

テストプロセス改善技術 入門ガイド [ver1.0.0]

これからテストプロセス改善技術を学ぶ方へ

NPO 法人 ASTER テストプロセス改善技術研究会

はじめに

本ガイドの目的

本ガイドは、これからソフトウェアテストプロセス改善技術（以降、テストプロセス改善技術）の情報収集や学習を考えている入門者を対象として作成しています。テストプロセス改善技術の概観を 30 分程度の読書ですらりとつかんでもらうことを目的として、記述量や内容を絞り込んでいます。

本ガイドを通読することで、テストプロセス改善技術の使い方や注意点などの概要を理解できます。また、参考となる文献や Web サイトの情報を得ることができます。

想定読者

これからテストプロセス改善技術の情報収集や学習を考えているプロセス改善入門者の方々を想定しています。

本ガイドの記述範囲

個別のテストプロセス改善技術の紹介は、それぞれ概要のみを記述しています。より詳しい情報は、それぞれの改善技術の「参考となる文献、Web ページ」をあたってください。

本ガイドの構成

1 章では、テストプロセス改善技術の概要を紹介します。

2 章では、ISTQB や SQuBOK Guide V3 で言及されているテストプロセス改善技術について解説します。本ガイドの 2.1 から 2.3 では、テストプロセスを改善するためのロードマップやアセスメント指標が提供されている TMMi、TPI NEXT、ISO/IEC 33063 を概説します。2.4 と 2.5 ではその他の技術として CTP と STEP について概説します。また、各テストプロセス改善技術の解説の最後に、参考となる文献、Web ページを紹介します。

本ガイドの用語定義

本ガイドでは、「テストプロセス改善手法」や「テストプロセス改善モデル」と呼ばれる用語を「テストプロセス改善技術」として統一しています。

各テストプロセス改善技術の説明では、参考文献で使用されている用語を使って記述しています。

改版について

適宜改版を予定しています。定期的に最新版をご確認ください。

目次

はじめに	1
本ガイドの目的	1
想定読者	1
本ガイドの記述範囲	1
本ガイドの構成	1
本ガイドの用語定義	1
改版について	1
目次	2
1. テストプロセス改善技術とは	5
1.1 テストプロセス改善技術とは	5
1.2 テストプロセス改善モデルの種別	6
2. テストプロセス改善技術概説	7
2.1 TMMi	7
概要	7
モデルの種別	7
モデルの構造	7
各要素の説明	8
使い方	9
注意点	9
参考となる文献、Web ページ	9
2.2 ISO/IEC 33063	10
概要	10
モデルの種別	10
モデルの構造	10
各要素の説明	10
使い方	12
注意点	13

参考となる文献、Web ページ	13
23TPINEXT	14
概要	14
モデルの種別	14
モデルの構造	14
各要素の説明	14
使い方	16
注意点	16
参考となる文献、Web ページ	17
24その他の技術（CTP）	18
概要	18
使い方	18
注意点	19
参考となる文献、Web ページ	19
25その他の技術（STEP）	20
概要	20
使い方	21
注意点	21
参考となる文献、Web ページ	21
参考文献	22
商標について	24
付録1テストプロセス改善事例集（作成中・一部公開版）	25
概要	25
事例集の目的	25
想定読者	25
本ガイドの用語	25
ご注意	25
① 利害関係者と積極的に情報共有せよ！	26
② ミーティングは目的とプロセスを定めよ！	27

③ コミュニケーションの齟齬を起こさないように用語を統一すべし！	28
④ ナレッジは積極的に蓄積せよ！	29
⑤ どういったテスト作業に時間がかかっているかを抑えるべし！	30
付録2テストプロセス改善スキル小説（作成中・一部公開版）	32
概要	32
小説の目的	32
想定読者	32
本小説の用語	32
ご注意	32
1 テストプロセス改善ストーリーの概要	33
1.1 登場人物および関連図	33
2 プロローグ	34
2.1 明日太さんの現状と課題	34
2.2 明日太さんのヒーロー	34
2.3 相談のセットアップ	35
3 アセスメント	35
3.1 第一歩	35
3.2 アセスメントってもんは	36
3.3 優しいクロスカウンター	37
3.4 アセスメント結果、再提出	37
巻末	39
テストプロセス改善技術入門ガイド執筆メンバー一覧	39
免責事項	39

1. テストプロセス改善技術とは

1.1 テストプロセス改善技術とは

テストプロセスの改善技術とはどういったものでしょうか。それは、テストプロセスの改善を促したり、そのための評価をしたりといった技術のことです。このような技術は、いくつかの知識体系で取り上げられています。

例えば、ソフトウェアテストに関する知識体系である、ISTQB (International Software Testing Qualifications Board)があります。また、ソフトウェア品質に関する知識体系である SQuBOKGuide (ソフトウェア品質知識体系ガイド) があります。

ISTQB では、Advanced Level Test Manager のシラバスの「5.テストプロセスの改善」で次のものを取り上げています。

- TMMi (Testing Maturity Model integration、テスト成熟度モデル統合)
- TPI NEXT (Test Process Improvement NEXT)
- CTP (Critical Testing Process assessment)
- STEP (Systematic Test and Evaluation Process)

SQuBOKGuide V3 でも、「2.3.KA:ソフトウェアプロセス評価と改善」で次のものを取り上げています。

- TMMi (テスト成熟度モデル統合) *1

また、発行時期の関係で ISTQB や SQuBOK Guide V3 でトピックとして取り上げられていませんが、次のものもテストプロセス改善技術のひとつです。

- ISO/IEC 33063 (ISO/IEC 33063: Process Assessment Model for Software Testing)²

本ガイドでは、次の5つのテストプロセス改善技術を概説します。

1. TMMi
2. ISO/IEC 33063
3. TPI NEXT
4. CTP
5. STEP

*1説明内で TPI および TPI NEXT にも言及しています。

*2SQuBOKGuide V3 では「プロセスアセスメントに関する規格 (ISO/IEC 33k シリーズ) は取り上げられています。

1.2 テストプロセス改善モデルの種別

第2章で紹介するテストプロセス改善モデルTMMi、ISO/IEC 33063、TPI NEXT の特徴を理解するポイントとして、モデルの種別を取り上げます。モデルの種別には「段階モデル」と「連続モデル」があります。

段階モデルは、組織のレベルを成熟度で表します。各段階の成熟度レベルごとに、改善すべき対象（例：テスト計画、テスト設計、テスト環境等）が個別に存在します。成熟度が上がるごとに、改善対象の範囲が段階的に広がっていくイメージです。着手順序がわかりやすいというメリットがありますが、画一的であるため改善の順序が組織に合わない場合もあり得ます。



図11段階モデルのイメージ図

連続モデルは、組織のレベルを能力度で表します。改善すべき対象（例：テスト計画、テスト設計、テスト環境等）すべてに対して、個々の能力度レベルを評価します。改善対象の範囲全体が最初から対象となっています。レベルが上がるにつれて、各改善すべき対象に求められる要求が高くなっていくイメージです。組織全体のレベルは一番低い改善すべき対象の能力度レベルであると判断します。一部の例外はありますが、改善すべき対象の着手順序を自由に選択できます。着手順序を柔軟に決められますが、組織にとっての最適な順序を考える必要があります。

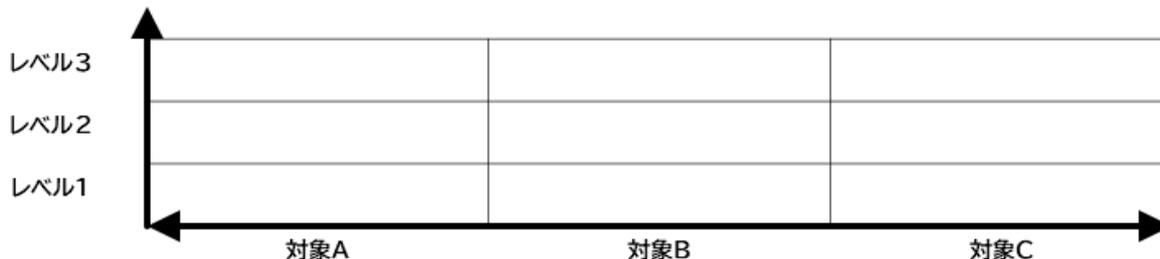


図12連続モデルのイメージ図

第2章では第1章で紹介したTMMi、ISO/IEC 33063、TPI NEXT、CTP、STEPを概説します。また、それぞれのテストプロセス改善技術について、次の項目を解説します。

- 概要
- モデルの種別※
- モデルの構造※
- 各要素の説明※
- 使い方
- 注意点
- 参考となる文献、Web ページ

※TMMi、ISO/IEC33063、TPINEXT のみに含まれる項目です。CTP と STEP には含まれません。

2. テストプロセス改善技術概説

2.1 TMMi

概要

TMMi は TMMi Foundation が作成しているテストプロセス改善技術で、その前身として TMM があります。TMMi の名前から想像される通り、CMMI (Capability Maturity Model Integration：能力成熟度モデル統合) を補完する位置づけです。すでに CMMI を導入している組織では比較的導入しやすいモデルです。

モデルの種別

TMMi は段階モデルです。成熟度レベルごとに改善すべき対象であるプロセスエリアが定義されています。成熟度レベルがあがるごとに、対象となるテストプロセスのスコープも段階的に広がっていきます。

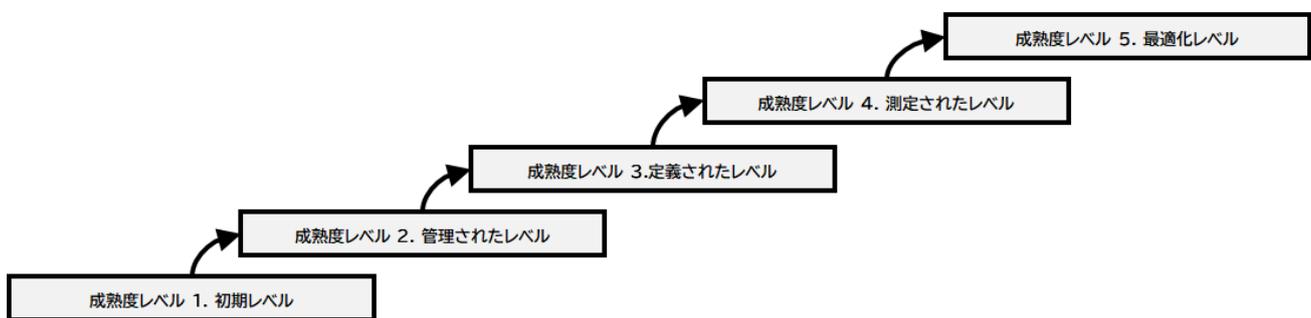


図 2.1 TMMi の段階モデル

モデルの構造

成熟度レベルには、1 から 5 までの 5 つの段階があります。この成熟度レベルは、テストプロセスや組織の成熟度を表しています。成熟度レベル 1 の初期レベルは、成熟度レベル 2 の管理されたレベルに達していない状態を示します。つまり、初期のレベル 1 の状態からレベル 2 の達成を目指します。次いでレベル 3、4、5 と段階的に改善活動を進めていきます。

成熟度レベルごとにプロセスエリアが定義されています。また、プロセスエリアごとに固有ゴールと共通ゴールが定義されています。これらを確認することで、取り組むべき改善活動を把握できます。なお、成熟度レベル 2 の管理されたレベルに達成していない場合を成熟度レベル 1 の初期レベルとしています。そのため初期レベルである成熟度レベル 1 にプロセスエリアはありません。

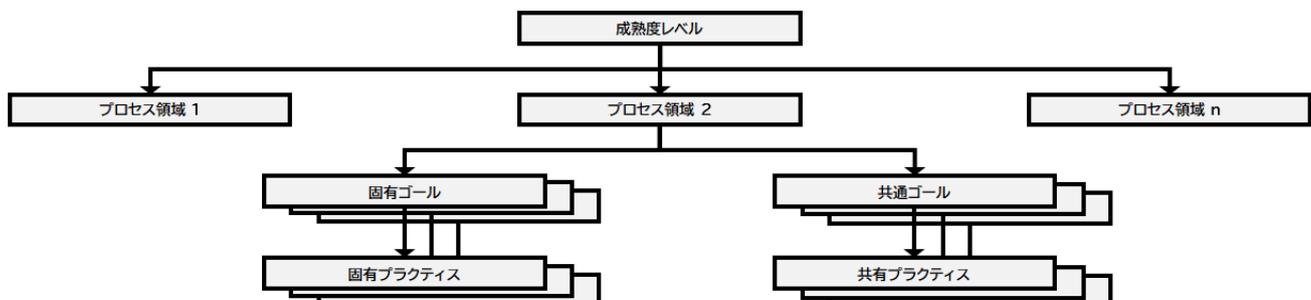


図 2.2 成熟度レベルごとの構成要素

各要素の説明

プロセスエリア

ある成熟度レベルを達成するために改善すべき対象であるテストプロセスをカテゴリに分けたものがプロセスエリアです。例えば、成熟度レベル 2 の管理されたレベルには、「テストポリシーと戦略」「テスト計画」「テストのモニタリングとコントロール」「テストの設計と実行」「テスト環境」の 5 つのプロセスエリアに分かれています。

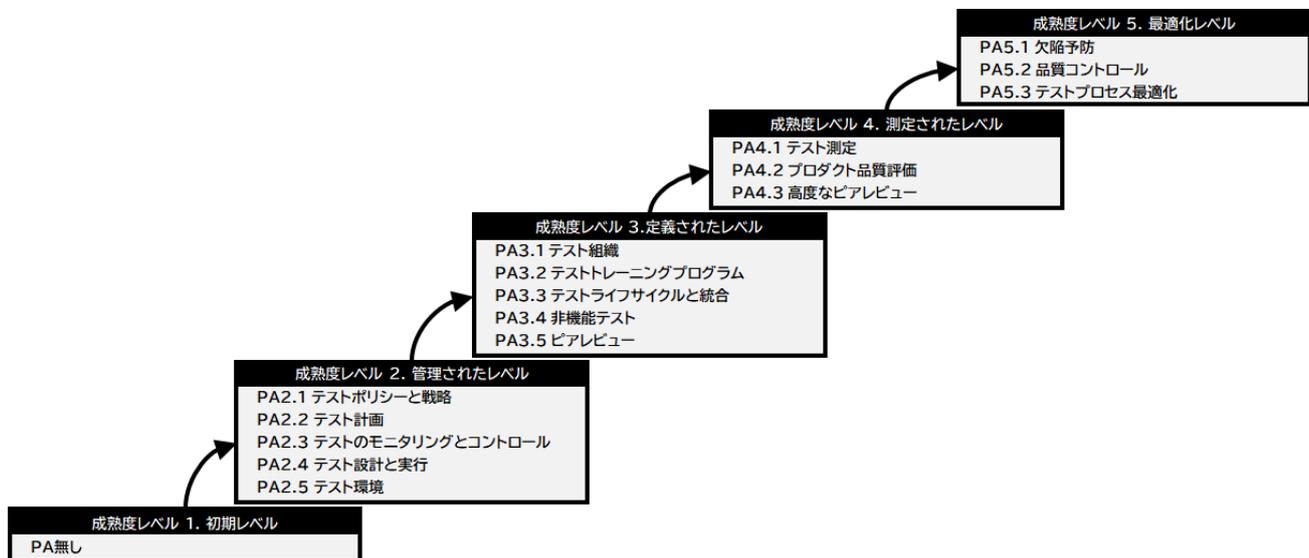


図 23 成熟度レベルごとのプロセスエリア

固有ゴール、共通ゴール

各プロセスエリアには、達成すべきゴールが設定されています。そのゴールには、固有ゴールと共通ゴールの 2 種類あります。特定のプロセスエリアを達成するためには、そのプロセスエリアに指定されたこれらのゴールを達成しなければなりません。

固有ゴールとは、そのプロセスエリア固有の目標です。例えば、レベル 2 の管理されたレベルのプロセスエリア「テストポリシーと戦略」では、「テストポリシーを定義する」(SG1)が固有ゴールとして設定されています。

共通ゴールとは、そのプロセスエリア固有ではないすべてのプロセスエリアに共通する目標です。例えば、「管理されたプロセスを制度化する」(GG2)が共通ゴールとして設定されています。

固有プラクティス、共通プラクティス

上述のゴールごとに、目標を達成するために必要なプラクティスが設定されています。そのプラクティスには、固有プラクティスと共通プラクティスの 2 種類あります。

例えば、上述の「テストポリシーを定義する」(SG1)という固有ゴールに対して、固有プラクティスとして「テストのゴールを定義する」(SP1.1)などが設定されています。また、「管理されたプロセスを制度化する」(GG2)という共通ゴールに対して、共通プラクティスとして「組織的なポリシーを確立する」(GP2.1)などが設定されています。

固有プラクティスでは、基本的に「成果物の例」と「サブプラクティス」の情報が提供されます。共通プラクティスには、基本的に「共通プラクティスの詳細」または「例」の情報が提供されます。また、各プラクティスではその他にも「注釈」「例」「リファレンス」などの情報が示されることもあります。

使い方

成熟度レベルの認定を公式に受ける場合は、TMMi Foundation から認定されたアセッサーによるアセスメントが必要です。しかし、公式な成熟度レベルの認定を必要としなければ、独自にアセスメントできます。ここでは、自身で簡易的に実施する場合の使い方をご紹介します。

まず、TMMiFoundation の Web サイトから TMMi Reference Model の PDF をダウンロードします。

現状のアセスメント

成熟度レベルとプロセスエリア、固有ゴールと共通ゴールを表にしたものを準備します。そして各ゴールで達成しているものを塗りつぶしていきます。これにより、できているところとできていないところが可視化されます。また、現在の成熟度レベルを把握できます。

改善活動

アセスメント結果を踏まえて、現在の成熟度レベルから次の成熟度レベルを目指します。次の成熟度レベルのできていないところについて、できるようになるための改善計画を考えます。改善活動の計画立案時は、固有プラクティスや共通プラクティスの情報が参考になります。

このようにして計画した改善活動を実行して定期的に再アセスメントすることで、継続的な改善を図ることができます。

注意点

- TMMi は日本語で提供されていません。英語の文書にあたる必要があります。
- 一度に多くの改善に取り組むのは難しい場合も多いです。そのため、次の成熟度レベルに向けて達成できていないと固有ゴールと共通ゴールのすべてを改善計画に含めるか、注意深く検討したほうが良いでしょう。
- 自分たちが直面している問題や課題に結びつくプロセスエリアを明らかにして、その中でも優先度が高く効果的なプロセスエリアから改善をはじめましょう。
- 公式な成熟度レベルの認定を受ける場合には、アセッサーの手配やアセスメントを受ける準備も必要であるため、組織的に取り組むと良いでしょう。

参考となる文献、WEB ページ

- TMMiFoundation の Web サイト
 - <https://www.tmmi.org/>
- TMMi の関連文書のダウンロードページ
 - <https://www.tmmi.org/tmmi-documents/>

2.2 ISO/IEC 33063

概要

ISO/IEC 33063 は、ソフトウェアテストのためのプロセスアセスメントモデルを規定した国際標準です。プロセスアセスメント領域を対象とする ISO/IEC 33K シリーズの 1 つです。

ISO/IEC 33K シリーズはもともと ISO/IEC 15504 シリーズとして標準化されていましたが、適宜 ISO/IEC 33k に置き換えられています。なお、ISO/IEC 33k シリーズは、SPICE（スパイスと読みます）の愛称で呼ばれることもあります。また、SPICE は Software Process Improvement and Capability Determination の略語です。

モデルの種別

ISO/IEC 33063 は、連続モデルです。改善順序は自組織で決定します。プロセス能力水準の低いプロセスの中からプロセス改善に着手すべきですが、同じ水準であれば、どのプロセスから着手すべきかを定めていません。

モデルの構造

プロセスアセスメントモデルは、その構造として横軸にプロセス参照モデルと縦軸にプロセス測定フレームワークを取ります。ISO/IEC 33063 では、横軸のプロセス参照モデルとして ISO/IEC/IEEE 29119-2、縦軸のプロセス測定フレームワークとして ISO/IEC 33020 を取ります（図 24 を参照）。

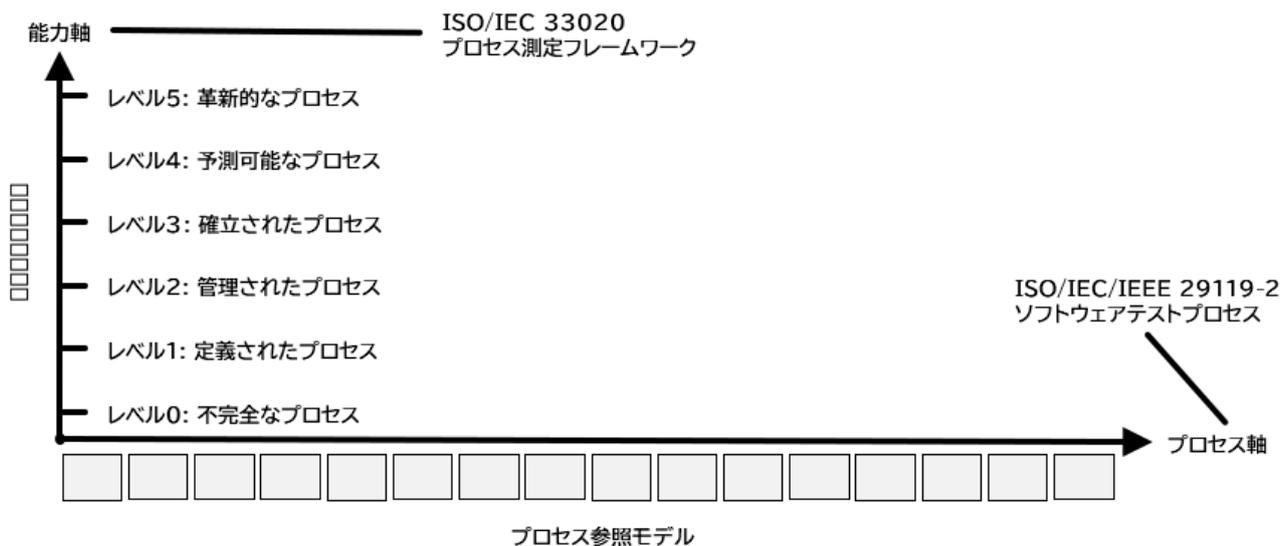


図 24 ISO/IEC33063 プロセス軸と能力軸

各要素の説明

横軸

ISO/IEC 33063 では、横軸のプロセス軸であるプロセス参照モデルとして、ISO/IEC/IEEE 29119-2 を使用しています。ISO/IEC/IEEE 29119-2 は、ソフトウェアテストのプロセスを規定した国際標準です。

ISO/IEC/IEEE 29119-2 には 3 つのプロセスグループに合計 8 つのプロセスが定義されています。ISO/IEC 33063 ではこの 8 つのプロセスを対象としています。付属書で 1 つのプロセスグループと 5 つのプロセスを参考情報として示しています（図 25 参照）。



図 25 プロセスグループ

縦軸

ISO/IEC 33063 では、縦軸の能力軸であるプロセス能力水準として、ISO/IEC 33020 を使用しています。ISO/IEC 33020 は、プロセス能力のアセスメントのためのプロセス測定フレームワークを規定した国際標準です。プロセス測定フレームワークは、プロセス能力水準として 0 から 5 までの 6 段階のレベルを規定しています。

プロセス能力を測定するためにはプロセス属性 (Process Attribute)を使います。プロセス属性とはプロセス能力を測定できる性質を定義したものです。具体的なプロセス属性の内容はプロセスアセスメントモデルである ISO/IEC 33063 で定義されています。このプロセス属性をどれくらい満たしているかどうかで、プロセス能力水準を測定できます。

全体のプロセス能力水準は、ISO/IEC/IEEE 29119-2 のプロセスの中で一番低いプロセス能力水準になります。例えば、テスト計画プロセスのプロセス属性をアセスメントした結果がレベル 1 であり、この値が一番低い結果であれば全体としてプロセス能力水準をレベル 1 と判断します。



図 26 プロセス能力水準およびプロセス属性

アセスメント指標

プロセス属性をアセスメントするために、アセスメント指標が用意されています。アセスメント指標には、プロセス能力指標とプロセスパフォーマンス指標の 2 種類あります。プロセス能力指標はプロセス能力水準のレベル 1 から 5 すべてのアセスメントに用いられます。プロセスパフォーマンス指標は、プロセス能力水準のレベル 1 のアセスメントにのみ用いられます。

アセスメント指標には、プロセス属性を達成するための情報が提供されています。それらは「プラクティス」「資源」「作業成果物」です。プラクティスを実践することでプロセス属性の達成を目指します。また、プラクティスを実践するためのインプットになる資源と、プラクティスを実践したアウトプットとしての作業成果物も示されています。

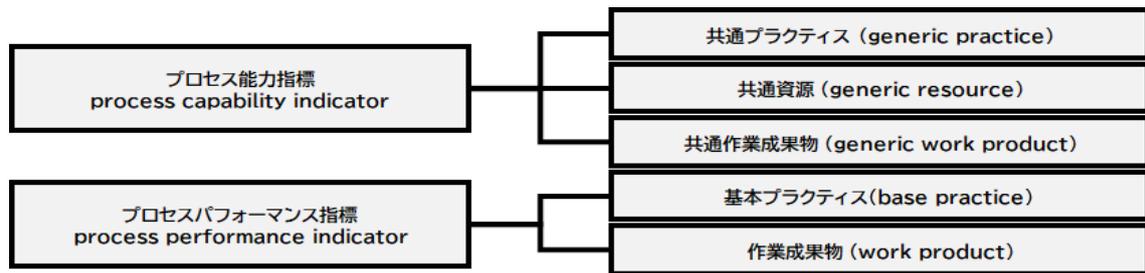


図 27 アセスメント指標

- **プラクティスの例:**「～の目標を定義する」「～のデータを分析する」「～の実行をモニタリングする」など
- **資源の例:**「人的資源」「ツール」「方法」「インフラストラクチャ」など
- **作業成果物の例:**「計画書」「報告書」「記録」など

使い方

本ガイドの執筆時点で、ISO/IEC 33063 のアセスメントを公式に提供している組織は少なくとも日本において存在しません。そのため、アセスメントは自分で行う必要があります。ここでは、自身で簡易的に実施する場合の使い方をご紹介します。

まず、JSA のサイトなどから ISO/IEC 33063 を購入してください。

現状評価

横軸の ISO/IEC/IEEE 2919-2 のプロセスごとに、プロセス属性を満たしているかアセスメントするために、基本プラクティスと共通プラクティスを表にしたものを準備します。そして、各プラクティスを達成しているものを塗りつぶしていきます。これにより、できているところとできていないところが可視化されます。また、アセスメントの結果から、プロセスごとのプロセス属性評定を求めます。プロセス属性評定の尺度には「十分達成している」「おおむね達成している」「部分的に達成している」「達成していない」の 4 種類があります (表 21 を参照)。より詳細な評定方法については、ISO/IEC 33020 を確認してください。

表 21 プロセス属性評定の尺度

尺度	意味	範囲	
N	達成していない (Not achieved)	x ≤ 15%	
P	部分的に達成している (Partially achieved)	P-	15.0% < x ≤ 32.5%
		P+	32.5% < x ≤ 50.0%
L	おおむね達成している (Largely achieved)	L-	50.0% < x ≤ 67.5%
		L+	67.5% < x ≤ 85.0%
F	充分達成している (Fully achieved)	85% < x ≤ 100%	

改善活動

アセスメント結果を踏まえて、現在のプロセス能力水準から次の水準を目指します。つまり、プロセス能力水準が一番低いプロセスを改善のスコープに設定します。一番低いプロセス能力水準のプロセスが複数有る場合は、改善のスコープとしてどこまで扱うかを整理します。そのなかのできていないプラクティスをできるように改善計画を考えます。

注意点

- あるプロセスの下位のプロセス能力水準が達成されていた場合に上位のプロセス能力水準のアセスメントを実施します。例えば、テスト計画プロセスのプロセス能力水準のレベル2が達成されていない場合は、テスト計画プロセスのレベル3についてアセスメントしなくてかまいません。もちろん、何ができていて何ができていないのかを可視化するためにすべてのプロセス能力水準のプロセス属性についてアセスメントしてもかまいません。しかし、アセスメントを行うにはかなりの時間を必要とします。特になれていない場合は、すべてをアセスメントする必要はないでしょう。
- ISO/IEC 33063 を実践するには、ISO/IEC 33063 に関連する規格についても理解を深めた方が良いでしょう。詳しくは「参考となる文献、Web ページ」を参照してください。
- ISO/IEC 33063 は日本語で提供されていません。英語の文書にあたる必要があります。

参考となる文献、WEB ページ

- ISO/IEC 33063:2015 Information technology — Process assessment — Process assessment model for software testing
 - <https://www.iso.org/standard/55154.html>
- ISO/IEC/IEEE 29119-1:2013 Software and systems engineering — Software testing — Part 1: Concepts and definitions
 - <https://www.iso.org/standard/45142.html>
- ISO/IEC/IEEE 29119-2:2013 Software and systems engineering — Software testing — Part 2: Test processes
 - <https://www.iso.org/standard/56736.html>
- ISO/IEC/IEEE 33001:2015 Information technology — Process assessment — Concepts and terminology³
 - <https://www.iso.org/standard/54175.html>
- ISO/IEC/IEEE 33004:2015 Information technology — Process assessment — Requirements for process reference, process assessment and maturity models
 - <https://www.iso.org/standard/54178.html>
- ISO/IEC/IEEE 33020:2015 Information technology — Process assessment — Process measurement framework for assessment of process capability⁴
 - <https://www.iso.org/standard/54195.html>

3 JIS X 33001:2017 - 情報技術－プロセスアセスメント－概念及び用語として JIS 化されています。また、JIS は JISC 日本産業標準調査会の Web サイトで閲覧できます。

4 JIS X 33020:2019 - 情報技術－プロセスアセスメント－プロセス能力のアセスメントのためのプロセス測定フレームワークとして JIS 化されています。また、JIS は JISC 日本産業標準調査会の Web サイトで閲覧できます。

2.3 TPI NEXT

概要

TPI NEXT は、Sogeti 社が作成しているテストプロセス改善モデルです。もともと作成していた TPI というテストプロセス改善モデルをアップデートしたもので、BDTPI とも呼ばれています。解説書が日本語に翻訳され、「TPI NEXT® ビジネス主導のテストプロセス改善」というタイトルとして出版されています。

モデルの種別

TPI NEXT は、連続モデルと段階モデルの特徴を併せ持つ混合モデルです。TPI NEXT では組織やテストプロセスを成熟度で表します。最初の成熟度のレベルからすべてのスコープを改善対象とすることは、連続モデルの特徴に当てはまりません。しかし、改善対象の着手順序は後述するクラスタによってガイドされているため、段階モデルの特徴も持っています。

	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		
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3
1 利害関係者のコミットメント	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3
2 関与の度合い	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3
3 テスト戦略	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3
4 テスト組織	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3
5 コミュニケーション	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3
6 報告	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3
7 テストプロセス管理	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3
8 見積もりと計算	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3
9 メトリクス	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3
10 欠陥管理	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3
11 テストウェア管理	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3
12 手法の実践	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3
13 テスト担当者のプロ意識	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3
14 テストケース設計	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3
15 テストツール	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3
16 テスト環境	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3

図 28 テスト成熟度マトリクスの例 (TPI NEXT)

モデルの構造

BDTPI モデルの要素は、下図のように構成されています。

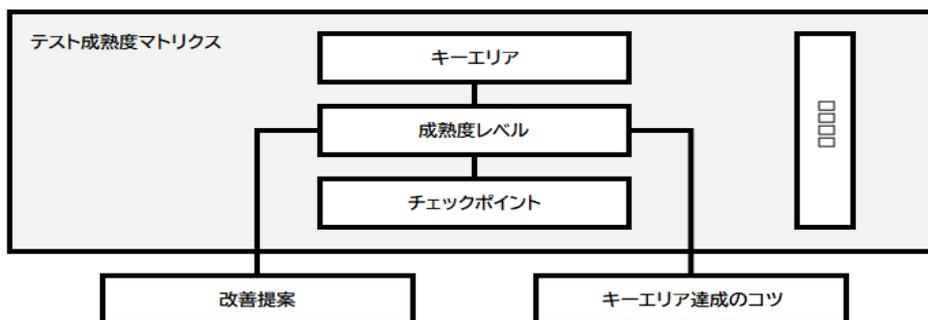


図 29 TPI NEXT の BDTPI モデル

各要素の説明

キーエリア

キーエリアとはテストプロセスのさまざまな側面を 16 個に整理したものです。この 16 のキーエリアは、次の 3 つに分類されています。

- SR(StakeholderRelation：利害関係者との関係)
- TM(Test Management：テスト管理)
- TP(Test Profession：テスト業務の専門性)

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
K01 利害関係者のコミットメント K02 関与の度合い K03 テスト戦略 K04 テスト組織 K05 コミュニケーション K06 報告	K07 テストプロセス管理 K08 見積もりと計画 K09 メトリクス K10 欠陥管理 K11 テストウェア管理	K12 手法の実践 K13 テスト担当者のプロ意識 K14 テストケース設計 K15 テストツール K16 テスト環境

図 2.10 TPI NEXT の 16 のキーエリアと 3 つの分類

成熟度レベル

キーエリアはそれぞれに成熟度レベルを持っています。成熟度レベルは次の 4 つのレベルで表されます。キーエリアの中で一番下位の成熟度レベルが、テストプロセス全体の成熟度レベルになります。

表 2.2 成熟度レベルとその概要

成熟度レベル	成熟度レベルの概要
初期レベル	アドホックな活動
コントロールレベル	適切なものごとを行う
効率化レベル	ものごとを適切に行う
最適化レベル	刻々と変化する状況に絶えず順応する

チェックポイント

チェックポイントには、何を達成すべきなのかという、BDTPI モデルの要求事項が示されています。各キーエリアは成熟度レベルごとに 2 個から 4 個程度のチェックポイントを持っています。このチェックポイントの 1 つずつについて要求事項を達成しているかを判断します。その結果によって、各キーエリアの成熟度を客観的に測定できます。なお、初期レベルにチェックポイントはありません。

改善提案

改善提案とは、満たせなかったチェックポイントをどのように満たすかを説明したものです。この改善提案の内容をヒントや助言として改善に活用できます。

なお、改善提案は満たすべき要求事項そのものではありません。また、包括的な情報を提供するものでもありません。TPI NEXT の書籍でも複数のソースから情報を得ることを進めています。その中でも Sogeti 社が提供している TMap NEXT というテストの方法論は TPI NEXT の内容を理解する上でも、実践していく上でもベースとなる情報を提供しています。

キーエリア達成のコツ

キーエリア達成のコツは、テストプロセスの改善を強化するとともに、ソフトウェア開発ライフサイクルと協調して成熟度を高める対策について示しています。ソフトウェアテストはそれのみでは語れず、必ず対応するソフトウェアの開発プロセスを考慮する必要があります。テストだけではなく開発側も考慮することで部分最適に陥らない改善活動を進められます。

クラスタ

クラスタは、どのキーエリアのチェックポイントから改善していくかを示しています。キーエリアを満たして成熟度レベルを上げていくために、こういった順番で取り組んでいくべきかのガイドを提供します。

使い方

個人やチームで独自にアセスメントする方法を紹介します。

チェックポイントの結果をプロットするには、Sogeti 社が提供している Excel で作られた TPI NEXT テスト成熟度マトリクスツールを使うと便利です。ツールにはユーザマニュアルも提供されています。なお、ツールとユーザマニュアルのどちらも日本語訳が提供されていますので、それらをダウンロードします。

現状評価

各キーエリアの成熟度レベルごとに、示されているチェックポイントを満たしているかどを判断します。その判断結果を TPI NEXT テスト成熟度マトリクスツールに記録します。その結果、それぞれのキーエリアごとの成熟度レベルとプロセス全体の成熟度レベルを概観できます。チェックポイントを満たしているか判断するためには TPI NEXT の書籍の内容を参考にします。もし判断に迷う場合はチーム内の話し合いで決めると良いでしょう。

	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		
1 利害関係者のコミットメント	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3
2 関与の度合い	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3
3 テスト戦略	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3
4 テスト組織	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3
5 コミュニケーション	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3
6 報告	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3
7 テストプロセス管理	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3
8 見積もりと計算	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3
9 メトリクス	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3
10 欠陥管理	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3
11 テストウェア管理	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3
12 手法の実践	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3
13 テスト担当者のプロ意識	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3
14 テストケース設計	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3
15 テストツール	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3
16 テスト環境	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3

図 2.11 組織の達成度を反映したテスト成熟度マトリクスの例 (TPI NEXT)

改善活動

現状評価の結果に基づいて、満たせていないチェックポイントを満たせるようにしていきます。どのチェックポイントから改善していくかを決めるために、クラスタの情報を利用します。クラスタはアルファベットで A から M までの 13 段階に分けられています。また、クラスタの優先順位は A が最初で M が最後です。優先順位が一番高いクラスタの中で、満たせていないチェックポイントから改善を始めます。改善するにあたっては改善提案やキーエリア達成のコツを参考にすることで、改善計画を立てやすくなります。なお、組織の現状に合わせてクラスタの優先順位をカスタマイズできます。詳細は TPI NEXT の書籍を参照してください。

注意点

- はじめて TPI NEXT を活用しようとする組織では、チェックポイントや改善提案を読み解くこと自体が難しい場合もあります。このため、活動の最初はテスト技術およびプロセス改善に通じた有識者やコンサルタントの協力を得ると良いでしょう。

参考となる文献、WEB ページ

- TPINEXT® ビジネス主導のテストプロセス改善
 - Alexandervan Ewijk, Bert Linker, Marcel van Oosterwijk, Ben Visser, Gerrit De Vries, Loek Wilhelmus, Rik Marselis (著) / 藪田和夫, 湯本剛, 皆川義孝 (訳) / トリフォリオ (刊) / 2015 年
- TPINEXT テスト成熟度マトリクスツールのダウンロードページ
 - <https://www.tmap.net/building-blocks/test-process-improvement-tpi>
- TMapNEXT® for result-driven testing
 - Tim Koomen, Leo van der Aalst, Bart Broekman, Michiel Vroon (著) / UTN Publishers (刊) / 2006 年

2.4 その他の技術（CTP）

概要

CTP (Critical Testing Process：最重要テストプロセス)とは、Rex Black 氏が考案したテストのプロセスモデルです。書籍「ソフトウェアテスト 12 の必勝プロセス」が出版されており、テストのプラクティスに関する詳細な説明を提供しています。

CTP ではテストプロセスを「目標を達成するために行われる一連の行動、作業、観察である」としています。その中でも最重要テストプロセスとして組織面、作業面、技術面の枠組みの中でテストチームが有益な情報やサービスを提供する能力に、直接大きな影響を与えるようなプロセスである 12 個のプロセスを示しています。

12 のプロセスは計画、準備、実行、改善の 4 つのステップに整理されます。この 4 つのステップをプロダクトのライフサイクルの間繰り返すことで、テストプロセスを改善していきます。なお、12 のプロセスの最初に示されるテストプロセスは、この 4 つのステップを包括するプロセスとして定義されています。残りの 11 のプロセスが各ステップに整理されています。

書籍の中では、架空のプロジェクトである Sumatra プロジェクトを使用してテストプロセスを説明しているため、具体的なイメージに基づいて読み進めることができます。

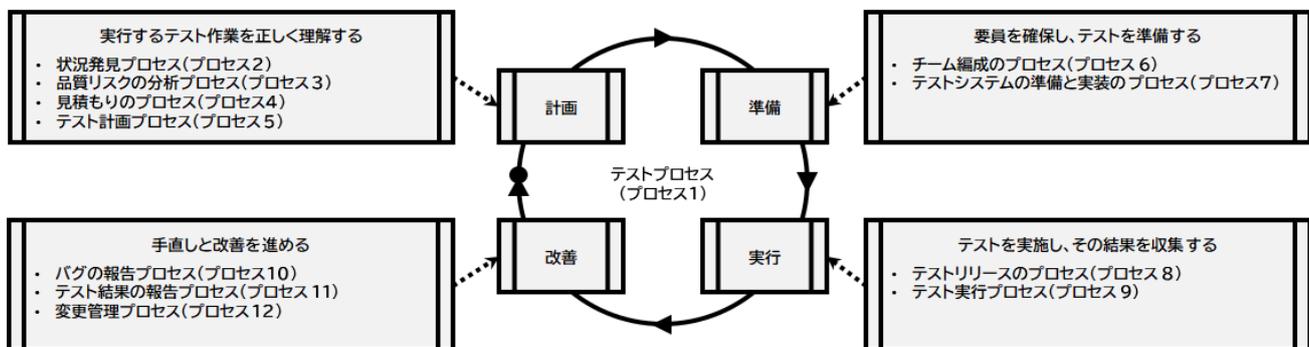


図 2.12 CTP12 の最重要プロセス

使い方

決まったアセスメント方法はありませんが、今回は書籍で提供されているチェックポイントを利用した方法をご紹介します。

現状評価

12 のプロセスそれぞれについて、チェックリストが示されています。そのチェックリストの中の各ステップを満たしているかどうかをアセスメントします。ステップを満たしているかどうかを判断するためには、書籍の内容を参考にします。もし判断に迷う場合は、チーム内で話し合っていると良いでしょう。なお、チェックリスト（英語）が Rex Black Consulting Services, Inc の Web サイトで公開されているので、利用すると良いでしょう。

改善活動

まだ満たしていないステップの中から改善するスコープを定めます。そのスコープに対する改善計画を策定して改善活動を実施します。改善するにあたっては、各プロセス⁵で示される「優れたプロセスの見分け方」「課題の克服」「改

⁵厳密にはプロセス 1 のテストプロセスとプロセス 2 の状況発見プロセスには優れたプロセスの見分け方「課題の克服」「改善へのヒント」はありません。

善へのヒント」を参考にして、チームで対策を話し合うと良いでしょう。また、Rex Black Consulting Services, Inc の Web サイトでテンプレートや例が提供されているので、それらも参考にすると良いでしょう。

注意点

- CTP に限りませんが、チェックリストを満たしたかどうかのみで善し悪しを判断しないようにしましょう。「優れたプロセスの見分け方」は、プロセスを実行した結果として得られることが期待される成果を示しています。チェックリストを満たしていると判断したときに、自分達は「優れたプロセスの見分け方」で示している成果を得られているかを合わせて考えると良いでしょう。
- プロセス同士は状況の依存性が高いため、仮にチェックリストを満たしているのに期待する成果が得られない場合は、他のプロセスの状況を考慮する必要があります。

参考となる文献、WEB ページ

- ソフトウェアテスト 12 の必勝プロセス
 - RexBlack(著)/テスト技術者交流会 (訳) /日経 BP 社 (刊) /2005 年
- RexBlackConsultingServices,Inc.(RBCS)の Web サイトで公開している CTP 関連資料のダウンロードページ
 - <https://rbc-us.com/resources/templates-and-examples/>

2.5 その他の技術 (STEP)

概要

STEP (Systematic Test and Evaluation Process)とは、Bill Hetzel 博士と David Gelperin 博士が開発したテストのプロセスモデルです。IEEE 829 Standard for Test Documentation を実装する手段として開発されました。書籍「体系的ソフトウェアテスト入門」でテストのプラクティスに関する詳細な説明を提供しています。

STEP ではテストレベルごとのテストプロセスをフェーズ、アクティビティ、タスクと段階的に詳細化しています (図 2.12 参照)。フェーズとタスクは明示的に定義されています (表 2.3 参照)。タスクを明示的に定義していませんが、書籍に記述する内容によって、何をすべきかを知ることができます。

STEP の重要な考え方の 1 つに「予防テスト」があります。予防テストとは、「システム開発ライフサイクルの非常に早い段階から実行すればテストには対象のソフトウェアの品質を実際に改善する効果がある」という考えに基づいています。つまり、「要件をテストするために (設計またはコーディングが完了する前に) テストケースを作成するプロセスを踏めば、要件定義中の欠陥を洗い出すことができる」ということです。STEP が開発された 1985 年当時としては新しい考え方でした。

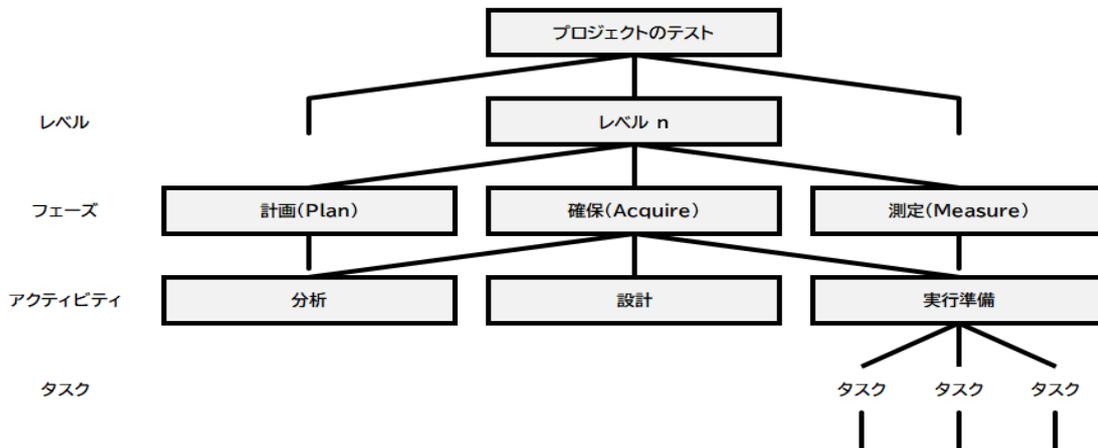


図 2.13 STEP のアーキテクチャ

ステップ1	戦略の計画
P1	マスターテスト計画の策定
P2	詳細テスト計画の立案
ステップ2	テストウェアの確保
A1	テスト対象事項明細一覧の作成 (要件ベース、設計ベース、実行準備ベース)
A2	テスト設計 (アーキテクチャと環境、要件ベース、設計ベース、実行準備ベース)
A3	計画および設計の実行準備
ステップ3	動作の測定
M1	テスト実行
M2	テストセットの妥協性のチェック
M3	ソフトウェアおよびテストプロセスの評価

表 2.3 STEP のフェーズとアクティビティ

使い方

決まったアセスメント方法はありませんが、本ガイドでは書籍で提供されている調査のテンプレートを利用した方法を解説します。

現状評価

書籍「体系的ソフトウェアテスト入門」の付録 B ではテストに関する調査テンプレートが提供されています。その付録には、「管理と測定」「評価プロセス」「テストのプロセスとアクティビティ」「テストおよび評価のツール」という 4 つの項目に対して「アクティビティ項目」が列挙されています。それぞれのアクティビティ項目について自己の組織での提供状況と、それが優れたソフトウェアを作り出すためにどの程度重要かどうかの価値を考えます。「適用状況」と「価値」をそれぞれ 0 から 3 の 4 段階で判定します。

改善活動

評価結果に基づいて、改善するスコープを定めます。例えば、より価値が高いと評価した中で、より適用状況が低いアクティビティ項目を優先するといった具合です。そのスコープにおける改善計画を策定して、改善を実施します。改善を進めるにあたっては、書籍の第 11 章「テストプロセスの改善」で示している改善プロセスを参考にするとよいでしょう。また、個々の改善項目については該当する書籍の記述内容を参考にすると良いでしょう。

なお書籍の付録 B では、「傾向と展望」の質問も提供されています。改善サイクルを回す度、質問内容を解答することで、現在の適用状況におけるソフトウェアテスト活動に関する傾向と展望を把握しやすくなります。

注意点

- STEP では IEEE 829-1998 のドキュメントテンプレート群を使用しています。しかし、IEEE 829-1998 は IEEE 829-2008 に置き換えられています。さらに、IEEE 829-2008 は ISO/IEC/IEEE 29119-3 に置き換えられています。
- IEEE 829-1998 は定義しているすべてのドキュメントを作成することを求めています。また、STEP は遵守すべきルールではなく、ガイドラインとして提示されているものです。組織やプロジェクト内容にあわせてテーラリングしましょう。

参考となる文献、WEB ページ

- 体系的ソフトウェアテスト入門
 - Rick Craig Stefan P Jaskiel (著) / 宗雅彦, 成田光彰 (訳) / 日経 BP 社 (刊) / 2004 年
- IEEE 829-1998 - IEEE Standard for Software Test Documentation
 - <https://standards.ieee.org/standard/829-1998.html>

参考文献

- CTP
 - ソフトウェアテスト 12 の必勝プロセス RexBlack（著）/日経 BP 社（刊）/2005 年
- ISO/IEC33001
 - ISO/IEC 33001:2015 Information technology— Process assessment— Concepts and terminology
 - <https://www.iso.org/standard/54175.html>
- ISO/IEC33004
 - ISO/IEC 33004:2015 Information technology — Process assessment — Requirements for process reference, process assessment and maturity models
 - <https://www.iso.org/standard/54178.html>
- ISO/IEC33020
 - ISO/IEC/IEEE 33020:2015 Information technology — Process assessment — Process measurement framework for assessment of process capability
 - <https://www.iso.org/standard/54195.html>
- ISO/IEC33063
 - ISO/IEC 33063:2015 Information technology— Process assessment— Process assessment model for software testing
 - <https://www.iso.org/standard/55154.html>
- ISO/IEC/IEEE 29119-2
 - ISO/IEC/IEEE 29119-2:2013 Software and systems engineering— Software testing— Part 2: Test processes
 - <https://www.iso.org/standard/56736.html>
- ISTQB Advanced Level シラバス テストマネージャ
 - テスト技術者資格制度 Advanced Level シラバス日本語版テストマネージャ Version 2012.J03
 - http://istqb.jp/dl/ISTQB-SyllabusAdvanced_TM_Version2012.J03.pdf
- ISTQB Expert Level シラバス テストプロセス改善
 - Certified Tester Expert Level Syllabus Improving the Testing Process (Implementing Improvement and Change) 2011
 - <https://www.istqb.org/downloads/send/12-expert-level-documents/76-expert-level-syllabus-test-management-2011.html>
- SQuBOK Guide V3
 - ソフトウェア品質知識体系ガイド(第3版):SQuBOK Guide V3
 - SQuBOK 策定部会（編）/オーム社（刊）/2020 年
- STEP
 - 体系的ソフトウェアテスト入門
 - Rick Craig, Stefan P Jaskiel（著）/成田光彰, 宗雅彦（訳）/日経 BP 出版センター（刊）/2004 年
- TMap NEXT
 - TMap NEXT® for result-driven testing

- Tim Koomen, Leo van der Aalst, Bart Broekman, Michiel Vroon (著) / Sogeti (刊) / 2014 年
- TMMi
 - Test Maturity Model integration (TMMi®) Guidelines for Test Process Improvement Release 1.2
 - <https://tmmi.org/tm6/wp-content/uploads/2018/11/TMMi-Framework-R1-2.pdf>
- TPINEXT
 - TPINEXT® ビジネス主導のテストプロセス改善
 - Alexandervan Ewijk, Bert Linker, Marcel van Oosterwijk, Ben Visser, Gerrit De Vries, Loek Wilhelmus, Rik Marselis (著) / 藪田和夫, 湯本剛, 皆川義孝(訳) / トリフォリオ (刊) / 2015 年
- TPINEXT テスト成熟度マトリクスツール
 - TPINEXT テスト成熟度マトリクスツール
 - <https://www.tmap.net/building-blocks/test-process-improvement-tpi>
 - TPINEXT テスト成熟度マトリクスツール 日本語ユーザマニュアル
 - <https://www.tmap.net/sites/tmap/files/2019-08/TPI NEXT Test Maturity Matrix tool User Manual v1.2.1 J.0.pdf>
- ソフトウェアテスト 293 の鉄則
 - ソフトウェアテスト 293 の鉄則
 - Cem Kaner, James Bach, Bret Petticord (著) / テスト技術者交流会 (訳) / 日経 BP 社 (刊) / 2003 年

商標について

下記の登録商標・商標をはじめ、本ガイドに記載されている会社名、システム名、製品名等は一般に各社の登録商標または商標です。なお、本文および図表中では、「™」、「®」は省略しております。

本文中では、次に示す登録商標に対して®の表記を省略しています。

- CMMI はカーネギーメロン大学の登録商標です
- ISTQB は International Software Testing Qualifications Board の登録商標です
- JSTQB は特定非営利活動法人ソフトウェアテスト技術振興協会の登録商標です
- STEP は Software Quality Engineering の登録商標です
- TMap, TMap NEXT, TPI, TPI NEXT は Sogeti Nederland B.V.の登録商標です
- TMMi は TMMi Foundation の登録商標です

付録 1 テストプロセス改善事例集（作成中・一部公開版）

概要

ASTER テストプロセス研究会では、テストプロセスの改善事例を収集し、それをノウハウとしてカタログ化する活動に取り組んでいます。現在 20 件ほどの事例を入手しノウハウ化の作業を進めておりますが、本ガイドではそのなかから一部を付録として公開します。

事例については随時募集しておりますので、公開可能な事例をお持ちの方がいらっしゃいましたら、事例のご提供いただけますと幸いです。また、メンバーとして事例集整備へご参画いただくことも大歓迎です。

★事例提供フォーム

<https://docs.google.com/forms/d/e/1FAIpQLSfs4WAaFeVH0FblDb1x5zwhzodD07txXLkdxQPqeZ8GFHZJg/viewform>

事例集の目的

本事例集は、ソフトウェアテストプロセス改善技術を導入、取り組んでいる方を対象に、実際に現場で実践したテストプロセス改善事例とその結果、改善案などの情報を提供し、テストプロセス改善成功への助けとしてもらうことを目的としています。

本事例集を参照することで、ソフトウェアテストプロセス改善に取り組むために必要となる情報を、実際の事例を通して学んでいただけます。

想定読者

ソフトウェアテストについての基礎的な知識を有しており（JSTQB 認定テスト技術者資格 Foundation Level を想定）、ソフトウェアテストプロセス改善の導入を検討、または取り組んでいる方々を想定しております。

本ガイドの用語

本事例集において使用する用語は次のようにしております。

- テストプロセス改善技術の説明においては、可能な限りその技術および書籍で用いられている用語を使用しています
- 各事例において、そのプロジェクトや組織固有の用語を使用している場合がありますので、適宜お読み替えください

ご注意

- 本事例集はサンプル公開です
- 事例は一部公開できない内容については編集しています
- 事例の構成や内容は、今後の作業により変更になることがあります
- 本事例の本書は、実際のプロジェクトでの事例情報を一般公開できるように編集し、テストプロセスの改善時の参考情報として利用できるように提供するものですが、現場の課題解決を保証するものではありません

① 利害関係者と積極的に情報共有せよ！

成功事例

■ 背景・プロジェクトの特徴

モバイルアプリケーション開発プロジェクトで、テストは1週間ぐらいのショートサイクルで進めていました。テストチームへテスト依頼や機能要件の共有は開発担当から急に行われることが多く、テストチームでは急ぎで機能詳細を開発者へヒアリングし、テストケースを作成する、といったことが頻繁に行われていました。大まかな機能要件はドキュメントで共有されるものの仕様書はあまり整備されておらず、仕様や詳細は開発者の頭の中です。

■ 発生した問題

開発担当者からのテスト依頼がいつも突発的に発生するため、無計画なテスト実施となり、常に場当たり対応となってしまいました。開発情報が十分に共有されないために、十分なテスト分析やテスト設計ができないままテストを実施するという状況になってしまいました。

また、ショートサイクルであるために、テストが終わったら次のテスト依頼が来るという状況で、十分に品質分析ができません。常にテスト実施に追われているため、マネージャへの報告も滞ることが多くなりました。

このような状況が利害関係者間で共有されていないため、プロジェクトマネージャやお客様といった利害関係者から頻繁にスケジュールについての問い合わせを受けることになりました。

品質状況が十分に共有されていないため、急なリリース日変更なども発生するようになりました。テストチームはそのテスト作業すべてがうまく進められない状況に陥り、混乱を極めました。

■ 実施した改善策とその結果

まずは、陥っている状況を正しくタイムリーに把握すべく、プロジェクトマネージャ、開発マネージャ、テストマネージャといった利害関係者で定期的な情報共有会を実施することにしました。

プロジェクトの全体状況や開発の状況、テストの状況を共有し、全体のプロジェクトスケジュールに対応した開発計画とテスト計画を合意し、毎週進捗を共有するようにしました。こうすることで、突発的なテスト依頼から計画的なテスト依頼に改善され、場当たり対応となっていたテスト作業が計画的にテストプロセスを回せるようになりました。このため、品質分析なども行えるようになり、その情報を利害関係者に毎週共有できるようになったため、プロジェクトスケジュールや開発スケジュールのコントロールにもよい影響がでました。マネージャからは情報共有会にてテストや品質の状況が得られるようになったため、頻繁な問い合わせもなくなりました。こうしてテストチームは計画的なテスト作業を行えるようになりました。

但し、情報共有会だけでは、開発チームからテストチームに詳細な仕様書がインプットされないということは改善されずそれは今後の課題として残りましたが、テスト計画に従って早め早めに情報をもらうというアクションが取れるようになりました。

■ 改善成功へのポイント

テスト作業を円滑に行うためには、利害関係者といかに密に情報共有し連携するかが大切です。テストチームは積極的にプロジェクトマネージャや開発チーム、社内企画やお客様といった利害関係者と情報共有しコミュニケーションを密にすることに取り組む必要があります。情報をもらうだけでなく情報を積極的に共有し、プロジェクトがよりよい方向に向かうように貢献することが大切です。

また、テストチームの活動について、スケジュールや予算といった面については、特に利害関係者の合意や支援が必要になります。理解を得るためにも積極的に利害関係者に情報共有することが大切です。

■ 関連するテスト TMMi のプロセスエリア、TPINEXT のキーエリアなど

本事例では、次にも改善のヒントがあります。改善成功へのポイントとあわせ参考にするとよいでしょう。

TMMi：「テスト方針と戦略」「テスト計画」「テスト監視とコントロール」

TPINEXT：「利害関係者とコミットメント」「テスト戦略」「コミュニケーション」

② ミーティングは目的とプロセスを定めよ！

成功事例

■ 背景・プロジェクトの特徴

組込み系の大規模開発プロジェクトにおいてテストを担当するチームとして開発に参画しています。プロジェクトの規模は大きく、複数の業務委託先が存在します。また、その一部はオフショア開発です。複数チームにより構成されるため、定例ミーティングを行って情報共有を行っていましたが、うまくいっているとは言えないと感じています。

■ 発生した問題

定例ミーティングを行うものの、いつしかチームリーダーが顔を合わせてただ会話するだけの場となってしまいました。このような状況なので、そこから特になにかアクションが生まれるわけでもなく、かといって他に得るものもありません。なんとなく他のチームの様子がわかるだけにすぎず、テストチームは時間を取って参加するモチベーションが失われてしまい、ついには欠席を重ねる状況になってしまいました。

■ 実施した改善策とその結果

まず定例ミーティングの目的を、課題共有とアクションアイテム決定の2点と定めることにしました。

ミーティングは事前にアジェンダを準備し、共有されるべき課題が漏れなく共有されるようにしました。また、課題の共有後にその場で対策を議論しアクションプラン化するように改善しました。アクションプランは具体性を持ち追跡可能とするために、SMART(※)を使って設定するようにしました。アクションの結果は、翌週以降の定例ミーティングで共有し、アクションアイテムの見直しが必要であれば実施するようにしました。また、ミーティングは開催者や司会、ファシリテーターを設け、開催連絡からフォローアップまでが確実に行われるように手順を定めました。

これらの改善により、定例ミーティングはたんなる会話の場でなくなり、課題や情報を横断的に共有できるようになったため、テストチームとしても再度積極的に参加するようになりました。

※SMART：Specific（具体的）、Measurable（測定可能）、Achievable（達成可能）、
Related（目的に沿った）、Time-bound（期限がある）

■ 改善成功へのポイント

ポイントは、ミーティングの目的とプロセスを定めたことです。

今回の事例では、目的が明確でなかったために、たんなる会話の会となってしまっており、集まる意味がなくなっていました。ミーティングは多くの方を物理的に拘束するもので大きなコストがかかります。ミーティングを開く場合には目的を定め、進行プロセスを定め、それを推進する担当者を決めることが大切です。

また、できる限り事前にアジェンダを定め、事前に資料を共有し、ミーティングはクイックに終了できるようにしましょう。共有された課題はきちんとアクションプラン化し、解決までフォローアップできるようにすることも大切です。

■ 関連するテスト TMMi のプロセスエリア、TPINEXT のキーエリアなど

本事例では、次にも改善のヒントがあります。改善成功へのポイントとあわせ参考にするとうよいでしょう。

TMMi：「テスト監視とコントロール」「テスト組織」「テストライフサイクルと統合」

TPINEXT：「関与の度合い」「テスト組織」「コミュニケーション」「テストプロセス管理」

③ コミュニケーションの齟齬を起こさないように用語を統一すべし！

成功事例

■ 背景・プロジェクトの特徴

あるプロダクトはその初回リリースから長年にわたって機能追加や保守開発を行ってきました。長期間にわたるため、開発初期から参画するベテランから、最近プロジェクトに参画した若手まで幅広いメンバーで構成されていました。

このプロジェクトは複数の外部のベンダーが参画しているほか、一部はオフショアも利用しており、メンバーの入れ替えも頻繁に発生していました。開発初期は少人数での開発でしたが、顧客と開発規模が大きくなるなか開発体制も大きくなり、それによってコミュニケーションに問題が生じるようになってきました。

■ 発生した問題

あるとき大きな機能追加が発生し、既存の体制では対応がむずかしくなったため、開発チームの増員の他、新たにテストチームを作ることになりました。このテストチームは、外部の第三者検証会社に参画新規参画して立ち上げることになりました。

テストチームでは早速テスト計画書やテストベースを確認し、開発担当者やこれまでのテスト担当者に問い合わせましたが、同種のドキュメントにおいて同じようなことを指していると思われる部分に異なる湯用語が使われており、自身の理解に確信が持てません。開発初期から参画しているベテランメンバーに確認すると、やはり同じことを指していることは分かりました。どうやら、歴代のメンバーによって使う用語が異なっていることが分かりました。

また、ミーティングや定例会でも、それぞれのチームや所属元の会社での用語や略語が使用されていたり、そうでなくても人によって用語がばらばらについていたりする状況でした。

その為、テストチームから様々な人に問い合わせても会話が噛み合いません。テストチームが使用する用語も伝わらないことが多い状況でした。

その結果、相互にコミュニケーションがうまくいかず、テストの開始時期やテスト対象、テストスコープなどの認識齟齬が発生し、テスト漏れや不要な追加テストが発生、頻繁な認識合わせを行う必要があり、大きな工数がかかってしまう状況となってしまいました。

■ 実施した改善策とその結果

テストチーム以外でも新たに増員された開発担当者もやはり同じ問題を抱えていました。このため、プロジェクトで用語を統一することにしました。テストの用語については ISTQB の用語集を標準用語集として採用しました。但し、それでもそのプロジェクトの特性によって独自の用語を作らざるをえなかったり、プロジェクトが属する企業特有の用語が存在したりするため、それらも用語集に追加することとしました。

この改善により、コミュニケーションミスによる認識齟齬が減少し、テスト漏れや作業の手戻り、本来必要ではなかった追加テストが少なくなって効率が向上しました。また、新たに参画したメンバーも用語集を活用することで立ち上がりが早くなりました。

ただし、この改善はプロジェクトを改善しましたが、関連する他のプロジェクトでは相変わらずの状況が続いています。どのように横展開していくかが課題として残りました。

■ 改善成功へのポイント

コミュニケーションにおいて、お互いに正確に理解や意図を伝えようとする場合、同じ理解の用語を使うことが大切です。このプロジェクトではコミュニケーション基盤の一つである用語集が整備されていなかったために無用なコミュニケーションミスやコミュニケーションロスを起こしていました。この問題は特に新しく参画する人が直面しやすく、開発体制の拡大時の大量増員時に顕在化します。また、複数のベンダーが参画するようなどきも起こりやすい問題です。

用語集の整備は地味で光があたりにくい施策ではありますが、コミュニケーションの基礎となるものですから、もし用語集がない場合は早急に整備するとよいでしょう。そして整備するだけでなく、お互いに用語集を意識しながらコミュニケーションをとることが大切です。

■ 関連するテスト TMMi のプロセスエリア、TPINEXT のキーエリアなど

本事例では、次にも改善のヒントがあります。改善成功へのポイントとあわせ参考にするとよいでしょう。

TMMi：「テスト組織」「ピアレビュー」

TPINEXT：「コミュニケーション」「テストウェア管理」

④ ナレッジは積極的に蓄積せよ！

■ 背景・プロジェクトの特徴

あるプロジェクトでは、システムテストの実施をオフショアベンダーに依頼していました。オフショアベンダーには仕様書とプログラムを送付し、テスト実施状況のレポートや故障レポート、終了レポートを受け取ります。オフショアベンダーではこのテストチームのアルバイトメンバーが大半であり頻繁なメンバーの入れ替わりが生じていました。

■ 発生した問題

頻繁にメンバーが入替わり、そのたびに同じような質問が依頼元に送られてきました。また、オフショアメンバー間でも長くチームに所属しているメンバーに内部での質問が集中し、本来実施すべきテスト作業がすまないうということも起こってきました。依頼元、およびオフショア先のテストチームの両方で工数が増大して、作業遅延も多く発生するようになり、お互いに不幸な状況になってしまいました。

■ 実施した改善策とその結果

まず手を付けやすいところとして、これまで設置されていなかった共通で利用する Q&A データベースを作りました。

新たな質問と回答はデータベースに登録し、質問者はまずデータベースを検索して同じ質問がない確認するように改善しました。データベース利用のプロセスフローも作成し、ベテラン担当者や依頼元に質問が集中するという状況も改善しました。ある程度データがたまってきたところで、質問者以外にも質問と回答を共有する場を作り、チーム全体での理解を深めました。

それにより、質問が大きく減少し効率が改善しました。依頼元では、データベースのデータを精査しナレッジ化することで依頼元にナレッジやノウハウを蓄積することができ、依頼するオフショアベンダーが変わっても同じような状況が起こることを防止できるほか、開発チームに展開して開発の参考にしてもらうなどの活用をしています。

■ 改善成功へのポイント

時間がない為に、ノウハウやナレッジの蓄積と共有を後回しにしてしまうことはよくありますが、この事例のように問い合わせが頻発して無視できない工数が発生することになってしまいます。情報のナレッジ化は一時的に工数がかかりますが、不必要な問い合わせが減少します。また、それを教材として教育をすることができます。また、ナレッジ化しておくことで、メンバーやベンダーが変わったとしても頼りとなる情報とすることができます。ただし、ナレッジの蓄積はだれかに情報を出してもらうことや、ドキュメント化してもらうことが必要なので、ナレッジ化することでベネフィットが得られるような仕組みにしておくことが大切です。

■ 関連するテスト TMMi のプロセスエリア、TPINEXT のキーエリアなど

本事例では、次にも改善のヒントがあります。改善成功へのポイントとあわせ参考にするとよいでしょう。

TMMi :

TPINEXT : 「テストウェア管理」「テストツール」

⑤ どういったテスト作業に時間がかかっているかを抑えるべし！

成功事例

■ 背景・プロジェクトの特徴

あるプロジェクトでは品質向上を目的にテストチームを立ち上げ、かつチーム規模を急拡大しています。立ち上げ時期ということもアリテストプロセスや体制に関してはまだ整ってはいません。規模が小さいため実際のテストにかけられるリソースも少なく、多くのメンバーは複数のテスト作業を兼務していることは当たり前で、テストマネージャもテスト実行を行うことがあるくらい、時間に追われる毎日でした。

■ 発生した問題

人的・時間的リソースが不十分ですが期限だけは決まっているため、目の前のテスト設計や実装実行を進めるのが精いっぱいな状況です。そこへプロジェクトマネージャから、品質状況の問い合わせがありました。ところが、テスト結果の分析ができていなかったため、品質がどのような状況であるか、どのようなリスクがあるのかなどを、メンバーの肌感覚でしか示せず、テスト完了の指標も明確でなかったために、リリース可否の判断が難しい状況となっていました。

■ 実施した改善策とその結果

改善すべきは「テストチームにテスト結果の分析ができるようにすること」です。

まずはテストチームがどういったテスト作業に時間を取られているのかを工数ベースのメトリクスを取得するようにして分析することにしました。これによって、時間に追われテスト分析や設計に時間をさかずにやみくもに全部の機能をテスト実行している状況がわかりました。本来テストしなくてもよい機能についてもテストしていたため余計な時間がとられていたのです。このため、テスト分析と設計に使う時間をシフトするという改善をしました。

これにより以前に比べて、明らかにテストが必要ない機能はテストを省くという高率改善ができ、ようやくテスト結果分析に時間を使うことができました。

テスト結果分析では、より全体の品質状況を把握するべく、テストの実施結果だけでなくデバッグプロセスも含めた品質分析とすることで、全体の品質状況がわかるようになりました。品質分析により、テストチームでは故障や欠陥の傾向といった情報も得られるようになったため、テスト計画時に実施プライオリティをつけられるようになったり、テスト分析や設計の際の「気を付けるポイント」として参考にすることができるようになったり、

より戦略的に、「やみくもに全部をテストする」から「戦略的に狙ってテストする」という体質に変わることができました。

■ 改善成功へのポイント

テストチームが円滑に作業を進めていくためには、どの作業にどれくらいの時間がかかっているかを把握することが大切です。

これにより、十分に頭・手をかけられているところ・不十分なところがわかりますし、必要以上に頭・手がかかっているボトルネックもわかるようになります。現実としてテストにかけられるリソースには制限があるため、こういった日々の作業の状況を抑えておくことはとても大切なことです。明らかにリソースが足りない場合体制の強化を行う必要がありますが、何人の増員が必要か、どういったテスト作業で手必要かの説明根拠となります。今回の事例では、まず工数の状況を手掛かりに改善を進めました。

その際、メトリクスという技術を使用しますが、メトリクスを計測するのはそれなりに大変なので、無闇にあれこれ取得するのではなく、現実的、継続的に取得し続けることが有益であるというメトリクスのみを、計測に関しておかかわる人で合意する必要があります。

計測するメトリクスをどう決定するかですが、そこには「目的」が必要です。どのような状況や傾向を知りたいのか、それによって計測するメトリクスは異なります。この目的から計測するメトリクスを決定するための手法としては、GQM という手法を用いるのがよいでしょう。

メトリクス取得とその分析によって、テストリソース配分の検討やテスト実施のプロセスを整備することが可能となります。

また、テスト作業を計測するうえで、テストプロセスが導入されていることも大切です。テストプロセスと作業タスクが標準化されていないと、メトリクスの計測や収集に支障をきたします。

※GQM：メトリクスを測定する目的や結果をどう解釈するかなどを明確にするモデルの一つであり、次の3段階でメトリクスを決めていきます。

① 目標を定義

目標とは、人がそこで何を知り、理解し、改善し、実行したいのかを定義するものです。

意図、論点、対象、観点を明確に定めておきます。

② 質問を定義

①で定義した目標に対して、誰が、何を、なぜ、どこで、どのように、といった質問を定義します。

③ メトリクスを定義

②で定義した質問に対するメトリクスを定義します。メトリクスは、質問に答えられるものである必要があります。1つの質問に対して複数のメトリクスが必要になる場合もありますし、逆に同一のメトリクスで複数の質問に答えることができる場合もあります。

■ 関連するテスト TMMi のプロセスエリア、TPINEXT のキーエリアなど

本事例では、次にも改善のヒントがあります。改善成功へのポイントとあわせ参考にするとよいでしょう。

TMMi：「テストライフサイクルと統合」「テスト測定」「製品品質評価」

TPINEXT：「テストプロセス管理」「見積もりと計画」「メトリクス」

付録2 テストプロセス改善スキル小説（作成中・一部公開版）

概要

ASTER テストプロセス改善技術研究会では、ソフトウェアテストプロセス改善技術を導入、取り組んでいる方を対象に、現場へのテストプロセス改善の進め方のよくあるケースとその取り組みをストーリー仕立てで説明することを目的とした小説の執筆活動に取り組んでいます。

現在ストーリーのシナリオ構想と執筆の作業を進めておりますが、本ガイドではその中からソフトウェアテストプロセス改善の序章となるストーリーを付録として公開します。

また、ストーリーのシナリオ構想と執筆にご参画いただく方を募集しております。ご自身のソフトウェアテストプロセス改善の体験談をシナリオとして書いてみたいという方も大歓迎です。

参加したい方は、次のリンク先のフォームで参加登録していただきますようお願い申し上げます。

★テストプロセス改善小説執筆活動参加登録フォーム

<https://docs.google.com/forms/d/e/1FAIpQLScCTiUofFLriSN1ZPso3k7i-Z1YonjsMCvT9P4YhSY04Km0xA/viewform>

小説の目的

本小説は、ソフトウェアテストプロセス改善技術を導入、取り組んでいる方を対象に、現場へのテストプロセス改善の進め方のよくあるケースとその取り組みをストーリー仕立てで説明したものです。

本小説書を参照することで、ソフトウェアテストプロセス改善を進める際のポイントを学んでいただくことを目的としています。

想定読者

ソフトウェアテストについての基礎的な知識を有しており（JSTQB 認定テスト技術者資格 Foundation Level を想定）、ソフトウェアテストプロセス改善の導入を検討、または取り組んでいる方々を想定しております。

本小説の用語

本小説中に利用されている技術用語は参考文献で使用されている用語を用いて記述しています。

ご注意

- 本小説はサンプル公開です
- 本小説の構成や内容は、今後の作業により変更になることがあります
- 本書のストーリーは架空の物語です。実在の企業・会社・個人名とは一切関係がありません。また、現場の課題解決を保証するものではありません。

1 テストプロセス改善ストーリーの概要

1.1 登場人物および関連図

明日太(あすた)：本作の主人公。社会人5年目。ソフトウェアテストを専門的に行う「テストエンジニア」として経験を積み、現在はテストチームのリーダーとして客先に常駐している。株式会社クオリティ所属。(架空の会社。ソフトウェアテストを事業として営んでいる。)

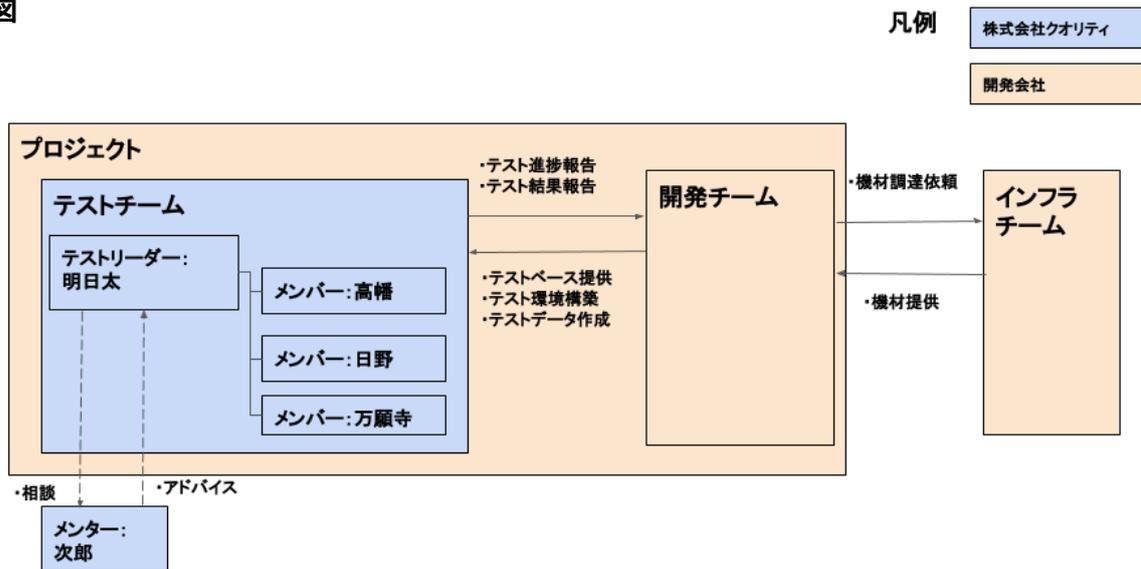
次郎(じろう)：明日太のメンター。ソフトウェアテスト歴20年以上のベテラン。豊富な知識と経験をベースに、明日太を精神的に支える。株式会社クオリティ所属。

高幡(たかはた)：明日太が率いるテストチームのメンバー。株式会社クオリティ所属。

日野(ひの)：明日太が率いるテストチームのメンバー。株式会社クオリティ所属。

万願寺(まんがんじ)：明日太が率いるテストチームのメンバー。株式会社クオリティ所属。

関連図



2 プロローグ

2.1 明日太さんの現状と課題



僕（明日太）は超困っていた。第一に、仕事の仕方が、判っているようで実は誰も何も判っていない。誰も、何もということはないが、テスト実行業務の手順書は、あると言われたものの実物を見てみれば16年も前の代物だったし、テスト設計書のテンプレート（雛形）とサンプルは、実は存在しなかった。判っているという連中は、先輩たちの仕事ぶりを見よう見まねで繰り返しているだけだった。細かい点で行き詰ると作業自体がストップする。なんてことだ。

僕は、本当に超困っていた。そんな実態が、そこそこ大規模・にこにこ短納期の開発案件で露呈した。調子がよかった.....いや、よく見えたのは自分が手がけたテスト計画づくりまでだった。直後から計画見直しの要求が頻発したことに不吉な予感を感じるべきだった。と思った時にはもう遅く、テスト活動の立て直しをする時間も取れなくなってしまっていたのは、痛いと言う以上の痛恨の極みだった。なんてことだ。

僕は、全くもってウソ偽りなく超困っていた。そんな状況で、なんとかどうにかテストの実行までこぎつけた。偉い。本当に偉い。偉いのだが、テストチーム全体がもう疲弊しているし.....そもそもテストをしても不具合（インシデント）はぼろぼろ出るし、開発チームだって満身創痍だ。そんな状態なのに、この時期に！ 仕様変更が涼しい顔をして滑り込んでくる。なんてことだ。

テストのチームリーダーをやるようになって2年。3つめのプロジェクトで、いい仕事をしてやろうと意気込んでいたところに、最悪が不幸の十字架を背負ってやってきたと思ったら全身から火を噴いた、みたいなテスト状況と来た日には、やる気のメーターがマイナスに振り切れるのも当然だろう。それも、自分がテストマネジメントを放置したせいならただの自業自得だ。そんなことはない。自分は一生懸命やってきた。そのはずだった。それにもかかわらず、テストの進捗は加速度的に遅れるし、インシデントレポートは一向に減らない。こんな状況でやる気など出るわけもない。元々ありもしない《やる気スイッチ》が跡形もなく破壊されているのだ。パニックを起こしていたとも言える。

一体、どうすればいいのだろう。いや、今この瞬間に、ではなくて。一体どうすればよかったのだろう。そして、この先、このプロジェクトが終わった後、どうすればいいのだろう。

いや、本当にどうしてこうなったということもあるし、そもそもなんでこんなことになったんだということもあるし、どうしていきのいいのだろうということもあるし.....。あれ、でも自分はどうなるのがいいと思っているのだろうというのもあるし、ホントにどうすればいいんですか。次郎さん。

2.2 明日太さんのヒーロー

僕には入社以来何かと頼っている次郎さんという先輩がいる。先輩と呼んでは申し訳ないほどのベテランエンジニアで、3年ほど前に超有名なテストベンダーから転職してきた。噂ではテスト歴25年、リーダー歴20年の歴戦の勇者だそうだ。

次郎さんが転職してきた当時、僕は入社3年目で、そろそろチームリーダーをやってみたらどうかと上司に言われていた。人を乗せるのがうまい上司で、僕もついその気になり、初リーダーの案件もトントン拍子に決まった。仕事を取ってくるのがうまい上司でもあった。僕は根拠のない意欲に満ちていたが、その一方で、もちろん、不安にまみれてもいた。

そんな時、心配なら次郎さんに相談してみたらと誰かに言われ、本社に話を聴きに行ったことが、つきあいができたきっかけだった。

本社の 5 階、喫煙所のすぐ近くの隅っこの机に、こじんまり座る次郎さんは、口数は少ないながら、僕が不安や疑問に感じるところについて、簡潔で、的確な答やアドバイスをくれた。そして、3 ヶ月後、僕は次郎さんの言葉の殆どが真実で、大半が役に立ったことを実感した。それ以来、自分や周りの仲間だけでは埒が明かない困りごとや、疑問点が溜まったら、次郎さんのところに行くようになった。

2.3 相談のセットアップ

本社の次郎さんにメールで訪問を伝えてから、このプロジェクトで起こった出来事、今の状態がどういう状態でどれ程困っているかを列挙しよう。できたら整理してみようと、僕は考えた。これは、次郎さんから、相談する時のルールとして言われていることだ。ルールと言うと固くなる。次郎さんもルールという言葉は好きじゃない。相談する時のセットアップ手続きってところか。パニックを起こした時に、自分で気づいてこのセットアップができていたら、もしかしたら、自分で解決の糸口が見つけれられたかも知れない。いや、そんなことはないだろうが。僕はいつも仕事に使っている A5 サイズのノートを広げ、4 色ボールペンをノックしつつ、この数ヶ月を振り返り始めた.....。

3 アセスメント

3.1 第一歩



何とかこの苦しい状況を打開する術はないのだろうか.....。考えながら自分なりに本を漁っていた。

「まずは自分で調べたら？」

と次郎さんに一蹴され、広大なネットの世界に打って出たところで挫折し、地元の本屋を徘徊した。

そこで一筋の光を見ることになる。

「テストプロセス改善.....これは.....っ！」

手に取った本を 1 週間もしないうちに読み切り、僕の心には完全に火が灯っていた。テストプロセス改善という名の火である。

(現状を知ることが大切と書いてあった。まずはアセスメントをする必要がある.....!!)

僕はアセスメントしてみることにした.....までは良かったが、具体的にどうしたら良いのか、本は診断のように、できているか、できていないか、と聞いてくるが、そもそも何を聞かれているのかわからない。自分なりに調べ、試行錯誤し、なんだかわかった気になってアセスメントを進めた。時に悩みつつ、論理的に組み立てて、何がどうできていないのかを考えていた。

(そもそも.....テスト設計だったら何ができていると OK なのか、ステークホルダーとの合意はしているのか。)

(テスト設計ってテスト設計できてればOK じゃないのか?)

(合意って.....お願いね。了解です。じゃダメなのか.....?)

(やっぱりわからない。何がどうするとアセスメントは完了なのか)

超困った。心に灯った火はまさに、風前の灯火だった。早すぎやしないか。風前に来るのは早すぎやしないか。そう思ったのだった。

「.....次郎さん」

やはり彼に聞くしかない。

3.2 アセスメントってもんは

いきなり、やってはみたけど、やっぱりよくわかりませんといっても答えてもらえないことはわかっている。次郎さんはそういう男だ。そんなことを考えつつも、何も思いつかない状態で勢いよく次郎さんの元へ飛び込んでしまうのだった。

「やっぱり読んだだけじゃダメだよな。」

「は、はい.....(それを先に言ってほしかった.....)」

普段口数の少ない彼が、ゆっくりと、論すように話し始めた.....と思ったら徐々に熱量を上げ、早口で、少々まくしたて気味に話し始めた。

「アセスメントするにも技術が必要なんだよね。ドキュメントやインタビューで情報収集して、その結果を文書化しているような角度、視点から見るとでしょ?そしたら例えばそれらの結果から強み、弱み、機会、脅威の判断とか SWOT 分析⁶するんだけど、その時にテスト関係者だけじゃなくてエンドユーザ、環境、保守メンバーとか色々関係してくる人も考慮して.....ああ、この辺の話は TPI NEXT に書いてあるから、それを見てもいいかも。あとはアセスメント結果を数値化すると説得材料にできるけど、その時は何をどう数値化するか、定性的に観察する方法も考えないといけないし、モデルも色々試してみるのもいいかもな.....ちょっと難しいかな。その上で今回のアセスメントやってみないと最初はつらいよね、きっと。」

最初は子守歌かと思ったが.....これは.....何を言っているのか。

「ちょっと、何を言ってるのか、わからないです。」

それを聞いて次郎さんは「ほほー」と顎を撫でながら、軽く突き放す。

「とりあえず、いいからやってみたら?」

その後は色々攻防(質問のやり取り)があり.....一応形になったのは数日後のことだった。

「いいんじゃないの。とりあえずこんな感じで」

資料を見ながら、次郎さんは僕に資料を戻したのだった。

⁶ 現状分析のためのフレームワークです。現在置かれている環境を、強み、弱み、機会、脅威の4象限に分けて分析します。

3.3 優しいクロスカウンター



次郎さんのアドバイスももらい、アセスメントを進めることができた。自分としてもよくできている気がしたし、読んで本達の理解が深まった気がしている。

なんだかカイゼンは進んでいないのに、とても清々しい気分になっていた。

意気揚々とアセスメント結果を僕とも歳が離れているわけでもなく、比較的気軽に話せるテストチームのステークホルダーに報告した。そんな僕はまたもカイゼンというものの難しさを思い知ることになる。それは悲しくも優しいクロスカウンターだった。

「うん.....アセスメント.....結果か.....」

「うん.....」

「これ、なんか、できてないことだらけで悲しくなるね。」

(ご、ごめんね.....)

そう喉まで出かかったが、強い意志を持って提示した僕としては引き下がれない.....と思ったが、できてないことばかりではなくて、できているところを提示できないか、ということに気付かされた。

悪い面ばかり目が行くが、何かできているか、良い面を見ることも大事だ。叱責されてばかりでは、大抵の人は委縮してしまおうし、反発も起きてしまう。

属人化しつつもうまくいっている部分もあるし、フォーマットは決まっていなくても合わせようという意識はメンバー内にある。メトリクスは組織的には取れていないが、個人で自分の作業に対する質を定量化している人もいる。カイゼンに向けてのポジティブな部分も実はあるのではないか。

僕は良い面を盛り込んだ形で再提出することにした。

3.4 アセスメント結果、再提出

アセスメント結果は具体的にすることを心掛けた。加えて、捻出できる工数、緊急性、重要度から3つを判断した。

「まず改善活動のための余裕を捻出することが必要.....だよな.....多分.....きっと」

長期戦になることは覚悟していた。

すぐに次郎さんに泣きついていた時とは違うのだ。気が早くも成長していることを実感している。

僕としては、次の点が現状最も表面化している問題点だと感じている。

表面化している問題点

- ネガティブな面
 - テストケースの問題
 - ◇ トレーサビリティ⁷が取れない
 - ◇ 構成管理ができていない
 - 時間外労働
 - ◇ 障害が多い
 - ◇ 仕様の問い合わせしっぱなし
 - ◇ 短納期のために稼働UP
 - コミュニケーションの問題
 - ◇ 仕様変更が急に来る（開発側との連携不足）
 - ◇ チーム間の連携不足、障害/仕様確認過多
- ポジティブな面
 - できないなりに、何とかしている
 - ◇ 仕様不明確だから、問い合わせしまくり
 - 期限は守れている
 - ◇ 内部では稼働がガンガン上がっている
 - ◇ ギリギリ保っているだけで、何か起きたらアウト

属人化したフォーマットも問題だが、その前にそもそもテストケースの構成管理もできていないし、トレーサビリティも取れない。時間外労働も増えていき、こういった面の手当でもできない。おまけに席が隣であれば問題ないが、ちょっと席が離れてしまうとすぐにコミュニケーションロスが発生する。

（これをもってテストプロセス改善計画を立ててみよう……。）

こうして、僕のテストプロセス改善は、ようやく第一歩を踏み出したのだった。

※つづく。。。 (今後公開予定)

⁷ 例えば、設計書に記載されている事物が、テストケースに記載されていることがビジュアルに確認できること。

巻末

テストプロセス改善技術 入門ガイド 執筆メンバー一覧

本ガイドは、ソフトウェアテスト技術振興協会（ASTER）テストプロセス改善技術研究会のメンバーで作成しました。作成に関わったメンバーを次に示します。（敬称略、50音順）

- リーダー
 - 池田暁（日立オートモティブシステムズ）
 - 山崎崇（ベリサーブ）
- パートリーダー（SWG リーダー）
 - 上田卓（ソーバル） *事例収集 SWG
 - 廣町幸治 *スキルガイド SWG
 - 藤原考功（ユーザベース） *スキルガイド SWG
 - 吉澤麻由（SHIFT） *初心者向けガイド SWG
- メンバー
 - 朝倉正範（ウフル）
 - 池田健太郎（SHIFT）
 - 上田卓（ソーバル）
 - 焰硝岩益土（日立産業制御ソリューションズ）
 - 大段智広（豆蔵）
 - 小野瞬（エスイーディー）
 - 近藤康二（SHIFT）
 - 佐藤浩一（日本ナレッジ）
 - 関仁慈（エイネット）
 - 泰楽無雅
 - 高坂茂樹（エスイーディー）
 - 田井康平（ノハナ）
 - 春田優子
 - 望月信昭（日本ナレッジ）

注釈:所属はガイド執筆時点の所属であり、現在の所属と異なる場合があります。

免責事項

本ガイドを利用したことで発生するいかなるトラブル、損失、損害などに対して、一切の責任を負いません。また、記載内容は執筆当時の情報です。特にWebサイトの情報は変更になっている可能性があります。

奥付

テストプロセス改善技術 入門ガイド

2020年12月15日Version1.0.0版発行

著者 ASTER テストプロセス改善技術研究会
発行 NPO 法人ソフトウェアテスト技術振興協会

Copyright(c) ソフトウェアテスト技術振興協会 (ASTER) All right reserved.