

システム開発におけるテストの取組み ～テストングフレームワーク～

Fujitsu's Efforts for System Development Tests: Fujitsu Efficient Testing Framework

● 神保寿久 ● 荒井一仁 ● 楠本博巳

あらまし

お客様システムの開発において、コストの多くがテストに費やされている。しかし、テストに十分な時間とコストをかけたとしても、必ずしもシステム品質が良くなるというものではない。このような中、富士通は効率的で高品質なテスト作業を可能とする Fujitsu Efficient Testing Frameworkを開発した。本フレームワークは、テストの作業手順やドキュメントが標準化されており、効率的かつ効果的なテストが可能である。また、テスト結果を見える化し、その出来栄を品質の出来栄として捉えて評価する考え方を導入しており、出来栄を分析・評価しながらテスト計画～実施～評価～見直しのPDCAによる品質ライフサイクルマネジメント(QLCM)が行える。さらに、時間・コスト・品質をバランス良くマネジメントしながら、システムのライフサイクルに沿って品質を確実に高めていくことが可能である。

本稿では、システムの品質を維持・向上していくために必要な要素と課題を明らかにし、その課題を解決するために開発したテストングフレームワークの機能と特長について紹介する。

Abstract

In the development of a customer system, a considerable amount of money is used for the testing processes. However, even if sufficient time and money are used, the system does not necessarily become a good quality one. Under these circumstances, Fujitsu has developed the Fujitsu Efficient Testing Framework which enables efficient and quality test work. Since the work procedures and documents of the test are standardized in this framework, an efficient and effective test becomes possible. Moreover, the framework allows us to perform Quality Life Cycle Management (QLCM) via the PDCA cycle of test planning, testing, evaluation, and reexamination, analyzing and evaluating the system's performance. Furthermore, it is possible to ensure higher quality over the life cycle of a system, maintaining a sufficient balance between managing time, cost, and quality. This paper clarifies the elements required to maintain and improve the quality of a system and the issues involved, and describes the functions and features of the testing framework that we developed to solve those issues.

まえがき

ビジネスにおけるシステムの重要性が高まるに従って、お客様（経営者層）にとってシステムの品質をどのように判断し、維持・改善・向上していけばよいか重要な課題になっている。

一方で、長引く経済不況やグローバル化の進展に伴い、システム開発を取り巻く環境は大きく変わりつつある。特に、オフショアなどによる分散開発および短期開発の傾向が強まる中で、システム開発における生産性向上や品質管理の重要性は更に高まっている。

経営者視点から求められる確かなシステム品質の確保には、これを担保するための業務アプリケーションのテスト仕様書作成やテスト実行にコストがかかり、大きな課題になっている。さらにシステムの規模が大きくなると業務改版時の再テスト量も多くなり、そのコスト増大も大きな課題になっている。

富士通は、金融・公共・通信などの分野において、多数の重要なミッションクリティカルシステムを、高信頼・高品質なソフトウェアによって構築、運用してきている。また、テストの結果がそのままシステムの品質を反映するような業務仕様にひも付いた、論理的かつ網羅的なテスト仕様の作成をはじめ、多数のミッションクリティカルシステム構築の経験に基づく、豊富な品質マネジメントプロセスと品質分析ノウハウを保有している。

これらのノウハウを結集して開発したFujitsu Efficient Testing Frameworkは、テスト工程の徹底した標準化と、携わるメンバの専門化・専任化を図り、テスト仕様書から実行、結果整理までを効率的に行うことができるようになっている。

本フレームワークの特長は、マトリックスやチェックリストを使った論理的な業務分析と整理手法、テスト観点別のテスト実績に基づく品質分析手法、テスト仕様書作成やテスト実行の自動化ツール、テスト資産（テスト仕様書、テスト結果、証跡など）の保管・維持・再利用ができるリポジトリなど、各種手法やツール群を備えていることである。

本稿では、システム品質に関する課題を、経営者サイド、および開発者サイドの両面から捉え、

その課題を解決するために開発したテストフレームワークについて紹介する。

現状の課題

本フレームワークの活用により短期間で効率的なテストが可能（当社従来比30～50%削減）となり、従来と比較してテストの負担を気にすることなく毎回多くのテストケースをこなすことができるので、確実な品質の担保ができるようになる。

システムの品質は、開発者サイドだけではなく経営者サイドの視点からも捉える必要がある。

経営者サイドは、指標値ベースの品質報告を受けても品質が良いのか判断がつかないため、品質に疑問を感じても的確な対応指示ができないといった課題がある。一方、開発者サイドでは、テスト資材の準備に時間を要することや、テスト実施要員の確保ができないといった課題がある。

本フレームワークにより、これら課題を以下のように解決する。

(1) 経営者主体の品質検証を実現

テストの出来栄を、品質の「出来栄」として捉えて見える化する。「出来栄」の分析から必要なアクションへつなげる品質ライフサイクルマネジメント（QLCM）を実現する。

(2) 開発者向けに効率的なテスト環境を提供

テスト項目の抽出や整理手法の徹底的な標準化により、テスト資材・手順を保管・再利用できる仕組みを提供する。さらに、テスト自動実行～証跡保管～報告書作成のツールと連携させることにより、担当者のスキルに依存しない効率的なテストを可能にする。

テストフレームワークの概要

Fujitsu Efficient Testing Frameworkに基づくテスト支援サービスが「Fujet」である。以降では「Fujet」を例に、新しいテストの取組みを紹介する。

Fujetサービスは、三つの層から構成されている（図-1）。

検証サービス層には、製造工程の早期品質検証を目的とする「プログラム品質検証」、業務機能の品質検証を目的とする「業務品質検証」、システム性能や運用性など非機能と呼ばれる要件の品質検証を目的とする「業務基盤品質検証」の三つの領

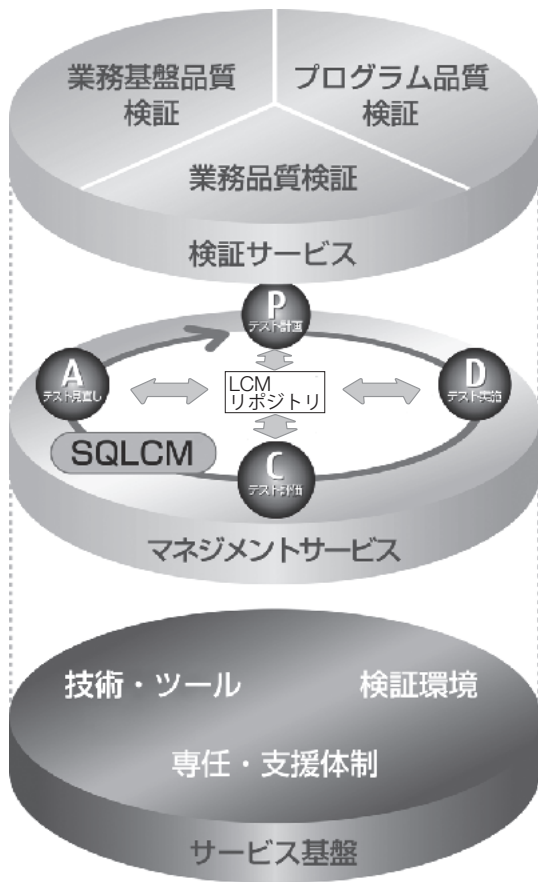


図-1 Fujetサービスの三つの層

域がある。

マネジメントサービス層は、検証サービス層で利用するテスト項目やテスト結果の保管と管理を行い、これら資材を使い品質に関するPDCAサイクルマネジメント（SQLCM：システム品質ライフサイクルマネジメント）⁽¹⁾を実現する部分である。

サービス基盤層は、テストフレームワークを支える基盤であり、テストケース抽出技法、品質分析技法、観点体系といった技術・ツール部分、テスト実施の技術を持った専任要員、検証を行う環境を意味する。

以下、各層の詳細を記述する。

検証サービス

● 業務品質検証

業務品質検証とは、業務分析～テスト項目作成～テストデータの設定～テスト準備～テスト実施～結果比較・保管までの一連の業務テストのことである。

(1) 業務分析～テスト項目作成

要件仕様書などからテスト項目の要素を抽出する。テスト項目の要素とは、テスト項目作成の切り口（観点）であり、要素と構成する値、値の範囲を使って、マトリックスなどの手法で一つのまとまったテスト項目群を作成する。こうすることで、テスト項目の組合せを見える化でき、作成したテスト項目をスキルに依存しない再利用可能な資産にすることが実現可能になる。

(2) テストデータの設定

バリエーション考慮時にペアワイズ法^(注)などの最適化手法を導入している。また、画面を中心としたテスト項目抽出では、投入項目の仕様に基づくバリエーションや、画面レイアウトを対象としたテスト項目を作成する。

(3) テスト準備

テスト項目実施の前提となる環境を整えたり、テスト結果の定義、テスト項目をいくつか組み合わせるテストシナリオ（テスト手順）を作成したりする。次に、テストシナリオに基づきテスト自動実行のためのスクリプトを作成する。

(4) テスト実施

実行、結果照合、テスト証跡保管、レポート作成までツールにより自動実行となる。夜間などにテストを実施し、翌朝に結果を確認するなど効率的なテスト作業が可能になる。

(5) テスト結果比較・保管

テスト項目ごとに、テスト項目内容、テスト結果、画面証跡がひとまとまりでWord形式文書で保管される。後でテスト内容を確認したり、リグレッションテスト時に前後でテスト結果を一括比較したりすることができるので、簡単に差異を確認することができる。

● プログラム品質検証

プログラム品質検証は、プログラム製造工程の段階から、ソースコードチェックや脆弱性チェックを静的に行い、プログラムの基本的な品質を検証するサービスである。複数の言語・プラットフォームに対応した静的解析ツールを、必要に応じて必要期間だけ利用することができる。開発規

(注) オールペア法とも呼ばれ、テスト項目の網羅を検討する際に、全網羅しなくて済ませる効果的なテストケースを作るための技法。

約準拠性、セキュリティホール、メモリリーク検出など212種の検証機能を備えている。

さらに、診断結果を基にしたプログラム改修方法などのコンサルティングサービスも用意している。

● 業務基盤品質検証

業務基盤品質検証は、非機能要件（信頼性・運用性・性能）に関する評価を行って、業務基盤の品質を検証するサービスである。

業務基盤品質検証には、二つのプロセスがある。

(1) 非機能アセスメント

ヒヤリングを行って、非機能の要求レベルを判定する。その後で、非機能の出来栄をチェックして具体的かつ実践的な評価レポートを作成する。

(2) 非機能検証

非機能アセスメントにより、品質不足と判定された部分について具体的なテストを行う。

マネジメントサービス

検証サービス層のテスト結果は、そのテスト項目とともにリポジトリに蓄積・格納されている。この情報を使って、以下の四つのプロセスに従ったシステム品質マネジメントを行う。

(1) テスト計画 (Plan)

テスト実施の方針を決定する。業務の重要性やリスク度を考慮してテストの範囲や重み付けを行う。さらに、テストの実施目標（項目の抽出観点やカバー率など）を決めてスケジュール化する。

(2) テスト実施 (Do)

テストを実施し、結果の記録や、トラブル状況や進捗情報を収集する。

(3) テスト評価 (Check)

テスト結果を見える化し、その出来栄を品質の出来栄と捉えて評価し、テスト計画に不備・不足がないか判断する。評価では、トラブルの発生状況や各評価領域におけるバグの収束性、リスク度、重要性を考慮しながら分析・評価を行う。

(4) テスト見直し (Action)

テスト評価の結果、テスト不足と判断された場合、テスト項目の追加や、対象（観点）の追加を考える。

本マネジメント (SQLCM) 層のもう一つの役割

は、テスト項目とテスト結果のトレーサビリティである。テスト結果から、関連するテスト項目、さらにプログラム、仕様書に至る情報を引き出すことができる。これにより、仕様変更に関わる関連テスト項目のみの抽出、バグに関わる関連テスト項目のみの抽出などが可能になるので、システム維持フェーズにおける品質マネジメントも容易にできる。

サービス基盤

Fujetフレームワークの初版は、端末を使ったテスト環境を対象にしている。

その特長は、テストのための環境として、Web端末上にテストツールを導入するだけで済むことである。サーバ側（被テスト側）には何も変更は必要ないので、本番環境を使ってテストすることもできる。

サーバ側の開発言語や、プラットフォームにも依存しないので、マルチベンダシステムであってもシステム全体をカバーできる共通テスト環境を構築できる。さらに、端末シミュレータを使って、外部（例えばクラウドセンター）から被テスト側をテストすることができるので、特別なテスト環境をサーバ側に用意する必要もなく設備費用を抑えることができる。

む す び

システムの品質を見える化して評価していく試みは過去からあった。しかし、そのためには膨大なテスト項目とその結果を蓄積・分析していく必要があり、期間とコストの関係から残念ながら断念されていた。

本フレームワークは、効率的なテストを可能とするために、テストに関連するドキュメントの標準化、作業プロセスの標準化、テストの自動化、専門要員の確保といったテスト基盤を整備している。

本フレームワークの活用により、膨大なテスト項目を蓄積し、繰り返し利用することが可能になり、テスト実績（動作を担保しながら）によってシステム品質を確認し、品質を積み上げていく新しいテスト手法が可能になった。

参考文献

- (1) 井上英明：バグを見つける富士通の新サービス。
日経コンピュータ 2011年9月15日号, No.791, p.16
(2011).

著者紹介



神保寿久 (じんぼ としひさ)

(株) 富士通ミッションクリティカルシステムズ
先端ビジネス本部第三統括部 所属
現在, テスト自動化サービスに従事。



楠本博巳 (くすもと ひろみ)

(株) 富士通ミッションクリティカルシステムズ
先端ビジネス本部 所属
現在, 新技術を使ったソリューション開発に従事。



荒井一仁 (あらい かずひと)

(株) 富士通ミッションクリティカルシステムズ
先端ビジネス本部第三統括部 所属
現在, システム品質検証サービスに従事。