

スモールスタートによるテストツール導入の提案

- 「広く浅く」を前提としたキャプチャ/リプレイツールの簡易的な導入 -

Test tool introduction by “Small Start”

- Simply introduction of capture and replay tools based on broad and shallow method -

主査 : 奥村 有紀子 (有限会社デバッグ工学研究所)
副主査 : 秋山 浩一 (富士ゼロックス株式会社)
堀田 文明 (有限会社デバッグ工学研究所)
研究員 : 牛尾 誠一 (株式会社インテック)
南野 大介 (T I S株式会社)

研究概要

システム開発においてキャプチャ/リプレイツールを効果的に導入するためには、プロジェクトやシステム全体を考慮した計画や準備が重要である。しかし、計画や準備に工数を要することが、テストツール導入の障壁を高く感じさせる要因ともなっている。テストツールの導入においては、テストの重要性が高く、効果を発揮しやすい範囲から段階的に自動化することがポイントであることに着目し、テストツール導入の障壁を低くする検討を行った。その結果、多くのシステム開発プロジェクトでテストツール導入を容易にする方法として、テスト範囲をシステムの全画面、テスト目的や観点を画面遷移の導通確認テストとする方法だと考えた。この「広く浅く」を基本とした、スモールスタートによるテストツール導入を提案する。

Abstract

In Web system development a rigorous plan with considering the whole project and system is required in order to use capture and replay tools effectively. However using test tools has some difficulty that taking time for planning and preparing. We focused on that the point of test automation is phasing in test tools, and examined a method which enables to infiltrate test tools into many projects. We propose the small start method for test tools introduction: limited to broad and shallow introduction at first.

1. はじめに

本研究の対象は、テストツールのひとつであるキャプチャ/リプレイツールの導入障壁を低くすることで、テストツール未導入のシステム開発プロジェクトが導入を行いやすくする方法とする。

テストに必要な作業ごとに、各種のテストツールが存在する^[1]。そのなかで、画面 UI を持つシステムに対して、手動の画面操作を記録し自動実行するツールが、キャプチャ/リプレイツールである。統合テストやシステムリリース前の確認において、画面の動作確認テスト、回帰テスト、外部データを使用したデータ駆動型の組合せテスト、OS やブラウザの種類やバージョンなど複数環境上での動作確認テストなどでこのテストツールを使用する。

システムを設計・開発・保守するシステムインテグレーションを主に行う研究員の所属組織においては、キャプチャ/リプレイツールの導入が進んでいない現状がある。キャプチャ/リプレイツールと同じく統合テスト時のテスト実施プロセスを自動化する(以降「テスト自動化」と略す)、性能テストツールやセキュリティテストツールは、研究員の所属組織において導入が進んでいる。個々に開発した複数のプログラムを結合する統合テストにおいては、機能間の整合性など多くの問題が表面化し、多忙となる状況が経験的に多い。統

合テストにおける問題を減少させるとともに、テスト実施プロセスの作業効率を向上させ作業工数を削減するため、キャプチャ/リプレイツール（以降「テストツール」と略す）の導入を行いやすくする方法を本研究の対象とした。

2. テストツールの導入状況と満足度

テストツール導入に関する調査結果を用いて、テストツールの導入状況と満足度を考察する。表 1. に、統合テストにおけるテストツール導入の状況調査結果を示す。

表 1. テストツール導入の状況調査（2011 年日経システムズ調査^[2]）

調査内容	導入済	未導入	不明
テストツール導入の状況	13.8%	73.7%	9.5%

表 2. に、表 1. の状況調査においてテストツール導入済みと回答した人を対象とした、テストツールへの満足度調査結果を示す。

表 2. テストツールの満足度調査（2011 年日経システムズ調査^[2]）

調査内容	満足	やや満足	やや不満	不満
テストツールの満足度	22.2%	58.9%	17.0%	1.9%

表 1. および表 2. より、テストツール未導入の人が 7 割であり、テストツール導入が進んでいない状況である一方、テストツール導入済の 8 割の人がテストツールに満足していることが分かる。テストツール導入済と未導入、各グループの特性が不明なため一概には言えないが、7 割のテストツール未導入の人がテストツールを導入した場合、そのうちの 8 割の人がテストツールに満足する可能性があると考えられる。

3. テストツール導入の障壁

2. に示したように、テストツール導入済の人はテストツールに高い満足度であるにもかかわらず、テストツール未導入の割合は高い。その理由について、2. の調査結果、研究員の所属組織における調査結果、テストツール導入に関する資料を用いて考察する。

3.1. テストツールを導入しない理由

表 3. に、2. の状況調査においてテストツール未導入と回答した人を対象とした、未導入である理由の調査結果を示す。

表 3. テストツールを導入しない理由（2011 年日経システムズ調査^[2]）

理由分類	テストツールを導入しない理由	割合
導入コスト	導入コストが高い。	40.1%
不信感/不安感	手作業で行った方が早い。	34.0%
知識不足	どんなツールがあるのか知らない。	27.6%
ツールの機能不足	必要な機能を備えていない。	23.2%
保守性	テストスクリプト/テストデータのメンテナンスが大変。	19.4%
ツールの教育	操作を覚えるのに時間がかかる。	18.6%
導入コスト	ライセンス形態が使いづらい（プロジェクトに限定されるなど）。	11.4%

表 4. に、研究員の所属組織においてテストツール導入に関してヒアリングした結果のうち、表 3. には出ていない、テストツールを導入しない理由、または過去に導入したが今後積極的に導入しようとは考えない理由を示す。

表 4. 研究員の所属組織におけるキャプチャ/リプレイツールを導入しない理由

理由分類	テストツールを導入しない理由
導入タイミング	プロジェクト計画時にテスト自動化を予定していないため、プロジェクト途中でテストツールを導入しにくい。

理由分類	テストツールを導入しない理由
導入タイミング	テストツールを使用する計画がないまま、統合テスト実施中にテストツールを導入して失敗した。
導入準備	テストスクリプトの標準や共通がなく、テストスクリプト作成の効率が悪かった。
再利用性	テスト対象の画面ごとに外観や操作が異なるため、テストスクリプトの使いまわしが行いにくい。
テスト対象の範囲	画面操作をどのレベルでテストするか曖昧だったため、テスト自動化のテスト範囲とレベル感に違いが発生した。その結果、テスト結果の意味あいまでが曖昧になった。
テスト繰り返し回数	自動テストの実施回数を、プロジェクト計画段階では判断できない。
テストの教育	人がテスト結果を判断するのは簡単だが、テストツールにテスト結果を判断させるには、厳密な判断方法と基準の指定が必要で、テスト設計が難しくなった。
他ツールとの関係	開発・テスト環境が複雑化し、環境構築・保守の担当者内で環境の知識やノウハウがブラックボックス化してしまった。
不信感/不安感	テストツール導入効果は、開発プロジェクト全体の一部の効果でしかないわりに、事前準備など色々とやらないといけないことが多そうで大変そう。
不信感/不安感	テスト自動化は品質向上にはなるだろうが、大規模で長期間の新規開発や、派生開発の頻度が高い保守プロジェクトでないと、工数削減の効果は出ないのではないかと。
不信感/不安感	自動テストを成功させた実績を社内で聞かない。
不信感/不安感	テストツール導入は顧客からの要望で、安心感向上のために行っている。

3.2. テストツール未導入理由の分析と導入推進方法論のマッピング

3.1. に示したテストツールを導入しない理由への対策として、多くの導入事例や導入推進の方法論が示されている。表 5. に、テストツール導入を推進する方法論を示す。なお、テストツールを導入しない理由のうち「導入コスト」と「不信感/不安感」に分類したものは、表 5. のすべての対策に当てはまる理由のため、表 5. の「対応する理由分類」列には明示しない。

表 5. テストツール導入推進の方法論

対策分類	内容	対応する理由分類
テスト自動化の計画	<ul style="list-style-type: none"> ・状況に応じて自動化の戦略を選択せよ [3]. ・テストの自動化計画はプロジェクトの早い段階に立てる [3]. ・自動化によって発生するリスクを明らかにする [4]. 	導入タイミング 導入準備
テスト自動化のプロセス	<ul style="list-style-type: none"> ・テスト実装工数を適切に確保する [1]. ・自動化しやすい仕組みを作っておく [5]. ・テストツールを使うための準備をプロセスに組み込む [6]. 	導入タイミング 導入準備
テストの範囲	<ul style="list-style-type: none"> ・システムの品質向上に、特に効果のある領域を狙う [7]. ・ビジネスに影響の大きい重要なアプリケーションを優先的に自動化する [4]. ・同一操作を繰り返し実施する回数が多いもの、ビルド変更やパッチ適用の度にテストを繰り返すものを狙う [4]. ・スクリプト作成時間が短い部分、単純な操作、アプリケーションの中で仕様が安定してきた機能を自動化する [4]. ・何度でも同じ事ができる、ミスをしない、人手ではやりにくいような条件下のテストも実行できる、自動化の恩恵が受けやすい所を自動化する [5]. 	テスト対象の範囲 テスト繰り返し回数
テストツールの選定	<ul style="list-style-type: none"> ・GUI テストツールは、互換性と習得性とサポートで選べ [3]. ・ツールでできることとできないこと、信用していい部分と確認が必要な部分、ツールの特性を見極める [8]. 	ツールの機能不足 知識不足

対策分類	内容	対応する理由分類
教育	<ul style="list-style-type: none"> ・テストの自動化に必要なのは、プログラミング、テスト、プロジェクト管理のスキル[3]. ・期待値が明確でない仕様の対処を明確にすること[1] 	ツールの教育 テストの教育
再利用性, 保守性	<ul style="list-style-type: none"> ・テストスクリプトは再利用, モジュール化を考慮すること[1] ・ユーザーインターフェース要素の認識方法を考慮すること[1] ・保守性向上にはスクリプティングのテクニックを駆使する[4]. 	保守性 再利用性
他ツールとの統合	<ul style="list-style-type: none"> ・ツールを組み合わせることで可能性が広がる. 継続的インテグレーション(CI)の導入により自動化を促進. 上流工程のツールとテストツールの連携. 開発だけでなく計画や運用も含めたライフサイクル全体をツールにより管理(ALM)する[8]. 	他ツールとの関係

3.3. 導入コスト障壁と心理的障壁

3.2.に示したように、多くのテストツール導入推進の対策が示されているにもかかわらず、テストツール未導入の人が多く存在する。このことから、初めてテストツールを導入する人には、計画・準備・教育・テストスクリプト開発・テスト環境構築などに工数を要する「導入コスト障壁」と、計画や準備など多くのことを行う必要があると思わせる「心理的障壁」が存在すると考察する。なお、本研究の目的および以降の4.で提案する内容では、テストツールの機能不足がテストツール導入の障壁にはならないため、考察対象からテストツールの機能不足は除外した。

初めてテストツールを導入する場合、部分的に少しずつ、テスト自動化の効果を確かめながらテストツールを導入する方法が推奨されている^[9]。しかし、テスト自動化で多くの効果を目指すあまりに、多くの対策を実施しようとするのだと推測する。このことが、新たに多くの作業を追加しないとテスト自動化がうまくいかないと思い込ませ、心理的な障壁を高める要因になると考える。この心理的な障壁が高まることで、テストツール導入に踏み切れない状況を作り出していると推測する。

4. スモールスタートを前提としたテストツール導入

テストツール導入の心理的障壁が要因となり、テストツール導入に踏み切れない人に対しては、まずはテストツールを実際に使用しテストツールを知ってもらうことが、心理的な導入障壁を低くする効果的な方法だと考える。そのため、本研究では、心理的障壁が要因となりテストツール導入に踏み切れないシステム開発プロジェクトを対象に、テストツールを知ってもらうことを目的に、まずはテストツールを使ってもらう「スモールスタート」によるテストツール導入を提案する。

4.1 「広く浅く」を基本としたテストツール導入

テストツールの導入においては、テストツールの効果が発揮される部分に絞り導入することが良い方法とされる。図1.に、テスト自動化を部分的に導入する2種類の方法を示す。

テスト対象とするシステムの機能を部分的に限定し、複数のテスト目的を実施する方法を「限定的導入」と定義する。また、テスト対象をシステム全体とし、テスト目的を簡易な動作確認に絞る方法を「横断的導入」と定義する。このうち、本研究におけるスモールスタートは、「横断的導入」の方法により、テストツール導入で効果がある部分を絞る方法である。

機能単位に多くのテストツール導入の効果を発揮させる「限定的導入」は、表5.に示したようなテスト自動化のための多くの対策を実施しようと考えさせ、結果的に導入障壁を高めやすくしがちな方法だと考える。逆に、システムの全機能に対してテストツールの効果を少しずつ発揮させる「横断的導入」は、テスト自動化のための対策を少なくさせ、導

入障壁を低くする方法だと考える。

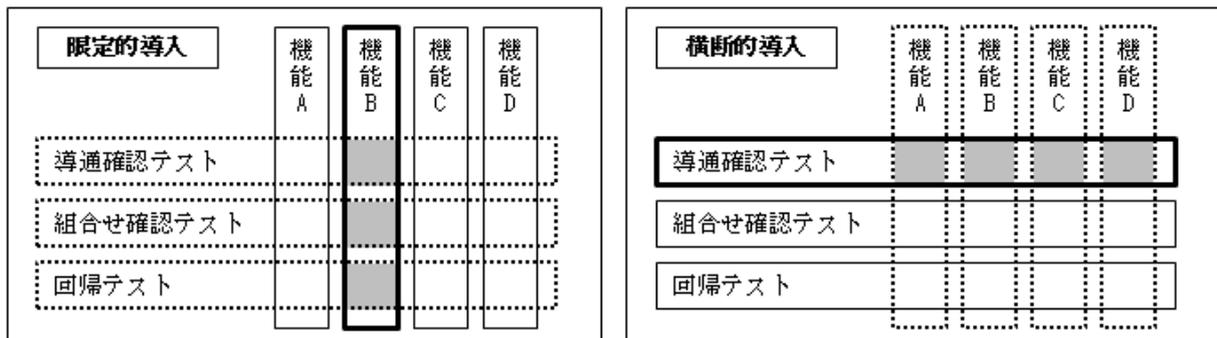


図 1. 限定的導入と横断的導入

「横断的導入」の方法で実施するスモールスタートは、以下の 2 点を特徴とすることで、テストツール導入に必要な追加工数の大部分を排除した方法である。

(1) システム全体の機能を対象とする（テストツールを「広く」導入）

テスト対象の範囲を、システムの全機能・全画面とする。また、テスト自動化を導入する対象システムを、画面 UI を持つ全システムとする。テスト対象の範囲を限定しないことで、テスト自動化の対象機能や画面を選定するための、調査や計画の工数削減が行える。

(2) テストの目的・観点を導通確認テストに限定する（テストツールを「浅く」導入）

テスト対象の範囲を限定しない一方、テストスクリプト作成に必要な工数を最小限にするようテストの目的と観点を限定する。具体的には、画面遷移が正しく処理されることを確認する、正常系の導通確認テストのみをテスト自動化の対象とする。テストケースの設計やテストスクリプトの作り込みを必要とする、異常系テストや回帰テストなどは対象としない。統合テスト実施のための受け入れテストとして、機能結合テストや統合テスト中のバグ修正後に正常系の導通確認テストを実施することを、スモールスタートでは重視している。

テストツールを導入する際、一般的にパイロットプロジェクトによる導入が行われる^[3]。これに対し、システムの規模や特性を選ばない点、事前の検討を経ずにすでに開発プロセスに入っているプロジェクトでも導入が可能な点、テストツール導入による効果の検証よりもむしろ「まずは使ってみる」「習うより慣れる」に重きを置いている点で、スモールスタートはパイロットプロジェクトとは異なる。

4.2. モデルプロジェクトによる考察

4.2.1. モデル設定

本研究で提案するスモールスタートによるテストツール導入の効果を定量的に分析するため、表 6. のようなモデルプロジェクトを設定し、スモールスタートによるテストツール導入において期待される具体的な効果を試算する。

表 6. モデルプロジェクト

プロジェクト要素	設定値
画面数	100 画面
工期	10 ヶ月
工数	100 人月
費用	100,000 千円（人月単価：1,000 千円）

4.2.2. スモールスタートにおける追加工数

図 2. に、モデルプロジェクトの開発プロセス例において、テストツールを導入するタイミングを示す。スモールスタートでは、統合テストの直前に、テスト実施担当者がテストスクリプト作成とテスト実施を行う。そのため、テストスクリプト作成とテスト実施のみが、スモールスタートのテストツール導入における追加工数となる。



図 2. モデルプロジェクトにおける開発プロセス例

図 3. に、モデルプロジェクトにおける画面遷移フロー例を示す。各画面遷移フローが 1 フローあたり 5 画面程度とすると、全体の 100 画面に対して必要なテストスクリプトは 20 本となる。また、画面遷移操作の自動記録のみがテストスクリプト作成に要する作業となるため、テスト実施を含め 1 スクリプトあたり 2 時間程度の工数に抑えられる。そのため、スモールスタート適用時の追加工数は、0.25 人月（全体工数 100 人月の 0.25%）程度に抑えられると試算した。

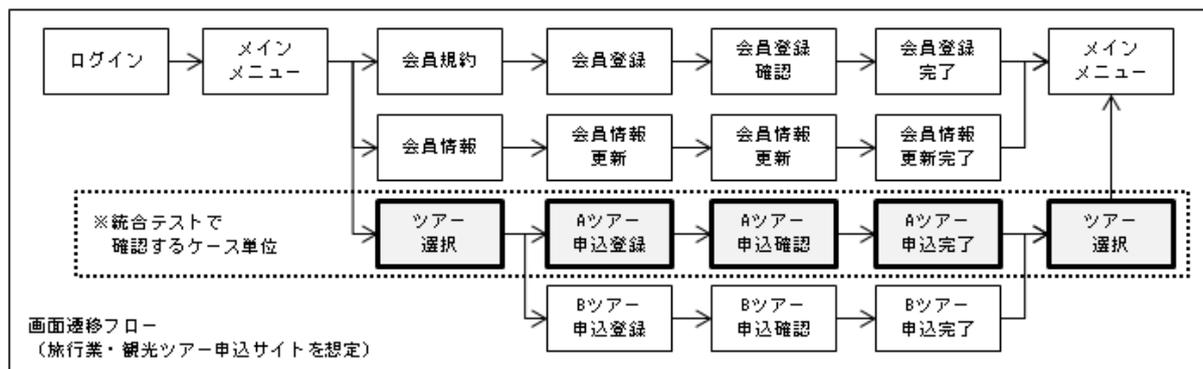


図 3. モデルプロジェクトにおける画面遷移フロー例

4.3. 期待される効果

4.3.1. 導入障壁の解消

表 7. に、3.2. に示したテストツール導入の障壁のうち、スモールスタートによるテストツール導入を行うことで導入障壁を解消できる内容を示す。

表 7. テストツール導入障壁の解消

対策分類	スモールスタートでの導入障壁の解消
テスト自動化の計画	テスト実施の直前にテストツール導入を決定可能なほど、簡易な計画と準備でよい。
テストの範囲	テスト目的とテスト範囲を限定しているため、テスト範囲の事前検討が不要。
テストツールの選定	テストツールの基本機能のみで実現できるテストのため、テストツールの選定が不要。
教育	テストを導通確認に限定するため、テストツールの高度な使用方法の習得が不要。
再利用性, 保守性	画面遷移フローや大きな変更が発生しない限り、テストスクリプトの改修が不要。

(1) テストの範囲

テストの範囲をシステム全体、テストの目的を導通確認テストと初めから割り切るため、テストツールを導入する対象システムの機能と、テスト方法の事前検討が不要となる。ま

た、テストツールで自動記録できる範囲の画面操作のみを導通確認テストの対象に限定するため、自動テスト箇所と手動テスト箇所を切り分けるための検討が不要となる。

(2) テストツールの教育

画面遷移フローの開始から終了までの、自動記録によるテストスクリプト作成とテスト自動実行のみを自動テストの実施範囲とするため、テストツール導入に関連する教育を簡易な内容で済ませることができる。簡易な内容の教育であるため、テストツールの理解が早く、教育期間が短く、教育工数も少なく済む。

(3) テストスクリプトの保守性

テスト自動化にあたって準備するテストスクリプトは、画面遷移を確認するのみの簡易な内容にとどめるため、画面遷移フローの変更を含む基本設計レベルでの大きな仕様変更が発生しない限り、テストスクリプトのメンテナンスが不要である。そのため、メンテナンス性を考慮したテストスクリプトの設計や実装、テストスクリプトのメンテナンスに要する工数の考慮が不要となる。

4.3.2. 副次的な効果

(1) 自動化テスト範囲の拡張性

スモールスタートでテストツールに慣れ、基本的なテストツールの操作を習得することで、テストスクリプトの作り込みや共通化、テストカバレッジの拡大などに応じやすくなる。そのことで、導通確認テスト以外の回帰テストや組合せテストの自動化に発展しやすくなる。

(2) 開発途中でのテストツール導入の決定が可能

3.2.に示したように、プロジェクト計画や要件定義などの段階で、通常はテストツール導入の可否を決定する必要がある。しかし、少ない追加工数で実施可能なスモールスタートでは、テストスクリプトを作成する統合テストの直前でテストツール導入の決定をしても、テスト自動化への対応が可能である。「途中からでも入れられる」気軽さも、テストツール導入の障壁を低くする要因となる。

(3) 後戻りが可能

少ない追加工数で実施可能なスモールスタートは、想定外の要因によりテストツール導入が不可能であることが判明した場合でも、それまでにテスト自動化に要し無駄となる工数が少ないため、導入を諦める選択が容易となる。「もし上手くいかなければやめてしまえばよい」気軽さも、テストツール導入の障壁を低くする要因となる。

4.4. 問題点と課題

(1) テストツールとテスト対象システムの特長

テストツールの利用ごとにライセンス費用が発生する場合、4.3.2.に示した「途中からでも入れられる」と「後戻りが可能」のスモールスタートの効果が無くなる。そのため、テストツールはフリーツール、もしくは所属組織内で標準ツール化され組織内の包括契約内で使用できるツールなどに限定される。また前提事項として、導入するテストツールはテスト対象システムの画面UI技術に対応している必要がある。

(2) テスト自動化の標準プロセスと標準基盤環境

テストツール導入に必要な作業を迷いなく計画および実施できるよう、所属組織の標準開発・テストプロセスに、スモールスタート用のプロセスを組込んでおく必要がある。ま

た、テストツール導入決定からテスト実施までを迅速に行えるよう、所属組織の共通開発・テスト基盤としてテスト自動化環境を用意しておくことが効果的である。

(3) テストデータの準備

テスト自動化を行う統合テストにおいては、画面遷移の確認を行うためのテストデータを準備しておく必要がある。導通確認においてテストデータが更新される更新系画面では、繰り返しテストを実施するためにはテストデータのメンテナンスが必要となる。参照系画面以外の更新系画面も導通確認テストの対象とする場合、自動テスト実施の前後でデータメンテナンス用スクリプトを実行するなど、何かしらのデータメンテナンスが欠かせない。

5. おわりに

今回の研究では、開発現場にテストツールを導入しやすくする手法として、スモールスタートを提案し、十分に効果を見込める手法であると結論づけた。プロジェクトマネージャの観点では、少ない追加工数でテストツールを導入できる効果、開発メンバーの観点では、テストツールの使用経験が培われる効果を見込んでいる。また、システム全体に渡って画面の導通が常に保たれることで、品質に対する安心感が高まると期待している。

ただし、システム開発プロジェクトにテストツールを導入した際に実際に現れる、具体的な効果や課題については、今後実地検証する所存である。

6. 参考文献

- [1] 特定非営利活動法人 ソフトウェアテスト技術振興協会 テストツール WG, テストツールまるわかりガイド (入門編), 2012, http://aster.or.jp/business/testtool_wg/pdf/Testtool_beginningGuide_Version1.0.0.pdf
- [2] 日経コンピューター, 開発支援ツール徹底調査 2011 年, 2011, <http://itpro.nikkeibp.co.jp/article/COLUMN/20110512/360285>
- [3] Cem Kaner, James Bach, Bret Pettichord, ソフトウェアテスト 293 の鉄則, 2003
- [4] 湯本剛, JaSST' 10 北海道 現場の力をメキメキ引き出すテスト戦略, 2010, <http://jasst.jp/archives/jasst10s/pdf/S1.pdf>
- [5] 細川宣啓, JaSST' 11 関西 テスト自動化の落とし穴, 2011, <http://jasst.jp/archives/jasst11w/pdf/S2-2.pdf>
- [6] 町田欣史, テストツール駆動によるソフトウェア開発プロセスの改善, ソフトウェア・テスト PRESS, Vol.10, 2009
- [7] 上甲貴広, JaSST' 10 東京 リスク・ベース・テストに基づくテスト自動化戦略の策定, 2010, <http://jasst.jp/archives/jasst10e/pdf/D5-2.pdf>
- [8] 町田欣史, JaSST' 11 関西 現場ですぐ使えるソフトウェアテストツール <http://www.jasst.jp/archives/jasst11t/pdf/s1.pdf>
- [9] 前川博志/井川将, JaSST' 13 関西 ようこそ TABOK の世界へ, 2013, <http://jasst.jp/symposium/jasst13kansai/pdf/S3-B1.pdf>
- [10] 湯本剛, JaSST' 11 九州 テスト自動化の前に押えておきたい 3 つのポイント, 2011, <http://jasst.jp/archives/jasst11k/pdf/S5.pdf>
- [11] 加藤大受, テスト自動化の見取り図, ソフトウェア・テスト PRESS, Vol.3, 2005
- [12] 町田欣史, 回帰テストの適用と実施 (デグレードへの対処はいつやるのか/どこまでやるのか), ソフトウェア・テスト PRESS, Vol.6, 2007
- [13] 日本科学技術連盟 SQiP 研究会 第 6 分科会 「派生開発」, 派生開発に XDDP を導入する際の障壁とその解消に向けたアプローチ, 2009