

# 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

2012/9/14

オムロン株式会社

田中桂三

# 自己紹介

## ■ 役割

- 産業用電子機器プログラミングツール（Windows ソフトウェア）開発のテストリーダー

## ■ 最近の注力活動

- テスト設計プロセス改善  
信頼性分析、グレーボックステスト
- 自動テスト普及・展開

## ■ 社外活動

- JaSST2012Tokyoにて論文発表  
“信頼性分析による品質確保の取り組み”
- 日科技連SPC研究会メンバ(2004年)  
“「リスク管理」の現状と分析”

# 目次

- はじめに
- 弊社開発プロジェクトでの従来の自動テストの課題
- 最適な自動テスト支援ツールの選定方法と実践
- 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
- 今後の展開
- **APPENDIX**

# はじめに

## テスト自動化対象：

- ・産業用電子機器向けWindowsプログラミングツールのバージョンアップを継続する開発プロジェクト
- ・自動化対象：GUIベースのテストで以下を自動化
  - ・過去バージョンの自動テスト資産をテスト開始迄にメンテナンス
  - ・不具合の再現手順提示および修正後の回帰テスト用に、短期間で操作手順を自動化

課題：自動テストの準備に時間がかかる。

⇒テストが遅れ、開発プロジェクトが計画通りに完了しない



課題解決のために、



取り組み：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

⇒テストを計画通りに完了、開発プロジェクト遅れを防止できた。



本論文では、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します。

# 弊社開発プロジェクトでの 従来の自動テストの課題

# 自動テスト導入当初の課題（ツール）

自動テスト支援ツールの条件として

「どんな処理でも自動化できる⇒高機能、高度なプログラミング可能」と考え、「UIAutomation」と「TestComplete」を選定。

これらのツールで、自動テストを実施した結果、

## UI Automation

オブジェクト指向設計の知識が必要  
⇒特定の人に負荷が集中

スクリプト変更時、都度コンパイルが必要  
⇒実行までに時間がかかる

## TestComplete

CPUとメモリを消費する⇒正確な性能測定ができない

デバッグ時、問題箇所の特定に時間がかかる

- Windows部品の指定方法が問題？
- TestComplete自身の制約？

定められた期間内に自動テスト準備が完了できない。  
⇒テストが遅れ、開発プロジェクトが計画通りに完了しない。

高機能なツールを使ったのに、なぜ失敗した？





# 自動テスト導入当初の課題考察

失敗原因を深掘りするために、各テスト種類の結果から課題を考察（表1）

## わかったこと

テスト種類により、自動テストの課題が異なる。

機能として現れにくい部分が原因の問題が発生した。

## 成功の力ギは？

テスト種類により最適な自動テスト支援ツールを選択すべき！

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が必要！

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ない。特定の人に負荷が集中。一部ケースを断念。
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ため自動テストを断念。
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った。自動テストを断念。
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

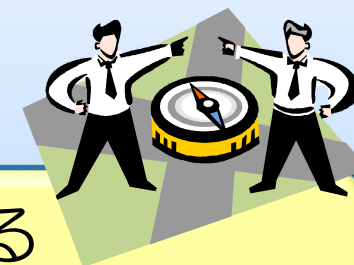


# 最適な自動テスト支援ツールの 選定方法と実践



# 自動テスト成功のための解決方針

## 解決方針



以下に着目して、自動テスト支援ツールを選択する

- ・各テスト種類に最適な自動テスト支援ツールを選択
- ・機能面に現れにくい部分も含めて、要件として満たしているかを確認

## アクション

「自動テスト支援ツールの選定手順」を作成し、  
それに基づいて選定。



＜目的＞

- ・最適な自動テスト支援ツールの選定を、「誤りなく」、かつ「客観的」「定量的」に行うため。
- ・ノウハウ化して、他の開発プロジェクトにも展開するため。

次頁以降に、選定手順を紹介します。

# 最適な自動テスト支援ツールの選定手順の流れ

以下の手順に沿って、最適な自動テスト支援ツールを選定。

手順 1

自動テスト支援ツールの要件を定義

手順 2

各テスト種類に着目し、必要な自動テスト支援ツールの要件が何か？を表に整理

手順 3

各自動テスト支援ツールに着目し、各要件を満たしているか？を表に整理

手順2と手順3で整理した表を照らし合わせて、

手順 4

各テスト種類で、必須要件を満たす「自動テスト支援ツール」を判断

手順 5

手順4で選定した自動テスト支援ツールを採点。  
各テスト種類で最も点数が高いものを選定。

次ページ以降に各手順について紹介します。

# 最適な自動テスト支援ツールの選定手順

## 手順 1

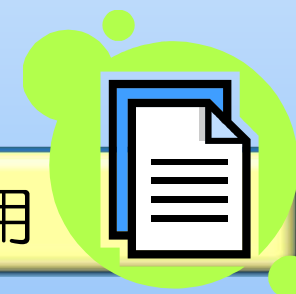
### 自動テスト支援ツールの要件を定義

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したい。

そこで、



「JIS X 0129-1 : 2003 品質特性 (※)」を引用



引用した理由：

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。

⇒観点が洗い出せ、機能面に現れにくい部分も定義できると判断。

(※) 以降便宜上「JIS品質特性」と記載。

APPENDIX 2「表2 JIS品質特性と自動テスト支援ツール要件」参照。

# 最適な自動テスト支援ツールの選定手順

## 手順 2

各テスト種類に着目し、必要な自動テスト支援ツールの要件が何か？を整理

対象となるテスト種類：

- 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

APPENDIX 3:「表3 各テスト種類の自動テスト支援ツールの要件」参照。



# 最適な自動テスト支援ツールの選定手順

## 手順3

各自動テスト支援ツールに着目し、各要件を満たしているか？を整理（表4）

表4：各自動テスト支援ツールの要件確認 YES：対応 一部YES：制限あり NO：非対応

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（将来の拡張、ソースの流用）	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っても実行できる）	NO	NO	YES	YES
	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：.Net Framework標準

※最適なツールが選べるよう、新たに「WSH」と、「UWSC」を選定候補とした。

＜選定候補とした理由＞

- ・手順2（表3.）の要件に該当する条件を機能的に概ね保有している。
- ・弊社で使用実績があり、使い慣れている。（機能に現れにくい要件も、概ね対応できている。）

APPENDIX 1: 各ツールの紹介については「自動テスト支援ツール選定候補の紹介」参照

# 最適な自動テスト支援ツールの選定手順

## 手順4

## 必須要件を満たす「自動テスト支援ツール」を判断

テスト種類毎に実施（表3の①～⑤）。表3と表4を照らし合わせることで、必要要件を満たす「自動テスト支援ツール」がどれかを判断。

方法：1）要件毎（行毎）に表3と表4を照らし合わせる。

2）表3の必須要件（＝◎）の行について、表4で満たしている（＝YES）となっているツールが必須要件を満たすと判断。

3）1）2）を行毎に実施。

4）テスト種類の◎がすべてYESになっているツールを選択候補とする！

表3. 各テスト種類の自動テスト支援ツールの要件

表4. 自動テスト支援ツール毎の要件確認

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 操作性	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（急ぎよ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	○	○	○	○	○
効率性	入力対象装置（マウス、キーボード）	○	○	○	○	○
	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）	○	○	○	○	○

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（将来の拡張、ソースの流用）	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
効率性	入力対象装置（マウス、キーボード）	一部YES	YES	YES	一部YES
	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
	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）	NO	NO	YES	YES
移植性	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：.Net Framework標準

1）要件毎（行毎）に照らし合わせ

2）表3の必須要件（＝◎）を満たしているツール（＝YES）を表4. で確認

3）1）2）を行毎に実施。

4）必須要件（＝◎）をすべて満たしているツール（＝すべてYES）が選択候補



# 最適な自動テスト支援ツールの選定手順

## 手順5

自動テスト支援ツールを採点。最も点数が高いものを選定。

- ・手順4. で必須要件を満たしている自動支援ツールを採点対象とする。
- ・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】表3.の要件の点数(A) と 表4.の対応度合いの点数(B) の積。  
(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：0.5点 NO：0点

表6. 各テスト種類での各自動テスト支援ツールの採点結果

自動テスト支援ツール	1位 (選定)	2位	3位	4位
テスト種類				
性能測定テスト	WSH (5点)	UI Automation (2.5点)	UWSC (対象外)	TestComplete (対象外)
回歸テスト (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)
不具合修正後の回歸テスト	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)

採点結果

各テスト種類で、1位のツールを選定。  
(表6)

APPENDIX5: 各テスト種類の選定結果は「表5－＊」参照。

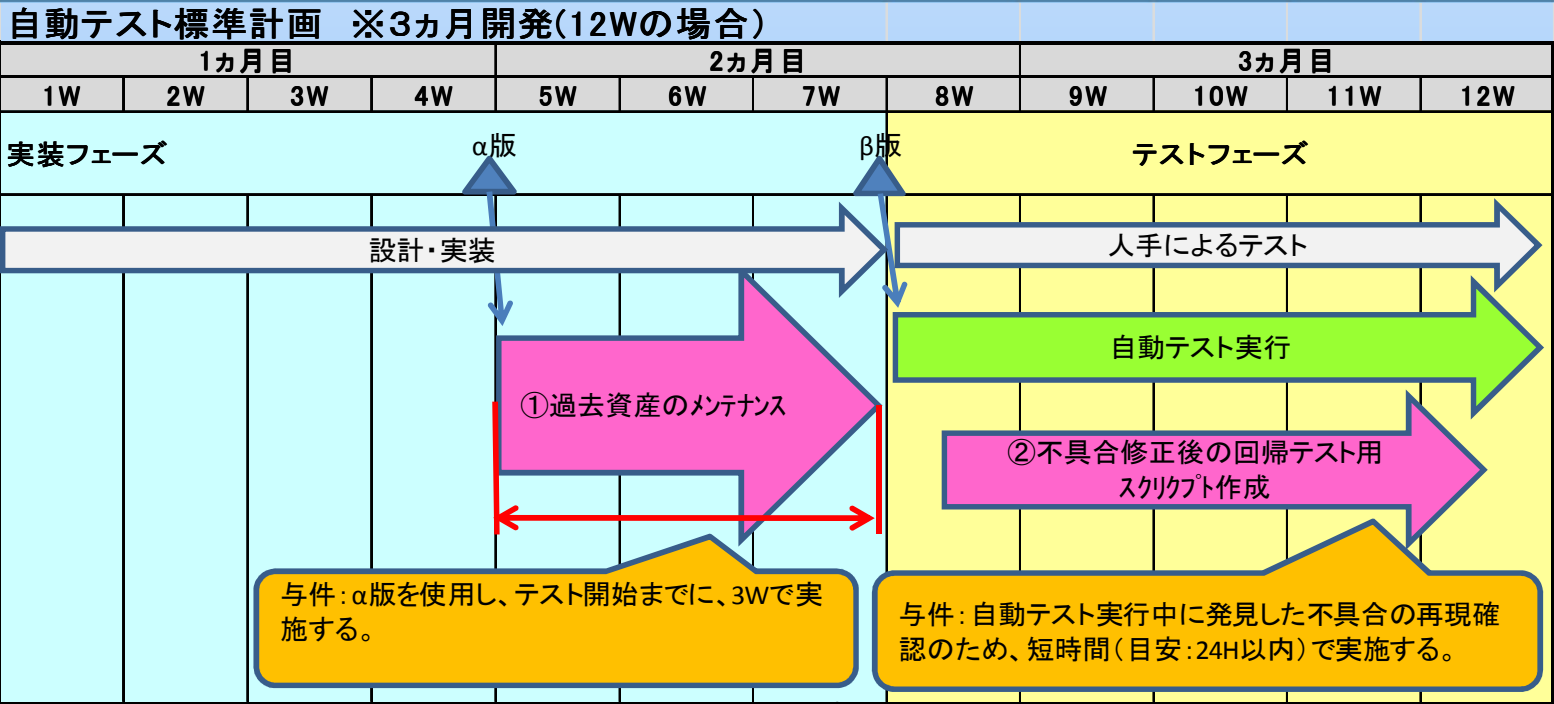
# 選定した自動テスト支援ツールでの 実運用結果、 および残課題の解決策

# 自動テスト標準計画の設定

From: 自動テストは、あまりプロジェクトとして進捗管理されていない。  
 To: **自動テスト標準計画**を設定。開発マスタ計画に含めた。

- 開発プロジェクトの中での**実施時期**を明確化。（テスト自動化準備遅れの防止、進捗管理）
- テスト自動化計画達成のための**与件**を明確化。（目的意識醸成）

**本計画をベースに**、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、**テストの自動化準備**を実施した。



# 選定した自動テスト支援ツールでの 実運用結果

- ・従来の自動テストの課題はクリア。  
いずれのテスト種類においても、選定ポイントにおいて効果が得られた。
- ・一部課題が残ったがそれぞれ解決策を施した。詳細は次頁。

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	標準計画を進める上での効果
性能測定テスト	① 3W以内でメンテ	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。）
regres テスト（一連の基本機能操作）	① 3W以内でメンテ	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 regres テスト	② 24H以内で新規作成	UWSC	操作記録がログとしてとれ、編集・再生できる	有り ・簡単操作で記録・再生が可能。24H以内にスクリプトが作成できた。 ・安定した動作が可能で再テスト回数が軽減できた。
			自動テストスクリプト作成の簡単さ	
			コンパイルなしで実行できる	
			CPU/メモリ使用量 が少ない	
ランニングテスト	① 3W以内でメンテ	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 3W以内でメンテ	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）

その結果、

自動テスト標準計画どおり、テストの自動化を実施できた。



# 残課題の解決策

2つの残課題に対し、それぞれ解決策を施した。  
⇒課題をクリアし、自動テスト標準計画どおり実施できた。

【課題1】連続操作で、**前操作から次操作への移行に失敗**する場合あり。  
・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、タイミングがずれしまい、途中で自動テスト処理が止まってしまう。



そこで、

【解決策1】  
前操作から次操作への移行の間に、**別機能（※）を定期的に呼び出す**。  
（起動できたら前操作が終了したと判断）  
※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

【課題2】性能測定テストにおいて、**WSHの標準機能では実現できない**場合あり。  
・プルダウンメニューの表示速度（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）  
・キーボードを押し続けている時間を計測



そこで、

【解決策2】WSH機能を補完するため、**WindowsAPI**を利用。  
・実現のため、プログラムの流れとして、WSH→ExcelVBA→WindowsAPIの順にコールした。  
⇒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。

# 本取り組みの成果と 今後の展開





# 本取り組みの成果（まとめ）

本取り組みにより、自動テストの課題がクリアできた

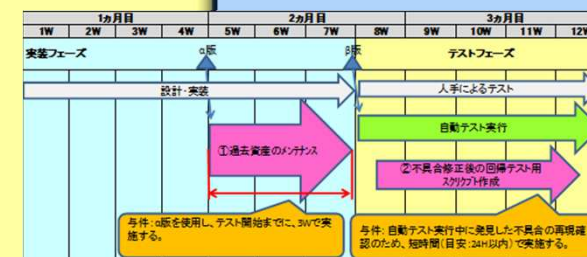
選定手順に基づき、各テスト種類で最適な自動テスト支援ツールを利用



自動テスト標準計画を設定、自動テスト準備を実施。

- ①過去資産のメンテナンス（3W以内で完了）  
⇒自動テスト準備中の問題が減り、スクリプトメンテナンス期間を3W以内で完了できた。
- ②回帰テスト用スクリプト作成：24H以内で完了  
⇒スクリプト作成が素早く実施できた。

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



自動テスト標準計画通りに自動テスト準備を完了！  
⇒テスト遅れによる開発プロジェクト遅れを防止

# 今後の展開



本取り組み後の、次のステップとして、

今後の展開

テスト自動化率の拡大、**テスト工数削減**

**効果を実感！**

目標値

**テスト自動化率：30%へ拡大（※1）**

（テスト実施削減工数換算：6人月）

テスト実施工数19人月  
分の約30%=6人月分

**これにより、全体テスト工数：5人月削減する！（※2）**



自動化による削減工数

自動テスト準備工数

目標達成の  
注力ポイント

※1: 一般的にテスト自動化率において、目標水準は「30%」と言われる。  
テスト全体の30%程度は、自動化が比較的容易で、かつ工数削減効果の大きいテストケース、つまり「**自動化による削減工数>自動テスト準備工数**」が成立するものである。

※2: 全体テスト工数削減（5人月） ← 表8(D)列の②  
= 自動テストによるテスト実施削減工数（6人月） ← テスト実施工数19人月分(表8⑦)の約30%=6人月分(D)列の⑧)に拡大  
- 自動テストによる全体テスト準備工数の増加分（1人月） ← 表8(D)列の③- (A)列の③  
表8.全体テスト工数の推移

(1) 各テスト種類で削減工数目標の設定（削減工数6人月の内訳を明確化）

(2) テスト自動化準備の効率化（テスト準備工数の増加分を+1人月に抑制）

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

# テスト工数削減のためのアクション

## (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今後の展開を設定。（表9）**  
それに基づき、自動テスト範囲を拡大。



表9. 従来の自動テスト結果と、今後の展開

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	1.25	
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	1.5	
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	生成データの妥当性確認にも拡大	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	別環境での回帰テスト、性能テストにも拡大	2.25	
合計（人月）			1	合計（人月）			3	合計（人月）	6



# 自動テスト範囲拡大のために、

各テスト種類での自動テスト範囲拡大に向け、**新たな対策案を設定。**  
解決に向け、WSH, UWSC, TestCompleteのスク립トをメンテナンス。

表10. 今後の展開に向けた新たな対策案

テスト種類	今回採用した自動テスト支援ツール	①今回の自動テスト支援ツール評価	②今後の展開（さらなる工数削減）に向けた新対策
性能測定テスト	WSH	以下の面で効果あり。 ・ <b>操作の終了タイミング取得ができる</b> ・ <b>CPU/メモリ使用量 が少ない</b>	「【課題1】連続操作で、前操作から次操作への移行に失敗」が、複数機能での性能測定で発生しないよう、【解決策1】以外の別の解決策を用意する。 例) 前操作終了で起動するプロセス(exe)をWatch (プロセスがあれば、前操作が終了したと判断)
回帰テスト（一連の基本機能操作）	WSH	以下の面で効果あり。 ・ <b>画面のスナップショットが取得できる</b> ・ <b>CPU/メモリ使用量 が少ない</b>	GUIテスト（メニュー起動、画面上の操作）への拡大に向けて、【解決策2】がどのメニュー、画面上の操作にも適用できるよう、汎用化。
不具合修正後の回帰テスト	UWSC	以下の面で効果あり。 ・ 操作記録がログとしてとれ、編集・再生できる。 <b>簡単な操作で24H以内に実現可能。</b>	生成データの妥当性確認自動化。正しいデータとの比較のため、DIFFツールを呼出し、結果の自動判断。
ランニングテスト	WSH	以下の面で効果あり。 ・ <b>ランニング停止がなく</b> 、余計なメンテナンス工数はかからなかった。	別環境でのランニングテストにも拡大できるよう、GUI部品を指定。(TestCompleteとの併用)
別環境でのGUIテスト	TestComplete	<b>GUI部品を指定することで</b> 、スク립トの変更なしで、 <b>他のPCへ動作可能</b> となった	別環境での性能テストにも拡大できるよう、 <b>上述の性能テストの対策を実施。</b> (WSHとの併用)

※今回実施済み【解決策1】前操作から後操作への移行の間に、別機能を定期的に呼び出す。

※今回実施済み【解決策2】WSH機能を補完するため、WindowsAPIを利用。

さらなる拡大  
に向けて

並行して、新たなツールの選定を検討。

- ・ 新課題の解決につながる要件を、表2. 表3. に追加
- ・ 他の自動テスト支援ツールを、本手順で再選定。



# テスト工数削減のためのアクション

## (2) テスト自動化準備の効率化

手動のテストケースには、**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、

テスト自動化準備の効率化に向け、**自動化しやすいように、「ケースの見直し」や「手順の変更」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。



以上で発表を終わります。  
ご清聴、ありがとうございました！



# APPENDIX



## APPENDIX I

# 自動テスト支援ツール選定候補の紹介

- 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。  
CPU/メモリ使用量が少ない。ツールへの依存性が少ない。GUIコントロール指定不可。
- 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。  
操作記録・再生が可能。4ツールのうち最も簡単に速く自動化できる。GUIコントロール指定不可。
- 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。  
GUIコントロールの直接指定が可能。多機能。CPU/メモリ使用量が多い。有料。
- 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  
GUIコントロール指定が可能、多機能。コンパイルが必要。

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「表4：各自動テスト支援ツールの要件確認」に整理。



## APPENDIX 2

# 表2：JIS品質特性と自動テスト支援ツール要件

## JIS品質特性毎に自動テスト支援ツールの必要な要件を列挙

要件の抽出観点

- ・表1に挙げた自動テストの課題を解決する条件を、各JIS品質特性の要件として列挙
- ・JIS品質特性を着眼として自動テスト実施を振り返った結果、「必要」と挙げたもの

結果：カタログに掲載されておらず、機能面に現れにくい部分も要件に定義できた。

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)



APPENDIX 3

表3：各テスト種類の自動テスト支援ツール要件

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（急ぎ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上がる要件      空白：特に不要な要件





## APPENDIX 4

# 手順4の詳細説明

## 「①性能テスト」を例に、手順4の詳細説明を行う。

方法：1) 要件毎（行毎）に表3と表4を照らし合わせる。

2) 表3の必須要件（＝◎）の行について、表4で満たしている（＝YES）となっているツールを確認する。

**要件「操作の終了タイミングが取得可能」が“◎” ⇒それに対し“YES”である、「WSH」、「UI Automation」が 該当**

3) 1) 2) を行毎に実施。

**要件「CPU/メモリ使用量が少ない」が“◎” ⇒それに対し“YES”である、「WSH」、「UWSC」、「UI Automation」が 該当**

4) テスト種類の◎がすべてYESになっているツールを選択候補とする！

**結果：必須要件◎のすべてを満たすのは「WSH」、「UI Automation」**



JIS品質特性	自動テスト支援ツール 要件	テスト種類	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			
		①性能			WSH	UWSC	TestComplete	UI Automation
機能性	スクリプト言語の選択肢が多い	○	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			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			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			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	◎	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			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	○		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	○	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			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）	○		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	◎	保守性	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）	○		ツール依存度（問題発生時に自己解決）	YES	一部YES	NO	NO
	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）			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
移植性	価格（参考情報）		移植性	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：.Net Framework標準

## APPENDIX 5-1

# 表5-1：「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

- ・手順2. (表3) 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。
- ・次に手順5. での採点結果はWSH:5点、UI Automationは2.5点。  
⇒よって「性能テスト」では、「WSH」を採用。

【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)		表4.の要件対応度合い(B)							
	テスト種類に必要な要件	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(急ぎ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	





# APPENDIX 5-2

## 表5-2：「回帰テスト（基本操作）」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

- ・手順2.（表3）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。
- ・次に手順5. での採点結果はWSH:4点、UI Automationは1.5点。  
⇒よって「回帰テスト（基本操作）」では「WSH」を採用。

【回帰テスト（基本操作）】 自動テスト支援ツール評価・ 採点	表3.の要件(A) 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)	
速度的変化に追従できる（タイミングがずれない）	◎	YES(1)	◎	YES(1)	◎	一部 YES(0.5)	×	YES(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(1)(1 0万円以 上)		YES(1)	
採点結果（合計）			4		×		×		1.5

## APPENDIX 5-3

# 表5-3：「回帰テスト（不具合修正確認）」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

手順2.（表3）の必須要件（◎）を満たしている自動テスト支援ツールはUWSCのみとなった。

⇒よって「回帰テスト（不具合修正確認）」では、UWSCを採用。

【回帰テスト（不具合修正確認）】 自動テスト支援ツール評価・採点	表3.の要件(A) 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)		YES(1)	1	YES(1)	2	YES(1)	
・操作記録がログとしてとれ、編集・再生できる（急ぎ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)	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)		YES(1)		一部YES(0.5)		YES(1)	
コンパイルなしで実行できる	◎	YES(1)	◎	YES(1)	◎	YES(1)		NO(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)		YES(1)	1	YES(1)		一部YES(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)		一部YES(0.5)	0.5	NO(0)		NO(0)	
他のPCでの実行ができる（画面サイズが違っても実行できる）		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(1)(10万円以上)		YES(1)	
採点結果（合計）			×		2.5		×		×

**APPENDIX 5-4**

# 表5-4：「ランニングテスト」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

- ・手順2.（表3）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UWSCのみとなった。
- ・次に手順5. での採点結果はWSH:5点、UWSCは3.5点。  
⇒よって「ランニングテスト」では、WSHを採用。

【ランニングテスト】  自動テスト支援ツール評価・ 採点	表3.の要件(A)		表4.の要件対応度合い(B)							
	必要な要件	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)	1	YES(1)		YES(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)	1	NO(0)	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)	◎	YES(1)	◎	一部YES(0.5)	×	YES(1)	◎	
コンパイルなしで実行できる	○	YES(1)	1	YES(1)	1	YES(1)		NO(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)	1	YES(1)		一部YES(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)	0.5	NO(0)		NO(0)		
他のPCでの実行ができる(画面サイズが 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(1)(10万円以上)		YES(1)		
採点結果（合計）			5		3.5		×		×	

## APPENDIX 5-5

# 表5-5：「別環境でのGUI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

- ・手順2. (表3) の必須要件 (◎) を満たしている自動テスト支援ツールは TestComplete と UI Automation のみとなった。
- ・次に手順5. での採点結果は TestComplete:3点、UWSCは1.5点。  
⇒よって「別環境でのGUIテスト」では、TestCompleteを採用。

【別環境でのGUIテスト】 自動テスト支援ツール評価・ 採点	表3.の要件(A)		表4.の要件対応度合い(B)							
	必要な要件	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)		YES(1)	2	YES(1)	1	YES(1)	1	
・操作記録がログとしてとれ、編集・再生できる(急ぎ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)		YES(1)		一部YES(0.5)		YES(1)		
コンパイルなしで実行できる	○	YES(1)	1	YES(1)	1	YES(1)	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)	1	YES(1)	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)	0.5	NO(0)	0	NO(0)	0	
他のPCでの実行ができる(画面サイズ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(1)(10万円以上)		YES(1)		
採点結果(合計)			×		×		3		1.5	

