



テスト工程は技術者の“誇り”をかけたプロセス。

# 「自動化という工夫を入れて充実を。」

システム開発環境が厳しくなっている。時間的、コスト的に大きなプレッシャーを受け、ものを出すのがとにかく先決、という状況だ。その余波を受けて削られているのがテスト工程である。しかし、十分なテストを経ないシステムの品質は、経営問題にも直結する火種となるだろう。テストを軽視するのは、自らを窮地に陥れる判断だ。

そうした中、米国 Spirent Communications 社は、ネットワーク環境を含みテスト工程が直面する課題を解決すべく「Spirent iTest」を提供している。

これは従来のテスト資産をそのまま活用しながら簡単にテストシナリオを作成、自動化できるというもの。時間が取れない状況でも、テストの充実を図れるという。



## 苛烈化の一途をたどるシステム開発競争

今日、IT なしにビジネスを進めることは難しい。社内で協業するにせよ、顧客に提案を行うにせよ、そこには何かシステムが介在するようになってきている。

それだけ IT が企業経営と不可分になったということだが、その裏ではシステム開発をめぐる厳しい案件獲得競争が展開されている。

ふた昔ぐらい前であれば、日本中の企業がいっせいに IT を必要としたため、そのニーズを満たすべく、システムインテグレータや小さなシステムハウスが山のように必要だった。それでも技術者は慢性的に不足している状況だったが、しだいに需要は満たされ

ていく。今や大半のシステムのニーズは一巡した。しかし、システム開発企業の数はい前のままで。

そうした中でリーマンショックが起こった。世界的な経済不況により、価格単価の下落も発生。人件費の安い国へ開発を発注するオフショア開発も増えた。

しかも、昨今はスピード経営時代となり、IT プロジェクトも従来のように 1 年、2 年とじっくり時間をかけられなくなった。長くて数ヶ月、場合によっては 1 ヶ月でサービスリリースなどという短納期開発案件もある。

このような状況になると、できるだけ少ない技術者でコストをかけずに、品質にも片目をつむって、とにかくモノを出すのが先決、ということになってしまう。

## 削られるテスト工程

テスト工程は、こういう状況下、真先に削られる工程である。ここは開発したシステムの品質を作りこみ、顧客に保証する、いわば“誇り”をかけた工程だが、時間がないためにかなり短縮される傾向にある。また最近では、金融機関の勘定系システムなど社会インフラ的なシステムはともかく、少々なら障害が発生しても業務に重大な支障をきたさなければよい、と品質問題を大目に見る風潮も出てきた。

しかし、やはりテスト工程を軽視することはよいことではない。

第一に、十分に検証されていない状況で本稼働を迎えると、当然障害が発生する確率が高くなる。もし運よく重大な障害

発生は免れたとしても、遅延、停滞は起こり、それによる損失も生じる。何より、失われた信頼はダメージとして長く残るだろう。

第二に、当のシステム開発企業にとって、手戻り仕事が発生する。それを予想して待機しているのならまだいい。しかし、たぶんすでに次のプロジェクトに入っている最中の割り込み仕事になるだろう。それも障害を受けての修正なので大急ぎだ。いきなりてんやわんやの状況になってしまう。また、システム開発の世界で広く知られる「1:10:100の法則」の洗礼をまともに受けることになる。これは設計の段階で修正すれば1、開発の段階で修正すれば10ですむものを、本稼動に入ってから修正すると100のコストが発生するという法則である。

つまり、手戻り仕事を発生させないという観点からも、コストという観点からも、早い段階で品質を作りこんでおくのが何より重要なのである。株式会社 東陽テクニカ 情報通信システム営業部 課長 徳道宏昭氏は次のように語る。

「テスト工程は保険のようなものだという方がいますが、保険は入らないという選択肢がありますから、この例えは違います。

このプロセスは開発とは切っても切り離せない不可欠な一部分です。本稼動で障害が発生したときの損害は、企業経営に大きなマイナスインパクトを与えます。

テスト工程の重視は、このマイナスインパクトを防止するための最低限の投資だと考えます」

## ● テストプロセスに自動化を導入する



情報通信システム営業部 課長 徳道 宏昭氏

しかし、テスト工程に余裕を持って時間が取れないのは、もはや容認せざるを得ない事実である。ここで品質を担保しようとするならば、短い時間でも必要十分なテスト内容を網羅できるよう、このプロセスをできるだけ効率的に進めることが大きなテーマになってくる。

そこで積極的な検討したいのが、テストの自動化である。行う必要があると考えるテスト項目を、テスト自動化ツールというシステムの力を使って自動的に行うのだ。システムは、これこそシステム開発企業が開発して顧客企業に価値を提供しているものだ。それを自らが享受して悪いわけはないだろう。

だからといって全自動化を進めているわけではない。むしろ全自動化はしない方がよい。これは製造業における工場生産を例にとればよくわかる。今や工場生産は機械化が進んでいて、部品の取り付けはロボットアームが行うが、しかし、その一方でボディラインの微妙なカーブは最終的に人間が目視と触覚でチェックしたりする。テストもそれと同じである。テストの中には、繰り返し試験など、人間が行うと間違えたり、退屈なあまりにモチベーションが保てないものがある。また、超大規模な負荷試験など人的リソースが少ない状況で簡単にできないものもある。そういう、ツールを使えば高い効果を得られる分野を選んで自動化というソリューションを導入する。そうすれば、ほんとうに限られた時間であっても最初に設定した品質レベルを断念しなくてもすむというわけだ。

## ● 自社開発ツールにはリスクがある

なかには、“そうしたツールはすでに自社開発で持っている”“ほんとうに必要と思えば自分で作るよ”という技術者の方もおられるかもしれない。確かに、顧客企業にシステムを提供できるのだから、自分たち向けにシステムを用意するのは一見簡単なことである。たぶん、それはテスト業務に経験豊かなエンジニアが、自分の精通している言語でテストスクリプトを組み、それをテストメンバーで共有するということになるだろう。確かに、それであれば最も低いコストで自動化に着手でき、その言語を

知っているエンジニアにとってはどんなテストを行っているかもわかる。

しかし、逆にいえばその言語を知らない他のメンバーにとっては、まったくブラックボックスになってしまうということである。そのため、スクリプトのメンテナンス、そのスクリプトを動かす環境のバージョンアップ対応などは、作成者本人が担当し続けることになる。本来業務のかたわら、ずっとボランティアワークを担わなければならない。彼がそれを苦もなくこなしたとしても、やがて人事異動や転職、退職が発生するものである。そうなったら、そのテストスクリプトは完全ブラックボックス化し、とたんに利用できなくなる。もちろん、基本的には自社開発のテストツールは役に立つ。特に、閉じた環境で使う、特定のメンバーで使う、短期間のみ使う、など、いくつかの条件下においては、手軽でコスト的にも割に合う。

しかし、その一方で、前述したとおり、テスト範囲の拡大や使用の継続性という点では、潜在的なリスクを有していることを忘れるわけにはいかない。中長期的な観点で仕組みとしてのテスト自動化を考えるならば、他の選択肢も視野にいれたいところだ。

## ● 技術者に選ばれる「Spirent iTest」

こうした中で、テストの自動化をツールソリューションとして提供、業界に広く愛用されているのが、米国 Spirent Communications 社のテスト自動化支援ソフトウェア「iTest」である。同社は、有線から無線、そして衛星に至るまで、通信ネットワークを利用するあらゆる顧客企業の QoE(Quality of Experience) を高めるソリューションを数多く提供している。「Spirent iTest」は、ネットワーク機器操作はもちろんのこと、Web や Java Swing といったアプリケーションテストの自動化にも対応、試験系全体を自動化することによって、試験精度や網羅性の向上、リソースの最適化にも貢献する。

この製品の最大の特徴は、ツールを使うためにツールに最適化されたテストシナリオとテストスクリプトを用意しなくてよいという点である。パッケージアプリケーションと

してのテスト自動化ツールは他にもいろいろあるが、多くは使い始めるためにテストシナリオを作るところから始めなければならない。しかし、この製品は、今まで実施していたテスト操作をこのツールで実際に行うだけでよい。特別なスクリプトを書く必要はまったくないのである。より具体的な流れを説明すると、

## 1 実行させたい動作を「Spirent iTest」上で一度行う

## 2 「Spirent iTest」が操作を全て記録、シナリオの自動作成

## 3 試験シナリオを実行、もしくは編集

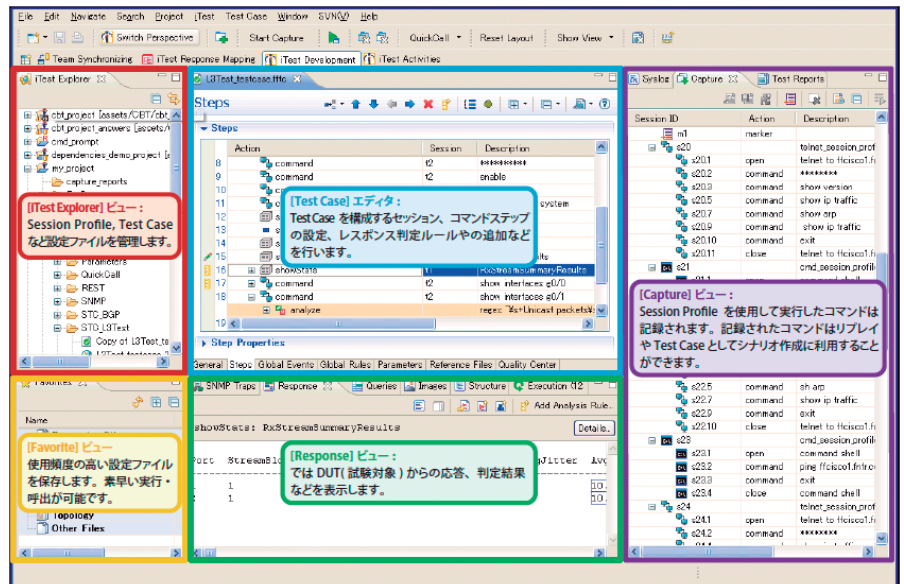
## 4 実行内容、結果を判別、合否判定や操作の条件分岐を行う

となる。マニュアルによる通常の機器操作が自動化シナリオとなる。これはつまり、テストを行うメンバー内の誰でもが自動化シナリオを作れて、読めるということだ。

### 「Spirent iTest」を使ったテストシナリオ作成方法

画面 1 は、「Spirent iTest」の典型的なメイン操作画面だが、上記の流れをこの画面を使って説明しよう(画面 1)。

iTest Explorer ビューやテストシナリオを管理するビューだ。新規にファイルを作成したい場合は、このビューに新たにファイルを設け、ここからテストしたい機器の操作を、Session Profile を作るという形で行う。たとえば、telnet 経由でルータ



画面 1 「Spirent iTest」メイン操作画面

に接続し、コマンドを実行して応答をみるという具合だ。telnet のみならず、Web ブラウザ、ターミナルソフト、コマンドプロンプト、ネットワーク機器アプリケーション、フリーツールなど、「Spirent iTest」はさまざまなアプリケーションやデバイスを制御することが可能である。

そのようにして実行したコマンドは、Capture ビューですべて時間も含めて記録され、操作が完了すれば、その記録がそのままテストケースとして再利用可能になる。すぐにも自動化の効果を実感したいのであれば、これをそのまま使えばよい。

テストによっては作ったテストシナリオをカスタマイズして別バージョンを作りたいという場合もあるだろう。そういうときは中央の Test Case エディタで編集を行う。コマンドの挿入や削除、テスト対象機器からのレスポンス判定ルールを設定したり、レスポンスの結果によって条件分岐を設定したりできる。GUI ベースで構成されているため、特に開発言語などの知識がなくても、直感的に操作することが可能だ。

Test Case エディタの下の Response ビューは、テスト対象機器からのレスポンス結果や判定の合否結果を表示させるもの。その左隣は Favorite ビューといい、頻繁に利用する設定ファイルを保存してすぐに呼び出すことができる。

さらに「Spirent iTest」は、同時に複数のデバイスやアプリケーションをテストして、その結果を各々表示させることもでき

れば、テストシナリオの一部を部品化して、ほかのテストシナリオからプロシージャとして呼び出すことも可能だ。また、他のテストツール上にあるテストシナリオをインポートして、この環境上で統合的に実行することもできる。

### 「Spirent iTest」導入で得られる効果

この製品を導入することによって、どのような効果が得られるか。株式会社 東陽テクニカ 情報通信システム営業部 SE グループ 主任 伊原有希子氏は、次のように語る。

「まずは、テストシナリオが簡単に作れて、そのシナリオが誰にでもわかるということです。たとえば、ルータに接続して、バージョンを見て、トラフィック量を見て、テストフレー



情報通信システム営業部 SEグループ主任 伊原有希子氏

ムを送って、と普段使用しているコマンドがそのまま見えるので、テストシナリオの内容が一目瞭然です。

自動化には不慣れな技術者であっても、中身を把握した状態でテストを進められ、また新たに作成することも難しくありません。自動化を適用するには、一部の人が読める、作れる、使えるのでは不十分です。グループ内全員の共通ツールとなり得ることが大切なのです。

第二には、再現性です。設定したコマンドを、順序通りに、設定したタイミングで、マルチスレッドなど幅広いスコープで、さまざまな値を適用し実行することが可能です。

テストには使用するコマンドの種類、値、順序、回数、タイミング、などさまざまな要素が絡みます。人間ならば、たとえ同じ担当者が行っても前回と同じ試験にならないこともあります。別の担当者となればさらに、再現試験は難しくなるでしょう。

自動化とは、そういった試験者によるテスト品質の“ばらつき”を吸収する最適な方法であり、『Spirent iTest』はそれを最少の工数で実現するツールです。」

最近では、この製品をテスト自動化のためだけに使うのはもったいないと、さまざまな場面での利用が増加しているという。そのうちの 하나가、ネットワーク性能監視ツールの代用として使われるケースだ。確かに、さっと各対象機器に入ってそこからの応答をリアルタイムに得られる機能などは非常に有用である。また、これを定期的に行う、性能管理する機器数を増やす、内容によってアラートをあげる、など、他にもいろいろ用途が考えられる。これは賢明なアイデアといえる。

## ネットワークテスト資源を予約 / 実行管理できる連携ツール

本稿では「Spirent iTest」と連携して利用できる配線 / 試験設備のマネジメントツール「iTest Lab Optimizer」も紹介しておきたい。

これは、一言にいうと、ネットワークテスト資源の予約 / テスト実行管理を統合に行うツールである。あらかじめ L1 スイッチに物理結線を含めてすべてのネットワークテスト環境を集約しておけば、その環境上で行うさまざまなテストが遠隔地からでも柔軟に行える。

具体的には「iTest Lab Optimizer」が、そのときにテストしたい論理トポロジーを予約時間に自動アクティベートし、その環境構築を自動的に構成する。その後、機器の設定変更などを含め「Spirent iTest」で自動テストを実行すれば、最後は「iTest Lab Optimizer」がテストに利用した論理トポロジーを開放する。

なんとといってもユニークなのは、利用したいネットワークテスト資源を、まるで会議室予約のように「iTest Lab Optimizer」上から予約できることである。これまでは、そうした機器は、その場へ出向いて周りのスタッフに聞かなければ、いつ誰が何時間ぐらい使う予定なのか把握できなかった。しかし、この製品によって、テスト環境が技術者のいるオフィスからどれだけ離れていようと、たとえそこが無人であろうと、自席に座ったまま資源の空いている時間が一目で見つけられるようになる。また、予約時間になると、テスト構成の物理結線を L1 スイッチが自動構成するため、テスト

ごとに物理的な結線変更作業を行う必要もない。

その結果、トータルのテスト実行時間を従来の 1/2 近くまで短縮することができる。

さらにこの製品は、自動アクティベート機能を活かして、頻繁に配線変更が発生するネットワークのコントロールに利用することも可能だ。

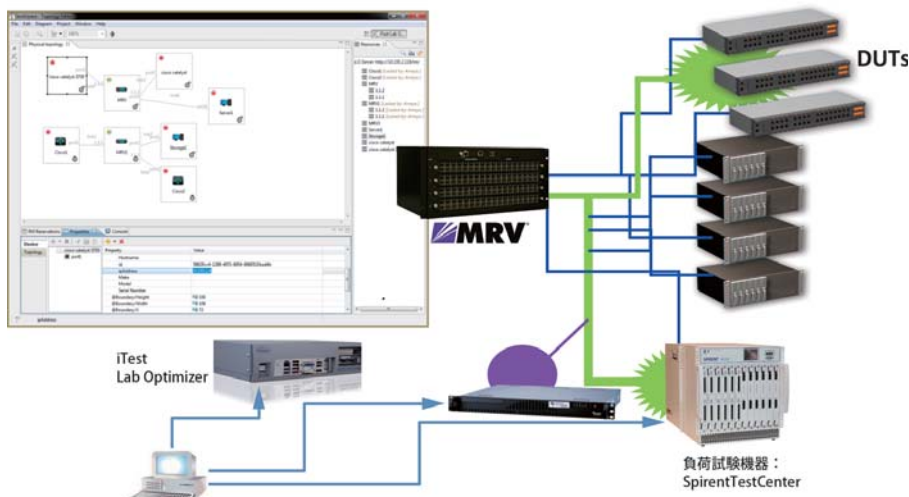
## テスト工程の“濃度”を高めるために

本稿では、システム開発とそのプログラム品質を担保すべきテスト工程の現状と、課題解決策としてのテスト自動化支援ソフトウェア「Spirent iTest」の概要と機能、その延長線上で利便性高く活用できる「iTest Lab Optimizer」を概観してきた。

いかがだろうか。テスト工程といえば、こなすべき課題が山のようにあるのはわかっていながら、時間やコストとの板ばさみとなって、どこをどう切り崩せば無事ゴールにたどりつけるだろうと暗澹たる心情に陥りがちである。

しかし、ここに「Spirent iTest」という自動化実現の道具を投入することによって、テスト工程に置くべき焦点が定まり、その実行内容にメリハリをつけることができる。その結果、時間やコストといった制約がある中でも、プロセス全体を濃度高く進めることが可能だ。

システム開発をめぐる環境は今後も激しく動き続けるだろう。しかし、テスト工程は常に“誇り”をかけた工程であり続けたい。幸い、私たちにはこれを実現する知恵があり、「Spirent iTest」という心強い味方がいる。テスト工程を充実させよう。



株式会社 東陽テクニカ 情報通信システム営業部  
Spirent Communications社製品窓口  
〒103-8284 東京都中央区八重洲1-1-6  
TEL.03-3245-1250 (営業直通) FAX.03-3246-0645  
URL. <http://www.toyo.co.jp/itest>  
E-mail. [spirent-web@toyo.co.jp](mailto:spirent-web@toyo.co.jp)