

アジャイルテスト実践 改善

AGILE TPI

Version 2.4 March 2017



バージョン情報

Version	Date issued	Remarks	Author
0.1	September 5, 2014	Initial setup and concept	Gerrit de Vries
0.2	September 24, 2014	Added contributions from: - Yvonne van Zaanen (Triodos) - Jos Punter (Sogeti NL) - Bert Linker (Sogeti NL) - Ben Visser (Sogeti NL)	Gerrit de Vries
0.3	September 25	Added contributions from: James Redpath (Sogeti USA)	Gerrit de Vries
1.0	December 2014	Final version	Gerrit de Vries
1.1	February 2015	Minor adjustments	Gerrit de Vries
1.2	April 2015	Added contributions from Arne Bergs (Sogeti DE)	Gerrit de Vries
1.3	July 2015	Synchronized with the ATIM tool	Gerrit de Vries
1.4	August 2016	ATIM changed to Agile TPI	Gerrit de Vries
2.0	August 2016	Synchronized with Agile TPI Tool Minor changes Input from Eric Belt and Ben Visser	Gerrit de Vries
2.4	March 2017	Some checkpoints adapted Synchronized with Tool 2.4 Input from Eric Belt and Ben Visser	Gerrit de Vries

概要

アジャイル、スクラム、TPI

アジャイル方法論は、テスト駆動開発や探索的テストといったテストの見地から、多くのすばらしい実践例を前面に押し出してきた。同時にプラクティスは、テストの全体的なアプローチやプロセスがアジャイルプロジェクトにおいてはまだまだ満足いくものとはいえないことを示している。

アジャイル方法論においても、ウォーターフォール開発と同様、テスト活動やテストプロセスの改善の必要がある。この側面については『TPI NEXT®ビジネス主導のテストプロセス改善』¹にも詳細に書かれており、「7.2 アジャイル開発手法への適用」のところで、アジャイル開発でどのようなモデルが使用できるかを説明している。

しかしながら、我々はアジャイル方法論を実践し、使用することに対して組織的な抵抗があることを学んだ。多くの状況では、「名ばかり」アジャイル、あるいはデイリースタンドアップ（会議）のようにスクラムの特定要素だけ適用したものにみえる。我々はまた、プロフェッショナルテスト担当者はその技能、そしてなんといっても経験を持ってアジャイルチームに貢献できるということも知った。テスト担当者としての仕事を通じて、彼はいわば「本拠地」となり、アナリスト、開発者、ユーザーと密接に協働する。この立場から、テスト担当者はテスト作業のメリットを他者に納得させ、他のチームメンバーや利害関係者とともに品質指向のマインドセットを作ることができる。

アジャイル環境におけるテスト経験とテスト活動を基に、本ドキュメントは「アジャイルソフトウェア開発宣言」（以下参照）を組織、プロジェクトチーム、そして個人のメンバーが適用するにあたって役立つと我々は考えている。

私たちは、ソフトウェア開発の実践あるいは実践の手助けをする活動を通じて、

よりよい開発方法を見つけだそうとしている。この活動を通して、私たちは以下の価値に至った。

プロセスやツール **よりも** 個人と対話を、

包括的なドキュメント **よりも** 動くソフトウェアを、

契約交渉 **よりも** 顧客との協調を、

計画に従うこと **よりも** 変化への対応を、

価値とする。すなわち、左側のことがらに価値があることを認めながらも、私たちは右側のことがらにより価値を置く。

「アジャイル」はさまざまな意味合いを持つが、我々はこのドキュメントではスクラムの用語を用いる。というのはスクラムが広く一般に用いられている方法論だからである。さらに、我々は「TMap NEXT® in Scrum」²を参照するが、そこでは著者はスクラム方法論を基本的な出発点として用いている。

開発者がテスト担当者か

理想的な状況では、スクラムチームは特に専門や役割を持たない開発者で構成される。すなわち、スクラムでは開発チームの中にサブチームを認識しない。テスト作業やビジネス分析のような解決する必要のある特定のドメインがある場合は別にして、このルールには例外はない。³

しかし、現実的な状況では、特にテスト担当者というところで、開発者は特別の役割を持っているようにみえる。スクラムガイドに刃向かうつもりはないが、実践的な状況でチームビルディングを行いながら、チーム内のテスト活動を支援し、改善したい。さらに本ドキュメントでは、我々は各チームメンバーが正しいやり方でテスト活動を行う能力があるという状況を目指している。つまり、テスト担当者を名乗るならば、それはテスト活動をするチームメンバーを意味する。スクラムチームにおけるテスト作業はチームの活動である。すなわちチームとしてテストシナリオを実装し、チームとして何を、どこを、そしてどうやって（マニュアルか自動か）テストすべきかを決定する。その活動から発生し、スクラムボードに表示されるタスクは、チーム内の誰でも選び取ることができる。

¹参照: TPI NEXT® - Business Driven Test Process Improvement; Gerrit de Vries et al. – UTN Publishers 2009.

²参照: TMap NEXT® in Scrum; Leo van de Aalst en Cecile Davis, Sogeti Nederland B.V. 2012.

³参照: The Scrum Guide, July 2013

読み方

次章ではテスト実践の改善の世界とアジャイルテスト改善へのアプローチ方法を紹介する。第 3 章はテスト担当者（またはテスト活動を行う者）としてのスキルに焦点を当てる。第 4 章では TPI NEXT®モデルのチェックポイントを元に、どのプラクティスがテスト活動に適用できるのかを述べる。これは 3 つの運用観点として扱われる。

- P: 個人レベルの開発者のためのテスト実践（Professional）
- T: チーム（Team）全体としてのテストプラクティス
- O: 組織（Organization）、プログラムまたはプロジェクトレベルのテストプラクティス

本ドキュメントは Appendix も含まれている。これは『TPI NEXT®ビジネス主導のテストプロセス改善』第 4 章の本文である。キーエリアまたは関連したチェックポイントに関する情報を入手するための参考資料として使用することができる。

ツール

このドキュメントには、プラクティスと組織をアセスメントし分析するのに使うことのできるアジャイル TPI ツールが付属する。

アジャイルの状況で TPI NEXT ツールを使ったが、アジャイル TPI ツールがあるので、簡単にテストのボトルネックや改善への示唆を得ることができるはずである。

このツール（とユーザーマニュアル）は www.tmap.net からダウンロードできる。

Appendix のユーザーマニュアルを参照するとツールを正しく使用できる。

目次

1	イントロダクション	1
1.1	改善の必要性?	1
1.2	区切りはあるが分離しないテスト活動	1
1.3	異なった視点とレベル	1
2	アジャイル TPI モデル	3
2.1	キーエリア	3
2.2	カテゴリ	3
2.3	チェックポイント	4
2.4	優先度の設定	4
2.5	クラスタ	5
3	改善のための戦略	6
3.1	己の立ち位置を決定せよ	6
3.2	スクラッチから始める	6
3.3	アジャイルとウォーターフォールのハイブリッド	6
3.4	クラスタによる段階的改善	6
3.4.1	初めのステップ: クラスタ A	6
3.4.2	次のステップ: クラスタ B	7
3.4.3	クラスタ C	8
3.4.4	クラスタ D	9
3.4.5	クラスタ E	10
3.4.6	クラスタ F	11
3.4.7	クラスタ G	12
4	テストの潜在能力の改善	13
4.1	テスト技術にフォーカス	13
4.1.1	イントロダクション	13
4.1.2	技術的なスキル	13
4.1.3	ソフトスキル	13
4.1.4	コーチとしてのテスト担当者	15
5	Appendix A: キーエリアとその重要性	17
6	Appendix B: AGILE TPI Tool のガイドライン	20

1 イントロダクション

1.1 改善の必要性?

テスト活動は各ソフトウェア開発プロジェクトの中で重要な役割を果たすが、いわゆるウォーターフォール開発環境であるか、よりアジャイルでイテレーティブな方法が採られる環境であるかを問わない。ソフトウェアプログラム、システム、または完全なエンドツーエンドの IT ソリューションであっても、組織には欠陥や不備のリスクがある。それがテストをする主な理由の一つである。テスト活動のもう一つの重要な原動力は、IT プロダクトにおいて、ユーザーとビジネスの観点から確信と信用を得ることである。ゆえに、「テスト」はプロダクトとソリューションの品質への深い洞察を生み出す必要があるのである。

そういう状況とは別に、効率的なテストを求める組織はますます増えている。必要な品質レベルに応じて、最小限の工数、時間、お金、リソースでテスト活動の結果が作られる。TMap®用語では、正しい道具を使い正しい時間に正しいやり方をする、リスクベースと人間駆動テスト⁴を意味する。

1998 年以来、組織は、テストの成熟度をアセスメントし、可能な改善をするために TPI NEXT® モデルを使用してきた。2009 年には、我々はもっとビジネスドリブンな方向にこのアプローチを拡大し、アジャイルのような環境にフォーカスするために、TPI NEXT®モデルを提供した。TPI NEXT®本はこれらの環境におけるモデルの使用を記述したセクションを提供している。このセクションの推奨が依然として妥当ではあるが、まだ改善の余地があると我々は感じている。少なくとも組織の中には未だにアジャイル環境におけるすばらしいテスト活動の利益を享受しようと戦っているものがあるからである。

1.2 区切りはあるが分離しないテスト活動

アジャイルテスト活動の改善に関する記事を提供することは自己矛盾のようにも聞こえる。

- “スクラム”⁵は、個人によって行われる仕事のいかんに関わらず、「開発者」以外のロールを開発チームメンバーに求めない。このルールに例外はない。
- スクラムは、テスト作業やビジネス分析のように解決すべき特定ドメインがあるかどうかに関わらず、開発チームにサブチームを作らない。このルールに例外はない。
- 個人の開発チームメンバーは専門スキルやフォーカスする領域があってもいいが、説明責任は開発チーム全体に属する。

ゆえに、テスト担当者、テストエンジニア、テストマネージャというロールはいうに及ばず、「テストプロセス」自体が存在しないといえるかもしれない。それでもなお、テスト「プラクティス」は存在するのである。というのも、テスト活動は他の開発活動と区別できるため、必要ならアセスメントを行い、評価し、改善することができるからである。さらに、先ほど触れた「専門スキル」は注目に値する。テスト担当者としての数年に及ぶ経験から、テスト担当者のタスクとアクティビティは、最終製品の品質にとってきわめて重要なだけでなく、「オペレーションの中心」の役割を果たす。つまりテスト担当者は要求、設計、中間成果物、ユーザーサポートと受け入れサポート、これらすべての品質に影響する。（参照：コーチとしてのテスト担当者）

この職責はテスト担当者（またはむしろテスト活動を行う開発者）に同僚の開発者たちをコーチするユニークな機会を与えるだろう。テスト担当者の経験から、自身の活動のみならず、チーム全体の活動を改善することができる。このため、アジャイル環境におけるテスト活動を区別したのである。それは、テスト活動を切り離したりするのではなく、他の開発活動と可能な限り同程度の強さを持たせたいからである。

1.3 異なった視点とレベル

アジャイル環境におけるテスト活動の改善は異なるレベルと異なるスタート地点から始められる。さらに、多くの組織は何か他の方法論からアジャイルへの転換への途中である。ここで注目したいのが、テスト活動の運用観点である。

- 一人またはそれ以上の**個人の**チームメンバーで行うテスト活動。
- 一つまたはそれ以上の**チーム**で行うテスト活動。

⁴ 詳細は www.tmap.net の TMap Human Driven testing を参照のこと。

⁵ From: The Scrum Guide, July 2013

- **組織的**観点から行われるテスト活動。

以下の段落では、改善とガイドのために TPI NEXT®モデルの要素を適用するのに、これらの区別を使う。

2 アジャイル TPI モデル

オリジナルの TPI NEXT®モデルはアジャイルテスト活動の改善を推奨する仕組みをつくる時の出発点として使われてきた。オリジナルモデルとの主な違いは成熟度レベルである。ある成熟度レベルが測定・改善可能として、前述した運用ビュー（個人・チーム・組織）を用いることにした。さらに、段階的改善を行うためにクラスタリングの方法が使われている。

2.1 キーエリア

オリジナルのモデルと同様に、異なるテストアクティビティと観点をグルーピングし、16 のキーエリア⁶に分けている。これらのキーエリアは 3 つすべての運用観点到に適用できる。

		個人レベル			チームレベル			組織レベル		
01	利害関係者のコミットメント									
02	関与の度合い									
03	テスト戦略									
04	テスト組織									
05	コミュニケーション									
06	報告									
07	テストプロセス管理									
08	見積もりと計画									
09	メトリクス									
10	欠陥管理									
11	テストウェア管理									
12	手法の実践									
13	テスト担当者のプロ意識									
14	テストケース設計									
15	テストツール									
16	テスト環境									

2.2 カテゴリー

このモデルと特に成熟度マトリクスにおいては、運用観点と運用カテゴリーがオリジナルの成熟度レベルと置きかえられている。オリジナルのマトリクスには 4 つの成熟度レベル（初期レベル、コントロールレベル、効率化レベル、最適化レベル）があった。アジャイル TPI モデルにおいては初期レベルが取り除かれ、他の 3 つのレベルに置きかえられている。（次の図を参照のこと）

- 個人レベル（P）
- チームレベル（T）
- 組織レベルまたはプロジェクトレベル（O）

アジャイル TPI モデルにおいては成熟度が示され、成熟度は運用観点ごとに示され、成長する（当然キーエリアごとにも）。達成され充足されたチェックポイントに従って、レベルはパーセンテージで示される。（例：個人レベル 55%）

		個人レベル	チームレベル	組織レベル
01	利害関係者のコミットメント			
02	関与の度合い			
03	テスト戦略			
04	テスト組織			
05	コミュニケーション			
06	報告			
07	テストプロセス管理			
08	見積もりと計画			
09	メトリクス			
10	欠陥管理			
11	テストウェア管理			
12	手法の実践			
13	テスト担当者のプロ意識			
14	テストケース設計			
15	テストツール			
16	テスト環境			

⁶ 詳細は Appendix A 参照のこと

2.3 チェックポイント⁷

ある観点からの成熟度を評価するため、そして改善提案を提示するために、アジャイルに適合させた TPI NEXT のチェックポイントが使用されている。下図においてチェックポイントはキーエリア（1 番目）、運用観点（2 番目）、通し番号（3 番目）として示される。
例：1.P.1：1 番目のキーエリア（利害関係者のコミットメント）に属する個人レベル（P）の 1 番目のチェックポイント

		個人レベル				チームレベル				組織レベル									
01	利害関係者のコミットメント	1	2		3		1		2		1								
02	関与の度合い	1	2	3	4		1				1		2						
03	テスト戦略	1	2	3	4		A		C		C		A		C				
04	テスト組織	1	2	3	4		D				F		D		G				
05	コミュニケーション	1			1		1		2		1								
06	報告	1	2		3		1		2		3		1						
07	テストプロセス管理	1	2		3		1		2		3		1		2				
08	見積もりと計画	1			2		1		2		1								
09	メトリクス	1				1				1				2					
10	欠陥管理	1	2	3		1		2		3		4		1		2			
11	テストウェア管理	1			2		1		2		1				2				
12	手法の実践	1				1				1				2		3		4	
13	テスト担当者のプロ意識	1	2	3	4		1				1				2				
14	テストケース設計	1	2	3		1		2		3		1							
15	テストツール	1			2		1				1				2		3		
16	テスト環境	1	2	3		1				1				2		3			

2.4 優先度の設定

チェックポイント間、または推奨事項間で、優先度が異なることがあるかもしれない。つまり、あるチェックポイントは他のチェックポイントよりも重要である。チェックポイントに対して優先度を設定する理由の一つは、このモデルが「ビジネス主導」⁸という側面を持つからである。アジャイル環境においては、多かれ少なかれアジャイルを強化するのに貢献しそうなキーエリアの重要度は以下のようになる。

	キーエリア
01	利害関係者のコミットメント
02	関与の度合い
03	テスト戦略
04	テスト組織
05	コミュニケーション
06	報告
07	テストプロセス管理
08	見積もりと計画
09	メトリクス
10	欠陥管理
11	テストウェア管理
12	手法の実践
13	テスト担当者のプロ意識
14	テストケース設計
15	テストツール
16	テスト環境

	高
	中
	低

⁷ AGILE TPI tool のテスト成熟度についてはチェックポイントがクラスタを示すアルファベット（A,B,C, ...）に置き換えられている。

⁸ 参照: TPI NEXT® Business Driven Test Process Improvement, Appendix B3.1. Categorized Key areas.

チェックポイントに対して優先度を設定するもう一つの理由は、チェックポイント相互の依存関係である。チェックポイントによっては、他のチェックポイントも満たす必要がある。

この優先度付けはクラスタという形で形式化する。

2.5 クラスタ

クラスタは特定のキーエリアと限られたチェックポイントの組み合わせである。初期レベルから始まって、より高い成熟度へと、改善の段階が A（第一段階）、B（第二段階）、…というように示されている。これは下図で示されているが、すべての 3 レベルのうち第一のクラスタ（A）が緑色で示されている。

		個人レベル				チームレベル			組織レベル			
01	利害関係者のコミットメント	C	D		D	C	E		C			
02	関与の度合い	A	A	B	C	A			A	B		
03	テスト戦略	A	B	C	D	A	C	C	A	C		
04	テスト組織	D	E	F	G	D		F	D	G		
05	コミュニケーション	A		C		B		D	C			
06	報告	D	F		F	D	E	F	E			
07	テストプロセス管理	F	F	G		B	B	E	E	F		
08	見積もりと計画	B		B		A		B	F			
09	メトリクス	B				F			G	G		
10	欠陥管理	C	D	E		A	C	C	D	D	E	
11	テストウェア管理	C		C		E		F	D	E		
12	手法の実践	B				C			A	B	B	C
13	テスト担当者のプロ意識	A	B	B	E	A			A	A		
14	テストケース設計	C		B	B	A	B	C	C			
15	テストツール	A		A		A			A	C	B	
16	テスト環境	A	C	C		A			A	A	A	

優先度の設定

各カテゴリにはクラスタのセットがあり、各クラスタにはそれぞれ異なるチェックポイントがある。

つまり、チームレベルのクラスタ A は個人レベルのクラスタ A とは異なる。クラスタは優先度の指標を与える。例えば、何を最初にして、何をあとにするか、といったことである。実践においては、異なる運用観点からチェックポイントの組み合わせを作ることで改善が実現される。

3 改善のための戦略

3.1 己の立ち位置を決定せよ

スタート位置を決定するのは常にお勧めである。現在までどのようにテストされているか、既知または隠れたボトルネックは何か、どの活動がうまくいっているか、迅速に勝ち得られそうなものは何か。アジャイル TPI アセスメントツールを使うと、すべてのチェックポイントの助けを借りてテスト成熟度を可視化することによって、このスタート位置を非常にうまく測定できる。アセスメントを使用することはテスト成熟度についての深い洞察を提供するだけではない。他の強みとの依存性を示し、短期と長期の改善への優先順位付けを可能にするのである。何を先に行い、何が次に来るのか（クラスタを参照のこと）。

3.2 スクラッチから始める

テスト活動をアジャイルチームの中で組織化し確立する必要がある時、すなわちチーム、プロジェクト、組織がすべてアジャイルに対して初心者であり、アジャイルがすべてのプロジェクトに対して用いるアプローチである場合には、アジャイル TPI モデルに広い焦点を当てて用いることをお勧めする。この戦略は、一般的なアプローチに注意を払いながら、3 つのすべてのカテゴリにおけるチェックポイントと改善ステップを統合する。すべてのカテゴリ（個人レベル（P）、チームレベル（T）、組織レベル（O））における第一の（A）クラスタ⁹を用いて状況のアセスメントを始めよう。そしてすべてのカテゴリのクラスタも参考にしながら、改善に必要なステップを決定する。改善ステップを網羅しながら改善計画を立てよう。そしてそれを見積もり、計画されたアクティビティ、アクション、タスク、責任事項に落としていく。

3.3 アジャイルとウォーターフォールのハイブリッド

ウォーターフォールとアジャイルの両者が混在している状況ではウォーターフォールのための TPI NEXT®モデルとアジャイルのための Agile TPI モデルを使用することを推奨している。両モデルのアセスメント結果を比較することで、何が必要な改善として求められているかという十分な判断材料を得ることができる。さらに、現実的な見積もりと計画に基づいた詳細なアクションプランに従って、両方の環境での改善アクティビティを実現してみよう。2 つのモデルを組み合わせ比較することで、お互いに学び、お互いのベストプラクティスを使うことが可能になる。マネジメントが両方、または片方の方法論に対して必要な変革と改善を示し、優先順位付けすることに対してコミットしているか、確認してみよう。

3.4 クラスタによる段階的改善

アジャイル環境におけるテスト活動の改善はコントロールされた方法で段階的に行わなければならない。それがまさにクラスタが使える場所である。改善ステップは初期レベルから始まりより高い成熟度になっていくと A（第一段階）、B（第二段階）、…のように示されることとなる。

改善の第一歩は、クラスタ A に属するさまざまなキーエリアのチェックポイントで構成される。改善の計画と戦略の前提として、改善のステップが実現可能であるかどうかを定義する必要がある。というのも、時には改善のためにチェックポイントを少なくして、ステップを小さくする必要もあるからである。次の段落はクラスタを用いた段階的改善を表す。

3.4.1 初めのステップ: クラスタ A

キーエリア	NR.	チェックポイント
関与の度合い	2.1.1	プロダクトリスク分析、計画、準備、テストケース設計のようなテスト活動は、テスト活動をスプリントのクリティカルパスとならないように、テスト実行よりも前に行われている。

関与の度合い	2.1.2	テスト担当者/開発者がテスト活動と他の活動を調整するために、プロジェクトおよびスプリントプランニングに参加している。
関与の度合い	2.2.1	開発者/テスト担当者が、テスト対象を記述するテストベースの最適化に（テストバリエーションのレビューにとどまらず）貢献している。
関与の度合い	2.3.1	見積もったテスト活動から学んだことが（一連の）将来のプロジェクトに生かされている。
テスト戦略	3.1.1	プロダクトリスクがプロジェクト期間に定義され管理されている。例：ストーリーのリスクポーカーによるリスク見積もり
テスト戦略	3.2.1	テスト戦略はテスト期間中モニタリングされ、必要に応じて適用している。
テスト戦略	3.3.1	プロダクトのリスク分析とリスク戦略のプロセスが、ふりかえりの中またはユーザーストーリーの詳細化の中で評価されている。
コミュニケーション	5.1.1	開発者/テスト担当者個人が、関連する情報を授受するためにデイリースタンドアップに（先を見越して）積極的に参加している。
テスト担当者のプロ意識	13.1.1	テストの役割を担う開発者が、特にテストエンジニアリングとテストケース設計に関するテストのトレーニングを受けている。
テスト担当者のプロ意識	13.2.1	チームにとって要求される専門知識 — 業界、ビジネス、かつ/または技術的 — が必要に応じて入手できる。
テストツール	15.1.1	回帰テスト自動実行のためのテストツールが使用されている。
テストツール	15.1.2	ツールは利用可能な状態にあり、ずっとではないにせよ必要なタイミングで使うことができる。
テストツール	15.2.1	（実装した）テストツールの正しい用法のトレーニングが提供されていて、すでに受けている。
テストツール	15.3.1	使用されているテストツールのメンテナンス、知識、サポートが利用可能である。
テスト環境	16.1.1	（自動化）テストケースがすべての関連する環境上で実行可能である。
テスト環境	16.1.2	要求されるテスト環境設定とテストデータがイテレーションの初めに明示的に定義されている。

3.4.2 次のステップ: クラスタ B

キーエリア	NR.	チェックポイント
関与の度合い	2.1.3	品質とテストという側面が、プロジェクト全体とスプリントのリスクの分析と軽減を進める中で、チームとプロダクトオーナーによって十分に考慮されている。
関与の度合い	2.3.2	トレーニングとコーチング設備がスプリント活動開始前から提供されている。
テスト戦略	3.1.2	テスト活動とテスト戦略（再テストと回帰テストを含む）がプロダクトリスクに基づいている。
コミュニケーション	5.2.1	関連情報（プロダクト品質とリスクなど）が定義され、チームと利害関係者の間で効果的にやりとりされている。

テスト担当者のプロ意識	13.1.2	TMap または ISTQB といったテスト手法の要点が開発者に周知され適用されている。
テスト担当者のプロ意識	13.1.3	個人の知識やスキルがチームに共有されている。例：他者にテストを任せる
テスト担当者のプロ意識	13.3.1	TMap NEXT® または ISTQB® といったテスト手法についてのトレーニングと資格認定の制度が組織によって提供されている。
テストツール	15.3.2	ツールの使用が、テストをより速く、より安く、よりよくすること、または、テストプロセスをより管理しやすくすることに対して評価されている。
テストツール	16.2.1	環境の利用可能性について外部サプライヤとの労働契約が十分に定義され調整されている。
テストツール	16.3.1	テスト環境の論理的または機能的な設計が組織によって提供され維持されている。アプリケーション、システム、コネクション、スタブとドライバ（モック）を規定している。

※訳注：下記の灰色のチェックポイントは、Agile TPI tool（2.5）に質問項目があるが本ドキュメントには記載されていないため補足したものである。

3.4.3 クラスタ C

キーエリア	NR.	チェックポイント
利害関係者のコミットメント	1.1.1	プロダクトオーナーがすべてのチームメンバーと交流し、よく知られている。
利害関係者のコミットメント	1.2.1	プロダクトオーナーがプロダクトリスク分析（テスト戦略のためのインプット）のオーナーである。
利害関係者のコミットメント	1.3.1	組織が（トレーニング、コーチング、分科会のような）テスト活動のリソースの最適化をしている。
テスト戦略	3.1.3	戦略は全体とプロダクトオーナーで合意されている。
テスト戦略	3.2.2	インターフェースと IT（統合テスト）が他のチームとともに計画され議論されている。
テスト戦略	3.2.3	リスクに基づき、一つ以上のテスト技法がテストケース設計に使われている。
テスト戦略	3.3.2	テスト後または市場で起こったインシデントが分析されテスト戦略に反映されている。
コミュニケーション	5.1.2	チーム外の利害関係者からの関連情報が、特にスクラムオブスクラムの状況下で、積極的に収集されている。
コミュニケーション	5.3.1	（手段としての）コミュニケーションとその効率性に関係したベストプラクティスや学んだことが、定期的に評価され、改善が実施されている。
メトリクス	9.1.1	情報とデータが収集され分析されている： （テストケースごとの）テスト設計と実行に要した時間。 1 日/1 スプリント/1 プロジェクトで準備、設計、実行されるテストケース数。 1 日/1 スプリント/1 プロジェクトのチーム活動合計に対するテスト活動の時間。

メトリクス	9.2.1	ふりかえり時に、スプリントを通して使用されるメトリクスが定義、議論、評価されている。
メトリクス	9.3.1	アジャイルチームをサポートするメトリクスを収集、蓄積、提供、管理する手段と手続きが文書化されている。
欠陥管理	10.1.1	欠陥のライフサイクルが可能な限り短い。欠陥はそれが埋め込まれたスプリント中に解決されるか、よりよいのは、リファインメントとプランニングあるいはユーザストーリーを作る間に解決される。
欠陥管理	10.2.1	欠陥の扱いは個別の役割に（開発者だけではなくプロダクトオーナーとスクラムマスターにも）アサインされている。
テスト担当者のプロ意識	13.3.2	テストコミュニティに参加したり、テストセミナーを受講したり、またはテスト文献を読んだり投稿したりすることができる状況がある。
テストケース設計	14.1.1	プロダクトリスクと戦略に基づいて熟考された選択がなされ、テンプレートやツールを使い、一定の論理的なレベルで文書化されている。
テストケース設計	14.2.1	テストベースのカバレッジレベルが明確に示されている（例えば要件定義のドキュメント内）。
テストツール	15.3.3	すべてのアジャイル活動のためにツールのポリシーが定義されている。
テストツール	16.1.3	事前に定義されたチェックリストが環境の（可用性と機能性）を確認するために用いられている。
テストツール	16.3.2	テスト環境のテクニカルデザインが入手可能で、テスト管理者またはテスト環境のスペシャリストによって公式に承認されている。
関与の度合い	2.1.4	リファインメントの段階で、変更（優先度、ユーザストーリー、機能的または技術的な解決法）の影響（リスク）に対する品質とテストの観点が多分に考慮されている。

3.4.4 クラスタ D

キーエリア	NR.	チェックポイント
利害関係者のコミットメント	1.1.2	プロダクトオーナーの自発的なタスクと責任が多分に果たされている。
利害関係者のコミットメント	1.1.3	すべての関連する利害関係者が定義されすべてのチームに周知されている。
利害関係者のコミットメント	1.3.2	（テストを含む）リソースの予算がプロダクトオーナーによって承認され交渉可能である。
テスト戦略	3.1.4	一つ以上のテストレベルが実行されている場合（ユニットテスト、機能テスト、受け入れテスト）テストはレベル（テストタイプ、テストカバレッジ、テスト深度）が分析されたリスクに基づき区別できている。
テスト組織	4.1.1	品質とテストのタスクと責任が明確であり、スクラムマスターとプロダクトオーナーとの調整ができている。
テスト組織	4.2.1	チーム内に一人以上のテスト担当者がいる場合または複数のチームにおいて：テスト活動が調整され、割り振られている。
テスト組織	4.3.1	テスト担当者に対するタスク、ルール、責任と期待がスプリントのアクティビティが始まる前に明確に定義されている。

コミュニケーション	5.2.2	異なるコミュニケーション手段が、チームの内部と外部で効果的なメッセージがやりとりできるよう考慮されている。
報告	6.1.1	スタンドアップの報告に時間、コスト、結果そして（上記すべての）プロダクトリスクという側面が含まれている。
報告	6.2.1	ふりかえりの報告はテストの時間とコストに関する傾向と推奨を含んでいる。
報告	6.2.2	ふりかえりの報告はプロダクト品質とプロダクトリスクに関する傾向と推奨を含んでいる。
欠陥管理	10.1.2	欠陥が次のフェーズまたはスプリントに引き継がれる場合、以下のデータが蓄積される： 関連するテストケース、欠陥の報告者、日付、重要度カテゴリ、欠陥の記述（欠陥の再現方法、期待結果と観察結果）、欠陥のステータス
欠陥管理	10.3.1	すべてのチームとプロジェクトは同じ欠陥管理システム（ツール）を使用するか、またはすべてのシステム間でシームレスな接続が提供されている。
手法の実践	12.1.1	テスト活動は規定されたガイドライン、提供するテスト成果物、仕事の仕方に対する追加要求に従って行われる。
手法の実践	12.2.1	ガイドラインやテスト手法の実践的な使用はチーム内で/チームによって評価される。
手法の実践	12.3.1	テスト手法はすべてのアクティビティのゴール、責任のある役割、使用されるツールと技術、前提条件を記述している。
テスト担当者のプロ意識	13.1.4	テストのスキルはチームとマネジメントとで定期的に期待値について議論され評価されている。
テストケース設計	14.1.2	それぞれドキュメントされたテストケースは以下の要素を含んでいる： - 初期状態（の説明） - 変更プロセス＝実行されるべきテストアクション - 予測される結果（!） - 必要なテストデータ
テストケース設計	14.2.2	チェックリストまたはレビューのような他の手段がテストケースではカバーできない品質（例えばユーザビリティ）について適用されている。
テストツール	16.3.3	テスト環境とテストデータの所有権限は限定された部署にアサインされている。

3.4.5 クラスタ E

キーエリア	NR.	チェックポイント
利害関係者のコミットメント	1.2.2	テスト活動とテスト対象の品質に関する情報が利害関係者に提供されている。
テスト組織	4.1.2	品質とテストのタスクと責任がチーム全体に対し明確にされている。
報告	6.3.1	チームからの情報がすべての品質、テスト活動、ソフトウェア開発ライフサイクルの改善に使用されている。

テストプロセス管理	7.1.1	バックログと完了の定義に、実施すべきテスト活動と、要求された品質を含む。
テストプロセス管理	7.1.2	スプリントプランニングかつ/またはバックログリファインメントをプロダクトオーナーと調整している。
テストプロセス管理	7.2.1	完了の定義がプロダクトオーナーと他の関連する利害関係者とともに議論されている。
テストプロセス管理	7.3.1	テストリソースの（再）割り当てがチームに委ねられている。
メトリクス	9.3.2	メトリクスを提供するための工数が、アジャイル活動の利益に対して測定されている。例：利害関係者からの情報提供ニーズに対してメトリクスがどれほど貢献しているか
欠陥管理	10.1.3	開発者/テスト担当者は、テストの観点から、欠陥のリスク分析に貢献している。
欠陥管理	10.2.2	チーム全体が欠陥の管理（ツールであることが好ましい）にアクセス権がある。
欠陥管理	10.3.2	欠陥はプロジェクト/プログラムのレベルで分析され、テスト実践の改善は将来の欠陥を防ぐために開始されている。
手法の実践	12.3.2	完全で包括的な一連のテンプレートがテスト手法の一部となっている。
テストケース設計	14.1.3	テストケースまたはテストスクリプトにおいて、具体的なシステムのふるまいを記述しているテストベース（ユースケースとユーザーストーリー）が参照できるようになっている。

3.4.6 クラスタ F

キーエリア	NR.	チェックポイント
テスト組織	4.1.3	品質と進行中のテストの仕事はどのチームメンバーによっても進められる。
テスト組織	4.2.2	チームはプロダクトとプロセスの品質の成功と失敗に対する説明責任があると感じている。
報告	6.1.2	プロダクトオーナーに対する報告（頻度と内容）は意思決定プロセスの基本的な要求に合致している。
報告	6.1.3	報告は利害関係者が要求する限り書面でなされる。
報告	6.2.3	利害関係者から求められる報告は、報告を提供するための工数に対してバランスがとれている。
テストプロセス管理	7.1.3	各テストアクティビティがチームにとって可視化され、監視され、必要に応じて調整されている。
テストプロセス管理	7.2.2	バックログリファインメントでバックログが予測と乖離しているかどうか、プロダクトオーナーと議論されている。
テストプロセス管理	7.2.3	バックログの調整は文書化されている。また必要であれば利害関係者が文書化を行う。
テストプロセス管理	7.3.2	テスト活動のコーディネートと管理は定期的に、チームによって内部的に、そして利害関係者とともに評価されている。

見積もりと計画	8.1.2	評価において考慮されるアクティビティとは： テスト計画、テストケースの定義、テストデータの収集、テストケースの実行、テストケースの保守である。
見積もりと計画	8.2.2	見積もり技法はふりかえりの中で評価される。
見積もりと計画	8.3.1	バーンダウン/バーンアップまたはベロシティのようなメトリクスは、チームの見積もりと計画の活動をサポートするために定義され維持される。
欠陥管理	10.2.3	欠陥は、テスト対象の品質を示すために、ふりかえりの中で分析されている。
テストウェア管理	11.1.1	テストベース、テスト対象そしてすべてのテストウェア（スクリプト、データ、計画）は名前とバージョンによって識別される。
テストウェア管理	11.2.2	テストデータ、テストウェア、テストベースそしてテスト対象を管理する手続きと論理的仕組みがある。
テストウェア管理	11.3.1	テストウェアの保守のためのガイドラインがプロジェクトを超えて提供されている。
手法の実践	12.3.3	テスト手法について、必須、条件付き、そして選択可能な要素が十分に示されている。
テストケース設計	14.2.3	テストケースの正確性と効率性が必要に応じて評価され改善されている。
テストケース設計	14.3.1	テスト設計技法とチェックリストが、再利用のために評価され、調整され、維持されている。

3.4.7 クラスタ G

キーエリア	NR.	チェックポイント
テスト組織	4.1.4	アサインされた役割、プロダクトそしてサービスがチーム外の利害関係者に明示されている。
テスト組織	4.3.2	品質とテスト対象の製品・サービスが組織レベルにおいて定期的に評価されている。
見積もりと計画	8.1.1	（テストの）工数見積もりにおいてプランニングポーカーのような簡易な技法が使用されている。
見積もりと計画	8.2.1	テストの多様性とチームとの依存性を示すようにプロットされた計画を作成する必要性につき、利害関係者と議論されている。
欠陥管理	10.2.4	欠陥の傾向が分析されている。そのためには、欠陥についてより多くの情報が記録されている：サブシステム、優先度、プログラムとバージョン、テストベースとバージョン、根本原因、すべてのステータスの遷移と問題解決者。
テストウェア管理	11.1.2	各テストケースはテストベースに対して透明性のあるやり方で簡単にトレースできる。
テストウェア管理	11.2.1	テストウェア管理下のアイテムに対し各チームメンバーがアクセスできる。
テストウェア管理	11.3.2	これから始まるプロジェクトに対してテストウェアの引き継ぎをするプロセスが定義されている。
手法の実践	12.3.4	テスト実践は定期的に評価されており、必要に応じて調整が行われている。

4 テストの潜在能力の改善

4.1 テスト技術にフォーカス

4.1.1 イントロダクション

スクラムガイドが示すように、個々のチームメンバーが専門スキルと技量を持っているかもしれない。この章ではテストスキルに焦点を当てる。

4.1.2 技術的なスキル

テスト担当者はテストの専門家であつたし、これからもそうあるだろう。準備、設計、実行、そしてテストケースの結果に関する報告のプロセスにおける専門家である。この点に関してはアジャイルの環境とその他の環境には差異がない。よって、次のスキルは、テスト責任者には常に当てはまるだろう。

- テストを実行し、技法を適用することを理解している。
- 論理的な順序でどのようにテストケースとスクリプトを設定するか知っている。
- テストの結果に関して明確な判断力を有している。
- リスク分析や利害関係者の管理のようなテストのコーディネート/タスクの管理を理解している。
- テスト活動の見積もりと計画をどのようにするか知っている。
- プロダクト品質に関するビジョンがある。
- テストをどのようにするか、何をテストするか、なぜテストするかを説明する能力がある。
- ツールとエンジニアリングに慣れ親しんでいる。
- (技術的な) 改善の必要性を明らかにする。

つまり、アジャイルな状況におけるテスト活動における技術的なスキルは、既知の技術的なスキルとそれほど変わらないのである¹⁰。

4.1.3 ソフトスキル

アジャイルにおいては、あらゆるレベルで、チームメンバーそして利害関係者といったすべての参加者間でコミュニケーションが求められることはいうまでもない。コミュニケーションにはまた、積極的な態度を求められる。誰かが情報がほしいというまで待っていてはいけない。自分または自分のチームに役立つならばどんなことでも共有する必要がある。

積極的なコミュニケーションはまた、情報に対する興味を持つことを意味する。よって、アイデア、意見、提案について聞くことに好奇心を持たなければならない。自分または自分のチームに役立つならばどんなことでも。

さらに、次の側面はチームの中のベストプレイヤーになることに役立つだろう。

リスクベーステスト

ウォーターフォールよりもアジャイル開発で求められる原則は「いいものはそこそこいいだけ」ということである。このことは、要件、あるいは要望としてのスコープや機能に達しないリスク、欠陥のリスクがあるだろうということである。テスト担当者として、喜んでチャンスを取り、リスクをとるべきであるが、よりよいのは、リスク駆動で考え行動することである。起こりうる影響を意識的に考えながら、ミスをする可能性を分析する。リスク駆動になることはテスト対象の製品やプロジェクトに関係するだけではない。それは、あなた全体の態度に関わる。つまり、何をし、何をしないのか、一緒に働く人、決断に対して優先度を決めなければならないと意識することである。

テスト実践:

- やろうと計画したテストをいくつかスキップすることを考え、チームメイトと可能性のあるリスクについて熟考せよ。そのことによってプロセスが加速しテストの負債を作るかもしれないが、(完了の定義において言及された) 定められた品質と動くソフトウェアの原則を危険にさらすことは決してしてはならない。常に品質はチーム全体としてアプローチし責任をとるものであると強調しよう。

¹⁰ 参考: *Tester professionalism in the TPI NEXT book*, Chapter 4.13

- プランニングボーカーでは「リスクさ」を見積もる：見積もりの中で自分自身と自分のチームにチャレンジしチャンスをつかむ。
- 欠陥を見つけた場合、修正しない（回避策の可能性）または修正後に再テストしないリスクも考える。

責任をとる

ウォーターフォール、または類似のプロジェクトや状況においては、ややもすると自分の専門性に逃げてしまったり、同時に完璧なテスト担当者を演じたりする。しかしアジャイルにおいては、さらにそれ以上が求められる。アジャイルチームの一員であることは、以下を意味する。すなわち、チームの結果に対して個人個人が責任を持つということである。同時に、チームは個人のメンバーの結果に対して責任がある。それは、前章で議論した技術的なスキルを超えたものを要求する。前面に出て、明示的なコミュニケーションを行い、より積極的に、そしてとりわけ、「チームプレーヤー」になれ、ということである。

テスト実践：

- （テストだけでなく）自分が焦点を当てる範囲をチーム内の他の専門分野にも拡張し、相互の責任とタスクを議論する。
- 最良の結果を得るためにチームメイトをどのように助けることができるか理解する。
- 自己批判的になり、チームメイトがどういうスキルを身につければ自分を助けてくれるようになるか議論しよう。
- 他のチームメイト、スクラムマスターやプロダクトオーナーに対しあなたがどうテストし、何をして何をしなかったか、そしてそれはなぜかを示す。例：ユースケースや要件のカバレッジ

技量とスキルへのチャレンジ

期待以上の成果を上げれば、チームの実績は大いに評価されるだろう。これは積極的な態度が要求される。すなわち、建設的な方法で、失敗から学び、進んでチャンスをつかみ、進んで安全地帯から外に踏み出そうとするということである。同時にチームメイトやチーム外の利害関係者に対してはオープン・マインドを持つ。チームに自分の知識を提供し、チームから知識を共有してもらえよう努力しよう。

失敗を恐れてはいけい。失敗をしてそこから学ぶことが最良の学習方法である。

テスト実践：

- 以前使ったことのないテスト設計技法の使用
- プロダクトリスク分析（PRA）の導入・実行・実践

整理する

アジャイルの状況ではテスト計画はほとんど作られない。活動とプロセスは短期間のサイクルであり管理可能である。とはいえ、仕事を計画、準備、構造化するのが賢明である。それは組織化を助ける、つまりめっちゃくちゃな机ではものをすぐに探せないのと同様である。さらに、自分の行動に責任を持つということは、これらの活動を適切にコントロールすることを意味する。すなわち、構造化し、整理し、コントロール下に置き、可視化することである。

テスト実践：

- テストをできるだけ早期に準備する。例えばプログラミング開始前や開発者とペアになっている時である。可能であればテストを単体テストに組み込んでもらおうとよい。これはすべてのユーザーストーリーごとに繰り返される。
- チームメイトにとって理解しやすいやり方でテストケースを設定する。
- どのような仕事をしているか説明しフィードバックを求める。

チームメイトへの信頼と尊敬

かつてはテスト担当者として批判的になり他の人が起こしたミス躍起に探そうとしていたかもしれない。それはよいことであり、それがテスト担当者として求められていたことであった。しかし、それをするにあたりそのミスをした人、アプリケーション内に欠陥を起こした人に対し常に敬意を払おう。そしてそういった人がこれらのミスから学び、成果物を改善できると信用しよう。さらに、仕事がうまくいった場合にはほめよう。そうすれば、よい品質を提供するための手助けに対して、あなたが尊敬されるようになる。

テスト実践：

- まずは他のチームメイトを信頼しよう。そうすることで、テストケースを実行してもらったり、自分自身で作ったものをテストしたりするように仕向けられる。

- もし欠陥を見つけたら建設的な方法で報告する。例：可能性のある解決策や回避策を提供
- 他の開発者とテストケースについて議論し、フィードバックを求める。

4.1.4 コーチとしてのテスト担当者

何年もの間、日々のテスト実践による進化を通じてテスト担当者はアプリケーションライフサイクルの中で運用の中心になってきた。一方で、アーキテクト、デザイナー、開発者、ビジネス（マネジメントとユーザー）、運用、保守、そしてサプライヤなど、他の専門分野からインプットする必要性にも駆られている。（例：テスト担当者はテストケースの参考文献が必要である。何が期待された出力結果なのか、どのアクションがこの機能を動かすために必要なのか。）他方では、他の専門分野の関係者が、テストの結果に左右されたり、積極的に結果を知りたがったりする。すなわち、（いつ）リリースするか、（中間の）プロダクト品質はどうか、などである。

このポジションと幅広い知識と経験に基づき、テスト担当者は他の専門分野に対してのコーチとしてふるまうことができる。アプリケーションライフサイクルの異なる局面において、テストエンジニア、テストマネージャ、テストチームのリードなどの役割からテスト担当者に得られる知識は、アジャイルチームと外部のテストチームにメリットをもたらす。下記の段落は「テスト担当者」がアジャイルチームに貢献する知識と経験の例である。

プロジェクトを始めるスキル：

- プロジェクトの確立と管理の方法。
- 予算または見積りの知識。
- テンプレートの開発と使用。
- プロダクトリスクの分析と軽減：
 - 誰がリスクアセスメントに参加するか。
 - リスクが何かをどう決定するか。
 - リスクの深刻性をどう決定するか（分類法）。
 - 緩和する活動。
- プロジェクトリスクの分析と緩和。

準備をするスキル：

- テストケースの重要な部分から得られる必要な機能性に対する知識、つまり機能、プロセス、アクティビティの望ましい結果や期待結果のことである。
- レビューに対して可読性と分析性がより高い設計（機能設計と技術的設計の両方）をする。
- ユーザーの願望を翻訳しユーザー受け入れテストにおいて不意打ちを避ける。

開発/実装をサポートするスキル：

- 欠陥管理とその修正：
 - 欠陥の根本原因分析の提供。
 - 実現可能な解決方法。
 - 回避方法。
 - 欠陥のリスクアセスメント。
 - 計画とスケジュール：今直すかあとで直すか。
- 再テストは必要か yes/no!
- ならびに：
 - 悪い知らせの報告方法。
 - 外交（チーム内外のやりとりや政治的交渉）。

受け入れをサポートするスキル：

- テストはエラーや欠陥を探すだけではない。テストはプロダクトの信用を提供する。
- ユーザーサポート：
 - 経験/結果の共有。

- 何がよくて何がよくないのか。
- テストケースの設計をサポート。
- (ビジネス) リスクのカバーと緩和に関する情報の提供。

運用をサポートするスキル：

- ユーザーサポート。
- ユーザーマニュアルの作成。
- トレーニングとして使われるテスト/テストとして使われるトレーニング。
- プログラム、機能、システムのロジックの説明。
- 「本番環境」でのテスト：
 - バグ修正のサポート。
 - すばやい修正。
 - かしこい再テスト。

5 Appendix A: キーエリアとその重要性

この付録は『TPI NEXT®ビジネス主導のテストプロセス改善』に詳細が記述されているキーエリアについて説明する。アジャイルでは、キーエリアの主要な要素は同じだが、用語が異なることに注意しなければならない。

(訳注：SR＝利害関係者との関係、TM＝テスト管理、TP＝テスト業務の専門性)

#	キーエリア	グループ	重要性
K01	利害関係者のコミットメント	SR	利害関係者のコミットメントと積極的な関与、効率的なコミュニケーションや協力関係を作ること： テスト作業に参加する個人のコミットメントがテストプロセスの円滑な運用をするための重要な条件となっている。利害関係者はプロジェクト管理者またはライン管理者であるが、（エンド）ユーザー、内部の経理担当、マーケティングということもある。利害関係責任者は（定量的かつ定性的に）よいテストを行う実施するために必要な時間、資金とリソースを提供する。
K02	関与の度合い	SR	テスト作業がプロジェクトに深く関与すると、開始当初からプロダクト品質が向上しやすくなり、テスト活動がプロジェクトのクリティカルパスになることを避けられる： 複数のテスト間で、タイムリーに準備と調整ができる。また、テストプロジェクトがプロジェクトのクリティカルパス上にある時間をできるだけ短くできる。テストチームの SDLC（訳注：ソフトウェア開発ライフサイクル）への早期の関与は、欠陥をできるだけ早くそして簡単に見つけることができ、さらに欠陥予防の手助けにもなる。早期段階では、テストチームがプロジェクトリスクの分析、および要件や設計のレビューを支援する。
K03	テスト戦略	SR	テスト戦略とは、工数やリソースをテストプロセスへ最適に割り振るための指針である： テスト戦略には、テストに掛ける工数の分配と、テストすべき部分か、またはテスト対象の側面に対するカバレッジについて定義する。テスト戦略の目的は、もっとも重要な欠陥を可能な限り早い段階で検出することである。「もっとも重要である」とは、ビジネスプロセスや重要成功要因、あるいは認識されたプロダクトリスクなどのテストゴールから導き出される。
K04	テスト組織	SR	テスト組織は、テストリソース、テスト関連成果物およびテストサービスに関するプロジェクトのニーズを満たす： テスト作業とは、人的資源による仕事である。各参加者はそれぞれ独自のスキル、タスク、責任、権限、力量、関心を持っており、関与するすべての人々に対してそれを明確に示す必要がある。テスト組織はコミュニケーションやプロセスをスムーズに進めるため、これらすべての側面や関係性を考慮する。
K05	コミュニケーション	SR	明確なコミュニケーションにより、関係者間で共通理解と期待値の調整ができる： テストプロセスでは、適切な条件の作成、テスト戦略の協調、活動の微調整、優先度の交渉などのために、関与する人々の間で明確かつ効果的なコミュニケーションが必要である。コミュニケーションには、チームから外部へ、外部からチームに向けて、あるいはチーム内など、さまざまな形がある。良好なコミュニケーションによって、対話とフィードバックが促される。
K06	報告	SR	報告は、利害関係者の意思決定プロセスやテストプロジェクト収支の把握に対して重要な知見を提供する： 報告は一方のコミュニケーションであり、その目的は、具体的な提案を提供し、テスト対象とテストプロセスの両方の品質を把握することである。また、報告はテストプロジェクトの予実も提示し、意思決定プロセスを支援する。これらの理由から、報告書はもっとも重要なテスト成果物の一つといえる。
K07	テストプロセス管理	TM	テストプロセス管理は、要求された時間とコストとリソースの中で、テストに課せられた任務の遂行を最大化する：

			テストプロセス管理によって、リソース、利害関係者、インフラ、技法、計画・準備・実行確認のサイクルをコントロールし互いに関連づけることができる。適切なテスト管理によって、テストプロセスが効果的かつ効率的なものとなる。
K08	見積もりと計画	TM	見積もりと計画の技法を正しく利用することにより、テストプロセスの計画や見積もりが予測可能で信頼の置けるものになる： テストの計画と見積もりによって、どの活動を実施すべきか、必要なリソース（人）は何か、いつ必要か、といった情報がわかる。見積もりと計画の技法を適切に使用することで、テストプロセスが予測可能で信頼の置けるものとなる。
K09	メトリクス	TM	メトリクスによって対象物を定量的にとらえられ客観性がもたらされる： メトリクスとは、プロダクトやプロセスの特性に関する定量的かつ客観的な観察である。テストプロセスをコントロールし、テストアドバイスを立証し、システムやプロセスの比較をするのに使われる。特にテストプロセス改善の際は、メトリクスによって改善活動の実施前後のデータを比較して結果を評価できるので、重要である。
K10	欠陥管理	TM	欠陥管理とは、個人レベルとグループレベルの両方で、根本原因分析やガイドラインを利用して欠陥に対処することである： 欠陥管理のルールを決め、それに従って管理することは、テストチームだけの問題ではなく、ソフトウェア開発プロジェクト全体にとっての問題である。だが、中心となって関与するのはテスト担当者である。優れた欠陥管理のルールによって、欠陥ライフサイクルの監視や、さまざまな（統計的）概観や報告（品質提案など）を提供できる。
K11	テストウェア管理	TM	テストウェア管理によって、テストの成果物間や、テストの成果物と関連する設計文書間の一貫性が確保できる： インプットプロダクト（設計やテスト対象など）や、テスト作業のアウトプットプロダクト（テストケース、スクリプト、テスト計画、レポート）は、保守と再利用が可能でなければならないため、管理が必要である。さらに、テストウェア管理によってテスト成果物同士やテスト成果物と関連する設計書の一貫性を確保する。
K12	手法の実践	TP	明文化したテスト手法は、テストプロジェクトの方向を定め、支援する： テスト手法は、活動、手順、規約、技法などから構成され、テスト活動やテストプロセスごとに、これを用いる。テスト手法はあらゆる状況に適用できるように十分包括的であるべきだが、同時に、日々のテストの実践を指揮し、支援できるように詳細な内容も含む必要がある。
K13	テスト担当者のプロ意識	TP	テスト担当者のプロ意識に含まれるものは、テスト活動を期待するレベルで行うのに必要な、さまざまなスキル、強み、規律、役割、知識などが正しく混ざっていることである： テスト作業には、ビジネスや組織に精通していることはもちろん、テスト対象に関する知識や一般的な IT のノウハウと技術的スキルなど、多くのスキルセットが必要となる。加えてテスト担当者は、手法に関する専門知識から利害関係者との効果的なコミュニケーション能力（社会的能力）にいたるまで、テストに特化したスキルも必要である。テスト担当者のプロ意識とは、これらのスキルと強みがすべて合わさって、期待するレベルのテスト活動を行うためのものである。
K14	テストケース設計	TP	テストケース設計とは、テスト戦略に沿って欠陥の検出ができるようにテスト実行を導くものである： テストケースは、元となる情報（テストベース）から導き出すのが標準的な方法である。これにより、テストのカバレッジや深さを把握でき、テストケースの（再）設計や（再）実行がしやすくなる。
K15	テストツール	TP	テストツールを使うと、特定のテスト活動を可能にし、加速する： テストツールは、テストプロセスの作業を自動化する、あるいはテストプロセスの管理作業へ人が介入することを不要にする。テストツールは、複雑さや量に依存せず、機械的に反復的な作業を行うことに秀でている。手動では時間がかかり実現不可能な場合や、エラーが起こりやすいといった場合にその効果を発

			揮する。テストツールは、特定のテスト活動を実現可能にする、あるいは加速させることによって、リードタイムの短縮、テスト深度の増大、テスト担当者のモチベーションの向上などをもたらす。
K16	テスト環境	TP	テスト環境は、各テストレベルのゴールを念頭において明確に設計、実装し、保守するものである： テスト環境は主に以下のコンポーネントで構成される。 • クライアント環境 • ネットワーク • ストレージ • （エンタープライズ）サーバ • ミドルウェア • テストデータ テスト環境はテストゴールを反映して構成し、セットアップすべきである。テスト環境は、テストプロセスの品質、リードタイム、コストに大きな影響を与える。テスト環境にとって重要なことは、責任、管理、いつでも利用できる十分な可用性、代表性※、柔軟性である。 （※訳注：そのテスト環境が、テスト対象のソフトウェアが実際に実行される環境をどれだけ反映しているかの度合いのこと）

6 Appendix B: AGILE TPI Tool のガイドライン

クイックスタート

すべてのマクロが（成熟度マトリクスを自動生成するような）動作するように、セキュリティ警告の後ろの「オプション」をクリックして「マクロを使用する」を選択してください。

[Information]シートをクリックし、アセスメント、分析、改善のプロジェクトなど固有情報を入力してください。ファイルに名前を付けて保存するのを忘れないようにします。

選択内容（個人レベル、チームレベル、または組織レベル）に応じてワークシートを選択するか、[All checkpointsMaster]シートを使用してください。これらすべてのワークシートは自動的に同期します。

チェックポイントは、はい[Y,Y] または、いいえ [n,N]が選べます。Notes の列はあとで推敲のために使用するか、アセスメント時に使用することが推奨されます。

最終的にチェックポイントは集計されて、[Test maturity matrix]シートに結果の概要が表示されます。

改善の目的のためには、埋められなかったチェックポイントが改善活動の参考になるでしょう。また、正しい改善の定義に関しては、クラスタが支援します。クラスタ A のチェックポイントから始め、次に B というようにできます。

以下のガイドラインでは、ツールの異なるステップ、アクティビティ、ワークシートの詳細を示しています。

詳細

全体

ファイルをロックせずマクロを有効にした状態で開くと、結果が [Test maturity matrix]シートに自動集計されます。マクロの保存を有効にし、適切な名前ファイルでファイルを保存します。（クイックスタートも参照のこと）

INFORMATION ワークシート

列	ガイドライン
C	適切な情報を入力するために、C 列の行のセルをクリックします。 B4 セルの {Project} と B8 セルの {Company name}の情報が成熟度マトリクスに追加されます。 C15 セルと C16 セルをクリックし、適切な日時を入力します。

ALLCHECKPOINTS ワークシート

列	ガイドライン
A*	Seq.Nr. [Seq.Nr.]と書かれたこの列はチェックポイントの数字を指します。連番は組み合わせとなっています。 位置 1: キーエリアの番号 位置 2: 運用観点での番号 1 = 個人レベル (P) 2 = チームレベル (T) 3 = 組織レベル (O) 位置 3: チェックポイントの連番 例: 2.1.1 は関与の度合い（キーエリア 2）の個人レベル (P) の 1 番目のチェックポイントであることを表します。
B*	Cluster この列の英字は 6 つのクラスタ (A, B, C, D, E, F) のうちの一つを指します。

例：9 行目の E は、チェックポイントがクラス E の中のものであることを意味します。

C*

Checkpoint

この列ではチェックポイントが定義されています。いくつかのチェックポイントには、一つ以上の変数が使われていることに注意してください。（例：～であること「かつ（and）」…であること、という言葉が使われている）すべての変数は、「例（Example）」を除き、分析と評価に適用可能です。

D

Fulfilled?

この列にはアセスメントまたは分析の結果を入れます。入力可能な値のチェックボックスをクリックしてください。デフォルトの値は「NA」で、[Test maturity matrix]シートのセルには無色で表示されます。選択肢は以下です。

- 'y' または 'Y' はチェックポイントが入力されこのチェックポイントの条件を満たした場合。この結果は[Test maturity matrix]シートのセルに緑色で表示されます。
- 'n' または 'N' はチェックポイントが入力されたがこのチェックポイントの条件を満たしていない場合。この結果は[Test maturity matrix]シートのセルに赤色で表示されます。

E

Notes

このエリアは分析結果の基として必要です。ここではそのチェックポイントの「満たされている」「満たされていない」「適用外である」をそれぞれ説明することができます。あり得る説明としては、文書やインタビューに証拠が見つけれなかった、などです。チェックしなかったものを除いて、すべてのチェックポイントに情報が覚え書きを残すようにするのがお勧めです。このことでマネジメントへの報告やプレゼンテーションの助けになるでしょう。また、この文書を他の分析や将来のアセスメントに変換しやすくなります。

* 注：列 A, B, C は固定され、修正できません。

TEST MATURITY MATRIX ワークシート

列 ガイドライン

全体 このシートは[All checkpointsMaster]シートの D 列に入力した分析の結果とチェックポイントの値を示します。成熟度マトリクス、カテゴリごとの結果と合計（絶対値）、キーエリアごとの結果（比率）のレーダーチャートを含んでいます。

A -AK

行 8-23

このセルはすべてのチェックポイントとその値を含みます。チェックポイントはクラスタのアルファベットと一致するように記されます。色については以下の意味があります。

灰色：チェックポイントが分析または入力されていません。'NA'を選択した箇所が該当します。

赤色：チェックポイントが分析されていますが条件を満たしていません。'n'または'N'を選択した箇所が該当します。

緑色：チェックポイントが分析され条件が満たされています。'y'または'Y'を選択した箇所が該当します。

セル（チェックポイント）の一つをクリックすると、[All checkpointsMaster]シートに記載された（行 4）テキスト内容、および、（行 5）Notes のコメント内容を見ることができます。

行 4 および 5

行 4 には、成熟度マトリクスのチェックポイントを選択してクリックするとそのテキストが表示されます。

行 5 には、[All checkpointsMaster]シートのチェックポイントに追加された Notes の内容が表示されます。

AN-AU

行 1-9

このグラフは絶対値で全チェックポイントの値の結果が表示されます。条件が満たされたチェックポイントがすべて緑色で表示され、条件が満たされていないか分析されていないものは灰色のまま表示されます。

左から 1 番目のグラフ：すべてのチェックポイント（108）のスコア

左から 2 番目のグラフ：個人レベル（Professional）のスコア

左から 3 番目のグラフ：チームレベル（Team）のスコア

左から 4 番目のグラフ：組織レベル（Organization）のスコア

行 11-23

レーダーチャートはキーエリアごとの[All checkpointsMaster]のチェックポイントの値の結果を合計に対する比率で表します。結果はチャートの青色の線で描画されます。