

## ソフトウェアテスト—標準モデルの必要性

—特別寄稿

### イントロダクション

ソフトウェア技術はつねにテストを必要としてきた。これは、(a) 用法シナリオが繁多で、(b) ユーザーが予期せぬアクションをすることがあり予測困難で、しかも(c) ソフトウェア開発者の視野は狭くなりがちで、上記の点の複雑さをコードから除外することが多い、といったソフトウェアテストに特有の込み入った事情によるものである。

ソフトウェアテストは、気になったら験するという簡単な点検から始まり、周到に考慮された方法論とテスターの活動を支援する様々なテクノロジーへと発展してきた。方法はますます特殊化され、ソフトウェアの高品質を保証するという目的は一つでありながら、開発チーム、組織、あるいは外注先で様々なものが使われている。テストはスライドショーから出発し、現在ではソフトウェア技術の主要な構成要素となり、さらには専門の産業にまでなろうとしている。

ソフトウェアテストのこの展開は、とりわけ専門語彙と多くのコンセプトを創造し、それらは広く受容されている。ユニットテスト、システムテスト、受入れテスト、性能テストといった用語は、ソフトウェア開発に従事する人々はほぼ内容を理解することができる。しかし、これはまだ第一歩に過ぎない。さらに標準化を進める必要があり、それはテスト従事者だけでなく、ソフトウェア開発関係者、さらにはソフトウェアユーザーにとっても大きな便益をもたらすと思われる。

多くの標準モデルと同様、テストモデル標準の最も直接的で明らかな利点は、成熟度を共有できるということであろう。このことは、テスト活動に関わる様々な当事者の中で、テストツールの中で情報をやり取りすることにより、全体の情報共有ができるということである。ユニットテストをサポートするツールは、統合テストをサポートするツールに情報を渡すことができ、その結果はさらにテストのカバレッジと結果を分析する第三のツールで使用できるようになる。

マイク・オーラ、ジェイディーブ・ミルチャンダーニ  
リラティビティ・テクノロジーズ

### テストコミュニティ

標準化は様々な当事者が対話しつつ協力していくためのものであるが、ここでテストに関わる当事者とテストプロセスにおける役割を明確にしておく必要があるだろう。またテストに関するニーズが少しずつ違っていることにも注意しなければならない。

システムアナリストとアーキテクト：要求を集約し、アプリケーションあるいはソフトウェア製品の機能と構造を設計する。彼らは関連するユースケースおよび実行される活動についての認識を持っている。主な関心事は、アプリケーションの機能的健全性を保証できるよう、これらのケースや活動がすべて適切にテストされるかどうかであり、同時にアーキテクチャ上の大きな決定を検証するための性能テストにも関心を持つ。

- ソフトウェア開発者：自ら開発した様々な部品のユニットテスト、およびその後の統合テストに関与する。主な関心事はバグの有無とアウトプットが予想通りのものかどうかである。
- QA チーム：ソフトウェアの品質に関わるすべての側面に目を配り、様々なテストの計画、設計、実施に当たる。またテスト結果を分析し、ソフトウェアが予め設定された品質基準に合格していることを保証する。
- エンドユーザー：ソフトウェアの最終受益者として、様々な機能性テスト、性能テストを行うが、それらは受入れテストと総称される。テスト結果はソフトウェアの受入れ、あるいは複数のベンダー間でのソフトウェア製品の選択の判断を助ける。
- セキュリティ専門家：権限のない人間が保護された資源にアクセスしないことを保証するため特別なテストを行う。
- システム監査者：アプリケーションがユーザーの基準を満たしていることを保証するために機能性テストを行い、正確な報告書を作成する。

今日のアウトソーシングの拡大傾向を考えれば、上で定義したコミュニティのいくつかは社内に取りまききれず、様々なベンダーやサービスに広がってしまうことも考えられる。

これらのコミュニティは、テストングの目標が違うので、異なったテストングツールを使うことになろう。ソフトウェアベンダーは、様々なテストングツールを開発・提供してきたが、内部で開発したものを使用することもある。

## テストングツール

ここでは提供するサービスによって分類された、様々なカテゴリのツールを検討する。複数のサービスを提供するツールはあり得るが、全部のニーズに対応するツールは稀であろう。

- ユニットテスト・ジェネレータは、ユニットテストのケースを作成するが、これはアプリケーションコードの詳細に関する知識に基づいた特殊なものである。あるものは、例えばプログラムの特定の演算処理に関するケースを生成する。
- 実行時環境において人が行う特定のアクションに基づいたテストケースをユーザーが取り込めるようにしたツールもある。ユーザーが特定の操作を実行すると、そのアクションは自動的に記録され、変換されてテストケースに収められる。
- 最近の多くのUMLツールは、ユーザーが要件やユースケース、アクション・ダイアグラムに基づいてテストケースを自動生成できるようにしている。
- テストケースの自動実行に特化したツールもある。バッチモードで膨大な数のケースを実行し、結果を記録する。
- テストケースおよびテストングプロセス全般の管理を行うツールもあり、さらに結果の分析や高度な要約レポートの作成を行うものもある。

## ベンチマーク

ベンチマークはテストケースのセットを標準化したもので、ソフトウェアプロダクトの性能を評価するのに使われる。テストケースは様々なソフトウェアプロダクトに対して同一なので、その結果を使って個々の性能を比較評価することができる。

## 協力の必要

同じプロセス（の異なる段階）で様々なコミュニティとツールが関係しているので、情報の交換が重要となる。例えば、QA チームはユニットテストの際に開発者が使ったテストを含めたいと望むかもしれない。もしユーザー受入れテストが

失敗したら、開発者は失敗したテストケースを見たいと思うだろう。これらはシナリオのほんの一部であり、さらに多くを考えることは容易である。

テストングにおける協力には2つのタイプがあると思われる。

- 垂直的協力は、ソフトウェアの構築、リリース、利用において同じプロセスの一部に関わる様々な当事者間で生まれる。古典的なウォーターフォール型アプローチでは、テストケースは開発フェーズ全体を通して、チーム間で受け渡すことになっていた。特に外部のベンダーが関わり、独自にツールを使用する場合には、テストケースを共有する能力は保証されていない。
- 水平的協力は、異なる組織が関わり、別の組織と同じようにテストを実行したい場合に生まれる。例えば、ある企業が別の企業にソフトウェアを販売する際に、テストケースも資産の一部として引き渡すような場合である。購入企業がすでに別のテストングツールを標準的に使っている場合に頻繁に生じることだが、テストケースの変換は非常に難しい。

テストモデルの標準は、上記のいずれの場合でも、共有と協力を促進するだろう。

## テストモデル標準の要素

標準は、OMGのような団体において、様々な当事者から出される複数の要求や提案に基づいて生まれてくるべきものである。われわれは、ここで必要となっている標準においてあるべき機能の一部をリストアップしてみたい。原則的には、標準モデルは独自の形式で記述されてよい。

- ユニット・テストケース：インプットと期待されるアウトプットとして表現される
- システムおよび統合テスト：一連のユーザーアクションと期待されるシステムのレスポンスとして表現される
- 予測され、また達成されたコード・カバレッジを示すコード・カバレッジ・モデル
  - 予測的カバレッジモデルは、実際に発生する前に有用なテストランのシミュレーションを提供する
- 「ベース」あるいは「スモークテスト」パッケージの定義：テストングの経済的条件が与えられると、基本的なレベルの機能を確認するために、基本的なテストスイートを頻繁に実行する必要性が生ずる。これを定義、管理、保守する方法についての標準があれば、テストケースの

作成者がテストスイートを効果的に適応させるのが容易になる。

- テスト可能性のデザイン：システム設計（ユースケースなどを含む）を、どこまでテストすることができるかで評価する能力は有効な尺度となる。究極の目標はエンドユーザーの満足でなければならないが、内部的にはアーキテクチャの健全性は、テストの徹底性と容易性から評価される。これを評価する標準的な尺度は、同時にテスト・プロフェッショナルにとっての効果的な尺度ともなり得る。
- データだけでなく、パフォーマンスその他の側面を含む個々のテストケースの結果
- テストに含まれる実際のコードへのポインタ、参照のオプション
- テストケースが導出された他のモデル（例えば、アクション図におけるアクション）へのポインタのオプション
- テスト結果の結論（カテゴリ別の失敗率と成功率）

#### 他の標準モデルとの関係

ソフトウェアテストは、任意の時点ソフトウェア品質の評価と記述であるから、テストモデルはソフトウェアを表現する他のモデルとリンクするのが自然である。こうしたリンクはすでに上述されているが、下記の要素への参照を含むことになる。

- プログラムのパーサツリーのノード(ASTM)
- シンタックスのコンストラクトその他ソフトウェア資産(KDM)
- ユースケース、アクション(UML)
- ビジネスルール(SBVR)

#### シナリオ

- 同じ開発プロセスに従事するチームは、テストケースを受け渡す。
- 一つのツールがテストケースの生成に使用され、別のものがそれを実行する。さらに別のものが結果を分析するが、個々のケースには最高のツールが使用される。
- ベンチマークは、多様な環境に容易に変換することができる。異なる技術の間で共有することができる。
- 企業は、(社内のテストの際には) テスティングツールとテストサービスベンダーの両方を変えることができ、特定のフォーマットに縛られることがない。

テストモデル標準形成の最適場としてのOMG

テストモデルのフレームワーク構築の知識と経験は、OMGが有している。OMGはオープンな非営利のコンソーシアムで、エンタープライズアプリケーションの相互運用性のための標準仕様を策定管理している。OMGのメンバーにはコンピュータ業界で最も成功し革新的な企業のほか、ITによって強固な競争力を築いている先進的なユーザー企業が数多く含まれている。すべてエンタープライズ、インターネット、リアルタイム/組込みシステムの将来に積極的にコミットしている。

OMGのモデリング標準は、UMLとMDAが実現した強力な機能が、ITシステムモデリング、ビジネスプロセス管理を含む、ソフトウェアその他のビジュアルデザイン、実行、ソフトウェア保守を含むプロセスをサポートしている。OMGのミドルウェア標準とプロファイルは、共通オブジェクトリクエストブローカ・アーキテクチャ(CORBA)に基づいており、広汎な産業をサポートしている。

#### 執筆

マイケル・オーラ/リラティビティ・テクノロジーズ社  
ジェイディーブ・ミルチャンダーニ/リラティビティ・テクノロジー

#### 解題

本稿は、MDAを適用したレガシーアプリケーション現代化ツールを開発・提供するリラティビティ・テクノロジーズ社CTO、マイク・オーラを中心に、OMGにおけるテスト標準化のためのホワイトペーパーとして書かれた。

参考までに、OMGにおけるテスト標準作業として、以下のようなものがある。

UML 2. Testing Profile (U2TP)：2006年採択

Model-level Testing and Debugging。：作業中

XMI/UML Testing Task Force：討議開始

Software Assurance ABSIG：討議開始

本論文が必要性について述べている項目の一部はU2TPでカバーされており、MLTDで実現するものもある。マイクラが拡張しようとしているのは、KDM (Knowledge Discovery Metamodel)に関連したもので、レガシー中の知識を抽出・再利用するためのものである。これは9月以降のTCミーティングで議論されることになる。◇