

# 結果の妥当性確認を分離した 汎用テスト自動化アーキテクチャの提案と適用検証

寺本 弦生(富士通株式会社)/ 渡邊 宏太郎(株式会社デンソー)/ 宮崎 大希(富士通株式会社)

## テストにおける課題

ソフトウェア開発ではテストの効率化が必要だが、自動化には課題が多い。

様々な開発対象や自動化の取り組みがある中、テスト結果の妥当性確認において共通課題がある。

- ・手動実施のため多くの工数を要する
- ・自動化済みでもレガシー資産となっている

## 手法・ツールの適用による解決

1. テスト自動化の構成要素をまとめた「汎用テスト自動化アーキテクチャ」を活用し、メンバ共通の課題を解決できるよう、結果の妥当性確認を分離するアーキテクチャ(cgTAA)を作成した。
2. 作成したcgTAAを基に、成熟度の異なる各職場ごとにテスト自動化ソリューションを作成/検証した。

## 汎用テスト自動化アーキテクチャの具体化と Assert分離の提案

ISTQBの汎用自動化アーキテクチャ[1]を具体化したcgTAA (Concrete Generic Test Automation Architecture)を提案。

- ・ 一般的なアーキテクチャ  
テスト記述はArrange-Act-Assertを揃える形で実現
- ・ 提案するアーキテクチャ  
「テスト結果の妥当性確認 (Assert)」について、  
Arrange, Actから分離する形で実現

[1] ISTQB, "Certified Tester Advanced Level Syllabus Test Automation Engineer Version 2016"

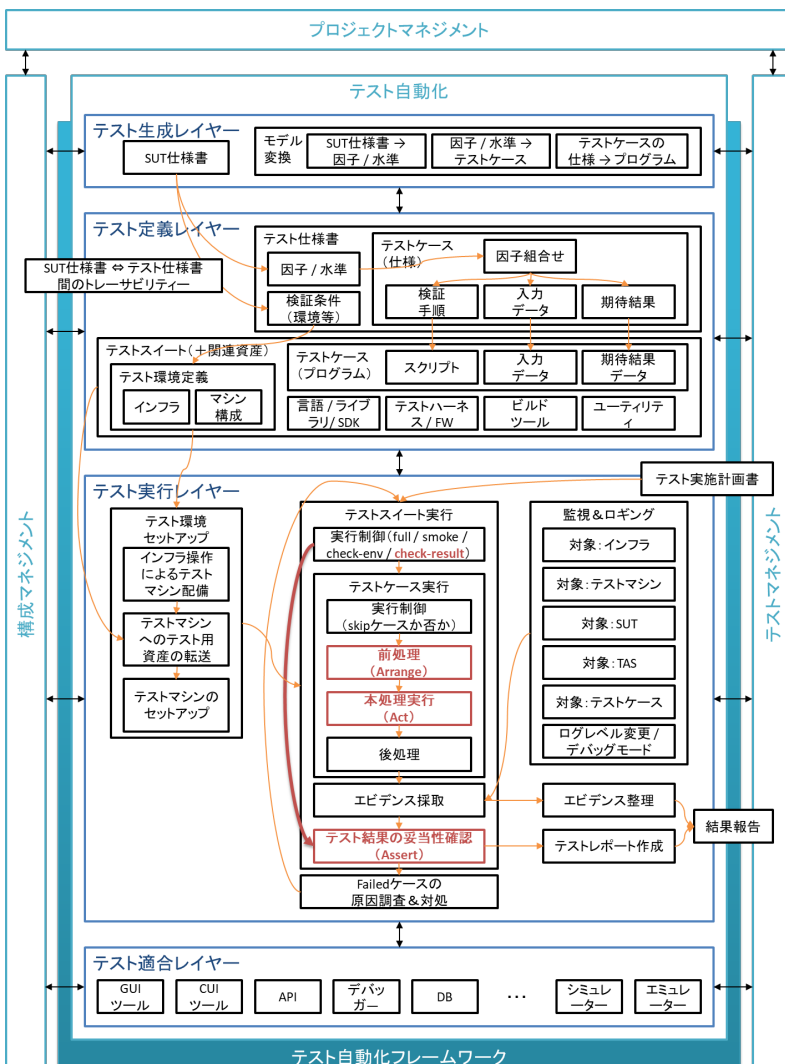


図1 テスト結果の妥当性確認 (Assert) を分離したcgTAA

## 本アーキテクチャの業務適用

開発対象や自動化成熟度の異なる3つの職場にて、本cgTAAに基づいてテスト自動化ソリューションを作成・検証した。

事例	開発対象	自動化の現況	テスト自動化ソリューションの効果
			Assert分離の効果
1	組込ソフト	全て手動	工数削減・検査品質の向上 手動のテスト実行処理から分離することで、Assertを全て自動実行できた
2	通信ソフト	一部自動化	工数削減 手動であったAssertの自動化について、既存資産に捉われることなく実装できた
3	Java実行基盤	一通り自動化	構成技術のモダナイズと工数削減 課題のあったAssertについて、既存資産に手を入れることなく改善できた

いずれの事例においても、一貫して提案アーキテクチャに沿った自動化/改善を適用でき、現実的なコストで実現できた。

## 本アーキテクチャの考察

一般的なテスト記述と比較したときに、Assertを分離して実現する本アーキテクチャの主なトレードオフを以下に示す。

### メリット

1. Assertの実装/変更時に、既存資産への副作用を抑制できる
2. 環境の制約が生じやすいArrange, Actの実行とは独立して、後でAssertを実行したり、別環境でAssertを実行したりできる

### デメリット

1. 記述が分散するため認知負荷増大の要因となる
2. Arrange, Actで有用な既存ユーティリティをAssertで活用し辛い (例: HTTPクライアントツール)