

# 「リスクゼロ」の実現に向けて 知識集約型開発における テスト自動化の狙い

開発要員の労力を上流工程に集中し、中流から下流は極力自動化することにより高品質化を目指す知識集約型開発。日本電気の大場氏は、以前よりその重要性を説いてきた。本セッションでは、知識集約型開発の中・下流工程で新たに実現可能となったテストシナリオの自動生成および自動テスト実行について解説。従来の「リスク回避型」から「リスクゼロ」のソフトウェア開発への転換に向けた取り組みを紹介した。



## 労働集約型から知識集約型開発へ

最近のアプリケーション開発の傾向として、開発期間の短縮化が挙げられる。シェア獲得のための競争が激化し、他社よりも優位性のあるサービスを早期に提供するために、短期開発のニーズはますます高まっている。一方で、仕様調整の難易度が高いシステムが増え、上流工程の期間およびそこにかかる費用は増大傾向にある。開発期間の短縮に対応するためには、中流から下流工程の大幅な効率化が不可欠であるのは明らかだ。

開発期間やコストのほかに、開発リソースの課題もある。この数年、SEは「新3K」のイメージが定着してしまい、不人気職種となりつつある。そのため、優秀なSEリソースの確保が非常に難しく、要員の育成もままならない企業も珍しくない。

こうした現状を打開するために求められているのが、従来の「労働集約型開発」から「知識集約型開発」への転換だ。知識集約型開

発とは、開発要員の労力を上流工程や評価試験の設計に注力させ、中流から下流工程はできるだけツールを活用して自動化する開発スタイルである。維持フェーズにおける効率化のために、設計情報を一元管理し、メンテナンス性の向上やデグレードの抑止も可能とする。

これらを実現するソリューションとして日本電気が推進しているのが、「知識集約型TCO削減指向AP開発ソリューション」だ。

## 仕様のゆらぎを防ぎ、追加・変更にも容易に対応

知識集約型TCO削減指向AP開発ソリューションは、様々なアプリケーション開発テクノロジーや開発手法、プロジェクトマネジメントの要素を連携したソリューションとなっている（図1）。その一要素が、アドバンスドDOAによる上流設計だ。これは通常のDOA（データ中心アプローチ）とは異なり、業務視点（ルールベース）のアプローチとデータ中心のアプローチを組み合わせたものである。ユーザー視点に立ち、見慣れた業務視点でユーザーの理解しやすい業務フローを主体としたアプローチで業務知識（ビジネスルール）をルールベース化し、Xupper IIのリポジトリで一元管理。同時に、データ中心のアプローチでビジネスルールとエンティティの関係を分析して、アプリケーション構造を確定する。

上流設計工程において仕様のゆらぎを防止したり、仕様の追加・変更に対応できるようにする上で有効な知識集約型TCO削減指向AP開発ソリューションの特徴としては、ほかに業務モデル化やイテレーション開発などがある。業務フローによって業務をモデル化することで、IOやデータ項目

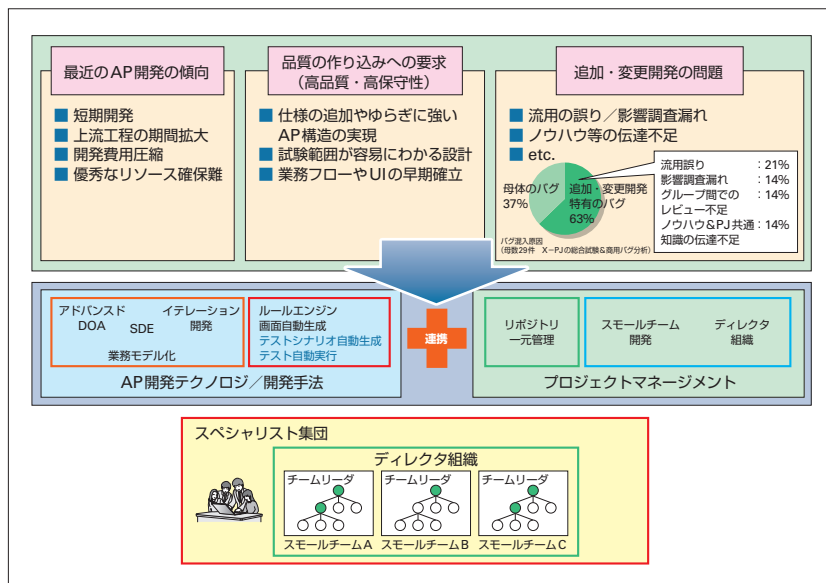


図1：知識集約型TCO削減指向AP開発ソリューション

などの影響ポイント（管理項目）を明確化できる。また、基本設計の確定後、業務要件の成熟度をイテレーション開発で段階的に上げていき、業務フローの要件やアプリケーション構造の確定を実施することで、仕様要件のゆらぎを抑えることが可能となる。

これらはいずれも、すべての設計情報をXupper IIのリポジトリで一元管理することが前提となっている。設計情報の一元管理（リポジトリ化）により、上流から中流・下流まで一貫して、設計情報を継承することができるのだ。

### テスト自動化の真の狙いとは？

例えば仕様の追加・変更時などは、テストの効率化や開発期間短縮のために、すべてのテスト項目をカバーするのではなく、サンプリングでテスト項目を設定する「リスク回避型」のレグレッションテストが行われるケースも少なくない。しかし、バグなどの問題は往々にしてサンプリングから漏れたところで発生する。そこで、知識集約型TCO削減指向AP開発ソリューションでは、「リスクを回避する」という考え方ではなく、「リスクゼロ」を目指すというスタンスをとっている。

そのために取り入れているのが、テストシナリオの自動生成およびテスト自動実行だ。Xupper II Ver9.0の新機能である自動テスト連携オプションを利用したのだが、元々これは大場氏の要望により開発が始まり、実装されたという経緯がある。

リポジトリの設計情報からテストシナリオを自動生成すれば、手作業で行う場合のような設計書からテスト仕様書への転記の誤りや漏れなどは発生しない。テスト観点の漏れや確認漏れの防止にもつながる。そして、テスト仕様にあるすべてのテストを自動実行することで、漏れの無いレグレッションテストを安価かつ短期間で実現できる。つまり、テスト自動化の目的はテスト工程の効率化や開発期間短縮だけではなく、真の狙いは「リスクゼロ」のソフトウェア開発を実現することにあるということだ。

### テストシナリオの自動生成からテスト自動実行の流れ

テストシナリオの自動生成とテスト自動実行は、Xupper IIとフロンテス社のSTAR-ATT (Auto Test Tool)、HP社のQTP (QuickTest

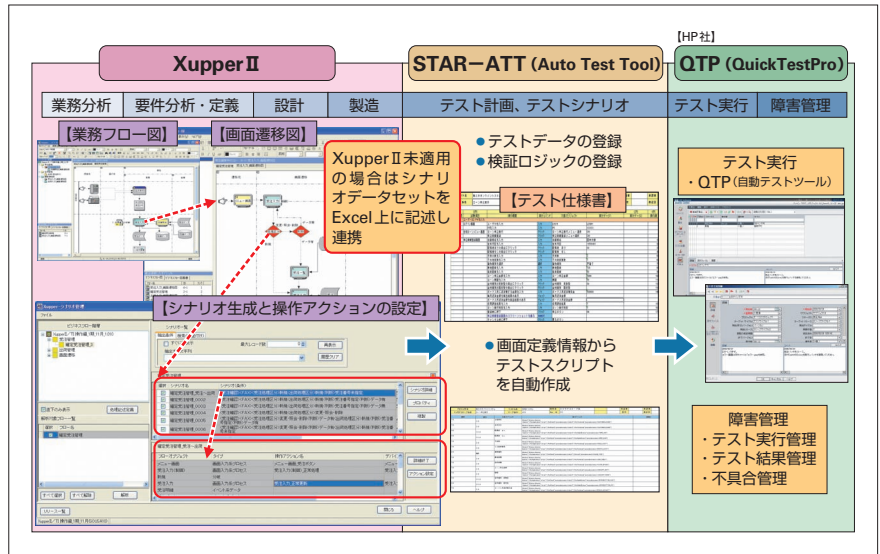


図2：テストツール連携全体図

Professional) という3つのツールの連携によって実現されている（図2）。

まず、Xupper IIでは、ビジネスフローに記述された機能（プロセス）に属する画面について、画面遷移図を定義する。続いて、1つひとつの画面設計と操作アクションの定義を行い、ビジネスフロー図および画面遷移図を解析してシナリオを抽出。シナリオが生成されたら、さらに詳細な操作アクションを定義する。なお、テストシナリオは画面遷移のパターンをすべて抽出して自動生成されるので、画面遷移図が正しく描けていれば、シナリオの漏れは決して発生しないという。

次に、自動実行用のテストスクリプトを作成するために、Xupper IIで作成したテストシナリオはSTAR-ATTに渡す。STAR-ATTでは、テストデータと検証ロジックを登録。検証ロジックについては、人間が作業する内容と同じ検査が実現できるよう豊富なコマンドが用意されている。テストシナリオにデータと検証ロジックを定義後、テストスクリプト（QTPスクリプト）が自動生成され、最終的にQTPと連動して自動テストが実行される。

テストシナリオ自動生成からテスト自動実行までのこうした一連の流れについて、セッション終盤では、携帯電話の端末を選んで購入するというサンプルアプリケーションを用いたデモを交えながら紹介された。

大場氏は最後に、こうした取り組みを更に洗練させていくことで、長らく変わっていないAP開発の「高度化」と新3Kからの脱却を推進し、日本のソフト開発産業の発展に寄与していくと述べ、講演を締めくくった。