品質と効率を追求した組み込みソフトウェア・テストの実現

中編　テストの無駄とむらを排除する

伊東 大助

前編（2009年9月号，pp.157-162）は，テスト計画時における最適化について述べた．今回は，テスト実行における工業化について解説する．

本稿でいう「工業化」とは，テスト実行時にいかに省力化するかということである．テスト・ツールによる自動化とメトリクス分析によるテスト実行/改善優先度付けについて考察し，テストの無駄とむらについて考えてみたい．

1．ハードウェアやツールがないとテストできない？

● ソフトウェアのみでテストする可能性

ハードウェアの開発が遅れるから，つまりハードウェアがないからテストができないという声をよく耳にする．しかし，ハードウェアがないと本当にテストはできないものなのだろうか？

確かに実機があればスムーズにテストができる．しかし，実機があったとしてもテストできないものも存在する．たとえば，条件分岐により深くネストしたロジックの検証において，それらすべての事象を実機で再現してテストすることこそ，まさに不可能ではないだろうか．

目の前に積まれたいくつもの（いや，やり切れないと言ってもよい）テスト項目を，ハードウェア（実機基板）を使って環境構築できたとする．しかし，ソフトウェアからすると，それらはしょせんレジスタやメモリの値がいくつになっているかであり，値を読み出した後は，特定の変数やパラメータとしてソフトウェアに渡され，処理されるものではないだろうか．

それならば，ソフトウェア的に変数に値を与えるとか書き換えることで，十分にシミュレーションできるし，ハードウェアを待つ必要もない．テスト内容をよくよく読んでみると，ある関数やクラス・メソッド，タスクの単体の機能テストであれば，さらにそのハードルは下がる．いや，むしろソフトウェアでテストした方が，時間効率も網羅性も格段にアップするのではないかと考えてみた．

● テスト・ツールは市販品でなくてもよいのでは？

ハードウェアの完成を待たずに，テストができそうだと気付いたものの，ではどうやってテストをするのか？　という次の疑問に当たってしまった．開発ツールに付属するデバッガ上で，変数の書き換えを行ってテストするという手を使ってみたが，これはもう一度同じ実行をするために，また一から人の手を介して行わなければならない．さらにその設定やシーケンスを，ドキュメントとして残すのも正直面倒くさい．

そこで筆者は，自作のテスト・プログラムを作成した．最初は，関数やクラス・メソッドの単体レベルのテストだけをプログラミングしていた．しかし，そのうちにソフトウェアが正しく動作するようになればなるほど，それらの組み合わせを変えることで何倍にもテストできる項目が増えてくることに快感を覚えてしまったのである．

余談ではあるが，その当時，自作のテスト・プログラムを後輩達に見せては，よく自慢していた記憶がある．これは，今になって思えば，どのようなテストをしているのかをトレースするのに役立っていた気がする．分厚いテスト仕様書を読めというよりも，テスト・プログラムとはいえ，実際のコードを読み，触れることで，モチベーションが上がったり，技術的なコミュニケーションの手段になったりと，テストの効率化以上の副産物があったような気がする．

● テスト・プログラムを作る時間がない

時がたち，筆者がプロジェクト・リーダとしてプロジェクトに参加するようになると，プログラミングの担当者た

KEYWORD ── ソフトウェア・テスト，テストの自動化，GUIのテスト，メトリクス分析