

ソフトウェア品質特性に基づいたシステム・テスト設計

岡崎 毅 久

日本 IBM(株)

アジア・パシフィック・プロダクツ, APP 品質

システム・テストとは、システム自身が製品目標との間に不一致があるかどうかを発見するプロセスである。したがって、システムとしての機能、性能、操作性などについて、特にユーザー視点を意識しながら、その妥当性確認(Validation)の立場からテストを行い、評価を行うことが基本となる。しかし、システム・テストが最も困難なテスト過程である理由は、その主要目的に対して、テスト設計手法が知られていないことである。今回、システム・テストを計画するに当たって、その主要目的をできるかぎり達成させるために、ソフトウェアを評価するための品質モデルの応用を検討し、その外部的な品質特性に基づいてシステム・テストを設計し、実施した。本稿では、その品質特性を利用した設計方法を紹介するとともに、その効用および課題についての考察を述べるものである。

System Test Design based on Software Quality Characteristics

Takehisa Okazaki

APP Quality, Asia Pacific Products
IBM Japan, Ltd.

This paper describes system test design technique based on the quality characteristics for the evaluation of software quality. System testing is a process to find or rule out gaps between users' requirements and programs from such user's standpoints as functionality, performance and so forth to end up with validation assuring the software is solving the right problem. With the view of making up for the blindness to achieve the objectives, we have come up with a technique that involves breaking programming objectives down into some pieces by means of quality characteristics, followed by putting some of them together considering testability and test efficiency. This technique counted for a great deal in terms of putting miscellaneous aspects on user requirements in clean-cut order, suggesting what should be tested as well as how the program should be evaluated.

1. はじめに

システム・テストは、もっとも誤解されやすく、困難な過程であるといわれる¹⁾。システム・テストは、完全なシステムまたはプログラムの機能に対し、単に様々なシステム環境下でバグを追及してテストする過程ではない。例えば、ユーザーが、キーボードからタイプする操作を記憶して、その再生を自動的に行う機能を提供するプログラムの開発において、種々のキーボードを用意して、プログラムと機能仕様との相違点を発見するテストは、本来、機能テストの過程である。

アメリカ国家規格である ANSI/IEEE 標準²⁾の「ソフトウェア工学用語集」(Std 729-1983)によれば、システム・テストとは、「ハードウェアおよびソフトウェアシステムを試験して、それらが規定された要件に適合することを検証する過程」とある。また、妥当性検査確認(validation)とは、「ソフトウェアの要求事項に従っているかどうかを確認するためにソフトウェアを評価する過程」としている。

システム・テストが、プログラムと製品目標との間の矛盾を発見する過程であるとすれば、それは、プログラムの外部仕様を設計す

る際、変換エラーを発見する過程ということになる。しかし、製品目標またはプログラム目的仕様書は、プログラムの外部インターフェイスに関する詳細な記述を含まず、したがって、一連のテストを設計する手段として適用できる明確な手法がない。また特に昨今の変化に富んだ時勢においては、製品の当初の目的仕様あるいはユーザー要求が、開発の最終工程で、大きく変わっていることも少なくない。

しかし何れにせよ、システム・テストでは、テストの結果から、何らかの妥当性評価が必要となるわけであるから、実際にシステム・テストを設計する段階で、各々のテストにおける計測尺度や評価基準などを定義しておく必要がある。

そこで今回、既存のソフトウェア品質モデルを吟味し、その中で定められている品質特性の定義に沿って、システム・テスト設計を行うことにした。本稿は、その品質特性を利用した設計方法を紹介するとともに、その効用および課題についての考察を述べるものである。

2. システム・テスト設計

2.1 システム・テストとS/W品質特性

ソフトウェア品質特性に関しては、過去に多くの研究がなされてきた⁸⁾。Wulfの品質モデル、Boehmの品質モデル、Walters&McCallのモデル、Metriqs社のSQM、日本電気のSQMAT、そして弊社のSQUALASなどが代表的なものとして知られている。いずれのモデルもソフトウェアの品質特性を、ある視点に立って評価するために定義し、体系化したものである。Walters&McCall、SQM及びSQMATのモデルでは、全体として開発者または製品自体の内部的視点に立ったモデルとなっている。日本IBMのSQUALASは、ユーザーのソフトウェア・システムに対する品質要求を、ソフトウェア開発の各工程でどのように実現し、どのように点検・測定・評価していくべきかの仕組み、方法について標準的なものを提供しながら、ソフトウェア開発プロジェクト管理を容易にするものである。

今回、システム・テストの設計を行うに当たって、その主要目的を達成するために、国際標準化機構のISO/IEC JTC1/SC7 専門委員会

がソフトウェアの品質評価プロジェクトの一環として定義している6つの品質特性⁴⁾(ISO/IEC 9126)、およびそれを支援するために、INSTAC/STD 下の作業グループが、「品質管理ガイドブック⁵⁾ - 第12章」の中で紹介している、下位レベルの特性(副品質特性)を用いることにした。そこに至った理由としては、次のようなことが挙げられる。

1. 客観的かつ正確な評価に対応するために、ユーザー視点の基準を有している。
2. 製品目標が反映できるべく特性を網羅しており、各特性の定義が明確で、誤解を生じ難い。
3. 同レベルの品質特性が、相互に排他的であり、その組み合わせが容易である。また、テスト項目を詳細に分類するものではなく、整理するのに役立つ。

ここで、システム・テストの種類として挙げられる一般的なものとISO/IEC 9126の品質特性間で、その目的および定義から見て最もよく対応するものを整理すると、表2のようにほぼ一応に分類することができる。以上から、今回適用した品質特性は、各テストの評価基準としての必要条件を満たしていると言える。

品質特性	システム・テスト種類(範疇)
1. 機能性	機能部分テスト、手続きテスト、機密保護テスト、説明書テスト、適合性テスト
2. 信頼性	負荷テスト、大容量テスト、信頼性テスト、回復テスト
3. 使用性	有用度テスト、導入テスト
4. 効率性	性能テスト、記憶域テスト
5. 保守性	保全性テスト
6. 移植性	環境(構成)テスト

表 1. 品質特性とシステム・テストの種類

2.2 テスト設計手順と実施過程

テスト設計、テスト・ケース作成までにとった手順は次の通りである。

1. 製品目標と照らし合わせて、対象となる品質特性を選出する。次に、それらに対応する下位の特性(副特性)を選出する。
2. ユーザー操作の視点およびテスト実行可能性・効率性の観点から、同時にテスト並びに評価が可能なものを組合せて、テスト設計を行う。
3. 上記の特性に基づいて、各テストの目的を整理する。
4. テスト実行およびテスト・ケース作成のための方法(アプローチ)を記述する。

5. 上記に基づいてテスト項目を整理し、テスト・ケースを設計する。

以上の手順を経て、ユーザーの要求事項を反映し得るソフトウェア品質特性を用いて妥当性評価基準を設定し、テストの設計を行う。テスト計画書作成から妥当性評価までの流れは、図1に示す通りである。この設計段階での明確な定義を、環境設定、ケース設計、テスト実行作業途中で、随時参照することによって、各作業の方向性を確認していくこともねらっている。

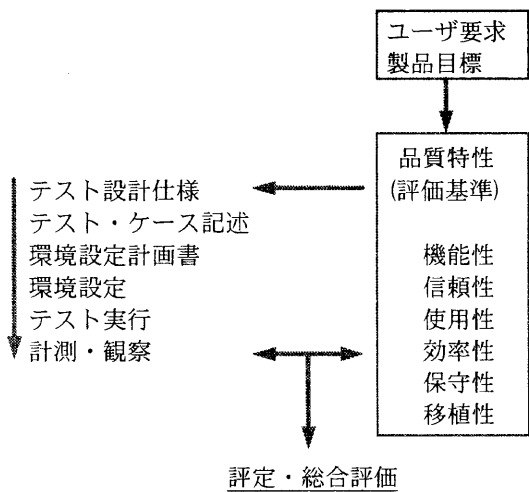


図1. システム・テストの実施過程

2.3 適用事例

本手法を適用したプロジェクトは、テスト実行・管理機能などを有する、三つの製品開発およびその統合環境を提供するAD分野の製品である。本手法によるテスト設計の結果を、実例の一つ挙げて説明する。

今回、想定ユーザ要求の1つとして、「複数のユーザが、複数の異なる環境 (H/W、S/W) 下でデータを作成しながら順次、一つのデータベースに登録して管理し、そのデータを集計したレポートが任意の時点で得られること」があった。この要求と照らし合わせ、品質特性および下位の副品質特性を検討していくと、以下のような品質特性が選出された。

① 上記の目的に対して必要十分な機能を備えていること、または備えられている機能の妥当性を確認する (機能性: 合目的性: 機能完全性)。

- ② ユーザ間のソフトウェアやデータへのアクセスを管理できること、または管理機能の妥当性を確認する (機能性: セキュリティ: アクセス管理)。
- ③ 製品間 (プログラムとユーザズ・ガイド) の機能に矛盾がないこと、またはその矛盾度合いの妥当性を確認する (機能性: 正確性: 無矛盾性)。
- ④ データ登録中のソフトウェアシステム故障がユーザに与える影響度は高いと考え、その故障が少ないこと、または故障頻度の受任限度の妥当性を確認する (信頼性: 信用可能性: 成熟性)。
- ⑤ 複数ユーザによる同時使用、大量データの扱いを考えると、ソフトウェアシステムがダウンする可能性は高く、そのような場合に、再稼働までの時間が短いこと、またはその時間の妥当性を確認する (信頼性: 可用性: 回復容易性)。
- ⑥ 上記同様、ソフトウェアシステムがダウンした場合、直接的な二次障害 (他データ破損など) の影響が小さいこと、またはその影響の妥当性を確認する (信頼性: 安全性: フェール・セーフ)。
- ⑦ 該当機能を備えたソフトウェアを実行するための準備 (動作環境、導入、導入結果確認) にかかる手間の妥当性を確認する (使用性: 運用性: インストール性)。
- ⑧ 今回の場合、データ集計に要する時間は、ユーザの扱うデータ量に大きく依存することが予想され、特にこの機能を、その処理時間の妥当性を検討する対象として注目することにした (効率性: 実行効率性: 処理速度)。
- ⑨ 複数の異なるハードウェア環境下での操作が必要となるので、各々のハードウェア環境に依存しないで動作できることを確認する (移植性: 移植環境適応性: H/W 環境独立性)。
- ⑩ 複数の異なるソフトウェア環境下での操作が必要となるので、各々のソフトウェア環境に依存しないで動作できることを確認する (移植性: 移植環境適応性: S/W 環境独立性)。

以上の品質特性を、テスト実行可能性・効率性および同時評価可能性などを考慮しながら組合せを行った結果、表1に示す三つのテストを設計した。

			Test ID		
品質 (副) 特性			1	2	3
機能性	合目的性	機能完全性	X		
	セキュリティ	アクセス管理	X		
	正確性	無矛盾性	X		X
信頼性	信用可能性	成熟性	X		
	安全性	フェールセーフ		X	
	可用性	回復容易性		X	
使用性	運用性	導入容易性			X
効率性	実行効率性	処理速度	X		
移植性	移植環境適応性	H/W環境独立性	X	X	X
		S/W環境独立性	X		X

表 2. 選出した品質特性と組合せ

- ・テスト 1 は、過去のプロジェクトにおける実データを用いて、その登録・管理、セキュリティおよび過負荷という観点から、そのシナリオ達成のために発生する作業を事前に整理し、サポートする代表的な環境下において、当該機能の妥当性 (十分性) 確認、シナリオ実行中の信用可能性 (MTTF、MTTB、障害収束率) の計測、ユーザーズ・ガイド中の記述の妥当性およびレポート集計処理時間の妥当性の検討を行うのが目的であった。
- ・テスト 2 は、前述のシナリオ中にシステム障害が発生した場合の安全性・可用性を、異なる環境下で確認するのが目的である。ここではシナリオ中の操作機能を抽出し、システム障害条件 (システム強制終了、電源オフなど) と合わせてテスト・ケースを設計した。またテスト実行効率性から、他のシナリオ (要求) の操作機能も同時にここで整理してケースを設計した。ここでの品質特性の組合せパターンおよびテスト・ケース設計は、従来の異常回復テストに相当し、テスト実行効率性、同時評価可能性からみて最も妥当な組合せであると判断し、他の二つの製品および統合環境に対する設計でも同様なアプローチをとった。
- ・テスト 3 は、当ソフトウェアの異なる環境下への導入容易性を確認し、また、導入説明書中の記述の妥当性を確認するのが目的である。今回のように各製品およびその統合環境の導入に対して共通のインストーラーを使用している場合、テスト実行可能性・効率性からみて、この品質特性の組合せパターンによって従来の導入テストのように一つに括るのが、最も妥当な組合せであると判断した。

以上のように、ユーザー要求を反映し得るソフトウェア品質特性を吟味しながら妥当性評価の基準を設定し、これに基づいて、評定、妥当性評価確認を行った。ここでの定義は、環境設定、ケース設計、テスト実行作業途中で、常に参照することによって、一貫した評価視点を保持しながら、各テスト目的の明確化に大きく貢献した。その明確化への貢献度を定量的に表すのは難しいが、今回評価基準として設定した特性のうち、約 67% に当たる特性は、常時参照しなくても、今回の関係者間で誤解や混乱の起り難いものであった。例えば、パフォーマンス・テストの中のターンアラウンドタイム (効率性の下位特性) などは、従来からよく採用されており、馴染みが深く、評価基準として一度設定した後は、特に参照するまでもないものであった。従って、残りの 33% が、明確な定義を用いて設定を行わないと、混乱の余地のあったもの、つまり、今回の定義を用いた結果の効用といえることができる。回復テストにおいて、何をもって回復というのか、何を測定または観察するのかなどは、従来関係者間で曖昧になることが多かったものの一つである。

テスト計画書作成以降の主な文書として、システム・テスト設計・手順仕様書を作成し、そこで、上述のソフトウェア特性およびそれらの組合せ、それに対するテストの取り組み方、テスト実行のための一連の行為等を規定した。章構成は以下のようなものとなった。

- はじめに
 1. 記述対象
 2. 文書管理
 2. 1. 文書作成者
 2. 2. 文書管理責任者
 3. 関連文書
 4. 各テストにおける記述構成要素
 5. 採用する品質 (副) 特性の定義
 6. 前版からの主な変更箇所
 7. 各テストと品質特性の対応マトリクス
- 製品 A に対するテスト設計
 2. 1. テスト A-1
 2. 1. 1. 概要
 2. 1. 2. 目的
 2. 1. 3. 該当する品質特性
 2. 1. 4. アプローチ、手順
 2. 1. 5. テスト・ケース設計

2. 1. 6. 準備、設定事項
2. 1. 7. 前提条件 (Entry Criteria)
2. 1. 8. 終了条件 (Exit Criteria)
2. 2. テスト A-2
3. 製品 B に対するテスト設計
4. 製品 C に対するテスト設計
5. 統合機能に対するテスト設計
6. システム・テストの実施管理
 6. 1. テスト進捗管理および報告手順
 6. 1. 1. テスト・ケースの実行進捗管理
 6. 1. 2. テスト実行日単位の記録事項
 6. 1. 3. テスト・ケース実行操作記録
 6. 1. 4. テスト・データ管理
 6. 1. 5. テスト終了報告
 6. 2. テスト・ケース ID 命名規則

今回の妥当性検査確認過程で、メトリクスを用いて定量的な計測を行ったものは、信頼性下位の信用可能性における、平均ダウン発生間隔 (MTTB: Mean Time To Breakdown)、その下位の成熟性における平均故障発生間隔 (MTTF: Mean Time To Failure) および更にその下位の無欠陥性における障害収束率と、効率性下位の実行効率性・処理速度におけるレスポンスタイムであった。この他については、有効なデータをとることができなかったため、定性的な妥当性確認報告のみに留まった。したがって、評価をできる限り客観性の高いものにするために、妥当性検査確認報告を行う際に、可能な限り多くの者が参加した。また、評定・評価は、テスト担当者全員によって行われた。

上記計測結果および妥当性確認報告に基づいて、以下の 5 段階の評定レベルを設定し、重要度などを考慮しながら、評定を行った。

1. ほとんど実用には耐えず、改善策も見当たらない。
2. 問題点や制限が、優れた点よりもかなり多く存在し、未成熟な状態。
3. 優れた点も多いが、同時に問題点や制限も多く見受けられ、改善の余地も未だ多い、または不安定。
4. 優れた点が際立っており、問題点や制限が解消されれば、かなり完成度の高いものとなる。
5. 問題点や制限が存在せず、完成しており、安定している。

以上のうち、評定結果が 4 に満たないものに

対しては、問題点・制限の内容を整理し対策をたて、フォローアップとして解決を図るようにした。

3. システム・テストの守備範囲

システム・テストが、もっとも誤解されやすい過程であることは、前にも述べた。その誤解を生じやすい要素として、機能テストとの区別、および検証と妥当性検査確認の区別という二つの点が挙げられる。最近な例を用いて、その二つの点について考察してみる。今、1000 個のオブジェクトを登録・処理できる機能仕様があるとする。標準的な操作、データおよび条件下でオブジェクトを 1000 個登録・処理することは、機能テストにおける境界条件下のテストに過ぎなく、システム・テストの大容量テストではない。品質特性上は、機能性下位の正確性の定義に相当する。一方、信頼性の概念は、“特殊なデータ、異常操作、過負荷、内部異常等に対する処置が適切である、および連続動作が可能”である。この概念から、表 1 の信頼性に対応したテストの視点がより明確になる。勿論、テスト実行効率や開発状況を考慮して、両者を 1 つのテストに代表させることは实际的であり、その場合、テスト戦略としてその旨が明確に定義されていれば問題ない。

次に、上の例で、実際にオブジェクトを機能仕様通り 1000 個登録できるかどうかを見るのが検証過程であり、仮に 900 個しか登録できなかった場合、果たしてそれが、ユーザーの立場から見ても問題なのか、どのくらい妥当なのかという視点で見ると妥当性検査確認である。また、1000 個登録が可能であった場合でも、それがその時点でのユーザーの要求と照らし合わせて妥当かどうかを確認する余地はある。システム・テストは、システム検証過程とシステム妥当性検査確認過程に大別されるが、システム・テストの主要目的からすると、後者により重点を置くべきである。

にもかかわらず、後者がしばしば軽視される理由について、前述の、「プログラムと製品目標との間の矛盾を発見するためのテスト設計手法が確立されていない」こと以外に、以下のことが挙げられる。

- ・システム・テスト自体が、一般に開発の最

終工程に位置づけられるために、開発上の都合などにより、実施するための十分な時間がとれないことが多い。

- ・同工程においても、その性質上、検証過程の後に位置し、検証結果やコード品質に大きく左右される。
- ・テストと品質評価の関連、ソフトウェアの製品目標、ユーザー要求と品質特性の関係など、評価技術上の課題が数多く存在する。しかし理由がどうであれ、妥当性検査確認という過程を抜きにして、多様化・急速化している実際の市場要求を満たす製品の開発は、不可能に等しい。初めの2つの要因に対しては、システム・テストの主要目的からすると、その工程が、必ずしも開発の下流のみに位置づけられる必要はない。ユーザーの立場に立って V&V (Verification & Validation) の考え方を開発計画の段階で適用し、各工程の任意の時点で実施することで回避できるものである。実際、弊社では、UCD (User Centered Design: IBM Corporate で開発された、ユーザーを第一に考えたソフトウェアづくりのアプローチ) 手法を開発プロセスの中に採用したものに対しては、開発期間を通して継続的に、ユーザーのフィードバックを得、ユーザー視点による妥当性の評価・確認を行っている。

何れにせよ、今後は、先に3つ目の要因として挙げた評価技術上の課題が、如何に実用的に達成されるかが、効率的且つ効果的なシステム・テスト設計、妥当性検査確認、ソフトウェア品質の評価を行うためのカギとなる。開発の任意の時点で妥当性確認作業を支援するような、内・外別、担当者別の品質特性の定義とその関係、および標準的な品質メトリクスを選定 (ISO/IEC JTC1/SC7/WG6 にて作業中) が、大いに望まれる。

4. 標準規格の企業内製品開発への応用

今回のシステム・テスト設計では、国際標準規格による品質モデルを採用したが、この他にも、テスト(手続きとコードの実行)に関連する基本的文書類については、アメリカ国家規格である ANSI/IEEE 標準の「ソフトウェア試験文書」(Std 829-1983)を参照している。本来、規格は、その汎用性が重要であることから、最低限の要求が定義されており、その

応用が容易でなければならない。上記の規格は、その適用範囲に関する記述のなかで、「各基本文書の目的、概要、内容などを規定したものであり、規格の文書は動的試験に焦点を合わせるものではあるが、その他、設計やコード審査に用いられる試験計画や試験事故報告などに応用してもよい」としている。また、「特定の試験方法論、取り組み、技法、施設、ツールを要求するものでない」ともしている。特に、企業内における製品開発プロジェクトへ標準規格を応用する場合、実際の作業の目的や体制、企業内標準等を考慮しながら、規格、規定のガイドとしての参照、及び必要に応じて適宜その内容の取捨選択が可能である。そのような規格を参照する最大の利点は、用語の明確な定義、各作業の定義、観点並びにその整理に効用を発揮し、作業品質の向上に大きく貢献し、結果的に、製品品質向上への貢献につながり、他のプロジェクトへの応用も可能なことにある。

5. まとめ

今回、システム・テストを設計するに当たって、その主要目的をできる限り達成させるために、ISO/IEC 9126 品質モデルをはじめとする品質特性を用いて設計し、テストを実施し、妥当性評価確認を行った。本手法で用いた品質特性は、システム・テストに必要な評価基準の定義を網羅しており、その定義が明確で、より排他的であるため、テストの設計、実施および製品の評価まで、一貫したユーザー視点を保持し、作業目的の明確化という点で、その効用を発揮した。

参考文献

- 1) G.J.Myers 著: "The Art of S/W Testing", 近代科学社 (Feb. 1989)
- 2) 東基衛監修: "ANSI/IEEE ソフトウェア規格集", 日本規格協会 (Feb. 1988)
- 3) B.Kitchenham, et.al.: "The Elusive Target", IEEE Software vol.13-no.1, pp.12-21 (Jan.1996)
- 4) International Standard ISO/IEC 9126: S/W product evaluation - Quality characteristics and guidelines for their use" (Dec.1991)
- 5) 森口繁一編集: ソフトウェア品質管理ガイドブック, 日本規格協会 (July, 1990)