ソフトウェアテスト自動化成熟度アセスメント手法による現状分析の実現

Achieve a current state of analysis using software test automation maturity assessment techniques

日本ナレッジ株式会社カスタマーサクセス部
Nihon Knowledge Co, Ltd. Customer Success Division
○加藤　大受 [1)]　　茂木　彩音 [1)]
○Daiju Kato[1)]　　Ayane Mogi[2)]

Abstract The fastest way to improve software development productivity is to automate software testing. To maximize the benefits of test automation, it is desirable to consider eliminating dependence on specific individuals and to develop and implement guidelines, but it is difficult to quantify the benefits of these measures. Therefore, we have defined a maturity model for the test automation process in development organizations, developed a checklist that can be used to assess the process, and developed a method for identifying areas for improvement in the test automation process. This paper explains how this methodology was developed and its benefits.

1.　はじめに

今日, IT 技術は私達の生活を支える必要不可欠な存在であり, その技術によって構築された様々なシステムが正常に稼働し，それらのシステムと連動したビジネスやサービスの提供が行われている．ICT の浸透によってビジネスのスピードはますます加速しており，その加速するスピードに合わせるように IT 技術もウォーターフォール開発からアジャイル開発［1]へと進化し, システムもオンプレミスからクラウドへと移行している．また，開発担当と運用担当が連携・協力し，フレキシブルかつスピーディーに開発するソフトウェアの開発手法である DevOps［2]も多くの開発組織で導入され, ソフトウェア開発のスピードアップを支えている．

ソフトウェア開発の生産性を高めるためには開発タスク全体の半分以上を占めると言われるソフトウェアテストの効率を高めることであり, その方法の一つにソフトウェアテストの自動化の実現がある. 一概にテスト自動化といっても図 1 のどのテストレベルのテストを自動化するかの検討が必要である(図 1)．

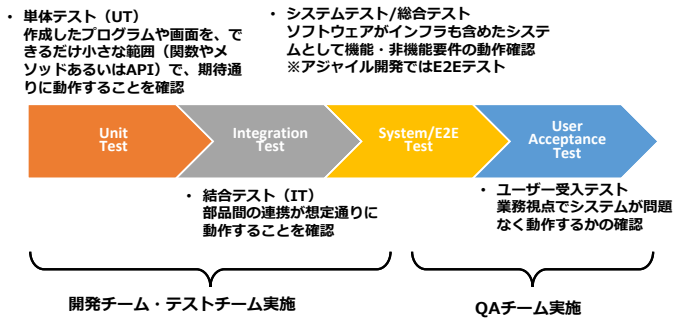

図 1. テストレベル

日本ナレッジ株式会社　カスタマーサクセス部
Customer Success Division Nihon Knowledge Co,　Ltd.

東京都台東区寿 3−19−5 JS ビル 9F　Tel: 03-3845-4781　e-mail:d-kato@know-net.co.jp
JS Building 9F 3-19-5,　Kotobuki taito-ku,　Tokyo Japan
1)日本ナレッジ株式会社 カスタマーサクセス部・エグゼクティブコンサルタント
Executive Consultant,　Customer Success Division,　Nihon Knowledge Co,　Ltd.
2)日本ナレッジ株式会社 カスタマーサクセス部・アシスタントセールス
Assistant Sales,　Customer Success Division,　Nihon Knowledge Co,　Ltd.

【キーワード】ソフトウェアテスト自動化，成熟度モデル，アセスメント，プロセス改善，品質管理

テスト自動化を考える場合，テストケースの望ましい比率を3つのテストレベルとしてピラミッド型で提唱したテストピラミッド [3]がある(図 2).

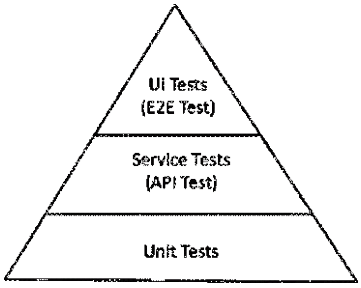

図 2 テストピラミッド

テスト対象をメソッドレベルなどの最小単位に分割してその振る舞いをテストするユニットテスト，複数の層が一つに繋がって動いていることを確認する API テストやサービステストは開発チームまたは開発チーム内のテストチームで自動化されることが多い．これはコード修正の度にテストを実行することで不安定なコードによるデグレードの摘出を可能とするためである．そのため，ビルド毎にこれらのテストは実行されることが望ましい．

一方，UI テストは自動テストの中で最もテストプログラムの構築や保守に時間がかかるテストである.UI の変更による影響を受けやすいので,UI の開発が落ち着いた開発の後半に向いており，回帰テスト等に利用されることが多い.システム全体をエンドーツーエンド(E2E)で利用し，ユーザーが実際にそのシステムもしくはアプリケーションを使用するときと同じ振る舞いをテストするため,自動化の実現には UI 操作を支援するキャプチャーリプレイ機能を持つ Ranorex [4]，UFT One [5]などの商用テストツールを使うことが多い.

そのため， ピラミッドの幅でテストケースを表しているテストピラミッドでは，上位のテストレベルであるほど，テストの記述(設計と実装)および実行にコストがかかるが，テストの忠実性が高くなる.同時にテストの実行速度は遅くなり，テストが毎回同じ結果を出力する度合いは低くなり，テストスクリプトの動作する順番などでテスト結果が不安定になるフレーキーテストが発生しやすい．Parry et al. [6]は，フレーキーテストを調査し，テスト結果が不安定になる要因を整理しており，フレーキーテストを減らすためにはテストが失敗したときの回復シナリオとなるスクリプトを整備し，実行することである.

なお，E2E テスト(エンドツーエンドテスト)とは，アプリケーションが期待どおりに動作し，ユーザーのタスクやプロセスの種類を問わずデータフローが適切に機能することを確認する手法である.

2. テスト自動化による生産性向上を実現するために必要な要素

2.1 テスト自動化の価値

筆者のこれまでの経験を踏まえ，テスト自動化の価値をまとめると次の 5 点となる [7].

① 品質向上
テスト自動化によって，製品の品質が上がる．開発者がコードを追加する度に細かくテストを行うことで，追加・修正・削除したコードが既存機能に影響を与えていないかの確認ができ，品質向上に貢献する.

② 正確性向上
手動でテストするのに比べ，テストの品質を常に一定に保つことができるため，テストの正確性が上がる.

③ 生産性向上
テスト自動化と CI(Continuous Integration:継続的インテグレーション)環境を組み合わせることで，自動化されたテストをビルド作業と組み合わせて実施することができ，ソフトウェア開発活動の生産性が上がるだけでなく，プロセス改善を実現する.

④ リソースの柔軟性
テスト自動化によって，そのテストを実施していた人が他の作業に取り掛かれ，人にしかできない作業に注力することができる.

⑤ 品質の見える化
テスト自動化によって，自動実行されたテストの結果を集計することで品質状況を把握で

き，品質の可視化ができる．

これらのテスト自動化の効果は自動化を行えば必ず実現できるわけではなく，ソフトウェア開発と同様にテスト自動化プロセスの構築にソフトウェアエンジニアリングの知識を必要とする．

2.2 テスト自動化を品質管理視点で捉える

テスト自動化は一般的に機能テストの効率性を上げるための手段である．しかし，テスト自動化の効果を国際規格の品質モデル（図3）[8]で整理することで，機能適合性以外の効果を検討しやすくなる．

一般的に，テストの自動化の効果を品質特性毎にまとめると図 4 のようになる．機能仕様の確認により機能適合性が確保されることは当然であるが，テスト実施中の各機能の処理時間や負荷状況を計測することで性能効率性を担保するデータの取得が可能である．自動テストの結果レポートを集計・分析することで品質確保の状況や推移，弱点となる機能を見出すことができるため，信頼性の確保にも貢献する．

自動テストの結果により確保できる品質が分類できることで，自動テストのパス率とカバー率等の品質データを品質管理の観点で活用できるようになる．

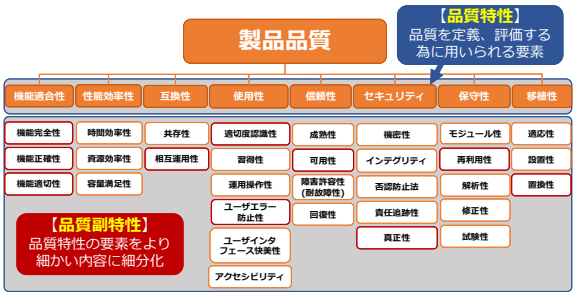

図 3．品質モデル

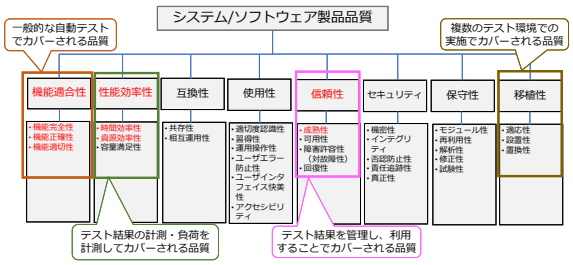

図 4．テスト自動化の効果

2.3 テスト自動化のガイドラインの整備

筆者の組織では，テスト自動化の実現や開発プロセスへの組み込みなどを様々な企業向けに支援している．テスト自動化に関する課題は企業毎に異なるが，これまでの筆者の経験を踏まえると，テスト自動化の悩みは次のとおり大きく 3 点に分類できる．

- 属人化の課題
- 運用・保守の課題
- 人材育成の課題

これら 3 点の課題の対策としてテスト自動化のガイドラインを作成し，プロジェクトメンバーはガイドラインを参照しながら自動化のタスクを実施している．

図 5 は Power Automate を使ったテスト自動化のプロジェクトで利用しているガイドラインの目次で 150 ページを超えるボリュームとなっており，このガイドラインの適用で属人化を排除している（図 5）．

また，ガイドラインとは別にフローや変数の命名規則を用意しており，ガイドラインとともに利用することで運用・保守の課題の発生を防いでいる．

人材育成の課題については教育用のシラバスを用意し，Power Automate による自動化の初心者向けの研修を整備するとともに，ガイドラインを参照したサンプル実装等の練習により，テスト自動化エンジニアの育成を行っている．

図 5．Power Automate を使ったプロジェクトのガイドライン例

ガイドラインは利用するテストツール毎に大枠を作成しており，それをプロジェクト毎にテーラリングして利用するとともに定期的に保守を行い，プロジェクトメンバーに共有している．

ガイドラインを整備することでテスト自動化の多くの課題は解決することができる．

しかし，テスト自動化の効果を長期的に維持し，自動テストのテスト結果を品質データとして品質管理で活用していくことを考えると，新たな課題に直面する．品質管理を継続するために開発組織としてテスト自動化の成熟度を計測し，改善していく仕組みが必要である．

この課題を解決するには，開発組織がプロジェクトにおけるプロセスを現状どの程度管理できているかをアセスメントし，そのアセスメント結果を踏まえて今後の改善策を示す CMMI(能力成熟度モデル統合) [9]のような成熟度モデルをテスト自動化の開発プロセスへ適用することである．そこで，筆者たちは品質管理での活用を踏まえ，この成熟度モデルの構築を始めた．

3. テスト自動化成熟度モデルの構築

3.1 モデル構築の準備

テスト自動化成熟度モデルは CMMI，テストプロセス改善のためのガイドラインおよび参照フレームワークとして TMMi Foundation によって開発された TMMi [10]，アセスメントの国際規格である ISO/IEC 33000 シリーズを参考に構築するものとした．特に CMMI を保管するモデルとして位置付けられている TMMi の成熟度モデル(図 6)を参考にテスト自動化成熟度のモデル化を進めた．

なお，モデル構築時はレベル判定のアセスメントができるように考慮した．

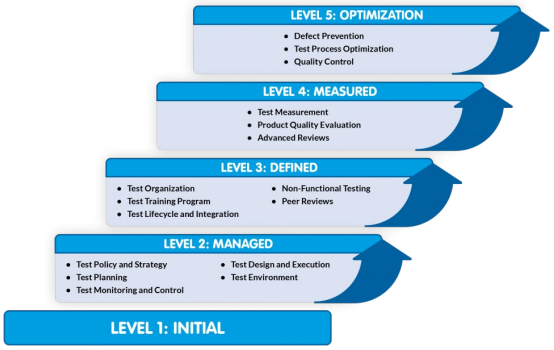

図 6. TMMi の成熟度モデル

3.2 構築したモデル

構築したテスト自動化成熟度モデルは 4 つのレベルを持ち，テスト自動化のプロセスの状態を示す形とした(図 7)．

また，次のレベルに到達するのに必要な要求事項を簡単に記載し，アセスメントを実施しなくても自組織が置かれているテスト自動化の状況が把握できることを心がけた．本モデルをツールベンダーが開催するテスト自動化の事例とともに紹介したところ，テスト自動化のレベル感を容易に把握することができるだけでなく，次のレベルに到達するための課題が見えるため，多くのセミナー参加者に好評を頂いた．

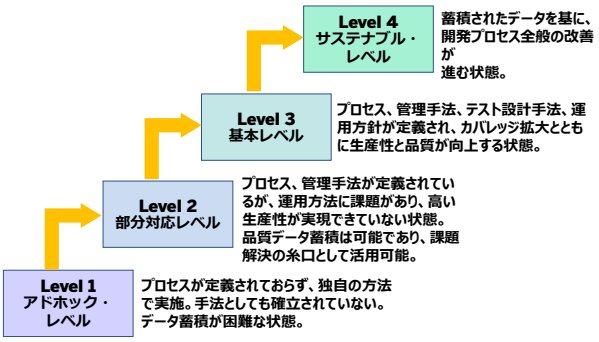

図 7. 構築したテスト自動化成熟度モデル

表 1 は各レベルの状態と次レベルに到達するために狙うべき状態である．

レベル 1 は自動化を行うエンジニアがルールなしで行っている状態，つまり属人化している状態である．この状態では作成した自動化スクリプトを他のエンジニアに引き継ぐことが難しいだけでなく，長期的に保守運用していくことは難しい．ソフトウェアコードの実装者が作成することの多いユニットテストのコードであったとしても属人化は防ぐべきである．また，テスト自動化の目的の明確化，チーム作業の意識，コーディング規約やテストの合否判定ルール等の最低限のルールを検討し，これらの情報をまとめたガイドラインを整備すべき状態である．

レベル 2 はガイドラインを整備し，単に手動テストを自動化することからテスト自動化の持つ品質改善や常にテストを一定の品質で実施できる効果などを意識した状態である．実装コードの

デグレード摘出のためにBVT(Build Verification Test)として活用するだけでなく，繰り返し実行できる運用環境の整備などが進んでいる状況である．

レベル 3 は，データ駆動テストと呼ばれる自動テストで使うデータを外部に持ち，反復テストを実施している状態である．データ駆動テストを実現するには自動化を意識したテスト設計とテスト実装が必須である．このような状態を実現するには開発組織全体またはプロジェクト全体でテスト自動化の効果を得るためにテスト自動化の目的が明確になっており，その目的を実現するための仕組み作りができている状態である．このような状態の組織では自動テストのカバレッジが上がるにつれ，より多くの効果を得られる状態であり，自動テストの結果を活用した品質の見える化，生産性の向上が実現できる．

最後にレベル 4 はレベル 3 の状態を維持しながら発生する課題を改善し，テスト自動化のプロセス改善が進んでいる状態である．この組織ではテスト自動化を行うことが当たり前であり，自動化の目的，テスト設計／実装／運用が属人化することなく共通認識の上で実施できている状態である．自動テストの結果は品質管理で活用され品質の見える化が実現できており，自動テストで発生する様々な課題は組織内で改善され，さらに自動化の効果を高めていくことが可能である．

表 1．各レベルの状態と次のレベル到達のために狙うべき状態

レベル移動	現状の状態	狙うべき状態
Level1(アドホック・レベル) から Level2(部分対応レベル)	自動化を担当するエンジニアが独自に設計･実装した状況から少しずつチーム開発へと移行している状況. 手動で実施しているテストをそのままスクリプト化することが多いため,自動化のメリットがあまり出ていない状態であり,生産性向上効果はほぼなし.	チーム作業として自動化を捉え，少しずつ自動化の目的と狙うべき目標が設定され，プロセスやスクリプト管理の課題が洗い出され，都度改善される状態.
Level2(部分対応レベル) から Level3(基本レベル)	開発進捗に合わせ,テスト設計･テスト実装を進めることができるようになり,デグレード摘出やビルドの品質チェックなどに活用できるようになった状態. テスト自動化による高い生産性を実現するには運用プロセスの確立を行うとともにテストカバレッジの拡大を進める必要がある.	社内でのテスト自動化の効果的な運用方法を検討していくこと，品質の見える化を少しずつ進めていくことでより高い生産性を得ることができる状態.
Level3(基本レベル) から Level4(サステナブル・レベル)	各プロジェクトで自動テストの目的と運用が明確になっており,品質の見える化を実現している状況.手動で行うテストと自動化するテストの切り分けがしっかりできているだけでなく,様々な運用および保守で自動テストの活用が行われている状況．高いレバレッジが得られる状態.	テスト自動化が当たり前の状況となっており，品質の見える化とともにプロセス改善が行われ，さらに自動テストの活用方法を見い出すことができる状態.

この成熟度モデルを活用するだけではレベル 4 の状態に到達することはできないため，CMMI や TMMi のようにアセスメントを行い，課題を洗い出し次のレベルに到達するための対策を検討する必要があると考えた．

4. アセスメント手法の構築

4.1 アセスメント手法の検討準備

アセスメントで使うチェックシートは成熟度モデルを構築するために参照した CMMI や TMMi だけでなく ISO 9001 の自己評価ツールとして品質マネジメントの状況を判断できるように活用することを目的に作成された国際規格［11]を参考に 8 つのカテゴリーの要求事項に対し，それぞれ 4～5 つの項目を洗い出した(表 2)．8 つのカテゴリーとしたのはテスト自動化は品質の定量評価を行う上で効果があり，ソフトウェア品質管理で活用して欲しいと考え，ISO 9001 に沿うようにテスト自動化プロセスの成熟度を評価したいと思ったためである．

表 2. アセスメント内容

#	カテゴリー	要求事項
1	目的・方針・価値	・ テスト自動化の目的は明示されているか ・ 目的の達成のためのクライテリアは設定されているか ・ 目的を実現するためのゴールは設定されているか ・ 目的達成のためのマイルストーンは設定されているか
2	ツール理解	・ 自動化の目的とツール理解 ・ ツールが提供する保守機能の理解 ・ ツールが提供する UI 情報の管理機能の目的と効果の理解 ・ ツールが提供する部品化機能の目的・効果・利用 ・ ツールが提供するレポート機能の理解と利用
3	テスト設計・実装	・ テスト設計の方針 ・ テスト設計レビューの方針 ・ テスト実装の方針 ・ テスト実装レビューの方針 ・ データドリブンの検討方針
4	ガイドライン・バリデーションルール	・ テスト自動化のガイドラインはあるか ・ 属人化しないルールは検討されているか ・ コーディングルールはあるか ・ バリデーションルールはあるか ・ ユーザーコード記載のルールはあるか
5	部品化・再利用性	・ 部品化のルールはあるか ・ スクリプトの再利用ルールはあるか ・ 部品化されたモジュールの利用ルールはあるか ・ 部品のライブラリ化ルールはあるか ・ テストスイートの利用/テストスイートでのライブラリの利用ルールはあるか
6	運用・並列運用	・ 自動テストの実施ルールはあるか ・ 自動テストの並列実施ルールはあるか ・ 自動テストの運用ルールはあるか ・ 自動テストの運用ルールに実施ルールはあるか ・ 自動テストの運用に並列実施ルールはあるか
7	プロセス	・ プロセスの決定 ・ プロセスの責任 ・ プロセスの権限 ・ プロセスのマネジメント ・ プロセスの維持

8	改善	・ パフォーマンスの評価 ・ パフォーマンスの分析 ・ パフォーマンス指標 ・ プロセスの改善

4.2 チェックシートの作成

アセスメントを行うためには要求事項の各項目に設問を用意し，スコア化できるようにする必要がある．各項目に 11 の設問を用意し，設問に解答していくことで要求事項の各項目をスコア化できるようにした(図 8)．

項番	アセスメント内容	スコア	判断結果	判断で利用した文書	判断理由
1-1	テスト自動化の目的は明示されているか				
1-1-1	明示されておらず各自で決めている	1			
1-1-2	プロジェクトの一部のメンバー内で目的が共有されている	2			
1-1-3	プロジェクトの一部のメンバー内で目的が共有されており、PM・PLなども理解している	2			
1-1-4	プロジェクトメンバー内で目的が共有されている	3			
1-1-5	プロジェクトメンバー内で目的が共有されており、PM・PLの方針であることが共有されている	3			
1-1-6	プロジェクトメンバー内で目的が共有されており、プロジェクト計画書などに明記されている	3			
1-1-7	プロジェクトメンバー内で目的が共有されており、プロジェクト計画書などに明記されており、担当する体制が明記されている	3			
1-1-8	テスト自動化に関するプロセスが規定されており、そのプロセスに基づき、プロジェクトメンバーに目的の共有とプロジェクト計画書などに明記されている	4			
1-1-9	プロジェクトメンバーだけでなく、品質管理チームなどのプロジェクト支援を行うメンバーに目的が共有され、プロジェクト計画書などに明記されている	4			
1-1-10	プロジェクトメンバーだけなく、品質管理チームなどのプロジェクト支援を行うメンバーに目的が共有されており、プロジェクト計画書などに明記されており、目的が第3者レビューされている	5			
1-1-11	テスト自動化に関するプロセスが規定されており、そのプロセスに基づき、プロジェクトメンバーだけでなく、品質管理チームなどのプロジェクト支援を行うメンバーと目的の共有されており、プロジェクト計画書などに明記されており、プロジェクト計画書などに明記された目的が第3者レビューされている	5			

図 8. 要求事項の設問例

成熟度モデルは 4 段階であるが，プロセスと改善のカテゴリーは ISO 9001 を意識しているため，チェックシートのスコアは ISO 9004 の附属書 A を参考に 1～5 としている．最終的にスコアの合計値で今回構築したテスト自動化成熟度モデル 4 段階に合うように修正し， 8 つのカテゴリーのスコアをレーダーチャートで表示することを考慮した．

5. アセスメント実施

作成したアセスメントチェックシートを用いて，テスト自動化の支援を行っているプロジェクトの開始時のヒアリング結果を参考に適用実験を行った．チェックシートの設問でスコア化することを考慮していたが，実際に各設問の判定を行うには全カテゴリーの要求事項の判定を行った後に，結果を再度見直す必要があることが分かった．これはガイドラインの有無やプロセス化の状況によって関連する設問の結果を見直す必要があるためである．たとえば，プロジェクトや組織にガイドラインがあったとしてもそのガイドラインを一部しか活用していなかったり，一部の人が守っていなかったりすることを考慮し，全体的にスコアを見直す必

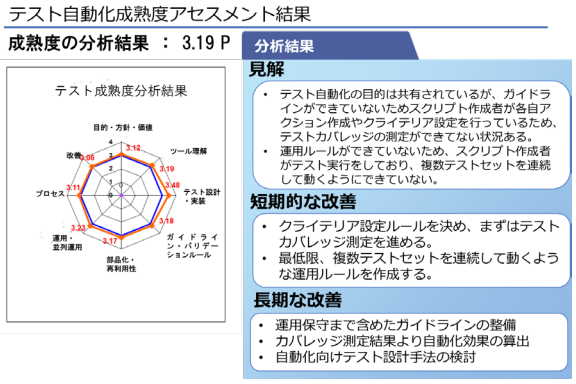

図 9. アセスメント結果の例

要がある．

また，アセスメントの結果から成熟度モデルのどこに位置付けるかについても判断が難しいことも判明したが，ガイドラインやバリデーションルールの有無でレベル 2 かレベル 3 かを判断することが望ましいと判断した．

実際のアセスメントの結果は図 9 のようになり，アセスメントによるスコアだけでなく，現状の状態を示す見解と課題については短期的と長期的に分けて記載することでアセスメントを行った組織が対策を検討しやすいようにした．

6. 結論

テスト自動化の効果をあげるためにガイドラインの整備や開発プロセスの改善を行っている組織は多いが，現状を分析する仕組みがないと次の目標を立てにくいだけでなく，組織としてどうあるべきかを判断することが困難である．この課題を解決するためにテスト自動化成熟度モデルを立案するとともに，チェックシートとヒアリングによるアセスメント手法を構築した．

本アセスメント手法はまだ開発したばかりであり，利用しているツールによって読み替える設問の準備やチェックシートを改善するためのデータ蓄積を行っている状況である．そのため，まだ PoC 段階であり，現時点では実際の効果を明確に判断するには時期尚早である．

しかし，ソフトウェア開発の生産性向上と自動テストの結果による品質の見える化を実現するテスト自動化の効果は非常に大きい．この効果を経営視点として開発組織の成熟度を判断する上で，テスト自動化成熟度アセスメントを実施し，次なる課題を明確にすることは大いにメリットがあると考えられる．また，コンサルテーションサービスを行う筆者の組織としても顧客にアセスメントを基に支援内容を提案できる効果は高く，顧客としても支援の必要性が理解しやすいといえる．

より多くの開発組織に本アセスメントを実施し，データを蓄積しながら，成熟度モデルとアセスメントチェックシートの改善を継続し，最終的に本アセスメントチェックシートを公開し，より多くの組織が自らアセスメントができるようにガイドラインの整備も含めて検討したい．

7. 参考文献

[1] アジャイルソフトウェア開発宣言(2001)， https://agilemanifesto.org/iso/ja/manifesto.html .

[2] DevOps， 10+ Deploys Per Day: Dev and Ops Cooperation at Flickr. 2009. https://www.slideshare.net/jallspaw/10-deploys-per-day-dev-and-ops-cooperation-at-flickr .

[3] Cohn， M. (2009): The Forgotten Layer of the Test Automation Pyramid, https://www.mountaingoatsoftware.com/blog/the-forgotten-layer-of-the-test-automation-pyramid/ .

[4] Ranorex， https://www.ranorex.com/ .

[5] UFT One， https://www.opentext.com/ja-jp/products/uft-one .

[6] Parry， O.， Kapfhammer， G.， Hilton， M.， Mcminn， P. (2021): A Survey of Flaky Tests, ACM Transactions on Software Engineering and Methodology， Vol. 31， No. 1， Article 17.

[7] DevOps ユーザー会 2022，セッション 1「テスト自動化を意識したテスト設計のとっかかり」，加藤大受， https://www.ashisuto.co.jp/seminar/report/detail/devops-user-kai_20221209.html .

[8] JIS X 25010:2013 『システム及びソフトウェア製品の品質要求及び評価（SQuaRE）−システム及びソフトウェア品質モデル』，日本規格協会.

[9] CMMI(Capability Maturity Model Integration: 能力成熟度モデル統合), https://cmmiinstitute.com/ .

[10] TMMi(Test Maturity Model integration: テスト成熟度モデル統合)， https://www.tmmi.org .

[11] JIS Q 9004:2018 『品質マネジメント－組織の品質－持続的成功を達成するための指針』， 日本産業規格.