

88

Session Based Test Management による 探索的テストの実践¹

～受託開発でも探索的テストを管理し活用できる～

1. 概要

当社はシステム・インテグレーターとして、多くの顧客に対して、システムの受託開発を行っている。昨今はビジネス環境の変化に伴い、システム開発に対して、開発スピードの向上とコストの低減がこれまでより強く求められるようになってきている^[1]。当社は顧客のニーズにこたえるため、ソフトウェア開発における生産性向上を目指し、多くの取組みを実施している。特にソフトウェアテストについては、バグの摘出効率が優れている探索的テストの活用を目指している。具体的には、従来から実施しているテストケース表を用いた記述式テストに対し、探索的テストを補完的に用いることで、バグの摘出工程を早め、手戻りを抑止することを推進している。

一般的には、探索的テストは、品質向上や生産性向上を目指す多くの組織に注目・活用されている。しかし、当社のようなシステムの受託開発を行う業者における、探索的テストの採用事例は少ない。探索的テストは、記述式テストとは異なり、事前にテストケースを作成しないため、テストの総量や実施範囲が不透明であり、テスト管理が難しいことが原因の1つと考えられる。本事例では、探索的テストに管理単位を与えるため「Session Based Test Management」^[3]と呼ばれるテスト管理の考え方をを用いて、受託開発の中で、探索的テストを活用できるようにした事例を紹介する。

2. 取組みの目的

2.1. 解決すべき課題と目標

記述式テストでは、多くの場合、事前に作成したテストケースの数をを用いて、テストの進捗や実施度合いを管理している。当社においても、プロジェクトマネージャーや、開発の委託元である顧客に対して、テストケース数をを用いた進捗報告やテスト結果報告を行うことが多い。一方、探索的テストでは、事前にテストケースを作成しないため、テストの総量や実施範囲が不透明となってしまう、前述のような報告を実施しにくい。特に受託開発においては、プロジェクトマネージャーや顧客が適時にテストの進捗状況や結果を把握することで、開発しているシステムの全体計画や、開発に関与する多数のチーム・メンバーと調整を行うことが多く、探

¹ 事例提供：株式会社 NTT データ 熊川 一平 氏

索的テスト特有の不透明さが受け入れられ難い。このような探索的テストにおけるテスト管理の問題を、JSTQB Foundation Level のシラバスに従い、テスト管理のカテゴリごとに整理すると表 88-1 のようになる²。

表 88-1 探索的テストにおけるテスト管理の現状

テスト管理のカテゴリ (JSTQB Foundation Level シラバスより)	探索的テスト実施時のテスト管理上の問題点
テスト計画作業と見積もり	●テストの内容や作業量はテスターに依存しており、テストの総量がわからない。
テスト進捗のモニタリングとコントロール	●総量が不明確なため、テストの進捗状況を把握することができない。 ●テストの実施内容が、テスターに依存しており不透明になっている。
テスト組織	●明確な役割（ロール）が存在せず、テストの管理はテスター任せとなっている。
構成管理	●どのバージョンのアプリケーションに対して、どの程度テストを行ったかわからない。
リスクとテスト	●テストの実施順序や優先度はテスター任せである。
インシデント管理	●テストに関する記録がなく、インシデントを引き起こした背景が伝わらない。 ●軽微なインシデントや、改善項目の報告が乱発しており、開発者の混乱を産んでいる。

そこで、受託開発において、プロジェクトマネージャーや顧客が、探索的テストの導入を前向きに検討できるように、記述式テストにおけるアウトプットや、テスト管理の方法を参考に、テスト管理のカテゴリごとに改善目標を定義し、取組みを行うこととした。具体的な改善目標は表 88-2 の通り。

表 88-2 探索的テストにおけるテスト管理の改善目標

テスト管理のカテゴリ (JSTQB Foundation Level シラバスより)	改善目標
テスト計画作業と見積もり	●事前にリソース割り当てを計画できること。 ●作業量の見積もりができること。 ●テストの開始/終了基準が明確にできること。
テスト進捗のモニタリングとコントロール	●進捗状況をメトリクスで確認できること。 ●テストレポートが残されること。 ●テストの結果やレポートによって方向性をコントロールできること。
テスト組織	●テストに関わる担当者の役割（ロール）が明確とできること。
構成管理	●テスト対象アプリケーションのバージョンごとに、テストを行った総量が確認できること。
リスクとテスト	●テストの優先度が設定できること。 ●リスクの再評価が可能であること。
インシデント管理	●開発担当者や関係者に必要に応じて問題の特定・抽出・解決に向けたフィードバックができること。

3. 取組みの対象、適用技術・手法、評価・計測

3.1. 方法論（どんな技術・手法を用いて実施したか）

表 88-1 および表 88-2 を俯瞰すると、テストの実行単位を持たないことが、探索的テストにおけるテスト管理の難しさであることがわかる。そこで、探索的テストに管理単位を与えるため、Session Based Test Management と呼ばれる手法に着目した[3]。Session Based Test Management では、テストをセッションと呼ばれる時間枠に分割し、管理単位としている。セッションは 30 分～2 時間程度の時間枠であり、セッションごとにテストの目的を定義する。また、探索的テストでは、テストチャーターと呼ばれるテストの範囲や目的・観点などを整理したドキュメントを用いることがある。Session Based Test Management では、セッションごとにテストチャーターを設定することで、探索的テストであっても、テストを実施した範囲や総量、観点などを後から把握することができる。そこで我々は、Session Based Test Management の概念を取り入れ、時間枠（セッション）ごとにテストチャーターとレポート様式を兼ねたセッションシートを準備することで、図 88-1 のように、テストの実施前にテスト管理者がある程度テストの内容を指示することや、テストの実施後に内容を振り返れる状態を目指した。



図 88-1 セッションシートを用いたテスト管理

また、テスト1人あたりの1日に実施するセッション数の目標値を定めることで、テスト期間内に実施可能な目標セッション数を割り出すことができる。目標セッション数に対するセッションシートの消化率を管理することで、テストの進み具合を把握することも目指した。具体的には、図 88-2 のように、テスト1日に実施する最大セッション数を3セッションと定め、テストの人数とテスト期間から目標セッション数を設定した。最大セッション数を3

セッションとしたのは、Session Based Test Management の考え方において、1セッションの最大時間を2時間とされているため、業務時間内で無理なく実施できるように考慮したことと、テストターの集中力が持続できる限界値を考慮したためである。探索的テストのようにヒューリスティックなテストでは、テストターは機械的な作業を行うわけではなく、テストに集中し、より多くのバグを見つけるために思考を巡らせる必要がある。これまでに探索的テストを実施したテストターにヒヤリングを行い、集中力の持続できる限界を検討した結果、1日3セッション程度までが妥当だと判断した。

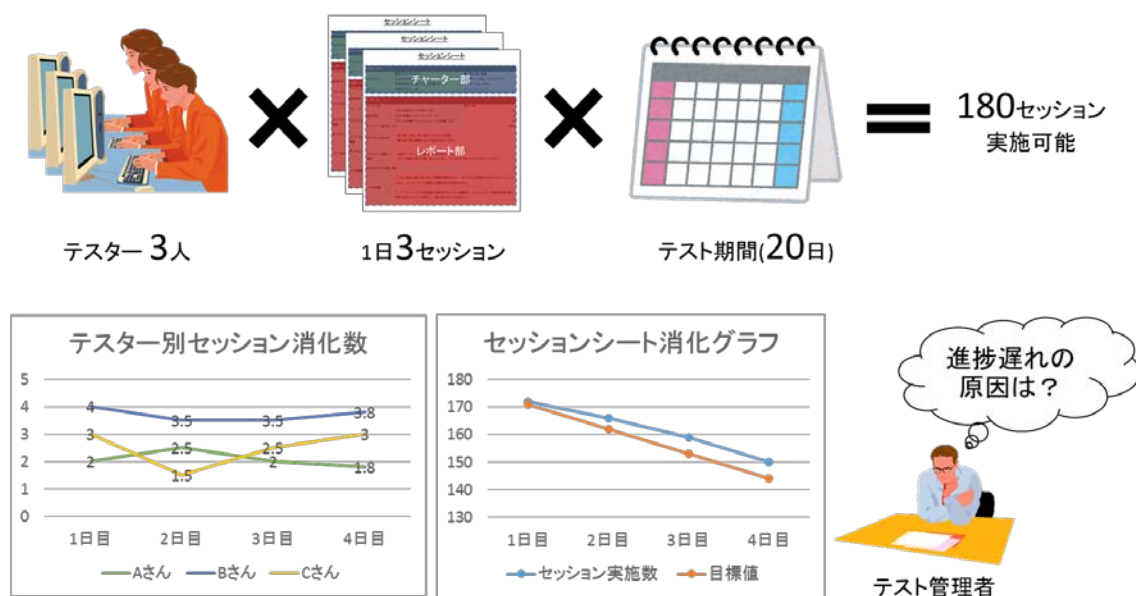


図 88-2 セッションによるテストの予実管理

記述式のテストでは、テスト対象ごとに実施したテストケースの数や、見つかったバグの数、バグの原因分類などによる分析を行い、弱点と思われる箇所に対して、品質を強化するために追加のテストを検討することが多い。一般的には、バグは偏在していると考えられるため、より効率的にテストを進めるためには、テストの実施状況や発見されたバグの原因の偏りを見て、テストを実施する対象や、テストの実行回数などを柔軟に変更できたほうが望ましい。また、重大なバグが見つかった場合に、他の機能でも同様のバグがないか確認できるのも重要である。

探索的テストにおいても、こういった状況にあわせたテストの方向性の制御が実現できることを目指した。具体的には、テストターに、セッション中に見つかったバグや、思いついたバグのヒントになりそうな点を、セッションシートに記入してもらうようにした。テスト管理者は、図 88-3 のように、日々セッションシートのレポートを確認し、テストターのレポートから気づきを得ることで、新たなセッションシートを作成し、随時テストの方向性を制御できる。例えば、類似した機能間で見つかったバグの傾向を見てセッションシートを追加することで、より効率的にバグを見つけられるように方向性を制御することができる。また、テスト対象ごとの

実施セッション数の偏りや、1セッションあたりのバグ数などを確認することで、バグのありそうなことを類推してテストの追加を検討できる。このようなテストの実施状況の確認や、バグの抽出傾向からの分析・方向性の制御は、テスト管理者とテスターの間で、日次の情報共有会を行い、その中で情報を収集することで実現した。

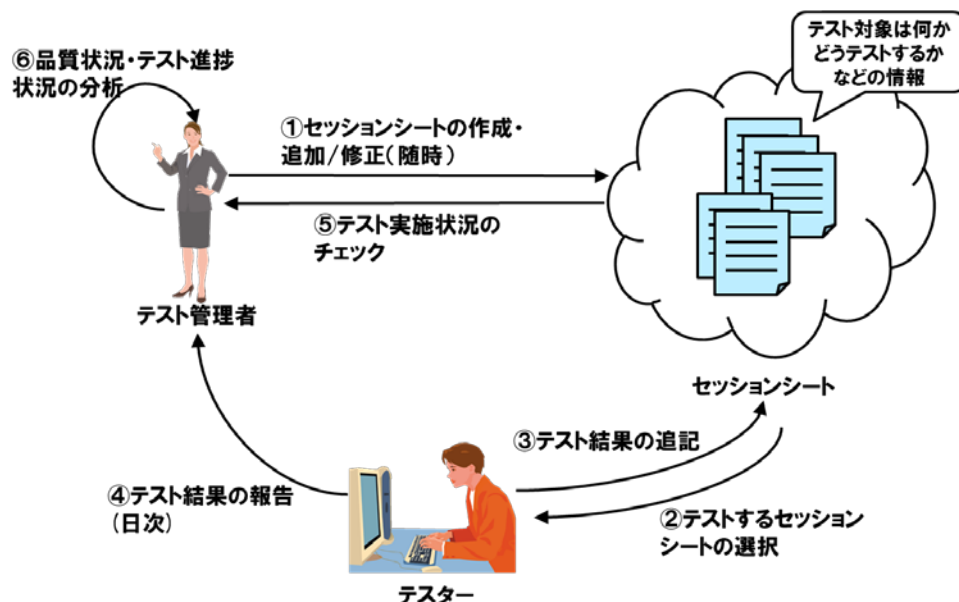


図 88-3 セッションシートによるテストの方向性制御

3.2. 対象プロジェクト

3.1 で示した方法論を用いて、以下の 2 つのシステム開発プロジェクトにおいて、探索的テストの導入を試みた。我々はプロジェクトにおけるテスト方針の検討支援メンバとして参画し、探索的テストの導入を推進した。具体的には、いずれのプロジェクトにおいても、あらかじめテスト管理者・テスターの両者に対して探索的テスト自体の考え方の研修を開催すると共に、Session Based Test Management の考え方を整理し、当社の開発標準に組み込むためのガイドラインドキュメントを作成の上、提供した。

(1) 金融機関向け審査システムの保守開発

当社にて保守を行っている Web システムに対しての機能追加を行うプロジェクトであり、リリース後のバグや仕様不備に対する品質強化施策として探索的テストを導入した。

(2) 教育機関向け学生管理システムのシステム更改

クライアント・サーバ型の現行システムから、パッケージソフトを利用した Web システムへと更改するプロジェクトであり、早期バグ抽出による後続のテスト工程の安定化を目指して探索的テストを導入した。

3.3. 取組みの評価

取組みの評価にあたっては、テスト実施期間中の日次の情報共有会や、プロジェクトの進捗会議模様を確認すると共に、テスト実施期間終了後にアンケートやインタビューを行うことで情報を収集した。ここでは、3.2 で示したプロジェクトのうち、(2) 教育機関向け学生管理システムのシステム更改での事例を取り上げる。

当プロジェクトでは、テスト全体の計画を行う際に、事前にアサインできるテスターの人数・テスト実施できる日数を定め、探索的テストを行う目標セッション数を導出した。テストの進捗管理は目標セッション数に対する差分を管理することで、進み具合・遅れ具合を評価した。具体的には図 88-4 のように、残存セッション数を週次単位でグラフによって図示することを試みて、進捗の遅れ具合を素早く察知できるようにした。当該プロジェクトでは、テスターの繁忙により、セッション数の消化に若干の遅延が見られたため、プロジェクトマネージャーにより、テスターの追加や期間の見直しなどの対処が指示され、進捗管理・モニタリングの観点から適切なマネジメントが行われていることが確認できた。

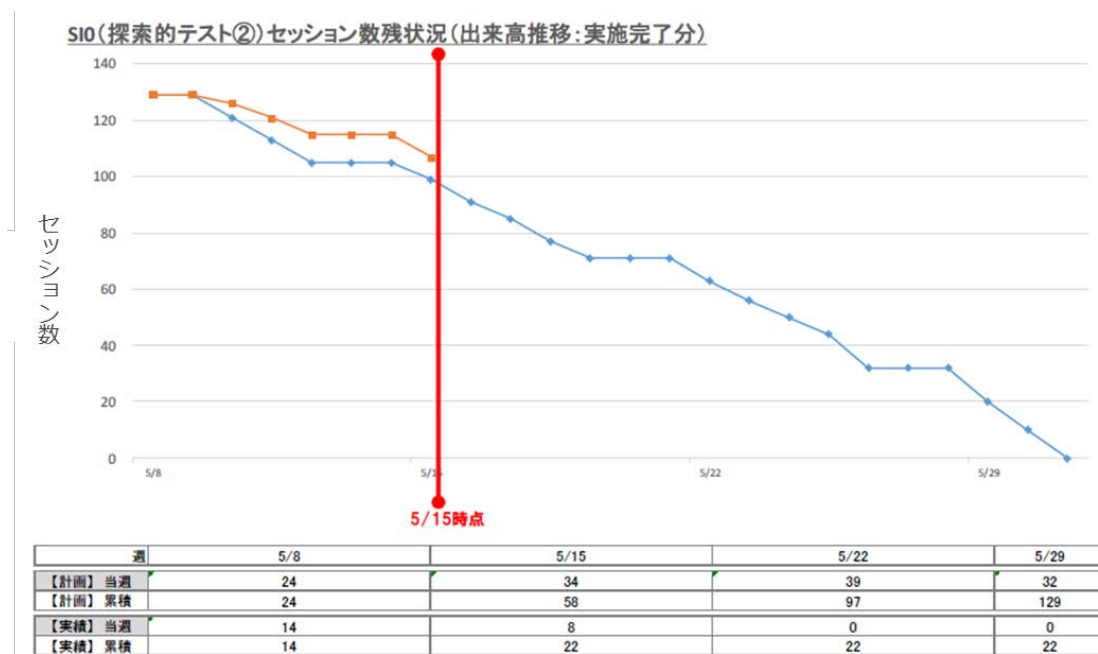


図 88-4 試行プロジェクトでのテスト進捗管理グラフ

また、テストの実施状況とバグの摘出状況の確認として、1 セッションあたりの摘出バグ数を機能ごとに図 88-5 のようにグラフで整理した。この結果、特定の機能でのバグが多く摘出されていることや、重大バグの発生状況の偏りを把握することができ、後続のセッションで実施するテストの内容を変更したり、前工程で実施していた記述式のテストや、設計工程の内容の見直しを行ったりと、状況に合わせた柔軟な対応を行うことができた。

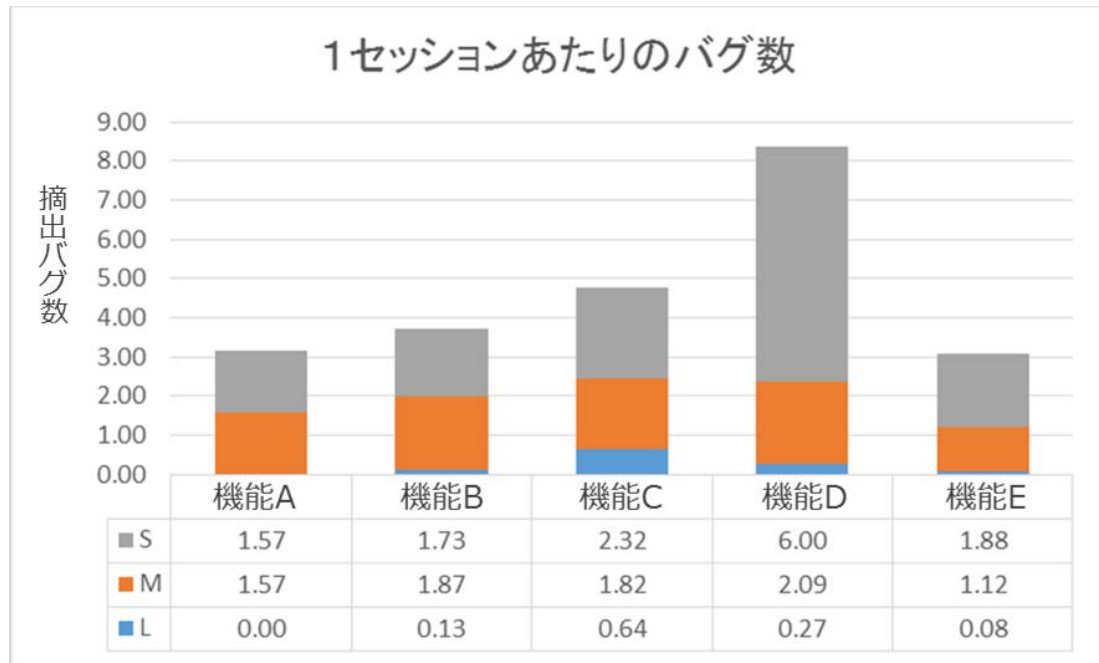


図 88-5 試行プロジェクトでのバグ抽出状況管理グラフ（一部情報をマスク）

このように観察されたテスト管理の改善状況に加え、テスト実施後のアンケートやインタビューの結果を踏まえて、表 88-2 に示した改善目標に対しての改善状況を表 88-3 の通り整理した。

表 88-3 目標に対しての改善状況

テスト管理のカテゴリ (JSTQB Foundation Levelシラバスより)	改善目標	実施結果
テスト計画作業と見積もり	● 事前に関リソース割り当てを計画できること。 ● 作業量の見積もりができること。 ● テストの開始/終了基準が明確にできること。	● セッションシートを使うことで、事前に関リソースの割り当てができた。 ● 1日あたりの目標セッション数を定義することで総セッション数（テストの終了基準）を明確にし、作業量の見積もりをすることが出来た。
テスト進捗のモニタリングとコントロール	● 進捗状況をメトリクスで確認できること。 ● テストレポートが残されること。 ● テストの結果やレポートによって方向性をコントロールできること。	● セッションシートの消化状況をメトリクスとして、進捗状況を把握できるようになった。 ● セッションシートのレポート内容に次に実施すべきテストの内容を検討し、セッションシートを追加することで、より効率的にバグを見つけられるようテストの方向性をコントロールすることができた。
テスト組織	● テストに関わる担当者の役割（ロール）が明確にできること。	● テスト管理者とテスターを明確に分けることができた。
構成管理	● テスト対象アプリケーションのバージョンごとに、テストを行った総量が確認できること。	● セッションシートのレポート部にテスト対象のバージョン番号などを記載することで、セッション実施数を把握することができた。
リスクとテスト	● テストの優先度が設定できること。 ● リスクの再評価が可能であること。	● セッションシートごとに、優先度を設定できた。 ● リスクの再評価に応じて、優先度の見直しができる。
インシデント管理	● 開発担当者や関係者に必要に応じて問題の特定・抽出・解決に向けたフィードバックができること。	● セッションシートに記録することで、問題のフィードバックが可能となった。また、テスト管理者が問題の一次受けをすることで開発者に多量のフィードバックが一度にわたらないようコントロールすることができた。

いずれの改善目標に対しても、本取組みの成果が発揮されていることがわかる。また、インタビューで得たテスト管理者・テストターの意見を下記に示す。特にセッションという管理単位が導入されたことで、テスト全体に統制がもたらされたといった意見や、テスト管理者・テストターだけでなく、プロジェクト全体を統括するプロジェクトマネージャーなどの理解を得やすくなったという意見が目立った。ここでは取り上げていないが、3.2 で示したもう1つのプロジェクトでも同様の状況が観察されており、本取組みの成果が狙い通りかつ十分に発揮できたと考えている。

- インタビューで得た意見

- テスト管理者

- ☆ セッションという定量的な見方ができることで、その日のノルマが見えるようになり、1日のテスト実施量をキープできた。
 - ☆ プロジェクトマネージャーや顧客に報告する立場として、セッション数という区切りがあったのがよかった。
 - ☆ 見つかったバグから記述式のテストに新たな観点を追加するなどの改善を行うことができ、システムの品質向上に寄与できた。

- テスター

- ☆ セッションごとに指示が明確であり、迷わずにテストを行うことができた。
 - ☆ セッションごとに報告を行うため、テストの内容が多様になりがちな探索的テストであっても、報告が容易であった。

また、テスト管理者・テストター間で行う日次の情報共有会において、各テストターが実施したテストの内容や、見つかったバグ・気になった箇所などの報告を行うことで、テストター間でノウハウの共有が行われた。これにより、バグの摘出効率を更に引き上げるという副次的な効果が得られたと推測している。本取組みの施策を適用しなかった探索的テストに比べ、取組みを適用したいずれのプロジェクトでも時間あたりのバグ摘出数が改善されていることが確認できた。

4. 取組み実施上の問題、対策・工夫

Session Based Test Management の考え方では、テストの管理単位となるセッションの定義や、内容が統制された状態となることが重要である。例えば、同じ1時間のセッションであっても、テスト実施のための準備（テストデータの作成や、環境の変更など）を行う時間が長いセッションと、短いセッションでは実施できたテストの濃度が異なる。また、探索的テストでは、テストチャーターによる指示があるとはいえ、テストターがバグを探索しているうちに、予めテストチャーターで指定されたテストのスコープをはずれてしまうことがある。このようにセッションごとの内容にブレが生じてしまうと、セッションあたりのバグ数などの情報が、

実際の状況とかい離してしまう可能性がある。そこで、当社では Session Based Test Management の導入時に、予め以下のようなルールを策定している。

- (1) テスターがバグの探索を進めるうちに、1つのセッションに2時間以上必要なことが分かった場合、テスターはその旨をテスト管理者に報告すること。報告を受けたテスト管理者は、セッションの分割を行うこと。
- (2) 原則として、セッション中は探索的テスト以外の業務を行わないこと。
- (3) セッション中にテスト実施にどの程度の時間をかけたか、概算で報告を行うこと。テスト管理者は報告された時間を確認し、当該セッションの取り扱いの濃淡を定めること。また必要に応じてテスト実施の時間ができるだけ多く確保できるように、テスターの困りごとを解決すること。
- (4) テスターは、バグが見つけれられそうであればテストチャーターにて指示されたスコープを外れたテストを行ってよい。但し、セッションごとにテストチャーターにどの程度したがってテストを行ったかを、「%」で示す概算で報告を行うこと。テスト管理者は報告された数値を確認し、必要に応じてセッションシートの内容を、テスターが実際に行ったテストに合わせるように書き換えること。

このようなルールを定めることで、実際に行われたテストの内容と、テスト管理のために、事後に収集する情報との間でかい離が起きないように努めている。

5. 今後の取組み

本取組みでは、探索的テストにおけるテスト管理の改善に着目し、受託開発において要求されるテストの客観性を担保することを試み、一定の成果を出すことができた。本書を参考に、探索的テストが管理できないテストであるという誤解が解決され、システムの受託開発における受発注者間で、探索的テストの価値が認められ、探索的テストの採用が進むことを期待したい。

また、システム開発のスタイルや方法は今後も大きく変化していくことが予想される。技術の進化に伴い、システムに求める要件を記載すれば、自動で対応するソフトウェアを作成するような未来もあるかもしれない。しかし、どれだけ開発技術が進化しようとも、システムやソフトウェアが、期待していたものと違いないことを確認するためには、その要求の出元である人間に依存せざるを得ない。探索的テストのようなヒューリスティックなテストは、そのような状況でも非常に有効な技術であると予想する。今後は、探索的テストの管理面だけでなく、ヒューリスティックな側面に着目し、その人のシステム・ソフトウェアに対する潜在的なニーズや要求をより効果的に引き出す方法などの、改善を進めていくことを検討している。

参考文献

- [1] 経済産業省, 平成 22 年度企業の IT 投資動向に関する調査報告書 (企業 IT 動向調査), 経済産業省 商務情報政策課, 日本, 2011 年.
- [2] International Software Testing Qualifications Board, テスト技術者資格制度 Foundation Level シラバス 日本語版 Version 2011.J02, Japan Software Testing Qualifications Board, 2012 年.
- [3] Bach James, Session-Based Test Management, <http://www.satisfice.com/articles/sbtm.pdf>, 2000

掲載されている会社名・製品名などは、各社の登録商標または商標です。

独立行政法人情報処理推進機構 技術本部 ソフトウェア高信頼化センター (IPA/SEC)