

テスト種類に着目した最適な自動テスト支援ツールの選定方法と実践

Selection method of optimal automatic test tools by Test type and its Execution

オムロン株式会社 オートメーション統轄事業部 ソフトウェア開発部 第2開発課
OMRON Corporation, Industrial Automation Business Company
Software Development Department, Group 2, Automation Systems Division, HQ. Development Center
田中 桂三¹
Keizo Tanaka

Abstract: In the case of GUI based test in Software version up development projects, we have to maintain automatic test programs made in the previous project in order to reuse automatic test programs. Also we have to prepare automatic test programs of regression test for defects found in the project in the short period of time. Considering the efficiency in these cases, we describe the procedure of the selection method of the optimal automatic test tools paying attention to test type and its requirements, and the effect by using the automatic test tools in this paper.

1. はじめに

ソフトウェアバージョンアップを継続する開発プロジェクトの GUI ベースのテストにおいて、過去バージョンの自動テスト資産を利用できるように、テスト開始迄に自動テスト資産をメンテナンスしておく必要がある。また、テスト実施中に発見した不具合の再現手順提示、および不具合修正後の回帰テストを実施するために、短期間で自動テストの準備（操作手順の自動化）を行う必要がある。

これらの課題を解決するために、テストの種類（テストの特徴）に着目した、最適な自動テスト支援ツールの選定方法を定義した。また実際に、弊社ソフトウェアバージョンアップ開発プロジェクトにおいて、定義した選定方法に基づいて自動テスト支援ツールを選択し、それらを活用した。その結果、上記課題を解決し、自動テストを計画通りに実施することができた。

本論文では、各テスト種類（テストの特徴）の必要な要件に着目した、最適な自動テスト支援ツールの選定方法の手順と、実運用上での工夫内容、および効果と今後の展開について紹介する。

2. 弊社開発プロジェクトでの従来の自動テストの課題と考察（自動テスト選定方法導入前）

弊社ソフトウェア開発プロジェクトでは、当初、高度な自動テストプログラミングができるものが高機能を保有しており、自動テストにとって都合がよいと考え、「高級言語を利用可能、プログラミング言語の選択肢が多い、関数が充実している。」という理由から、自動テスト支援ツールとして、UI Automation と TestComplete を利用した。これらの2つのツールは、機能面において、弊社ソフトウェア開発プロジェクトでの全てのテスト自動化を実現できる機能を保有していた。しかしながら、一部の自動テストケースにおいて以下の問題が発生したため、定められた期間内に自動テスト準備が完了できず、その結果、自動化を断念し人手によるテストに切り替えた。

テスト自動化が進まなかった原因を、自動テスト支援ツールの観点で以下のように考察した。

<UI Automation>

- ・オブジェクト指向設計のスキルが必要で他のテストの実施が難しく、特定の人に負荷が集中してしまった。

¹ 〒525-0035 滋賀県草津市西草津 2 丁目 2-1
2-2-1, Nishi-Kusatsu, Kusatsu-City, Shiga-Pref, 525-0035 JAPAN
Tel: 077-565-5225 E-mail: keizo_tanaka@omron.co.jp

- ・スクリプト変更時、都度コンパイルが必要であり、自動テスト実行までに時間がかかった。

<TestComplete>

- ・TestComplete 自身が CPU とメモリを消費するため、正確な性能測定ができなかった。
- ・.NetFramework の WPF 部品の一部で、うまくオブジェクト情報を取得できない部品があった。
- ・デバッグ時に、問題箇所の特定制約が難しく（WPF 部品の使用方法に問題があったのか、それとも TestComplete 自身の制約なのかの切り分けがつかず）、調査に時間がかかった。

また、さらに原因を深堀りするために、テスト種類に着目して課題を考察した（表 1.）。

表 1. 弊社開発プロジェクトでの従来の自動テストの課題

テスト種類	弊社開発プロジェクトでの従来の自動テストの課題		
	自動テスト支援ツール	結果	結果説明
性能測定テスト	TestComplete	×	TestComplete 自身が CPU とメモリを消費するため、正確な性能測定ができなかった。
回帰テスト（一連の基本機能操作）	UI Automation	△	オブジェクト指向設計のスキルが必要で簡単に自動テストスクリプトが作成できず、特定の人に負荷が集中。一部ケースを断念した。
不具合修正後の回帰テスト	TestComplete	×	いち早く（目安として 24H 以内に）スクリプトを作成する必要がある。しかし .NetFramework の WPF 部品でオブジェクト情報を取得できず、原因の特定ができなかったので自動テストを断念した。
ランニングテスト	UI Automation	×	ランニング中にタイミングのずれが発生し、処理が止まってしまい、自動テストを断念した。
別環境での GUI テスト	UI Automation	△	パソコンスペックの違いによるタイミングのずれ調整でスクリプト変更を行う必要があった。

△：一部のケースが期間内で自動テストできず ×：問題が大きく自動テストを実施できず

まとめると、以下の 2 点に課題があることがわかった。

- ・自動テスト支援ツールの機能面だけに着目しては自動テストの実現は難しい。逆に使い勝手や簡単さなど、機能面に現れにくい部分も考慮して選択する必要がある。
- ・テスト種類によりそれぞれ自動テストを実施する上での課題が大きく異なる。

3. 最適な自動テスト支援ツールの選定方法と実践

上述の考察の結果、自動テストを成功させるための方針として、自動テスト支援ツールを選定する際、機能面に現れにくい部分も要件として満たしているかを確認することとした。またこれらの要件を満たす最適な自動テスト支援ツールを各テスト種類でそれぞれ選定することとした。なお、客観的かつ定量的な選定を行うため、選定手順を作成した。以下に、各テスト種類で最適な自動テスト支援ツールの選定手順を説明する。

<手順>

(1) 短時間で自動テスト資産のメンテナンスと自動テストスクリプトの新規作成を行う為の、自動テスト支援ツールの要件を定義した。

- ・機能面だけではなく機能面に現れにくい部分も要件として定義するために、ソフトウェア品質の評価指標として体系的に定義されており、世間一般にも普及している「JIS X 0129-1：2003 品質特性」（以下、JIS 品質特性）を引用した。
- ・JIS 品質特性に該当する項目として、表 1 に記載した自動テストの課題を解決するための条件を自動テスト支援ツールの要件として挙げた。
- ・同様に、JIS 品質特性を着眼点として弊社開発プロジェクトでの従来の自動テスト実施を振り返り、必要であると挙げた条件についても要件として列挙した。

「GUI 部品をスクリプトから指定できる。」「スクリプト言語の選択肢が多い」
「画面のスナップショットが取得できる。」「自動テストスクリプト作成の簡単さ」 など

表 2. JIS 品質特性と自動テスト支援ツール要件

JIS 品質特性	JIS 品質特性 概説	自動テスト支援ツール 要件
機能性	目的から求められる必要な機能の実装の度合い	スクリプト言語の選択肢が多い (将来の拡張、ソースの流用)
		操作記録がログとしてとれ、編集・再生できる (急きょ調査が必要になった場合の利用)
		GUI 部品をスクリプトから指定できる
		画面のスナップショットが取得できる
信頼性	機能が正常動作し続ける度合い	操作の終了タイミングが取得できる
		長時間実行しても安定している (処理が止まらない)
		速度の変化に追従できる (タイミングがずれない)
使用性	分かりやすさ、使いやすさの度合い	コンパイルなしで実行できる
		自動テストスクリプト作成の簡単さ
		入力対象装置 (マウス、キーボード)
効率性	目的達成のために使用する資源の度合い	CPU/メモリ使用量 が少ない (テスト対象ソフトの性能・安定性に影響を与えない)
保守性	保守 (改訂) 作業に必要な努力の度合い	ツール依存度 (問題発生時の自己解決)
移植性	別環境へ移した際そのまま動作する度合い	他の PC での実行ができる (画面サイズや処理スピードが違う場合でも実行できる)
		価格 (参考情報)

(2) テスト自動化対象となる各テスト種類に必要な「自動テスト支援ツール要件」を整理した。

- ①性能測定テスト
- ②回帰テスト (一連の基本機能操作)
- ③不具合修正後の回帰テスト (再現手順の早期確立、修正者への伝達、修正後の修正確認)
- ④ランニングテスト
- ⑤別環境での GUI テスト (別パソコンや画面サイズが違う環境での入出力確認)

表 3. 各テスト種類の自動テスト支援ツールの要件

JIS 品質特性	自動テスト支援ツール 要件	テスト種類				
		① 性能	② 回帰	③ 不具合	④ ランニング	⑤ 別環境
機能性	スクリプト言語の選択肢が多い (将来の拡張、ソースの流用)	○	○	○	○	○
	操作記録がログとしてとれ、編集・再生できる (急きょ調査が必要になった場合の利用)			◎		
	GUI 部品をスクリプトから指定できる					◎
	画面のスナップショットが取得できる		◎			◎
信頼性	操作の終了タイミングが取得可能	◎	◎		○	
	長時間実行しても安定している (処理が止まらない)				◎	
	速度の変化に追従できる (タイミングがずれない)	○	◎		◎	
使用性	コンパイルなしで実行できる	○	○	◎	○	○
	自動テストスクリプト作成の簡単さ			◎		
	入力対象装置 (マウス、キーボード)	○	○	○	○	○

効率性	CPU/メモリ使用量 が少ない (テスト対象ソフトの性能・安定性に影響を与えない)	◎		◎	◎	
保守性	ツール依存度(問題発生時の自己解決)	○	○	○	○	○
移植性	他の PC での実行ができる(画面サイズや処理スピードが違う場合でも実行できる)					◎
	価格(参考情報)					

◎：自動テストに必須要件。これを満たさないと自動テストケースが実施できず回避策もない。

○：自動テスト作業効率が上がる要件 空白：特に不要な要件

(3) 「自動テスト支援ツール」候補を再選定した。

「TestComplete」「UI Automation」だけでは、各テスト種類の自動化に適応できないことから、自動テスト支援ツールの選定候補を追加した。マニュアルや製品情報から、表 3. の要件に該当する機能を概ね保有しており、かつ弊社で使用実績のある WSH、UWSC を選定候補とした。これら 4 つの自動テスト支援ツールにおいて、各要件を満たしているか否かを整理した。

- ・ WSH (WindowsScriptHost) : Microsoft Windows においてテキストファイルに記述したスクリプトを実行する、スクリプト実行環境。Microsoft Windows に標準でサポート。
- ・ UWSC : マウスやキーボードの操作を記録して、記録した操作を再現するフリーソフトウェア。
- ・ TestComplete : 米 SmartBear 社の有償自動ソフトウェアテスト支援ツール。GUI コントロール部品を指定したり、操作記録を再生可能。
- ・ UI Automation : .NetFramework に含まれる、GUI に対するイベントの起動を自動化するためのフレームワーク

表 4. 各自動テスト支援ツールの要件確認

JIS 品質特性	自動テスト支援ツール 要件	自動テスト支援ツール			
		WSH	UWSC	TestComplete	UI Automation
機能性	スクリプト言語の選択肢が多い	YES	YES	YES	YES
	操作記録がログとしてとれ、編集・再生できる	NO	YES	YES	NO
	GUI 部品をスクリプトから指定可能	NO	NO	YES	YES
	画面のスナップショットが取得できる	YES	NO	YES	YES
信頼性	操作の終了タイミングが取得可能	YES	NO	一部 YES	YES
	長時間実行しても安定している	YES	YES	一部 YES	一部 YES
	速度の変化に追従できる	YES	YES	一部 YES	YES
使用性	コンパイルなしで実行可能	YES	YES	YES	NO
	自動テストスクリプト作成が簡単	一部 YES	YES	一部 YES	NO
	入力対象装置 (マウス、キーボード)	一部 YES	YES	YES	一部 YES
効率性	CPU/メモリ使用量 が少ない	YES	YES	NO	YES
保守性	ツール依存度(問題発生時に自己解決)	YES	一部 YES	NO	NO
移植性	他の PC での実行ができる(画面サイズや処理スピードが違う場合でも実行できる)	NO	NO	YES	YES
	価格(参考情報)	YES : indows 標準	YES : フリーウェア	一部 YES (10 万円以上)	YES : .NetFramework 標準

凡例：YES：対応 一部 YES：制限あり NO：非対応

(4) (3) の各自動テスト支援ツールが (2) の必須要件 (◎) を満たしているかを確認し、すべての要件を満たしている自動テスト支援ツールのみを選択候補とした。(YES のみを選択。一部 YES

は選択対象外。)

- (5) (4)で必須要件を満たしている自動支援ツールにおいて、(2)の各テスト種類に必要な要件が、(3)の各自動テスト支援ツールでどれだけ満たしているかを定量的に評価。テスト種類毎に各自動テスト支援ツールを採点し、最も点数の高いものを採用した。採点方法は以下の通りである。

【採点方法】表 3. の要件の点数(A) × 表 4. の対応度合いの点数(B)

(A) ○ : 1 点 ※◎は(4)で選択候補となった時点で満たしているため、採点不要。

(B) YES:1 点 一部 YES(1) : 0.5 点 NO:0 点

例としてテスト種類が「性能テスト」での各自動テスト支援ツールの採点結果を表 5. に示す。

(4)で(2)の必須要件(◎)を満たしている自動テスト支援ツールは WSH と UIAutomation のみとなった。次に(5)での採点結果は WSH:5 点、UI Automation は 2.5 点。よって WSH を採用した。

表 5. 「性能テスト」での各自動テスト支援ツールの要件対応確認と採点結果

【性能テスト】 自動テスト支援ツール評価・ 採点	表3. の要件 (A) テスト種類に必 要な要件	表4. の要件対応度合い (B)							
		WSH (Windows Script Host)		UWSC		TestComplete		UIAutomation	
		評価	採点	評価	採点	評価	採点	評価	採点
スクリプト言語の選択肢が多い (将来 の拡張、ソースの流用)	○	YES (1)	1	YES (1)		YES (1)		YES (1)	1
操作記録がログとしてとれ、編集・再 生できる (急きょ調査が必要になった 場合の利用、スクリプト作成支援)		NO (0)		YES (1)		YES (1)		NO (0)	
GUI部品をスクリプトから指定できる		NO (0)		NO (0)		YES (1)		YES (1)	
画面のスナップショットが取得できる		YES (1)		NO (0)		YES (1)		YES (1)	
操作の終了タイミングが取得できる	◎	YES (1)	◎	NO (0)	×	一部 YES (0.5)	×	YES (1)	◎
長時間実行しても安定している (処理 が止まらない)		YES (1)		YES (1)		一部YES (0.5)		一部YES (0.5)	
速度の変化に追従できる (タイミング がずれない)	○	YES (1)	1	YES (1)		一部YES (0.5)		YES (1)	1
コンパイルなしで実行できる	○	YES (1)	1	YES (1)		YES (1)		NO (0)	0
自動テストスクリプト作成の簡単さ		一部 YES (0.5)		YES (1)		一部YES (0.5)		NO (0)	
入力対象装置 (マウス、キーボード)	○	YES (1)	1	YES (1)		YES (1)		一部YES (0.5)	0.5
CPU/メモリ使用量 が少ない (テスト 対象ソフトの性能・安定性に影響 を与えない)	◎	YES (1)	◎	YES (1)	◎	NO (0)	×	YES (1)	◎
ツール依存度 (問題発生時の自己解決)	○	YES (1)	1	一部 YES (0.5)		NO (0)		NO (0)	0
他のPCでの実行ができる (画面サイズや 処理スピードが違う場合でも実行でき る)		NO (0)		NO (0)		YES (1)		YES (1)	
価格 (参考情報)		YES : Windows標準		YES : フリー ウェア		一部YES (10万 円以上)		YES	
採点結果 (合計)			5		×		×		2.5

上述の手順を実施した結果、各テスト種類の採点結果は以下の通りとなった。各テスト種類で最も点数の高い1位のツールを最適な自動テスト支援ツールとして選定した。

表 6. 各テスト種類に最適な「自動テスト支援ツール」の選定

自動テスト支援ツール テスト種類	1 位 (選定)	2 位	3 位	4 位
性能測定テスト	WSH (5 点)	UI Automation (2.5 点)	UWSC (対象外)	TestComplete (対象外)
回帰テスト (一連の基本機能 操作)	WSH (4 点)	UI Automation (1.5 点)	TestComplete (対象外)	UWSC (対象外)
不具合修正後の回帰テスト	UWSC (2.5 点)	WSH (対象外)	TestComplete (対象外)	UI Automation (対象外)
ランニングテスト	WSH (5 点)	UWSC (3.5 点)	UI Automation (対象外)	TestComplete (対象外)
別環境での GUI テスト	TestComplete (3 点)	UI Automation (1.5 点)	WSH (対象外)	UWSC (対象外)

4. 選定した自動テスト支援ツールでの実運用結果、および残課題の解決策

計画された開発期間内でテストを実施完了するために、図 1. のとおり自動テスト標準計画を設定し、計画達成のための与件を明確にした。

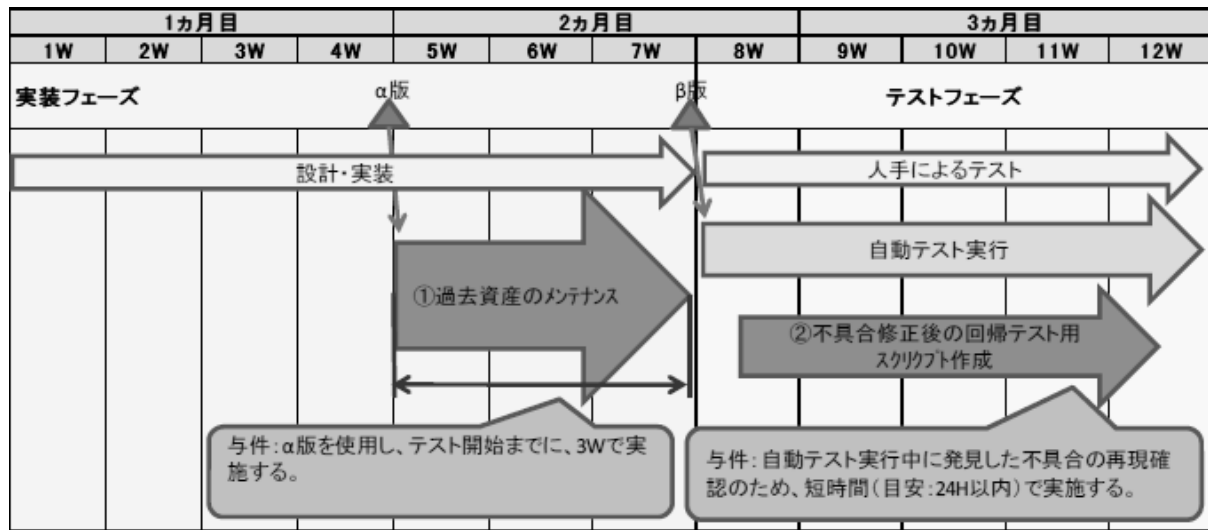


図 1. 弊社ソフトウェア開発プロジェクト 自動テスト標準計画 ※3 カ月開発(12W) の場合

実際に弊社ソフトウェア開発プロジェクトで、各テスト種類に選定した「自動テスト支援ツール」を用いて、テストの自動化を実施した。その結果、従来の自動テストの課題は解決し、いずれのテスト種類においても、ほとんどの選定ポイントにおいて効果が得られた。このことから、自動テスト支援ツールの選定の際、「機能面ではなく機能面に現れにくい部分も考慮して、自動テストに必要な要件を列挙し、最適なツールを選定する。」「テスト種類により最適なツールを使い分ける。」という方針の効果が出たと言える。

ただし、一部課題が残ったが、それぞれ以下の解決策を施した。実施結果は表 7 にまとめる。

【課題 1】連続操作において、前操作から次操作への移行に失敗する場合があった。

- ・前操作の終了時に終了ダイアログが出ない場合、操作の終了タイミングが取得できない。
- ・データのサイズによって、タイミングがずれてしまい、途中で自動テスト処理が止まってしまう。

【解決策 1】前操作から次操作への移行の間に、別機能（※）を定期的呼び出し、起動できたら前操作が終了したと判断し、次操作を開始するようにした。

（※）操作に関係がなくかつ CPU 使用率が低い機能。例：ツール | オプション画面の起動など

【課題 2】性能測定テストにおいて、プルダウンメニューの表示速度（※）や、キーボードを押し続けている時間などを計測することが、WSH の標準機能では実現できない。

（※）例：a を入力して、a が先頭の文字列一覧を選択候補としてプルダウンメニューに一覧表示する

【解決策 2】WSH から、WindowsAPI を利用できるよう、ExcelVBA を呼び出した。

プログラムの流れとして、WSH(VBA)→ExcelVBA→WindowsAPI の順にコールした。

この構成により、特別な開発環境がなくても Windows の設定を操作し上記計測を行うことができた。

表 7. 各テスト種類に最適な「自動テスト支援ツール」の選定後の実運用結果

テスト種類	適用箇所	自動テスト支援ツール	選定ポイント（自動テストに必要な要件）	標準計画を進める上での効果
性能測定テスト	①	WSH	操作の終了タイミング取得ができる	一部【課題 1】【課題 2】が残ったが、【解決策 1】【解決策 2】により解決。

			CPU/メモリ使用量 が少ない	有り (CPU/メモリ使用量負荷調整によるメンテナンス工数はかからなかった。)
回帰テスト (一連の基本機能操作)	①	WSH	画面のスナップショットが取得できる	有り (テスト結果の確認工数が減った。)
			操作の終了タイミング取得ができる	【一部【課題 1】が残ったが、【解決策 1】により解決。
			速度の変化に追従できる (タイミングがずれない)	
不具合修正後の回帰テスト	②	UWSC	自動テストスクリプト作成の簡単さ	有り ・簡単操作で記録・再生が可能。すぐにスクリプトが生成でき、24 H以内に実施することができた。 ・安定した動作が可能で再テスト回数が軽減できた。
			コンパイルなしで実行できる	
			操作記録がログとしてとれ、編集・再生できる	
			CPU/メモリ使用量が少ない	
ランニングテスト	①	WSH	長時間実行しても安定している (処理が止まらない)	有り (ランニング停止がなく、余計なメンテナンス工数はかからなかった。)
			速度の変化に追従できる (タイミングがずれない)	【一部【課題 1】が残ったが、【解決策 1】により解決。
			CPU/メモリ使用量 が少ない	有り (CPU/メモリ使用量負荷調整による、メンテナンス工数はかからなかった。)
別環境での GUI テスト	①	TestComplete	GUI 部品をスクリプトから指定できる	有り (GUI 部品を指定することで、スクリプトの変更なしで、他の PC へ動作可能となった)
			画面のスナップショットが取得できる	
			他の PC での実行ができる (画面サイズが違って実行可)	

① テスト開始までに、3W でテストスクリプトを作成する。

② 自動テスト実行中に、短時間 (目安: 24H 以内) で実施する。

5. 今後の展開

最適な自動テスト支援ツールの選定により、4 章に述べたとおり、自動テストの導入目的である、決められた期間内での自動テスト資産のメンテナンスと新規操作の記録用の自動テストスクリプトを期間内で作成することができた。

一方、本取り組みにより全体テスト工数の削減にもつながったものの、削減工数はわずか 1 人月と効果は少ない (表 8. の (C) の②)。

一般にテスト自動化の成功目安は 30%といわれている。今後の展開として、テスト自動化 30%を元に設定した表 8. の (D) を目指して、自動テストによる削減工数の範囲を 19 人月分 (⑦) の約 30%である 6 人月分 (⑧) に拡大していく。そのために、以下の 2 点を重点に進めていく。

(1) 各テスト種類で、削減工数目標の設定

各テスト種類で、テスト削減工数目標と自動テストの拡大方針を設定し、それに基づき自動テスト範囲を拡大する。(表 9. 参照)。

(2) テストケースの見直し

もともとの手動のテストケースをそのまま自動化すると、冗長な操作がある場合や、複数の操作を実現する際、次の状態への移行が難しい場合がある。そのために、そのまま自動化するとスクリプトが長くなり複雑になる。今後の展開として、自動化しやすいように以下のように手順の変更やテストケースの見直しを行う。

- ・複数操作のテストケースの順番を変えて、操作を単純化する。

例：余計な画面のオープン・クローズ回数を減らす。

- ・テストケースの期待結果を、自動テストで確認しやすいテスト結果に変える。

例：複数設定データの確認を、一つずつ設定をクリックして確認するのではなく、期待結果の画面をあらかじめ用意しておき画面単位で比較する。

表 8. 全体テスト工数の推移

工数 \ 効果の推移	(A) 自動テスト導入前 ※効果判断の基準値	(B) 自動テスト導入 当初 ※テスト自動化5%	(C) 本取り組み 実施後 ※テスト自動化15%	(D) 今後の展開 ※テスト自動化30%
①全体テスト工数 (③+⑥)	27	30	26	22
②自動テスト導入による効果 (削減工数) 各列の① - (A) の① ※マイナス値が効果あり	-	3	-1	-5
③全体テスト準備工数 (④+⑤)	8	12	10	9
④手動テスト準備工数	8	7.5	7	6
⑤自動テスト準備工数	0	4.5	3	3
⑥手動テスト実施工数 (⑦-⑧)	19	18	16	13
⑦手動テストで実施した場合の 全テスト工数	19	19	19	19
⑧自動テストによる効果 (削減工数) ※マイナス値が効果あり	0	-1	-3	-6

※同規模のテストを実施した場合の概算値 ※単位：人月

表 9. 従来の自動テストと、最適な自動テスト選定後の効果比較

テスト種類	従来の自動テストの結果			取り組み結果			今後の展開	
	自動テスト支援ツール	自動化範囲	削減工数 (人月)	自動テスト支援ツール	自動化範囲	削減工数 (人月)	今後の展開	削減工数 (人月)
性能測定テスト	TestComplete	なし	-	WSH	単機能の性能計測	0.5	複数機能の性能計測にも拡大	1
回帰テスト (一連の基本機能操作)	UI Automation	ファイル オープン クローズ 画面オープン クローズ	0.75	WSH	左記+データ転送	1	GUIテスト (左記以外機能のメニュー起動、画面上の操作) にも拡大	2
不具合修正後の回帰テスト	TestComplete	なし	-	UWSC	GUIベースの回帰テスト	0.25	生成データの確認にも拡大	0.5
ランニングテスト	UI Automation	なし	-	WSH	メモリ使用量推移のテスト	0.25	別環境でのランニングテストにも拡大	0.5
別環境でのGUIテスト	UI Automation		0.25	TestComplete	別環境でのGUIテスト	1	別環境での性能テストにも拡大	2
合計 (人月)			1	合計 (人月)			合計 (人月)	6

参考文献

- [1] ソフトウェア製品の品質 第1部 品質モデル、JIS X 0129-1 : 2003 品質特性、JIS 規格
- [2] ソフトウェア品質評価ガイドブック、日本規格協会
- [3] ソフトウェア品質保証の考え方と実践、日科技連
- [4] 増田 隆、UWSC を用いた自動回帰テストの評価、JaSST' 11 Tokyo、2011