

チームで継続的に探索的テストの効果を出す工夫

Small Actions and Continuous Improvement of Exploratory Testing

東京エレクトロン株式会社 システム開発センター 次世代 APP 開発グループ
TOKYO ELECTRON LIMITED System Development Center Advanced Application Development Group

○中嶋 信 柴田 威¹⁾ 高木 進也²⁾
○Makoto Nakakuki Takeru Shibata¹⁾ Shinya Takagi²⁾

Abstract Development of embedded software for manufacturing equipment requires a long development period from the initial development of equipment to maintenance, during which function addition and function modification are necessary. Especially improvements are frequently made in new products and software is released on a regular basis.

In the pre-release test, we have done two different tests, which specification based test and a defect detection test. The specification base test is tests to confirm that the created features behave according to specifications. The defect detection test is a test focusing on defect detection.

Although these test activities are carried out, defects may be detected after release. Among several causes, we paid attention to the fact that the defect detection testing is done by an ad-hoc manner. In this paper, we introduce how to improve the defect detection test from ad hoc manner to exploratory test.

1. はじめに

製造装置の組み込みソフトウェアの開発は、装置の初期開発から保守にいたるまで開発期間が長く、その間に機能追加、機能改修を随時行う必要がある。特に新規装置では改良が頻繁に行われ、ソフトウェアは定期的にリリースされる。この定期的なリリースでは、開発から実機組み込みまで複数のテストフェーズがある。

- ・実装時のテストコードを使ったテスト
- ・実装後の結合動作を確認するテスト
- ・リリース前の総合的なテスト

リリース前のテストでは、目的の異なる2つのテストを行っている。仕様確認のテストと欠陥検出のテストである。仕様確認のテストを主とし、欠陥検出のテストは仕様確認のテストを補完している。

- ・仕様確認テスト : 実装した機能が仕様通りに動くことを確認するテスト。
事前にテスト仕様書を作成し、仕様書にしたがってテストを行う。
- ・欠陥検出テスト : 欠陥検出を主眼としたテスト。
仕様書を用いず、テスターが気になる箇所を確認する。

これらのテスト活動を行っているが、リリース後に欠陥が検出されることがある。いくつかある原因の中で、欠陥検出のテストがアドホックな方法で行われている点に着目する。本稿では、欠陥検出テストのアドホック方式から探索的テスト^[1]への改善を紹介する。

東京エレクトロン株式会社
TOKYO ELECTRON LIMITED

北海道札幌市中央区南1条東1丁目 Tel: 011-252-2711 makoto.nakakuki@tel.com
Higashi1 Minami1, Chuo, Sapporo, Hokkaido Japan

- 1) 東京エレクトロン株式会社 次世代 APP 開発グループ チームリーダー
TOKYO ELECTRON LIMITED System Development Center Advanced Application Development Group Team Leader
2) 東京エレクトロン株式会社 次世代 APP 開発グループ
TOKYO ELECTRON LIMITED Advanced System Development Center Application Development Group

【キーワード:】 アドホックテスト、探索的テスト、セッションベースドテスト、テストチャータ

2. 課題と取り組み

2.1 課題

欠陥検出のテストについて把握している具体的な問題を次に列挙する。

A) 何をどのようなテストをすると良いかわからないため

- ① ただ触るだけで時間が過ぎる。
- ② 重点的に確認すべきポイントが外れている。
- ③ 重要な欠陥を検出できず、スペルミスなどの比較的軽微な欠陥しか検出できていない。

B) 責任範囲が不明確のため

- ④ だれが何をテストしたかが不明で、複数人で同じ内容をテストしている、もしくは、だれもテストしていない可能性がある。
- ⑤ 参加が消極的で参加するメンバーが決まっている。

2.2 施策

各問題に対する施策を以下に述べる。①～⑤は「2.1 課題」の内容と対応している

① ただ触るだけで終わってしまう場合、テスターがその機能を知らない、ドメイン知識の不足、テストの勘所のようなノウハウがないことが考えられる。

【施策】

1. テスト実施前：対象機能に対するテストの方針、テストケースを抽出し、テスターに情報を提供する。
2. テスト実施後：テスト実施ログ、検出欠陥をもとにチームメンバーでノウハウを共有する。

② 重点的に確認すべきポイントを外している場合、そのリリースで追加、または、変更された機能を知らないことが考えられる。

【施策】

1. テスト実施前：対象機能の説明と実装者の不安を抽出し、テスターに情報を提供する。

③ 重要な欠陥を検出できず、軽微な欠陥のみが検出される場合、対象がどのような欠陥があるといけないのか？という狙いがわからないと考える。

【施策】

1. テスト実施前：テストで見つきたい欠陥について、事前に認識合わせを行う。

④ だれが何をテストしたか不明の場合、重複してテストする場合ならばまだ良いが、誰もテストしていない場合を避けたいといけない。

【施策】

1. テスト実施前：テスト範囲ごとに複数のテスターをアサインする。
2. テスト実施時：実施内容を記録する。

⑤ 基本的には任意参加になるため、他の作業を優先する傾向がある。

【施策】

1. テスト実施前：メンバーに担当範囲と実施予定時間を宣言してもらう。

これらの施策を実行するために、大きく次の2つこのことを行った。

- ・テスト実施前後にキックオフミーティングと振り返りミーティングを開催する
- ・チャーターを利用する

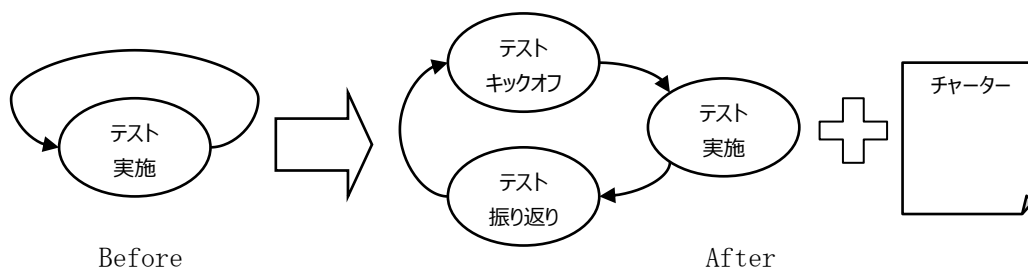


図 2-1 テストプロセスの変更

(1) キックオフミーティング

- ・テスト対象の明確化
対処案件から変更箇所をピックアップし、テスト対象を明確にする。(施策②-1)
ここでテスト対象は明確にするが、ピンポイントにそこだけをテストするというのではなく、最低でも明確にした対象を外さないようにするのが狙いである。
- ・方針のアウトライン決め
案件担当から気にしてほしいポイント、不安ポイントを拾い上げて、どのようなテストをすると良いか、方針のアウトラインを決める。(施策②-1)
- ・検出する欠陥の認識合わせ
エラー推測などの観点を利用して、検出しておきたい欠陥を確認する。(施策③-1)
重要な欠陥の検出を意識付けるが、決して重要度の低い欠陥を無視するということではない。
- ・テストケースの抽出
先に抽出したテスト対象、テスト方針をもとに、どのようなテストをすると良いか、論理的なテストケース出しを行い、テストする内容を明確にする。(施策①-1)
テストケースの抽象度はテストをするきっかけになる程度にしており、先にあげた方針で十分な場合は省略している。
- ・タイムボックスの設定
テスト対象ごとに数人を仮アサインし、テスター自身にテスト対象ごとにテストする時間(≒タイムボックス^[2])を宣言してもらうことで積極性を促す。(施策④-1、⑤-1)
ただし、このテストはあくまでも任意参加のため、これは仮アサインとし、アサインを気にせずに行うことができる人が積極的にテストを行う。

(2) チャーター^[3]の利用

- ・チャーターの作成
実施前にテスターがキックオフに抽出したテスト対象ごとにテスト方針、テストケースを整理し、チャーターを作成する。(施策①-1)
必ずしも事前に整理したことだけをテストするのではなく、気になることがあれば自主的に範囲を広げられる。
- ・テストの実施内容の記録
テスト実施期間中に誰がどのようなテストを行ったかをメンバー間で確認できるようにテストの実施内容を記録(ロギング)する。(施策④-2)
このログを見ることで他のテスターが同じテストを実施しなくて済む、もしくは、未着手のテストを確認できる。また、ログをインプットとして他のテスターは気づきを得られる。
ログの記入は、テスト実施中/実施後のどちらでも良く、テスト実施の流れを止めないようにする。

機能グループ名

対象と概要

チケット	タイトル	概要	実装担当	実務者コメント
#101	〇〇機能の追加	〇〇を××で変更する	A君	関係機能があるため、機中機により、想定外の結果にならない心配。
#114	〇〇するところ××にならなくてほしい	〇〇するところ××にならないように△△する	B君	前提条件が不明です。考えを最終段階の条件確認時に確認させていただきます。仮に××××が原因であれば対応いたします。

テスト方針

- ・ #101と関連ありそうも実施と判断もする
- ・ #114の優先度は低いテストも中心に実施する

テストケース

- ・ A機能、B機能と#101機能の組み合わせ

担当

担当	見積もり(時)	実績(時)	実施概要
A君	60		
B君	30		

ログ

No.	日付	記入者	検出方法	操作	気になったこと	コメント	チケット	ステータス
01	11/10	D君	仕様確認	機能Aの手順で機能Bを実行	機能Aの手順で機能Bを実行するとエラーが発生する	11/10 A: 〇〇でしようかと確認します。 11/10 A: 〇〇は仕様と異なります。 チケット登録をお願いします。 B: 〇〇でチケット登録しました。	#100	〇 既知 〇 対応中 〇 終了
02	11/10	D君	テスト方針より関係機能の確認	機能Aの手順で機能Bを実行	機能Aの手順で機能Bを実行するとエラーが発生する	11/10 A: 〇〇でしようかと確認します。 11/10 A: 〇〇は仕様と異なります。 チケット登録をお願いします。 B: 〇〇でチケット登録しました。	#101	〇 既知 〇 対応中 〇 終了

機能グループ

テスト対象のグループ分け

対象と概要

テスト対象の詳細情報

テスト方針

テスト対象の情報から決めた方針

テストケース

方針をもとに実施する抽象的なテストケース

担当

テストを実施する担当者を記入、チャーター作成時に実施予定時間を記入し、実施後に実績時間、実施したことを記入する

ログ

実施時の記録。「気になったこと」「やってみたこと」を記録する。

図 2-2 チャーターのイメージ

(3) 振り返りミーティング

振り返りでは施策の実施というよりも、取り入れた施策をよりよくすることが中心であり、改善すべきものは次のキックオフへのインプットとする。

- ・ ログの共有
検出した欠陥の内容とどのように検出したかを共有する。(施策①-2)
- ・ KPT^[4]の実施
定性的な振り返りでテスト実施に関する気づきや問題点を共有する。
- ・ メトリクスの利用
定量的な振り返りで改善経過を確認する。

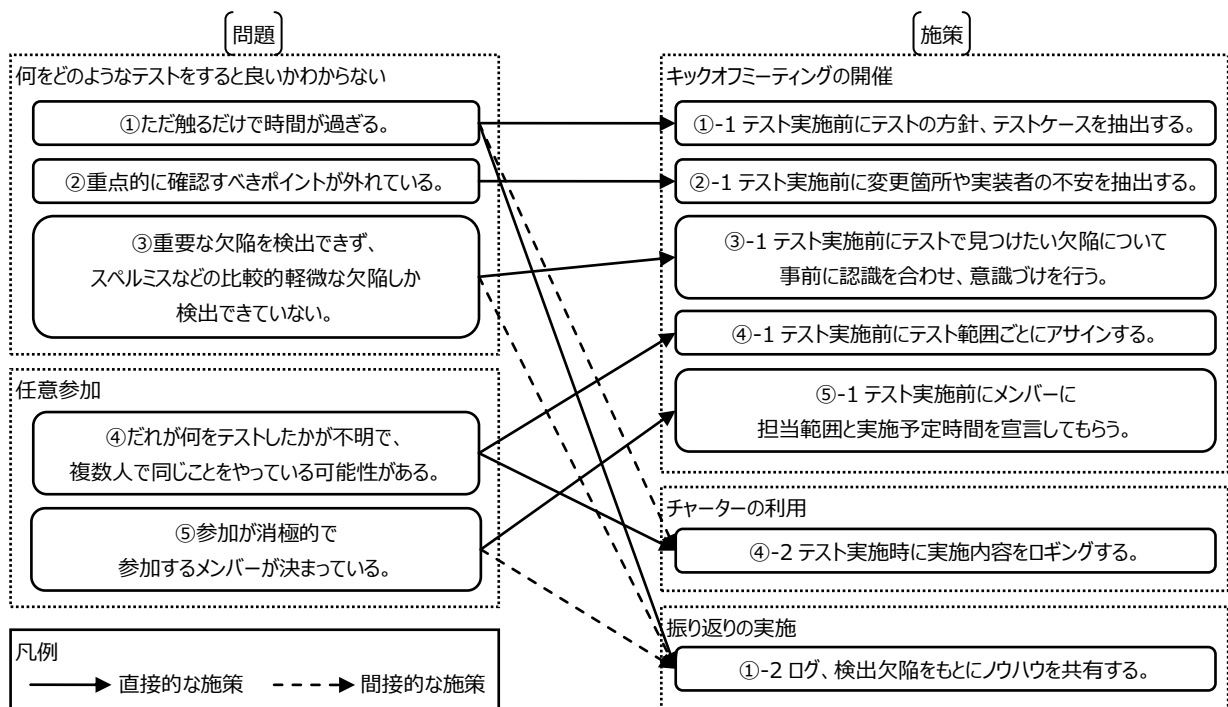


図 2-3 問題と施策の関係図

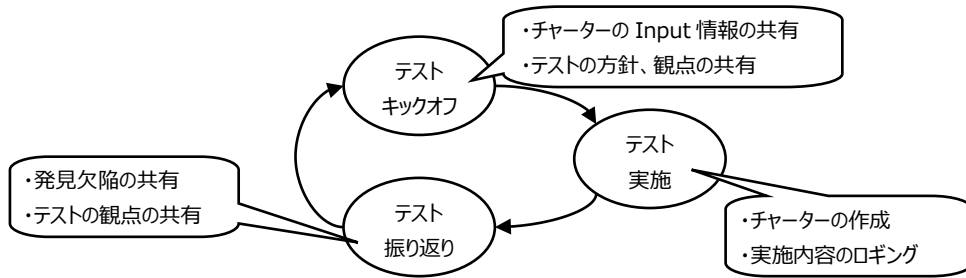


図 2-3 施策とプロセスの関係図

3. 結果と考察

本施策を 1 年間続けた効果を施策適応時に決めたゴール指標と定性的な効果を用いて考察する。

3.1 定量的な結果

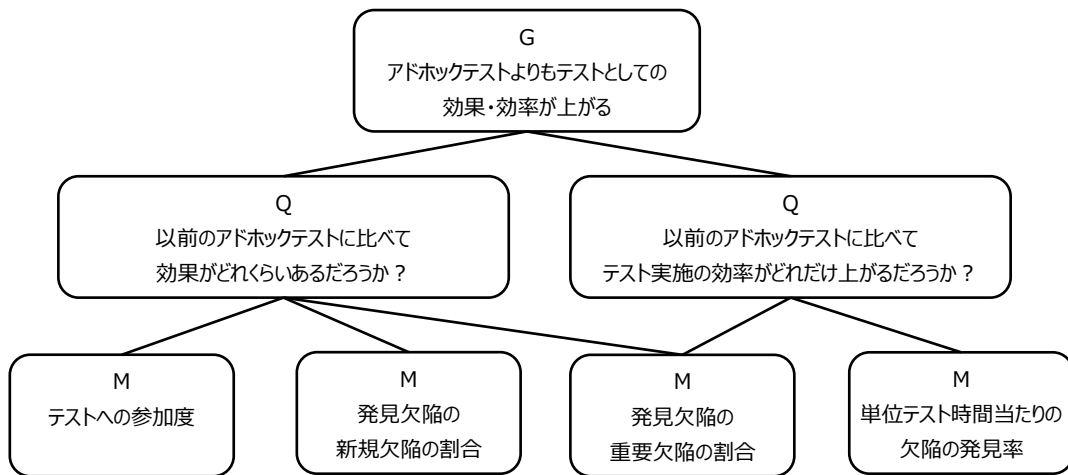


図 3-1 GQM^[5]

(1) テストへの参加度

【仮説】

対象範囲のテスターを明確にしてテスター自身にタイムボックスを宣言してもらうことで、テストの時間を確保してテストが行われるようになると考えるため、参加度が上がる。

【考察】

参加率だけをみると After2~4 は施策前よりも下がっているが、参加度に注目すると全体的に施策前よりも参加度が上がっているため、改善していると言える。

参加度：

参加率と参加者一人当たりの実施時間を掛けた値。チームメンバー一人当たりのテスト時間と等価。

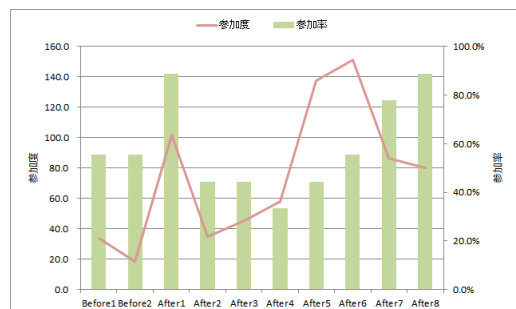


図 3-2 参加度と参加率

(2) 発見欠陥における新規欠陥の割合

【仮説】

リリースされる機能を共有し、変更による影響を考慮したテストをするようにしたため、潜在欠陥よりも変更内容による新規欠陥の検出が多くなる。

【考察】

新規欠陥の割合を見ると施策取り込み直後は、取り込み前よりも新規欠陥の割合が低くなっている。これは、取り入れた施策の副次的な効果として、今までよりも多くの欠陥を発見できるようになり、これまで取りこぼしていた潜在欠陥を検出できるようになったと考える。そして、潜在欠陥がある程度取り除かれた時点（After4以降）で新規の割合が徐々に増えていると考える。これより、新規欠陥の発見に関しては改善していると言える。

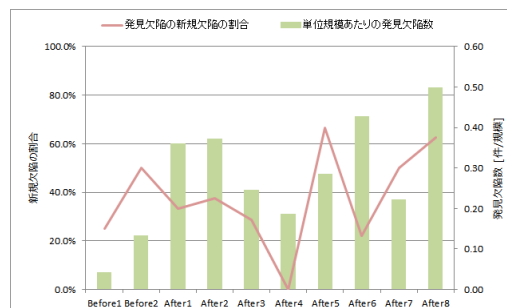


図 3-3 新規欠陥の検出割合

(3) 発見欠陥における重要欠陥の割合

【仮説】

どのような欠陥を発見しておきたいか周知し、意識付けを行ったことで、発見欠陥のうち重要欠陥の割合が増える。

【考察】

重要欠陥の割合を見ると Before よりも下がっている。これは施策以前で発見できた欠陥数自体が少ないため、重要欠陥1件が占める割合が大きくなっていると考えられる。一方 After では発見欠陥数が Before より増えているため、発見した重要欠陥が占める割合が下がったが、欠陥検出としては良い結果が出ていると言える。

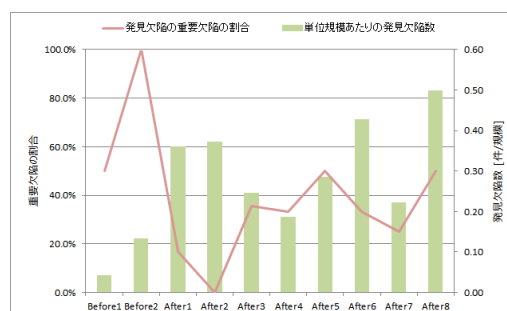


図 3-4 重要欠陥の発見割合

(4) 単位テスト時間当たりの欠陥の発見率

【仮説】

事前にテストケースを抽出するため、単位時間当たりで発見できる欠陥が多くなる。

【考察】

単位時間当たりの欠陥発見数は下がっている。しかし、単位規模あたりのテスト時間が増えているため、一つの案件に時間をかけるようになり、この影響で単位時間当たりの発見数が下がっていると考えられる。良い方向に進んでいるがムダに時間をかけていないか、テストにかかる時間が適切か、を見極める必要があり、今後の課題である。

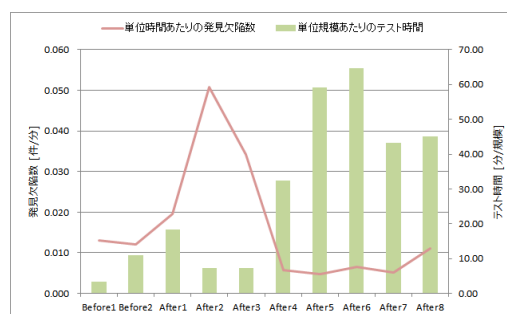


図 3-5 単位時間当たりの欠陥発見数

3.2 定性的な効果

- ・ 取り組み前はテスト範囲が途方もなく広大な範囲を手探りで欠陥を探している感じで、挙句の果てに何も見つからないと徒労感しか残らなかったが、取り組み後ではそういう感じがなく、「またやろう」となる。
- ・ ワイワイ相談しながら楽しんでテストできる。
- ・ 複数人で狙う範囲をカバーするため、安心感がある。
- ・ 対象と実施時間を宣言することで参加する意識が高まった。
- ・ 狙うポイントを明確することで、テストに取り掛かりやすく(敷居が低く)なった。

3.3 結論

収集したメトリクスは、改善した、もしくは、改善してきているという結果読み取れる。これにより GQM モデルの G を達成したと考えられる。

定性的な効果を取り上げると、想定していなかった「またやろう」というテストに対するモチベーションアップやテストが楽しい、安心感があるなどの意見が出ている。

このことから、今回の取り組みを通して、先にあげていた問題がほぼ改善し、想定していた以上の成果を得たと言える。

4. まとめ

結論では想定以上の成果を得ていると述べているが、これは一回の取り組みでできたのではなく、キックオフ→振り返りというループを重ねることで改善が進んだ結果と考えている。先にあげていた問題が徐々に改善されて行く状況は定量データから見ることはできるが、探索的テストの運用自体の改善が進んだことも影響していると考えている。

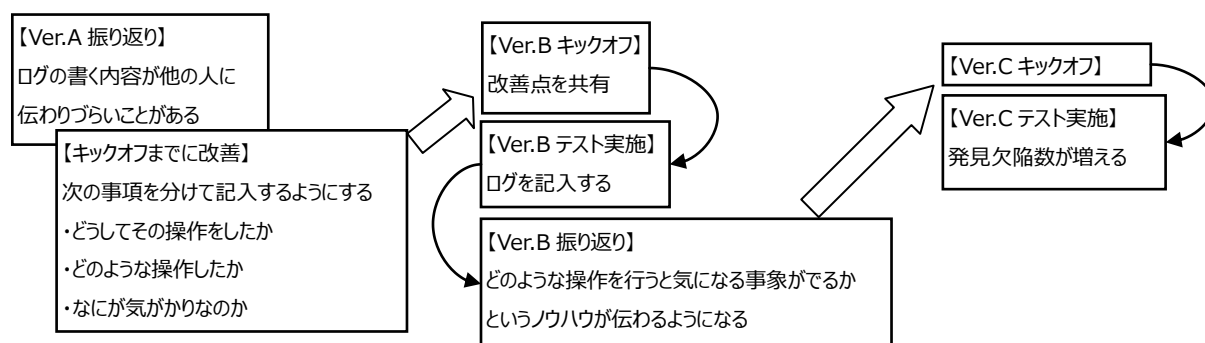


図 4-1 運用の改善による効果の例

また、今回の取り組みを続けるために気を付けたポイントがいくつかある。次に列挙する。

取り組みのポイント

- ・ キックオフ、振り返りのミーティングを短時間（30分程度）に抑えることで、時間的な負担を感じさせない。
- ・ ログを書くのが手間にならないように、文言などは気にさせずに書けるように気軽さを重視した。
- ・ 振り返りで出たものを次のキックオフの入力にすることで、振り返りの意義を実感できるようにした。

このようなことを気にかけることでチームとしての継続的に運用できたと考えている。そして、これらの結果は、個人のテストでは得られず、チームでキックオフや振り返りという共有する場を設け、チャーターを活用した結果と言える。

5. 今後の課題

以下に課題を述べる。

- ・ 探索的テストで見つかった欠陥が、「探索的テスト」で見つかった理由を探り、チームが対象としているドメインや開発プロセスと照らし合わせ、開発の改善につなげる。
- ・ この運用は一部のチームだけにとどまっているため、他のチームへ展開する。
- ・ 良い方向に進んでいるが過剰にテストに時間をかけていないか、テストにかかる時間が適切か、考慮することが必要である。

6. 参考文献

- [1] 高橋 寿一、知識ゼロから学ぶソフトウェアテスト、2013 年
- [2] Jonathan Bach、Session-Based Test Management、2000 年
- [3] Japan Software Testing Qualifications Board 技術委員会、ソフトウェアテスト標準用語集 日本語版 Version 2.3.J02、2015 年
- [4] 天野 勝、これだけ! KPT、2013 年
- [5] Victor Basili、Adam Trendowicz、ゴール&ストラテジ入門: 残念なシステムの無くし方、2015 年