアの振る舞いの正当性の確認
ストレステスト	システムの許容範囲を越える負荷を掛けた場合のシステムの挙動
ロングランテスト	連続稼働させた状態での振る舞いとシステムリソースの状況を確認
障害リカバリテスト	障害発生から利用可能になるまでの手順とシステムの挙動の確認
基盤系運用テスト	システム基盤上で稼働するアプリケーションに依存しない基盤としてのバックアップやリカバリ、ログの監視手順の確認
レスポンステスト	非機能要件のシステムの応答性の確認
スループットテスト	非機能要件のシステムの処理能力の確認
セキュリティテスト	システムがセキュリティ要件を満たしていることを確認
他システム接続テスト	開発システムと他システムが業務的に連携することの確認
運用慣熟テスト	本番運用を繰り返し、本番運用の操作を関係者が周知していることの確認
移行テスト	移行作業実施に支障のないことの確認
ユーザビリティテスト	操作性の確認

#### 5.4 テスト工程別テスト計画書とテストの種類の組み合わせ

本プロジェクトで作成したテスト計画書は表2の通りである。プロジェクト計画のテスト工程毎に前節の表1に示したテストタイプをテスト戦略書で対応づけた。なお、テスト工程以外に各種管理計画書と各開発計画書が存在したが、テスト戦略書はテスト工程のみを対象とした。

表2 本プロジェクトで作成したテスト計画書

計画書名	テストの観点	対象としたテストタイプ
G1 機能テスト計画書	機能性	機能性テスト
G2 機能テスト計画書		
G3 機能テスト計画書		
G3 性能確認計画書	性能性	レスポンステスト
インフラシステムテスト計画書	安定性	ストレステスト、ロングランテスト
	性能性	スループットテスト、レスポンステスト
	運用性	基盤系運用テスト
R1 機能テスト計画書	機能性	機能性テスト
R2 機能テスト計画書	性能性	レスポンステスト
R3 機能テスト計画書		
システムテスト全体計画書	機能性, 信頼性, 安定性, 性能性, 運用性	全てのテストタイプ
インフラレイヤ接続テスト計画書	接続性	他システム接続テスト
他システム接続テスト全体計画書		
移行システムテスト計画書	移行性	移行テスト

本プロジェクトでは早い段階での性能確認を重視し、実装時の静的検証に性能の観点を含めるだけでなく、機能テストでも性能評価を行った。GROUP3 開発で一通りの機能の実装が終わったことを受け、G3 機能テストとは別に G3 性能確認を行い、現行改修取込である R1 ~ R3 開発では機能テストと同時に個別機能の性能確認を行っている。

テスト戦略書を作成する上では各テスト工程で実施するテストの考え方や基本方針を定めると同時に、利用するテストタイプ別に、テストすべき観点についても記載し、各テストの計画時に具体的なテストケースを作成する上で指標となる条件が明確になるよう心掛けた。

### 5.5 テスト工程の依存関係の定義

プロジェクトにおけるテストタイプ（どのようなテストを行うか）とテスト工程（テスト戦略書ではシステムの品質レベルに合わせて、テストレベルと呼んだ）を定義した後は、その相互の関連（どの工程の後にどの工程が来るか、並行して行われる工程はどれか、その場合の影響範囲はなにか）を戦略書に定義した。そのテスト工程の依存関係を表し、どのようにテストが推移し、システムに期待される品質レベルに到達していくかを可視化するのがテスト戦略書の一番重要な役割であると同時に、各テストはどのような品質レベルのアプリケーションやその部品を入力とし、テスト完了時点でどのような状態になっていることが期待されているかを明確にすることで、そのテスト工程の具体的な作業に落とすことができる。また、同時に実行される他のテスト工程の存在を理解することで、共有すべき情報（テスト環境の調整、テスト対象に加えられる変更の有無など）も明らかになる。図6は品質レベルがLEVEL0からLEVEL4へと成熟度が業務品質に至る過程でシステムを構成する要素との連携の拡大を表し、そのLEVELに到達していることを検証するテスト工程を対応づけたものである。戦略書では品質レベルを以下の通りの定義とした。

- LEVEL0 コードレベル - サービス単位評価可能な状態
- LEVEL1 サービスレベル - 機能単位評価可能な状態
- LEVEL2 単一機能レベル - 全体評価可能な状態
- LEVEL3 全体評価レベル - 業務品質評価可能な状態
- LEVEL4 業務品質レベル - 業務適用可能な状態

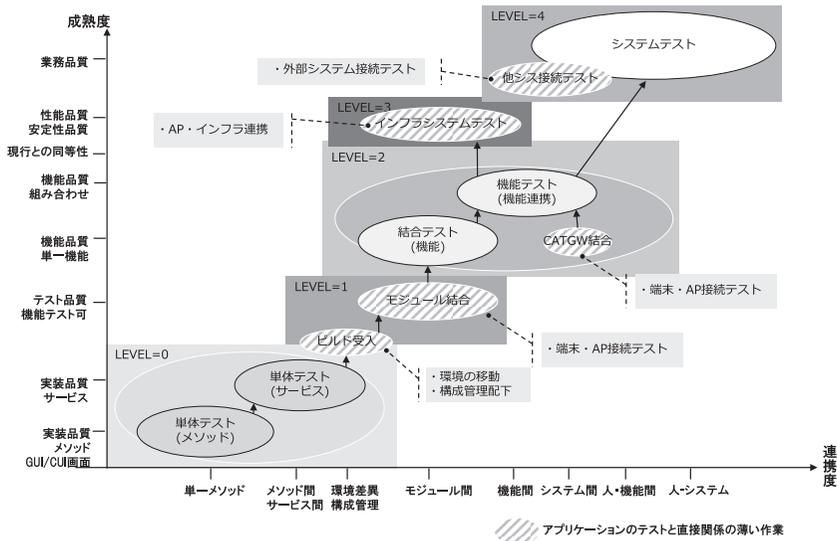


図6 品質レベルとテスト工程の推移

## 5.6 アプリケーション以外のテストの記述範囲

アプリケーション以外のテストについてのテスト戦略書の記載範囲は、基本的な考えと、アプリケーションが影響を受ける範囲とした。本プロジェクトで記載した具体的な範囲は次の通りである。

- ・インフラ基盤の運用機能（障害リカバリ、ログ管理、リリース運用、監視運用）のテスト
- ・端末、他システム接続ゲートウェイ開発の機能テストとインフラレイヤ接続テスト
- ・他システム接続テスト（他シス系のテストの組み立て、詳細は他システム接続テスト計画書に記載した）
- ・移行開発機能テスト、移行システムテスト

## 6. テスト戦略書を活用するテスト統制チーム

テスト戦略書の作成に関する活動は、ステークホルダーのレビューを受け、プロジェクトメンバーに周知することで終了するが、その真価は実際に使われて初めて発揮される。テスト計画を作成する時にテスト戦略書を参照することで、当該テスト工程を全体的な位置づけを理解した上で漏れなく、重複なく、作成することが可能になる。本章ではテスト計画の作成及びテストの推進を担ったテスト統制チームの活動について記述する。

### 6.1 個別テスト計画書の作成・引き継ぎとテスト実施の統制

アプリケーション関連のテスト計画書はテスト戦略書の方針に従いテスト統制チームが作成した。該当するテスト工程のテストチーム・開発チーム・プロジェクトマネージャをヒアリングした上で作成し、テスト対象の要素が必要とする具体的なテスト内容を反映した。テスト戦略書にはテストの運営に関する事項も記載されているので、テスト運営の立ち上げもテスト統制チームの重要な役割である。テスト運営はプロジェクトによるテスト計画書の承認を得た上で、テストチームに引き継がれた。テスト工程の開始後は、PMOからテストの進行に関する情報を週次で得て、プロジェクトに進捗を報告することを、テスト統制チームの役割とした。

### 6.2 他チームによるテスト実施の推進活動

設計・開発メンバーがテストも実施した基盤系、ネットワークゲートウェイ系、そして移行系は、テスト統制チームが提供するテンプレートを基にテスト実施チームがテスト計画書を自ら作成し、内容のレビューのみテスト統制チームが参加した。レビューでは当該計画書がテスト戦略書に準拠しているかを評価するため、以下の点について確認した。

- 1) テスト戦略書の記載に準拠しているか、準拠していない場合はそれが当該テスト計画内で閉じた事項であり、基本方針に抵触せず、準拠しない理由が明示されているか
- 2) 当該テスト工程の範囲として記載されている事項が計画書に記載されているか
- 3) テスト戦略書と重複した（無駄な）記述はないか

特に、1)の「非準拠」項目はテスト戦略書の基本方針に抵触するものでなく、他のテストに影響を与えるものでない限り、当該テスト計画のレビュー内でその項目を許容するかの判断はテスト実施チームに一任した。3)はテスト戦略書が改訂された場合、重複した記載の改訂が必要になるのを避けたこともあるが、より重要なのは当該計画書のレビューにて、既に確定している戦略レベルの内容を再び説明する手間を省き、本来レビューすべきテスト内容のみに注力

し、効率的なレビューを実現するためでもある。

テスト計画作成後、テスト統制チームは実際の各テストの推進をテストチームに任せ、状況を把握する役割に留まった。そのような第三者的な立場でいることで、応援を必要としている状態の発生を素早くキャッチし、対応する準備を整えることができた。

### 6.3 システムの特性とテストの方向性

テストの計画を作り、推進することがテスト統制チームの重要な役割であるが、そのためにはテスト戦略書の作成後、プロジェクトの承認とステークホルダーの理解を得ることが必要となる。特に、ステークホルダー向けには本プロジェクトで開発するシステムを七つの特性で捉え、それぞれの特性にどのような考え方をもち取り組むかを全体的な指標として示した。システムの特性はプロジェクト毎に異なるため、個別に検討する必要があるが、本プロジェクトでは表3の通り、それぞれのシステム化要件毎に考え方を定義した。

各特性から導き出された基本方針が、表3の右欄に表されるどのテスト工程やテスト運用に対応付けられるかをテスト戦略書に明記したことにより、ステークホルダーの観点で全体としてのテストの網羅性が把握でき、各テスト計画時には、テスト戦略書から指定された基本方針をどのようにテストとして組み立てるかの具体的な作業に集中することができた。

ステークホルダーの理解を得られたのは「テスト戦略」の議論の場を得ることができたのと、テスト戦略書に「ステークホルダーの視点」を取り入れることができたのが要因であったと考えている。

### 6.4 テスト統制チームのその他の役割

テスト統制チームはテスト戦略書、テスト計画書の作成だけでなく、テストの管理プロセス定義、ツール仕様の作成、テスト実施の立上支援、およびリハーサル実施の役割も担った。

## 7. テスト戦略書活用の効果と留意点

### 7.1 テスト戦略書の適用により得られた効果

テスト戦略書を作成したことにより、次の効果が発揮できた。

- 1) 何故このテストを行うのかと、どの範囲でどのようにテストを実行するのかは「テスト方針」としてテスト戦略書に記載し、具体的なテスト内容はテスト計画書に記載することにより、プロジェクトの全体像を以てステークホルダーと議論し、目指すものの共有ができただけでなく、具体的な計画書作成段階で目指すものも明確となり、テスト計画が効率良く推進できた。
- 2) 個々の議論が時間切れで、消化不良のままテスト工程に入ることなく、テストに必要な作業を計画に盛り込むことができ、計画書の品質が向上した。
- 3) 用語、テストの観点、記述レベルの統一が図れ、計画書のレビューでは具体的な作業内容の確認に集中することができ、時間だけでなく、内容も充実した。
- 4) テスト工程間の役割が明確となり、テスト対象の状況が明確で、次工程が期待している品質も把握してテスト計画を作成したため、重複する不要なテスト作業を除外することができた。

表3 システムの特性に対応するテスト方針とテスト工程

特性(1) 顧客の要求する「六つのシステム化要件」を満たしているかの「検証」がされること		
	100%確実な移行	
1)	移行時間を意識した手順を確立させる	移行システムテスト
	作成した手順通りに実施させる	移行リハーサルとチェックポイント
	但し、「移行」に関するテスト方針はテスト戦略書ではなく「移行全体計画書」に記載した	
	現行と同等の業務品質の維持	
2)	機能要件書通りに実装されていることを確認する	機能テスト
	他システムと接続した状態で確認する	他システム接続テスト
	業務の観点で検証する	システムテスト
	現行と同等のサービスレベル(処理性能)の実現	
3)	現行から移行したデータを対象に要求される処理量を実際に処理する	ラッシュテスト
	現行と同等のサービスレベル(可用性/信頼性)の実現	
4)	HW 冗長化設計, 障害時の縮退運用設計で確認する	インフラシステムテスト, 障害リカバリテスト
	アプリケーションとも連動させる	ロングランテスト, ストレストテスト
	さらに, 基盤系の静的テストとして, 上記以外に顧客の Weblogic と Oracle の DB 固有の構築ガイドラインを利用し設計内容の技術評価を実施した。	
	現行と同等の保守体制の実現	
5)	障害発生から障害対応, リリースと確認まで網羅する	システムテスト
	稼働直後の保守体制を強化する	臨戦リハーサル
	但し, 臨戦リハーサルは移行計画の一環として定義し, 全体移行計画内で方針を記載した	
	顧客ガイドラインに準拠したセキュリティ対策	
	セキュリティ対策を考慮した	開発ガイドライン
	セキュリティ検証を行う	セキュリティテスト
さらに, 開発ガイドラインが顧客のセキュリティガイドライン準拠であることは顧客による技術レビューにて確認した。また, 開発ガイドラインに準拠して AP が開発されていることは静的チェック(FindBugs/CheckStyle+レビュー)にて確認した。		
特性(2) 大規模でミッションクリティカルなシステムであること		
1)	大規模=開発機能が多い	開発管理計画書
2)	大規模=トランザクション数が多い	パフォーマンステスト
3)	大規模=利用者数が多い=同時アクセス数が多い	パフォーマンステスト
4)	ミッションクリティカル=安定稼働が責務	ロングランテスト
5)	障害時の迅速な対応が用意されている	臨戦リハーサル
6)	障害の影響度による重要性が定義されている	臨戦リハーサル
7)	過去の重障害の経験が活かされている	システムテスト
特性(3) 新 OPEN アーキテクチャが採用されたこと		
1)	業界における世界初の基幹業務用 OPEN システム	ベンチマークテスト
2)	短縮されたバッチや新バージョンのリリースサイクル	インフラシステムテスト
3)	複数のソフトウェアバンダー, ハードウェアバンダーの連携	バンダーコンソーシアム
特性(4) 社会基盤としての堅牢性, 安定性の保証が求められていること		
1)	外部への影響を最小限にし, 既存システムからシームレスな移行	全体移行計画書
	但し, 全体移行計画書は別資料として作成し, テスト戦略書では関連について述べた	
特性(5) 現行チャネル, 機能によるリスク分散開発を考慮していること		
1)	現行と同じインタフェースとサービスの提供	他システム接続テスト
2)	移行を想定した	移行リハーサル
3)	CUI 画面採用による現行操作性の継承	テスト手法: CUI 端末を用いた打鍵
4)	現物での確認テストを行う	テスト手法: 実機を用いた打鍵
特性(6) 開発期間中に現行の改修案件の取り込みが計画に盛り込まれていること		
1)	リグレッションテストの採用	テストプロセス: 自動リグ環境
2)	テストケースも保守対象	テストプロセス: テストスクリプト
特性(7) 個人情報保護の観点も重視すること		
1)	セキュリティガイドラインに準拠	インフラシステムテスト

- 5) テスト戦略書の方針に従わない内容が明確となり、その有効性判断に重点を置くことができ、前工程のテストで積み残した内容について取り込む計画とするなど、プロジェクトとしての弱点や遅れを補完することができた。

## 7.2 テストを効果的に進めるために

テスト計画の作成とテスト運営を行う上でテスト戦略書に記載して効果のあった事項について記載してきたが、その他にテストを効果的に進めるポイントについて、以下にまとめる。

- 1) テスト方針、内容、観点以外にテスト環境、テスト体制と役割、テスト運営、テストの開始・終了基準といった本来はテスト計画書に記載する内容もテスト戦略書内に記載する。

ここで定義した環境が物理的にどのサーバを用いるか、役割を担う体制はプロジェクトのどのメンバーを充てるかなど、テスト計画の作成を始めてから決まる事項も存在するが、基準とすべきテスト環境や体制、運営方法などをこの時点で定めることで、各テスト計画を共通の視点で具体化することができる。

- 2) テスト計画書で具体化する事項を明確にする

テストの密度を定義する品質メトリックスはテスト戦略書で定義したテストの観点を基に、各テスト計画書で定義することで、テストの進み具合と、障害の発生状況から見直しを行う運用が可能となる。前のテスト工程の結果報告を基に、次のテスト工程の品質メトリックスの具体値を定義することができ、弱点と判断される分野に注力できる。

- 3) テスト戦略書の方針と異なる内容のテスト計画書を許容する

テスト戦略書の方針に準拠することで、テスト計画の作成だけでなく、顧客や他のステークホルダーの理解を得やすいことは前章で述べたが、前の工程の状況やプロジェクトの進捗具合によっては、テスト戦略書作成時に定めた方針がテスト計画作成時にはそぐわなくなっている可能性がある。この場合は、テスト戦略書の方針と異なる部分は何か、何故それが必要になったのかを計画書に明示することで、計画のレビュー時にプロジェクトとして容認するかの判断ができるようになった。

- 4) ソフトウェアライフサイクルにおけるテスト戦略書の適用範囲を定義する

本プロジェクトでは実現しなかったが、ソフトウェアが利用され、保守される期間を通じテストは繰り返し行われる。テスト戦略書内に開発するソフトウェアのライフサイクルにおけるテストの考えを記載することで、テスト戦略書を開発時点だけの資料とせず、ソフトウェアが生きている間にも活用できると考えている。

- 5) リスクベースのテストの考え方の活用

業務へ与える影響度（他の手段で代替可能か、利用頻度や利用者への影響など）を定め、テストの充実性を測る尺度を持つことは重要であった。本プロジェクトでも仕様書の網羅性では不足するテストの観点として、性能評価やシステムテストのテストケース抽出、リグレッションテストなどテストの十分性を定める文章がない場合に有効であった。

## 7.3 本プロジェクト外でのテスト戦略書の適用への期待

ProjectAIは大規模で長期にわたるプロジェクトであったことから、テスト戦略書の作成は

要員面とスケジュール面で優遇されていたと言える。さらに、イテラティブ開発手法故に、同類のテスト工程が繰り返され、テスト戦略書の記載内容がテスト計画書の作成やそれに従い実施したテストに与えた影響とその効果を把握することができた。本稿に記載したような体系的なテストを推進していくことは全てのプロジェクトが行うべき振る舞いであり、そのためにはテスト戦略書は作成すべき資料である。しかし、多くのプロジェクトでは本プロジェクトのような恵まれた環境になく、少ない要員で期限内に有効な戦略を定義することが求められるため、プロジェクトとしてはテスト戦略書の制定に二の足を踏むことが想像される。プロジェクトとは独立した品質保証部門がテスト戦略書の作成を請け負う仕組みを設けるのも、テスト戦略書を小規模のプロジェクトに広めるのに効果があると期待している。

## 8. おわりに

大規模開発であったが故に「テスト戦略書」の作成に力を入れることができ、実際に作成し、その効果が確認できた。テスト戦略書の作成に道筋を示してくれた日本ユニシス(株)品質保証部各位にこの場を借りて感謝したい。また、本稿が読者のテスト活動の手助けになることを望むものである。

- 
- \* 1 ProjectAI とは、全日本空輸株式会社の新たな国内線旅客システム“ANACore”の開発プロジェクト名である。
  - \* 2 AirCore は米国 Unisys 社製のエアラインパッケージであり、同社の商標である。
  - \* 3 他シスは、外部システム連携を表わす ProjectAI 固有の略語で、主に AP (アプリケーション)、FW (フレームワーク)、command と組み合わせて用いる。
  - \* 4 ISEP は Information Services Engineering Process (業務ソリューションエンジニアリングプロセス) の略である。
  - \* 5 RUP は International Business Machines Corp. の商標である。
  - \* 6 JTest は Parasoft の商標である。
  - \* 7 SITestTool は Excel に記載された項目と入力データを基にサービスインタフェースを呼び出すツールである。
  - \* 8 CUITestTool はテキストファイルに記載された画面入力コマンドから CUI インタフェースを呼び出すツールである。
  - \* 9 ESTestTool は外部システムに送付する電文を入力ファイルとして他シスインタフェースを呼び出すツールである。

**参考文献** [1] 大塚 俊章, 荻野 富二夫, ソフトウェアテスト技術, ユニシス技報, 日本ユニシス(株), Vol.27 No.2, 通巻 93 号, 2007 年 8 月, P80-83

**執筆者紹介** 平松 敦郎 (Atsuro Hiramatsu)

1977 年日本ユニシス(株)入社。サポートセンターにて ASM, FORTRAN, COBOL, RPGII などで作成されたメインフレーム上のシステムコンバージョンに従事。C 言語で UNIX ワークステーション上の CAD/CAM システム開発に携わり、X Window, PC-UNIX の環境へ、SYSTEM v で CORBA を経験以降 PSA, SRM, SCM, ERP, リバースオークションなどの Web 技術の技術評価後、AirCore の受け入れ担当、ProjectAI のテスト統制チームに参加。

