

# テスト技術者認定資格制度

## Expert Level シラバス日本語版 テストマネジメント

(テスト、テスト担当者、テストステークホルダーのマネジメント)

2011.J01

---

International Software Testing Qualifications Board

---



### 著作権

本書は、出典を明記する場合に限り、全文を複製すること、または一部を抜粋して使用することができる。

Copyright © International Software Testing Qualifications Board (hereinafter called ISTQB®).

Graham Bath, Rex Black, Debra Friedenberg, 大西建児, Marcel Kwakernaak, Martin Klonk, Judy McKay, Gary Mogyorodi, Erik van Veenendaal (2010 – 2011)

Translation Copyright © 2005-2025, Japan Software Testing Qualifications Board (JSTQB®), all rights reserved.

日本語翻訳版の著作権は JSTQB®が有するものです。本書の全部、または一部を無断で複製し利用することは、著作権法の例外を除き、禁じられています。

## 改訂履歴

### ISTQB®

バージョン	日付	備考
2011Bata	2011年4月14日	ベータ版
2011Release	2011年11月11日	リリース版

### JSTQB®

バージョン	日付	備考
Version 2011.J01	2025年12月9日	2011の日本語翻訳版

## 目次

改訂履歴 .....	3
目次 .....	4
謝辞 .....	7
1. 本シラバスの紹介 60 分 .....	8
1.1 The International Software Testing Qualifications Board .....	8
1.2 本シラバスの目的 .....	8
1.3 ソフトウェアテストにおける認定テスト担当者 Expert Level .....	8
1.3.1 期待する姿 .....	8
1.3.2 受験資格および更新要件 .....	9
1.3.3 知識レベル .....	9
1.3.4 資格認定試験 .....	9
1.3.5 認定審査 .....	9
1.4 規範的な部分と参考的な部分 .....	9
1.5 詳細レベル .....	10
1.6 本シラバスの構成 .....	10
1.7 用語と定義 .....	10
2. テストミッション、ポリシー、戦略、ゴール 555 分 .....	11
2.1 イントロダクション .....	12
2.2 ミッション、ポリシー、成功のメトリクス .....	12
2.3 テスト戦略 .....	13
2.3.1 分析的戦略 .....	13
2.3.2 モデルベースの戦略 .....	14
2.3.3 系統的戦略 .....	14
2.3.4 プロセス準拠または規格準拠戦略 .....	15
2.3.5 対処的戦略 .....	15
2.3.6 コンサルティブ戦略 .....	15
2.3.7 リグレッション回避とテスト自動化戦略 .....	16
2.4 テストポリシーとテスト戦略の組織との整合 .....	16
3. テストチームのマネジメント 855 分 .....	18
3.1 イントロダクション .....	18
3.2 テストチームの構築 .....	19
3.2.1 ジョブディスクリプション .....	19
3.2.2 履歴書 .....	19
3.2.3 面接 .....	20
3.2.4 新規メンバーの受け入れ .....	21
3.2.5 雇用の終了 .....	22
3.2.6 契約関係の終了 .....	22
3.3 テストチームの育成 .....	22
3.3.1 個人の育成 .....	22
3.3.2 ゴールと目標の設定 .....	23
3.3.3 明確な役割と責任の定義 .....	23
3.3.4 個人の性格とチーム内での役割 .....	24
3.3.5 スキル開発 .....	24
3.3.6 トレーニングの機会 .....	25
3.3.7 メンタリング .....	25

3.3.8 パフォーマンス評価とフィードバック .....	25
3.4 テストチームのリード .....	26
3.4.1 情報共有とコミュニケーション .....	26
3.4.2 忠誠心と信頼の醸成 .....	26
3.4.3 チームビルディング .....	26
3.4.4 テストチームのモチベーション向上と挑戦の機会の提供 .....	27
3.4.5 分散したチームのマネジメント .....	27
4. 対外関係のマネジメント 330 分 .....	28
4.1 イントロダクション .....	29
4.2 対外関係のタイプ .....	29
4.3 契約上の問題 .....	29
4.4 コミュニケーション戦略 .....	30
4.5 外部ソースからの統合 .....	30
4.6 テスト戦略のマージ .....	31
4.7 品質の検証 .....	32
5. 組織横断マネジメント 780 分 .....	34
5.1 イントロダクション .....	35
5.2 テストチームの価値の提示 .....	35
5.2.1 テスト組織の宣伝と価値の提示 .....	35
5.2.2 テストの価値の売り込み .....	35
5.2.3 説明責任を果たせるチームの構築 .....	36
5.2.4 チームの保護と支援 .....	36
5.3 テストチームの配置 .....	38
5.4 ステークホルダーコミュニケーション .....	38
5.5 関係の形成と発展 .....	39
5.6 組織全体にわたる品質活動の推進 .....	40
5.7 組織全体にわたるツールの統合 .....	41
5.8 倫理的問題への対処 .....	42
5.8.1 チームにおける行動規範のマネジメント .....	42
5.8.2 テストのステークホルダーとのやりとり .....	43
5.8.3 レポート結果 .....	43
5.8.4 テストマネジメントの倫理 .....	43
6. プロジェクトマネジメントエッセンシャルズ 615 分 .....	44
6.1 イントロダクション .....	45
6.2 プロジェクトマネジメントタスク .....	45
6.2.1 テスト見積り .....	45
6.2.2 テストスケジュールの決定 .....	47
6.2.3 予算編成とリソース割り当て .....	47
6.2.4 プロジェクトのマネジメントと追跡 .....	48
6.2.5 トレードオフへの対処 .....	49
6.2.6 変更マネジメント .....	49
6.2.7 タイムマネジメント .....	50
6.3 プロジェクトリスクマネジメント .....	50
6.3.1 プロジェクトリスクのマネジメント .....	50
6.3.2 プロジェクト全体のリスクマネジメントへの参加 .....	51
6.4 品質マネジメント .....	51
7. テストプロジェクトの評価とレポート 510 分 .....	53
7.1 イントロダクション .....	53

7.2	追跡する情報 .....	53
7.3	情報の評価と活用 - 内部報告 .....	54
7.4	情報の共有 - 外部への報告 .....	55
7.5	テスト結果のレポート作業と解釈 .....	56
7.5.1	計画、モニタリング、コントロール .....	56
7.5.2	分析と設計 .....	57
7.5.3	実装と実行 .....	57
7.5.4	終了基準と報告の評価 .....	58
7.5.5	テスト終了作業 .....	59
7.6	統計的品質管理手法 .....	59
8.	ドメインとプロジェクト要因に関するテスト上の考慮事項 270 分 .....	60
8.1	イントロダクション .....	61
8.2	ライフサイクルモデルのテストマネジメントに関する考慮事項 .....	61
8.2.1	ライフサイクルモデルの比較 .....	61
8.2.2	アジャイル手法 .....	62
8.3	部分的なライフサイクルプロジェクトのマネジメント .....	64
8.3.1	統合プロジェクト .....	64
8.3.2	メンテナンスプロジェクト .....	65
8.3.3	ハードウェア/ソフトウェアおよび組込みシステム .....	65
8.3.4	セーフティクリティカルシステム .....	66
8.4	リリースに関する考慮事項 .....	66
8.4.1	市場の需要 .....	66
8.4.2	メンテナンスの容易さ .....	66
8.4.3	インストールの容易さ .....	67
9.	有効性と効率性の評価 195 分 .....	68
9.1	イントロダクション .....	69
9.2	テストプロセスの有効性、効率性および満足度メトリクス .....	69
9.3	テストポリシーの目的に対する有効性、効率性、満足度のメトリクス .....	70
9.4	プロジェクトのふりかえり .....	71
10.	参考文献 .....	72
10.1	ISTQB ドキュメント .....	72
10.2	商標 .....	72
10.3	書籍 (日本語翻訳が出版されているものは追記している) .....	73
10.4	論文/記事 .....	74
10.5	Web 参考文献 .....	74
11.	付録 A – 教育機関への通知 .....	76
11.1	研修時間 .....	76
11.2	職場における実践演習のガイドライン .....	76
11.3	コース計画の例 .....	77
11.3.1	コース計画例 - 職場での学習の目的なし .....	78
11.3.2	コース計画例 - 職場での学習の目的を含む .....	78
11.4	e ラーニングのルール .....	78
11.5	標準の利用 .....	79

## 謝辞

このドキュメントは、International Software Testing Qualifications Board の Expert Level 作業部会の執筆チームによって、モジュール「テストマネジメント (Test Management)」のために作成した：

Graham Bath (ISTQB Chair, Expert Level)

Rex Black (Chair for the Expert Level Test Management syllabus)

Debra Friedenberg

大西建児

Marcel Kwakernaak

Martin Klonk

Judy McKay (Vice-Chair for the Expert Level Test Management syllabus)

Gary Mogyorodi

Erik van Veenendaal (ISTQB Vice-Chair, Expert Level)

執筆チームは、レビューチームとすべての国内委員会の提案と意見に感謝する。

このモジュールの Expert Level レベルシラバスが完成した時点で、エキスパートレベル作業部会 (テスト マネジメント) のメンバーは以下の通りである。(アルファベット順)：

Graham Bath

Debra Friedenberg

Kari Kakkonen

Vipul Kocher

Gary Mogyorodi

Andrea Rott

Rex Black

Simon Frankish

Marcel Kwakernaak

Steen Lerche-Jensen

Joanna Nowakowska

Dakar Shalom

Monika Boegge

Leanne Howard

Martin Klonk

Judy McKay

Thomas Rossner

Erik van Veenendaal

次のメンバーが、本シラバスのレビュー、意見表明、および投票に参加した。(アルファベット順)：

Graham Bath

Monika Boegge

Bernard Homés

Beata Karpinska

Alon Linetzki

Ingvar Nordstrom

Andy Soundranagam

Gualtiero Bazzana

Piet de Roo

Leanne Howard

Marcel Kwakernaak

Judy McKay

Thomas Rossner

Jurian van de Laar

Rex Black

Debra Friedenberg

Kari Kakkonen

Martin Klonk

Gary Mogyorodi

Andrea Rott

Erik van Veenendaal

本ドキュメントは、2011年11月1日にISTQB®の総会によって正式にリリースされた。

日本語訳については、以下の日本語翻訳ワーキンググループメンバーにより行われた。

日本語翻訳ワーキンググループメンバー

亀井亮介(株式会社マネーフォワード)

森田麻沙美(株式会社 Voicy)

高木 陽平(株式会社 AGEST／JSTQB 技術委員)

湯本剛(株式会社 ytte Lab／JSTQB 技術委員)

## 1. 本シラバスの紹介 60 分

キーワード

なし

本シラバスの紹介の学習の目的

なし

### 1.1 The International Software Testing Qualifications Board

The International Software Testing Qualifications Board(以下 ISTQB®)は、世界中の国または地域を代表する委員会によって構成されている。ISTQB®の構成および会員資格の詳細については、[ISTQB-Web]で参照できる。

### 1.2 本シラバスの目的

本シラバスは、テストマネジメント向けの国際ソフトウェアテスト認定資格 **Expert Level** のベースとなる。ISTQB®は、このシラバスを以下の主旨で提供する：

- メンバー委員会に対し、各国語への翻訳および教育機関の認定の目的で提供する。メンバー委員会は、本シラバスを各言語の必要性に合わせて調整し、出版事情に合わせてリファレンスを追加することができる。
- 認定委員会に対し、本シラバスの学習目的に合わせ、各国語で試験問題を作成する目的で提供する。
- 教育機関に対し、教材を作成し、適切な教育方法を確定できるようにする目的で提供する。
- 受験志願者に対し、認定試験準備の目的で提供する(研修コースの一部として、または独立した形態で実施)。
- 國際的なソフトウェアおよびシステムエンジニアリングのコミュニティーに対し、ソフトウェアやシステムをテストする技能の向上を目的とする他、書籍や記事を執筆する際の参考として提供する。

ISTQB®では、事前に書面による申請があり ISTQB から承認された場合に限り、第三者がこのシラバスを先に定めた以外の目的での使用を許諾することがある。

### 1.3 ソフトウェアテストにおける認定テスト担当者 **Expert Level**

#### 1.3.1 期待する姿

**Expert Level** 資格は、ソフトウェアテストの分野すでに高度なキャリアを築いており、特定の領域における専門性をさらに高めたいと考えている人々を対象としている。**Expert Level** で提供されるモジュールは、テストに関する幅広いトピックをカバーしている。

テストマネジメント分野におけるテスト専門家(エキスパート)とは、テスト全般に関する幅広い知識を持ち、特定のテスト領域において深い理解を有している人物のことである。ここでいう「深い理解」とは、テストに関する理論

および実践に関する十分な知識を持ち、テストプロセスの策定、実装、実行において、組織やプロジェクトの方向性に影響を与えるられるレベルであることを意味する。

Expert Level モジュールの概要[ISTQB-EL-Modules]では、このモジュールのビジネス成果を示している。

### 1.3.2 受験資格および更新要件

Expert Level の受験資格は、ISTQB のウェブサイト[ISTQB-Web]の Expert Level ページに記載がある。

これらの受験資格に加え、Expert Level のテストマネジメント資格を取得するためには、次のいずれかの認定資格を保有していなければならない：

- Advanced Level Test Manager (CTAL-TM)
- ISEB Practitioner (2008 年 3 月 31 日までに受験)
- ISEB Test Management Practitioner (2008 年 4 月 1 日以降)

上記を除き、他の ISEB 認定資格は、Expert Level テストマネジメント認定資格の受験資格としては認められない。

特定のモジュールの Expert Level 保有者は、ISTQB Continuous Learning Concept [ISTQB-EL-CEP] の最低単位数を達成し、定期的に認定を更新しなければならない。

### 1.3.3 知識レベル

本シラバスの各章の学習の目的は、明確に識別できるように各章の冒頭に記載されている。シラバスの各トピックは、そのトピックに割り当てられた学習の目的に従って評価する。

学習の目的に割り当てられた知識レベルは、ISTQB Web サイト[ISTQB-Web]に記載されている。

### 1.3.4 資格認定試験

すべての Expert Level 試験は、本シラバスに加え、Advanced Level シラバスのテストマネジメントモジュール、および Foundation Level シラバスに基づくものとする。試験問題に対する解答は本シラバスの複数の節を基にしている場合がある。

試験の形式は、ISTQB の Web サイト[ISTQB-Web]の Expert Level ページに記載がある。また、ISTQB の Web サイトには、受験者に役立つ情報もいくつか掲載されている。

### 1.3.5 認定審査

各国の委員会は、このシラバスに従ったコースを提供する教育機関を認定することができる。

ISTQB の ISTQB のウェブサイト[ISTQB-Web]の Expert Level ページには、コースの認定に関して教育機関に適用する具体的な規則が記載されている。

## 1.4 規範的な部分と参考的な部分

シラバスの規範的な部分は、試験の出題対象となる内容です。具体的には以下が該当する：

- 学習の目的
- キーワード

- 職場での実施が求められる演習

シラバスの残りの部分は参考的な部分で、学習の目的に対する理解を深めるための補足説明である。

## 1.5 詳細レベル

このシラバスの詳細レベルは、国際的に一貫した指導と試験を可能にする。このゴールを達成するために、シラバスは以下のように構成されている：

- 知識分野ごとに、認知的な学習成果と達成すべきマインドセットを記述した学習の目的(これらは規範的なものである)。
- 教えるべき重要な概念の説明、一般に認められた文献や基準などの情報源、そして必要に応じて追加情報源への参照を含む、教えるべき情報のリスト(これらは参考情報である)。

シラバスは、テストマネジメントの知識分野全体を説明するものではなく、認定された **Expert Level** の研修コースでカバーされる詳細レベルを反映したものである。

## 1.6 本シラバスの構成

大きく分けて 8 つの章がある。最上位の見出しあは、その章の時間を示している。例えば

2. テストミッション、ポリシー、戦略、ゴール 555 分

この例は、「第 2 章」の内容を教えるのに 555 分 を想定していることを意味している。また、各章の冒頭には、その章に対応する具体的な学習の目的が列記されている。

## 1.7 用語と定義

ソフトウェア関連文献で使われる多くの用語は、しばしば同義語として使われている。本 **Expert Level** シラバスにおける定義は、[ISTQB-Glossary] に準拠している。

本 **Expert Level** シラバスの各章の冒頭に挙げられているキーワードは、すべて [ISTQB-Glossary] にて定義されている。

## 2. テストミッション、ポリシー、戦略、ゴール 555 分

### キーワード

分析的テスト戦略、コンサルティブテスト戦略、系統的テスト戦略、モデルベースのテスト戦略、運用プロファイル、運用プロファイリング、プロセス準拠テスト戦略、対処的戦略、リグレッション回避テスト戦略、標準準拠テスト戦略、テストミッション、テストポリシー、テスト戦略

### テストミッション、ポリシー、戦略、ゴールの学習の目的

#### 2.1 イントロダクション

本節の学習の目的はない。

#### 2.2 ミッション、ポリシー、成功のメトリクス

- LO 2.2.1 (K4) 組織の品質ゴールについて議論し、そのゴールに合致したテストチームのミッションを定義する。
- LO 2.2.2 (K6) ソフトウェア、品質、依存関係(組織的、経済的、技術的)に長期的な効果をもたらす可能性のあるテスト目的、優先度、テストゴールを定義、記述、評価する。
- LO 2.2.3 (K3) 状況に応じて、テストプロセスの有効性、効率性、満足度のメトリクスを定義し、改善の機会を識別する。

#### 2.3 テスト戦略

- LO 2.3.1 (K4) 特定の状況を分析し、その状況においてどの個別戦略または融合戦略が最も成功しやすいかを判断する。
- LO 2.3.2 (K4) 特定のテスト戦略が失敗した状況を分析し、失敗の可能性が高い原因を識別する。
- LO 2.3.3 (K6) 特定の状況について、重要な要素を含むテスト戦略文書を作成し、成功要因を促進し、失敗の原因を軽減する。

#### 2.4 テストポリシーとテスト戦略の組織との整合

- LO 2.4.1 (K6) テストプロセスとテスト組織、人、ツール、システム、および技法を含む、企業、組織、および/またはテストチームのテストポリシーと戦略を長期的および短期的に定義、記述、評価、および改善する。

## 2.1 イントロダクション

たわいもない娯楽を除けば、すべての旅には意義があり、目指す場所があり、目的がある。複雑度の高い旅には、目指す場所に到着するための地図が必要である。テストの意義は、そのミッション(「なぜ」)であり、テストポリシーは目指す場所と目的(「何」)を定義し、テスト戦略は目指す場所までの地図(「どのように」)を提供する。

## 2.2 ミッション、ポリシー、成功のメトリクス

テスト活動そのものは、ビジネスとしての価値があるわけではない。テストが価値あるものと見なされるのは、それが、対象となるシステムの品質および/またはテストに関心を持つ1人以上の主要なステークホルダーにとって、価値ある何かと結びついているときである。テストマネジメントの最も本質的な活動は、この結びつきを確立するテストマネージャーの能力に関するものである。このようなつながりがなければ、テストチームが技術的な能力としていかに優れても、テストマネージャーによるテストチームのマネジメントがいかに優れても、テストチームとテストマネージャーは失敗する。従って、テストマネージャーは、ステークホルダーの目的とテストとのつながりを注意深く評価し、その目的に関連するテストのゴールとミッションを定義しなければならない。

テスト組織の具体的なミッションと目的はさまざまであるが、典型的な目的には以下がある:

- 顧客満足度またはユーザー満足度に影響する可能性のある欠陥を発見、そして開発者(または他の作業成果物作成者)がリリース前に修正できるようそれらの欠陥に関する十分な情報を提供する。
- 主要な品質リスクに関連する重要なテストをリスク順に実行することでリスクをマネジメントし、リリース前にシステムの品質リスクを既知の許容レベルまで低減する。
- 共通の手順、品質ゲート、サービスレベルアグリーメント(SLA)、引き渡しポイントに関して、他のプロジェクトステークホルダーとの整合性を確保する。
- 品質、テスト、リリース準備に関する重要な情報をプロジェクトチームに提供する。
- 致命的でない欠陥を発見した際には、それらに対する回避策の確立を支援する—リリース前に欠陥が修正されなかつた場合でも、カスタマーサポートやヘルプデスクがユーザーの問題を解決できるようになるためである。

テスト組織のミッションと具体的な目的は、より広い文脈によって異なる。例えば:

- 市場ベース、あるいは規制によるプロダクトのリリース期限は、テスト組織が本来達成できるはずの欠陥検出率を低下させてしまう可能性がある。
- 選択したソフトウェア開発ライフサイクルは、テスト組織が達成できるリスクカバレッジの範囲を制限することがある。
- ビジネス上の制約や優先度によって、テスト組織が実施できるテスト量が制限されることがある。

これらの変数の範囲内で、テストマネージャーはテスト組織の目的を最大限に達成することを追求しなければならない。そのためには、テスト組織の目的およびその目的に対するゴールを設定する際に、これらの変数を注意深く評価し、再評価する必要がある。

テスト組織のミッションと目的を明確にすることは重要である。この明確性を確立するための実践事例としては、テストポリシー文書を作成することがある。この文書には、組織の品質ポリシーとの関係を含め、テスト組織の適切なミッションとゴールを反映させるべきである。テストマネージャーは、予算、スケジュール、ツールなどの制約を考慮した上で、このポリシーはテスト組織が実際に達成できることを反映していることを確実にしなければならない。さらに、テストマネージャーは、ある組織や状況について、定期的にテストポリシーを評価し、可能な改善を特定する必要がある。

テストポリシーでは、測定可能な形でミッションと目的を定めるべきである。テスト組織のために定義されたすべての目的には、有効性、効率性、および/またはステークホルダーの満足度に関する明確なメトリクス(とゴール)

を含めるべきである。ゴールが測定可能でない場合、テストマネージャーはしばしば、成功の判断が困難で、ステークホルダーの期待が達成不可能で、失敗する可能性をはらんでいるという状況にあることを認識する。

基本的なテストプロセス全体にわたる有効性、効率性、およびステークホルダーの満足度に関するメトリクスの例としては、以下のようなものがある：

- テストの分析と設計：テスト組織は、最も重要な品質リスクをカバーする一連のテストケースを開発する（有効性）。
- テストの実装と実行：テスト組織は、自動テストケースを実行することによって、リグレッションリスクのカバレッジを高める（すなわち、リスクを低減する）（効率性）。
- テスト結果のレポート作業と終了基準の評価：プロジェクトチームは、報告されたテスト結果が、プロジェクトを成功に導く上で役立つことを認識する（満足度）。

テストマネージャーは、主要なステークホルダーと協議の上、これらのメトリクス（ゴール）に対する目標値を設定し、時間の経過とともに改善するためのゴールを設定すべきである（テスト技術者資格制度 Expert Level シラバス テストプロセス改善を参照）。

## 2.3 テスト戦略

テスト戦略は、テスト組織がテストポリシーを実装するための手段である。つまり、テスト戦略は、**どのようにテストを行うか**を説明し、テストポリシーは、**なぜテストを行うか**を説明する。各プロジェクトには固有の特性があるため、テストアプローチは、特定のプロジェクトに対するテスト戦略の実装である。

テストマネージャーのエキスパートは以下のことができる：

- 特定の状況を分析し、どの戦略が最も成功しやすいかを判断する。
- テスト戦略の利点を評価し、さまざまなステークホルダーに説明する。
- テスト戦略が、失敗の原因となり得る要因を特定できなかった状況を評価する。
- 選択した戦略が、組織のスキル、リソース、形式的なレベルに適していることを確認する。

以下の項では、最も一般的なテスト戦略、その利点、成功要因、リスク、および各戦略と主要なテストステークホルダーとの関係について検討する。テストマネージャーは、さまざまな戦略の利点を最大化し、リスクを最小化するために、複数の戦略を組み合わせることがプロジェクトにとって最良のアプローチであると判断する場合がある。

### 2.3.1 分析的戦略

分析的テスト戦略には、最も一般的なテスト戦略である、要件ベースのテストとリスクベースドテストの 2 つが含まれる。テストチームは、分析的テスト戦略に従って、テストベースを分析し、カバーすべきテスト条件を識別する。

以下は、分析的テスト戦略の利点、成功要因、リスクである：

- 利点：明確に定義されたテストベースとの整合性、テストベースに対するテストの測定可能性、テストベースの早期分析による欠陥防止。分析セッションに参加するステークホルダーは、テストプロセスおよびその結果について、一定の透明性と洞察を得ることができる。
- 成功要因：文書に焦点を当てた分析的テスト戦略（要件ベース、設計ベース、契約ベースなど）の場合、テストチームはベースとなる文書を受け取らなければならない。ただし、これらの文書は改訂や反復を経ることもある。ステークホルダーに焦点を当てた分析的テスト戦略（リスクベースドテストなど）の場合、テストチームはステークホルダーからインプットを受けなければならない。いずれの場合も、分析はテストプロセスの早い段階で行わなければならない。

- リスク: 文書に焦点を当てた分析的戦略の場合、文書がマネジメントされずに変更されると、戦略が台無しになる可能性がある。文書が存在しないか、または品質が低い場合、この戦略が実行不可能になる可能性がある。リスクベースドテストの場合、リスク識別とアセスメントのプロセスにステークホルダーを関与させることができないと、リスクリストが不完全になり、リスク評価が不正確になり、戦略の有効性と効率性が低下する。

要件ベースのテストについては、[Craig02]を参照のこと。リスクベースドテストについては、[Black09]を参照のこと。

### 2.3.2 モデルベースの戦略

モデルベースのテスト戦略には、運用プロファイリングがあり、信頼性テストや性能テストによく用いられる。モデルベースのテスト設計は、UMLのような形式モデルを活用することもできる。これらのモデルは、信頼性よりもむしろ機能的な側面をカバーするために使用されることが多い。テストチームがモデルベースのテスト戦略に従うのは、システムが存在する環境モデル、システムに加えられる入力と条件、およびシステムがどのように動作すべきかのモデルを開発するときである。このモデルは通常、実際の状況、または予想される状況の分析から始める。

以下は、モデルベースのテスト戦略の利点、成功要因、リスクである:

- 利点: システムの実際の使用方法と一致した方法でのシステムのテスト。
- 成功要因: 実際の使用状況の正確なモデル、そしてほとんどのケースでは、そのモデルに従ってテスト実行を自動化できるツール。
- リスク: 正確なモデルを構築するためのデータが不十分であること、モデルの統計的な不正確さ(おそらくモデル作成者のスキルが不適切であることが原因)により重大な欠陥が本番環境に見逃されること、テストを自動化するためのツール選定が不適切であること、ポジティブパスに集中しがちであること。

モデルベースのテスト戦略では、少なくともモデルの妥当性確認において、主要なテストステークホルダーが関与するべきであるが、通常はその実際の構築にも関与する。モデルベースドテストの詳細については、[Uutting06]を参照されたい。

### 2.3.3 系統的戦略

系統的テスト戦略には、標準的なテスト条件セットをテストベースとして使用することが含まれる。テストチームは、あらかじめ定められたテスト条件セットを使用する場合、系統的なテスト戦略に従っているといえる。これには、品質規格(例:ISO 9126)、チェックリスト、あるいは特定のドメイン、アプリケーション、またはテストタイプ(例:セキュリティテスト)に関連する汎用的で論理的なテスト条件の集まりなどが含まれる。テスト条件のセットは、イテレーションごとに、あるいはリリースごとに変わるものではない。

系統的テスト戦略の利点、成功要因、リスクを以下に示す:

- 利点: 製品またはシステムで定義された属性が一貫してテストされることを保証する。
- 成功要因: 適切かつ最新のテスト条件、チェックリスト、またはテストケース、および時間の経過とともに大きく変化しない安定したテスト対象。
- リスク: テストベースが不十分または古いため、カバレッジにギャップが生じ、重要な欠陥がテストから見逃される可能性がある。

系統的テスト戦略では、主要なテストステークホルダーはテスト条件セットの初期定義にのみ関与が多い。

### 2.3.4 プロセス準拠または規格準拠戦略

テストチームは、標準化委員会などの他者によって定義された一連のプロセスに従うことを決定した場合、プロセス準拠または規格準拠のテスト戦略に従う。

これらのプロセスは通常、ドキュメント作成、テストベースおよびテストオラクルの適切な識別と使用、テストチームの組織に関する事項を扱う。

プロセス準拠または規格準拠のテスト戦略の利点、成功要因、リスクを以下に示す：

- 利点: 規格を開発した人々、例えば社内の標準化委員会、IEEE 標準を構築した人々、またはアジャイルソフトウェア開発手法を定義するアジャイルマニフェスト[Agile-Manifesto-Web]を考案した人々のスキルと経験を活用する。
- 成功要因: 選択した規格またはプロセスが、解決すべきテスト問題と整合している。
- リスク: プロセスまたは規格の不適切な理解、プロセスまたは規格の不適切な実装、異なるテスト問題へのプロセスまたは規格の適用。

プロセスや規格によって、これらの戦略には多かれ少なかれステークホルダーの関与が伴うかもしれない。例えば、アジャイルでは、主要なステークホルダーとプロジェクトチーム全体の日常的な関与が求められる。プロセスに準拠した戦略の詳細については、[Crispin09]と[Drabick03]を参照。

### 2.3.5 対処的戦略

対処的テスト戦略は動的(すなわち、テストアプローチの急速な進化)であり、ヒューリスティック(すなわち、テストケースは標準準拠のような形式的モデルではなく、実務経験に由来する)である。対処的テスト戦略を使用する場合、テストチームは、テストアイテムを受け取るまで、テストの設計と実装を待つ。シーケンシャルなライフサイクルであれイテレーティブなライフサイクルであれ、対処的なテストは実際のテスト対象システムに対処することである。Whittaker のソフトウェア攻撃メソッドのような、ある程度の既存の構造が関係するかもしれない。

以下は、対処的テスト戦略の利点、成功要因、リスクである：

- 利点: 他の戦略では見逃す可能性のある欠陥を発見できること、発見した欠陥 1 件あたりのコスト効率が高いこと、問題のある領域に対してテストの焦点を継続的に再調整できること、不完全なテストベース文書の状況下でも堅牢に機能することである。
- 成功要因: ビジネス領域および使用技術に関して深い理解を持つ、高度なスキルと豊富な経験を有するテスト担当者。
- リスク: テスト担当者のスキル不足、システムおよび対象分野に関する知識が不十分なテスト担当者、カバレッジの追跡やテストオラクルの決定が困難であることである。

対処的テスト戦略は、テストチームの知識を活用しつつカバレッジを確保するために、より形式化された戦略と組み合わせて使用されることが多い。対処的戦略については、[Whittaker02]を参照されたい。

### 2.3.6 コンサルティブ戦略

テストチームは、カバーすべきテスト条件を決定するにあたり、1 人以上の主要なステークホルダーからのインプットに依存する場合、コンサルティブテスト戦略に従う。

コンサルティブテスト戦略における利点、成功要因、リスクは以下の通りである：

- 利点: テストチームが必要なテストを適切に設計、実装、実行した場合、相談相手となったステークホルダーは、自ら要求した通りのテストカバレッジを得ることができる。
- 成功要因: 相談相手となったステークホルダーは、何を、どの程度、どの順序でテストすべきかについて、正確な認識を持っていなければならない。

- リスク: 相談相手となった複数のステークホルダー間で優先順位が競合すること、テスト条件の定義に誤りがあることが挙げられる。ステークホルダー間で期待が矛盾する場合もある。重要なテスト条件について相談相手となったステークホルダーの判断が誤っている場合、その方向性を自己検証する手段がこの戦略には含まれていないため、テスト活動自体も誤ったものとなる。すべての適切なステークホルダーに相談できていない場合、テスト活動は十分な支援を得ることができない。

相談相手からのインプットが不十分または不正確であるリスクがあるため、コンサルティブテスト戦略は通常、テストの抜け漏れリスクを低減するために他の戦略と併用する。

### 2.3.7 リグレッション回避とテスト自動化戦略

リグレッション回避、およびテスト自動化戦略も、広く用いられている。テストチームは、リグレッション回避戦略に従う場合、リグレッションのリスクをマネジメントするために、特に1つ以上のレベルでのテスト自動化を広範に活用するなど、さまざまな手法を用いる。

イテレーティブなライフサイクルにおいては、シーケンシャルモデルよりもリグレッションリスクが高まる傾向にあり、そのため継続的なリグレッションリスクのマネジメントが、より一層重要かつ切実になる。

リグレッション回避とテスト自動化戦略における利点、成功要因、リスクは以下の通りである：

- 利点：システムまたはプロダクトの新バージョンをリリースしつつも、1つ以上の重要領域におけるリグレッションリスクを最小化できる。
- 成功要因：ユニット、統合、システム、またはシステム統合レベルにおいて、保守可能なテストを成功裏かつ効率的に自動化できる能力である。
- リスク：不十分または利用できないツール、自動化が困難なテスト、テストチームのスキル不足、新機能のテスト不足、製品のスコープの変化。

リグレッション回避のテスト戦略は、現在正常に動作している機能が、今後のリリースでも引き続き正常に動作し続けるという、主要なテストステークホルダーの目的を満たすものである。

テスト自動化戦略はさまざまなプロジェクトで使用されるが、短いイテレーション内で十分なテストが求められるという理由から、アジャイルプロジェクトの成功には特に不可欠である。

アジャイルチームには常に開発者が含まれているため、彼らはテスト可能なソフトウェアの設計に貢献し、自動化アーキテクチャーおよびスクリプトの開発にも頻繁に参加することができる。テスト自動化戦略に関する詳細は、[Graham99]を参照されたい。

## 2.4 テストポリシーとテスト戦略の組織との整合

実際に運用するには、テストポリシーとテスト戦略の両方が、実用的で、完全で、一貫していなければならない。実用性とは、ポリシーと戦略が適用される、すべてのテストレベルで実際に実装できることを意味する。これは多くの場合、異なるテスト戦略が異なるテストレベルに適用されることを意味する(例えば、ユニットテストレベルではテスト自動化を徹底し、システムテストレベルではリスクベースドテストを実施する)。完全性とは、そのような作業成果物にとって重要な要素を文書が扱っていることを意味する。一貫性とは、テストポリシーとテスト戦略との内部的な一貫性だけでなく、テストポリシーとテスト戦略と、より広範なプロジェクト、プロセス、製品、組織的な要請との外部的な一貫性の両方を意味する。

ポリシーと戦略の成功を評価する上で、究極的には、そしておそらく唯一意味のある尺度は、主要なステークホルダーがテストチームに対して求める「有効性」「効率性」「満足度」に基づくものである。エキスパートテストマネージャーは、これらの分野のメトリクスを評価し、必要に応じてテスト戦略を改善することができるべきである。これらの評価と改善は、短期的と長期的の両方の視点、プロセスと組織、そして人、ツール、システム、テクニックを考慮する必要がある。

主要なステークホルダーたちはテストに対して異なる目的を持つため、テストポリシーおよびテスト戦略は組織間で一様ではなく、同一組織内においても部門やプロジェクトごとに異なる場合がある。ある組織に適したポリシーや戦略が、他の組織では効果がなかったり、逆効果になることさえある。そのため、テストマネージャーは、ポリシーと戦略の適切性を評価し、戦略の適切な組み合わせを決定し、必要な場合には改善の可能性を特定すべきである。

テスト戦略の統合と調整のプロセスは、整合性が達成されるまで継続すべきである。これは、テストポリシー、テスト範囲、会社のゴール、テストチームとの整合性を意味する。エキスパートテストマネージャーは、テスト戦略を評価し、整合の齟齬を特定し、解決策を提案することができる。さらに、事実および制約条件が特定の場合に、エキスパートテストマネージャーは、適切な要素を考慮し、テストポリシーおよび組織の整合によりリスクを最小化し、利益を最大化するテスト戦略を作成することができる。詳細については、[Riley09]第2章を参照されたい。

テストマネージャーは、新しいプロジェクト、プログラム、組織の変更に適応性を持たなければならないが、同時に、複数のプロジェクトやプロジェクトチーム間での一貫性を持たせなければならない。よく定義されたテスト戦略は、このようなタイプの変化に対応する適応性を持つ。

将来的には、ISO IEC 29119 (part3)が、組織レベルのテストポリシープロセス、組織レベルのテスト戦略プロセス、プロジェクトレベルのテストマネジメントプロセスをサポートする予定である。本シラバス執筆時点では、この標準は作業草案(Working Draft)として参照可能であった。

## 3. テストチームのマネジメント

855 分

### キーワード

マイヤーズ=ブリッジス・タイプ指標 (MBTI®) 、RACI マトリクス、S.M.A.R.T.の法則、テストアーキテクト

### テストチームのマネジメントの学習の目的

#### 3.1 イントロダクション

本節の学習の目的はない。

#### 3.2 テストチームの構築

- LO 3.2.1 (K5) 特定の求人に対して、履歴書一式を評価し、最適な候補者を決定する。
- LO 3.2.2 (K6) 特定の求人に対して、候補者を面接するための適切かつ効果的なアプローチを考案し、面接を実施する。
- LO 3.2.3 (K3) 新規メンバー研修プログラムの主なポイントの概要を説明する。
- LO 3.2.4 (K2) 従業員または請負業者を解雇する際に考慮すべき事項をリストアップする。

#### 3.3 テストチームの育成

- LO 3.3.1 (K2) 個々のテスト担当者について、育成計画を作成し、ゴールや目的を設定することの意義について説明する。
- LO 3.3.2 (K4) 個々のテスト担当者について、S.M.A.R.T.の法則をレビューし、そのゴールがテスト担当者の履歴書と組織のテスト戦略に基づいて適切かどうかを判断する。
- LO 3.3.3 (K2) テストチームにおける個人の役割と責任を理解することの重要性を説明する。
- LO 3.3.4 (K2) 特定のマイヤーズ=ブリッジス・タイプ指標のプロファイルが、個人への役割の割り当てにどのように影響するかを説明する。
- LO 3.3.5 (K3) テスト担当者がスキルを見直し、強化すべき領域を特定し、トレーニングや指導を含む適切な学習オプションを選べるようにする。
- LO 3.3.6 (K5) 高いパフォーマンスを発揮している個人の成長を促すことを目的とし、その成果を認識し報奨するためのパフォーマンス評価を実施する。
- LO 3.3.7 (K5) 低いパフォーマンスである個人の取り組みの方向性を修正し、パフォーマンスの向上を促すこと、および達成可能な短期ゴールと長期ゴールを設定するためにパフォーマンス評価を実施する。

#### 3.4 テストチームのリード

- LO 3.4.1 (K5) 特定の状況において、テストチームが必要とする情報とガイダンスを評価し、最も効果的なコミュニケーション方法を決定する。
- LO 3.4.2 (K3) チーム内での忠誠心や信頼感を育むため、さまざまなチームビルディング活動についてブレーンストーミングを行う。
- LO 3.4.3 (K2) テストチームのモチベーション向上に関して、短期的および長期的な課題を説明する。
- LO 3.4.4 (K4) オフショアを含むチームビルディング活動の計画を作成し、コミュニケーションの改善に役立てる。

#### 3.1 イントロダクション

ピープルマネジメントは、テストマネージャーの役割の重要な部分である。すべてのテストマネージャーは、組織を構築し、発展させ、統率するために必要なスキルに精通しているべきである。テスト組織の規模に応じて、

テストマネージャーは、個人、複数のテストチーム、オフサイトまたはオフショアリソースのマネジメント、そして/またはマネージャーのマネジメントを担当することがある。この章では、1つまたは複数のテストチームの内部マネジメントについて述べる。

大規模な組織はもちろん、小規模な組織であっても、通常、正式な雇用契約の処理、福利厚生の管理、従業員と経営陣の双方に対する助言やサポートを行う部署がある。この部署は、一般に人事部または人事課と呼ばれる。本節では、その部署が組織内に存在することを前提とする。実際の部署が存在しない組織の場合は、同様の責任を負う個人が存在する可能性がある。そのような部署や個人が存在する場合は、以下の節に記載されているように、その部署や個人に相談すべきである。組織内にそのような機能がない場合、テストマネージャーは、ここで扱ういくつかの課題について、マネジメントチーム、法務部門(存在する場合)、および過去の慣例に依存して対処することとなる。

## 3.2 テストチームの構築

テストチームのために、適切な採用判断を行うことは、テストマネージャーの重要なスキルである。間違った人材を採用すると、生産性や士気に悪影響を及ぼし、組織の成功にも影響する可能性がある。一般的に、採用プロセスには複数の人が関わるが、オファーを出すかどうかは、最終的にはテストマネージャーの判断となる。

テストマネージャーは、詳細な採用計画を策定し、採用するポジションを明確にし、そのポジションに必要なスキルを理解する必要がある。テストマネージャーはまた、可能な限り最善の採用判断を下すために、熟練した面接官でなければならない。多くの企業の人事部門は、採用や面接に関する研修を提供しており、[McKay07]や[Rothman04]のようなテストチーム向けに特化した書籍も多数存在する。

### 3.2.1 ジョブディスクリプション

組織の規模によっては、テスト担当者の役割と経験レベル、およびテスト組織のニーズに基づいて、いくつかの専門的なテストポジションを定義することができる。

テスト組織内の各特定ポジションには、そのポジションの具体的な責任や権限、そのポジションに必要な経験、給与範囲、必要なスキルやコンピテンシーなどの概要を説明した詳細なジョブディスクリプションが必要である。ジョブディスクリプションに記載すべき事項の詳細は、企業の人事部門が定める場合がある。

ジョブディスクリプションを作成したら、求人掲示板、企業の採用サイト、その他組織が使用するあらゆる媒体に掲載し、資格のある求職者が反応できるようにする。

### 3.2.2 履歴書

募集ポジションの適任者を見極める最初のステップは、提出された履歴書または職務経歴書のレビューである。人事部門はフィルターとして機能し、明らかに不適格な候補者を拒否することができる。これが不可能な場合は、テストマネージャーまたはその代理人が個々の履歴書をレビューし、関心のある候補者を特定する必要がある。

多くの企業では、候補者選考プロセスのマネジメントについて厳格なガイドラインを定めている。テストマネージャーは、人事部門に確認し、履歴書選考と面接の両プロセスにおいて、どのレベルの文書が必要かを判断する必要がある。

履歴書をレビューする際、テストマネージャーまたはその代理人は以下の指標を確認すること:

- 候補者は要件となるスキルと経験を有しているか？
- 候補者は、ISTQB 認定テスト技術者資格 Foundation Level など、要件または望ましい認定を取得しているか。

- 候補者の給与要件は、募集ポジションに合致しているか？
- 履歴書は整理されていて読みやすいか。
- 履歴書にスペルミスや文法ミスはないか。(多くのテストマネージャーは、履歴書に間違いがある候補者を選考対象としない。)

### 3.2.3 面接

技術職の面接は、採用する人材が開発者、ビジネスアナリスト、ソフトウェアテスト担当者のいずれであっても、基本的に同じ構成に従う。複数回の面接、異なるバックグラウンドを持つ面接官チームの編成、明確に定義された面接プロセスなどのよい面接の実践は、すべてテスト担当者の面接にも当てはまる。

テスト担当者候補と面接する際には、候補者の長所を探る必要がある。具体的には、優れた問題解決能力と批判的思考能力、優れた文章力と口頭でのコミュニケーション能力、チーム内で協力する能力、好奇心旺盛な性格、技術的知識、ドメイン知識、そのポジションに必要な経験／経験といったことをテスト担当者は示せる必要がある。面接での質問は、(最低限)以下を見極めることを目指すべきである：

- その職務および組織に適した姿勢を持っているか？そのグループに適しているか？その職務をうまく遂行できるだろうか？テストの役割を真剣に受け止めているか？スケジュールが厳しくなっても、前向きな姿勢を維持できるか？
- 自信と知識を適切に表現し、自己能力を過剰に主張していないか？自身の成果を正当に主張すべきときに主張できるか？間違っていることを認め、成熟した態度で対処できるか？必要であれば、テストチームの立場を代表してプロジェクトミーティングに参加できるか？
- 職務に必要な組織スキルを備えているか？頻繁に中断が発生する環境で働くことができるか？中断した場合、最小限のダントンタイムで元の仕事に戻れるか？欠陥を発見した場合でも、自分が実行していたテストに戻り、そのテストを完了することを覚えているだろうか？
- その役割に必要な成熟度を持っているか？その職務に必要なリーダーシップポテンシャルがあるか？起り得る対人関係の問題に効果的に対処できるか？
- 他者に共感的に接する能力を持っているか？開発者に不快感を与えることなく欠陥を提示できるか？議論の両方の視点を理解できるか？欠陥の優先度を決める際に、優先度評価に影響する複数の要因を考慮できるか？

これは全数リストを意図したものではなく、テストマネージャーが面接質問を準備する際に、手始めとして利用することを意図したものである。面接の質問例やテクニックについては、[McKay07]を参照されたい。

面接や面接の質問には多くのタイプがある。面接プロセスの一環として、面接チームおよびテストマネージャーは、以下を含めることを検討してもよい：

- 電話スクリーニング – 多くの場合、短時間で済むよい一次面接であり、面接官は口頭でのスキルをチェックすることができる。また、候補者が現地に来られない場合には、唯一の選択肢になることもある。
- グループ面接とパネル面接 – 複数の人が候補者を一度に面接する。質問と質問の間の「思考時間」が短くなるため、候補者にとってはよりストレスの多い面接となる。
- 行動に関する質問 – ある問題に直面したら候補者はどうするか？
- シチュエーションに応じた質問 - 候補者に、プロジェクトの状況とその結果の実例を挙げてもらう。
- 技術的な質問 – 候補者にコーディングやスクリプトのスキル、データベースのスキルなどを示させる。
- パズル問題 – 問題解決能力と迅速な思考能力を評価するために、候補者に頭を使うパズルを提示する。
- 特定のドメインに特化した質問 – 職種によって異なる(ソフトウェアセキュリティ、規制の厳しい業界など)。
- チームダイナミクスに関する質問 – 候補者は他のテストチームとどのように調和するか？候補者はテストチームの他のメンバーとうまくやっているか？

- 筆記試験 – 企業によっては、特定のポジションの候補者全員に標準試験を課すことがある。このような試験では、技術的知識、自己組織化能力、問題を構造化する能力をチェックし、小論文形式の問題が含まれる場合は、筆記によるコミュニケーション能力も評価する。一般的に、試験を実施する場合は、一貫して実施する必要があり、組織の人事部の承認が必要な場合もある。面接プロセスにおける試験の使用に関する法的問題や方針については、人事部に確認すること。

面接に加え、アセスメントも候補者の評価に用いられる。アセスメントは通常グループで実施する。実務スキルの実演(特定のテスト条件に対するテストケースの作成)、トピックに関するプレゼンテーションの実施、グループでのトピックに関するディスカッション、分析スキルの実演(一連の要件に対する制御フロー図の作成)、その他の適切かつ実証可能なスキルなどのタスクについて、候補者のパフォーマンスを評価する。

候補者にポジションを正式にオファーする前に、テストマネージャーはリファレンスをチェックし、組織が要求するすべてのプロセスに従う必要がある。これには、身元調査、信用調査、セキュリティクリアランスの確認、在留資格の確認などが含まれる。

チームメンバーの面接や選考を行う際に重要なのは、結束力のある効果的なチームを作ることがゴールであることを忘れないことだ。個人のスキルが完璧にフィットすることはめったにないが、ゴールは、その完璧なチームを作るために必要なスキルを身につけるポテンシャルと能力を持った人材を発見することである。異なる経験レベルや異なる知識レベルは、チームに多様な視点をもたらし、チームがさまざまな仕事に取り組むことを可能にする。

### 3.2.4 新規メンバーの受け入れ

前節では新規採用について述べたが、テストマネージャーが常にチームメンバーを採用できるとは限らない。テストマネージャーは、既存のチームを引き継いだり、複数の既存チームを統合したり、他者によって採用された人員を受け入れたりする場合もある。いずれの場合であっても、テストマネージャーの役割は、チームを構成する個々のメンバーを集めて、効果的で一体化したワーキングチーム作り上げることである。

新しいテストチームメンバーが加わった場合、そのメンバーは、テスト組織のプロセスや手順について訓練を受け、部門およびプロジェクトで使用しているツールにも慣れる必要がある。テストマネージャーは、組織向けに確立された新規採用者研修プログラムを用意しておくべきである。このプログラムには、組織のテスト実践およびプロセスに関する文書、チームメンバーによる実演や実地研修、人事部門による企業方針に関する研修などが含まれ、新規採用者が迅速にチームに適応できるよう支援する。複数の従業員を同時に採用する場合、彼らと一緒に研修に参加させることで、結束を高める効果も得られる。

研修資料そのものに加え、新規採用者向けの研修スケジュールも策定しておくべきである。これにより、新規メンバーが採用されるたびに、担当するチームメンバーが何をいつ実施すべきかを把握できる。例えば、雇用初週にはプロジェクト資料へのアクセス方法を学習し、第2週には欠陥追跡システムやテストマネジメントシステムの使用方法を学習するといったタスクをスケジュールに含めることができる。テストマネージャーは、こうした研修スケジュールにおいてマイルストーンを設定し、達成状況を検証すべきである。

テストグループ向けの新規採用者研修プログラムは、既存のチームメンバーによって共同で作成させることもできる。これにより、メンバー間で知識を共有できるだけでなく、新規メンバー研修への関与を通じて既存メンバーのモチベーション向上にもつながる。新規採用者向け研修に加えて、テストマネージャーは新規テストチームメンバーに対してメンターを任命し、新規採用者が新しい組織に慣れる過程で質問や相談の際に頼れる一次連絡先を確保しておくとよい。

### 3.2.5 雇用の終了

チーム作りの一部として、いつ、どこでチームを縮小すべきかを知ることがある。残念ながら、テストマネージャーは、低いパフォーマンス、事業方針の変更、または人員削減(レイオフ)のために、チームメンバーを手放すという難しい判断を下さなければならないときがある。

テストマネージャーは、人事部門と協力して、組織の方針と手続き、法的な影響、適用される労働組合の規制、雇用終了と解雇に関する国内労働法または協定を理解する必要がある。雇用終了手続きを開始する前に、必要となる文書を収集しておかなければならぬ場合もある。

ある従業員のパフォーマンスが許容範囲を下回った場合には、マネージャーは直ちに人事部の助言を求め、パフォーマンス改善計画または雇用終了計画に何が求められるかを理解すべきである。

雇用終了が避けられない場合、テストマネージャーは、必要なスキルや知識の移転作業が行われ、会社の知的財産がすべて保持されるようにすべきである。テスト担当者は、文書化されていない設定手順、パスワード、データの場所など、システムに関する独自の知識を保持している場合が多く、それらは、退職する人のタスクを引き継ぐ人にとって重要な情報となる。雇用期間中に業務横断的なトレーニングを行うことで、重要かつ独自の知識を持つ人材の退職リスクを最小限に抑えることができる。

### 3.2.6 契約関係の終了

多くのテストチームは、スキルのギャップを埋めるために、そしてまたは、有能なテスリソースでチームの人数を迅速に増やすために、個人またはアウトソーステストベンダーのような業務請負を利用することで、一時的に労働力を増やすという利点を活用している。

ほとんどのテストチームや企業は、あらかじめ決められた期間、請負業者やアウトソーシングできるテスト会社を利用する。その後、契約は終了し、リソースは利用できなくなる。テストマネージャーは、スキルやリソースの移行計画書を作成することで、人員や専門的なスキルの喪失に備え、十分な事前準備を行うべきである。この計画には、スキルのギャップをどのように埋めるか、既存のリソースをどのように再配分するかについての情報を含めるべきである。

請負業者またはアウトソーシングできるテスト会社が十分なパフォーマンスを発揮していない場合、早期終了が正当化される場合がある。テストマネージャーが契約を解除する権限は、契約書、合意書、その他の法的文書によって制限される場合がある。従業員の雇用終了と同様に、請負業者またはアウトソースグループの契約終了は、人事部およびまたは法務部および調達部と協力して処理すべきである。詳細は、[Black09]の第 10 章を参照のこと。

## 3.3 テストチームの育成

テストマネージャーの役割の 1 つは、チームメンバーに必要なツールと機会を提供することである。そうすることで、チームメンバーは、現在の職務を成功裏に遂行し、時間とともにスキルと能力を向上させることができる。ソフトウェアチーム開発に関する詳細については、[DeMarco99]および[McConnell99]を参照のこと。

### 3.3.1 個人の育成

テストマネージャーは、部下の育成に責任を持つ。もちろん、これは個人だけでなく、テストチーム全体にも利益をもたらす。なぜなら、既存の従業員を成長させることは、グループ内のスキルギャップを埋めるのに役立つからである。

多くの組織では、マネージャーや従業員に対して、改善すべき分野の概要を説明し、その変更を達成するためのロードマップを含む、能力開発計画を作成することを求めたり、奨励したりしている。ロードマップには、正

式なトレーニングコースへの参加、学位取得に向けた学習、特定のトピックに関する自習、新しい役割や仕事への挑戦、テスト認定に向けた学習などのタスクが含まれる。これらの能力開発タスクを成功裏に完了することが、従業員の定期的なゴールまたは目的の一部となる。

### 3.3.2 ゴールと目標の設定

多くの組織では、マネージャーに対して、チームメンバーの定期的(通常は年次)なゴールと目標を設定することを要件としている。これらは従業員のパフォーマンスを評価するために使われ、個人のパフォーマンス評価や給与見直しに結びつく場合もある。テストマネージャーは、個々のテストチームメンバーと協力して、個々のゴールと成果物について相互に合意し、ゴールと目標に対する進捗を定期的にモニタリングする責任がある。

チームメンバーがゴールや目標(目標によるマネジメント)を達成したことに基づいて評価される場合、ゴールは測定可能である必要がある。テストマネージャーは、S.M.A.R.T.の法則(Specific, Measurable, Attainable, Realistic, Timely)の使用を検討すべきである。

ゴールを設定するとき、テストマネージャーは、具体的な成果物(何を行うのか、どのように達成するのか)そのゴールや目標を完了するのかのタイムフレーム(いつまでに完了するのか)の概要を説明すべきである。テストマネージャーは、達成不可能なゴールを設定すべきではないし、無意味なほど単純で漠然としたゴールや目標を使うべきではない。これは、タスクの測定が難しい場合、テストマネージャーにとって特に課題となる。例えば、テストケース一式を完了させるというゴールを設定しても、タイムフレームが短すぎたり、ソフトウェアの品質が十分でない場合には、現実的ではないことがある。テストゴールは、評価される人にとって公平で、コントロールの範囲内であることが非常に重要である。発見された欠陥数、テストケースの合格率、本番で発見された欠陥数などのメトリクスは、テスト担当者個人のコントロールの範囲ではない。テストマネージャーは、誤ったデータを測定することで、誤った行動を助長しないよう、細心の注意を払わなければならない。たとえば、記載した欠陥の件数に基づいてテスト担当者を評価した場合、報告した欠陥の品質を考慮しないと、開発者によって却下される欠陥数が増加する可能性がある。

テストチームメンバーの個人のゴールの設定に加え、テストチーム全体またはテストチーム内の特定の役割について、グループのゴールを設定することもできる。例えば、テストチームのゴールは、新しいツールの導入に取り組み、テストチームのスタッフがそのツールのトレーニングを完了し、そのツールを使ったパイロットプロジェクトを成功させることである。ゴールは常に、組織の上位レベルで設定されたゴールの達成に貢献する形で構築されるべきである。個人のゴールは、マネージャーのゴール、さらにその上位のゴールへと支援が続いていくべきである。

### 3.3.3 明確な役割と責任の定義

テストマネージャーの仕事の一部は、テストチームのメンバーに、組織内での自分のグループの役割と、個々のグループおよびプロジェクトレベルの責任を理解させることである。テストマネージャーは、その人の役割と責任が定義され、パフォーマンスベンチマーク(またはゴールと目的)が設定されない限り、個人のパフォーマンスを適切に評価することはできない。

組織の規模に応じて、いくつかの専門的なテストポジションが定義されることがある。これらのポジションは、多くの場合、テスト担当者の役割と経験レベルに基づいている。テスト組織に含まれる専門職の例としては、以下のようなものがある:

- ブラックボックステスト担当者(テストアナリスト)
- ホワイトボックステスト担当者(テクニカルテストアナリスト)
- 性能テストエンジニア(テクニカルテストアナリスト)
- テスト環境とデータアドミニストレーター
- ツール開発者

- テストリード(テストマネージャー)
- テストアーキテクト
- テスト自動化スペシャリスト(テクニカルテストアナリスト)
- セキュリティテストエンジニア(テクニカルテストアナリスト)

これらの専門的な役割のそれについて、役割と責任のマトリクスを作成し、各ポジションに含まれる、活動やタスクのタイプを概略的に示すことができる。全体的なマトリクスは、基本的にテスト組織自体の役割と責任を定義する。この種のマトリクスは、個々のテスト担当者が自分に何が期待されているかを知ることができるので、社内でも価値がある。また、スキルのギャップを特定するのにも役立ち、そのギャップをトレーニングや採用によって埋めることができる。スキルマトリクスが、各個人の目標設定の期待値と一致していることを確認することが重要である。このマトリクスは対外的にも価値があり、テストチームの役割や、テストチームが実行できるスキルや能力を有して実行できる活動のタイプについて、組織がよりよく理解できるようになる。RACI マトリクスのような、より形式化されたマトリクスは、定義された領域について、誰が実行責任者、説明責任者、相談先、または報告先であるかを定義するものであり、有用であろう。

### 3.3.4 個人の性格とチーム内での役割

テストチームマネージャーは、個々のテスト担当者がテストチームにどのような貢献をしているかを理解し、チーム内で各自がどのような役割を果たすかを理解することが重要である。チーム内では、各自が自分の役割と能力だけでなく、同僚の役割と能力も理解する必要がある。他のチームメンバーのスキルを尊重し、スキルセットの相互補完性を理解することで、チームがユニットとして効果的に機能する。

テストマネージャーとテスト担当者間のやりとりは複雑であることが多く、必要なスキルがないと、誤解や情報の隠蔽、あるいは間違った情報や偽の情報を得る結果になりかねない。テストマネージャーは心理学者である必要はないが、共依存的な行動を理解することから生まれる優れた対人スキルが必要である[Copeland Paper 01]。

テストマネージャーは、テストマネージャーとテストチームメンバーの両方が個々の行動をよりよく理解できるように、個々のテスト担当者の性格プロファイリングに投資したいと思うかもしれない。マイヤーズ=ブリッジス・タイプ指標 (MBTI®) のような専門的に実施された性格テストをこの目的のために使用することができる [Myers&Briggs95]。

個人の性格プロファイリングに加え、テストマネージャーは、チームの関係性や働き方をよりよく理解し、チーム環境における自分自身とチームメイトの行動をより深く理解するために、テストチームの役割タイプ評価に投資することもできる。また、系統的なモデルを使用して、個人が好む学習方法、定義が不十分な問題に対処する能力、ストレスの多い状況に対処する能力などを理解することもできる。

### 3.3.5 スキル開発

テストマネージャーは、テストチームメンバーが時間をかけてスキルを向上させることを奨励することが重要である。これは、個々のチームメンバーだけでなく、チーム全体にとっても有益であり、チーム内のスキルギャップを埋める助けとなる。スキル開発のために考慮すべき領域は、以下の通りである：

- ツールのスキル
- 技術力
- テスト方法論のスキル(スクラムなど)
- テクニカルテストスキル
- ドメイン知識
- ソフトスキル(コミュニケーションなど)

### 3.3.6 トレーニングの機会

テストマネージャーがチームを育成する最良の方法ひとつは、テスト組織内でトレーニングの利用を推進することである。トレーニングは、非公式なもの（例えば、他のテストチームメンバーによって行われる資料やトレーニング、オンラインジョブトレーニング）、または正式なもの（例えば、専門のトレーニング組織によって開催されるトレーニング、オンラインまたはeラーニングシステムを使用したトレーニング）がある。

組織の規模や使える予算に関係なく、内部リソースは常に特定することができ、テストチームメンバーのスキルレベル向上に役立てるべきである。テストマネージャーは、手元にあるリソースを考慮し、グループ内の知識を活用するだけでなく、開発者、DBA（データベースアドミニストレーター）、ビジネスアナリスト、その他の技術専門家の知識も活用すべきである。テストチームメンバーがグループ内のさまざまなプロジェクトに配属されるように、技術分野とドメイン分野の両方で業務横断的なトレーニングを行うことは、重要なトレーニングゴールである。テストアナリストを他のグループに組み込んだり、開発者を一定期間テストグループに入れたりすることも、他のグループとの関係を構築し、学習機会を提供する優れた方法である。この業務横断的なトレーニングを実施する方法として、より正式な形であるジョブローテーションが考えられる。

### 3.3.7 メンタリング

テストチームメンバーを育成するために、テストマネージャーは、メンターの利用を奨励することができる。調整したい特定のスキルがある場合、メンターと一緒に仕事をし、フィードバックやアイデアを得ることが理想的かもしれない。テスト組織内の上級者は、経験の浅いテスト担当者や、性能テストのような分野の新しい側面を学ぶことに関心のあるテスト担当者のメンターとして働くことができる。

チーム内のメンターを活用するだけでなく、他チームのメンターを活用することも有効である。例えば、テストチームの新しいリーダーやテストマネージャーは、新しい視点を得たり、新しいマネジメントスキルを学んだりするために、別のグループのマネージャーの指導を受けることができる。同様に、テクニカルテストアナリストは、開発者と一緒に仕事をすることで、ホワイトボックステストのスキルを高めることができる。

### 3.3.8 パフォーマンス評価とフィードバック

多くの企業では、各従業員に対して年に一度の業績レビューを受けることを義務づけており、その中で合意されたゴールや目的について見直しや議論が行われる。従業員の給与、賞与、または雇用形態がこのレビューに関連している場合もある。テストマネージャーは、パフォーマンス評価の具体的要件について、企業の人事部に問い合わせるべきである。

パフォーマンス評価の準備をする際、テストマネージャーは、評価期間におけるテストチームメンバーの作業に関する情報を収集する必要がある。これは、テスト担当者が作成した作業成果物（テストケース、ステータスレポートなど）を利用して行うことができる。さらに、テストマネージャーは、レビュー対象の個人と密接に仕事をしたことがある他の従業員の意見を求める検討するといい。これは、従業員に関する具体的かつ自由回答式の質問を盛り込んだ質問票を作成することで実施できる。パフォーマンス評価には、作業成果物やテスト業務のパフォーマンスといった客観的評価に加え、パフォーマンスに関するマネージャーによる主観的評価の両方が含まれる。すべてのレビューは、事実と事例によって裏付けられていることが重要である。

マネージャーが遭遇する可能性のある困難な問題のひとつは、従業員が仕事に関して、または組織内での昇進能力に関して、非現実的な期待を抱いている場合である。この場合、テストマネージャーは、職務要件、従業員の発揮能力、および両者の差分を明確に定義すべきである。従業員の願望は、事実に基づいて明確に扱うべきである。

### 3.4 テストチームのリード

優れたマネージャーであり、効果的なリーダーであるために、テストマネージャーは、適切なマネジメントとリーダーシップ両方の原則と技法を使うように努力すべきである。これらの多くは、テストマネジメントに特化したものではなく、むしろ技術チームやチーム全般のマネジメントに広く使われる一般的な原則や技法である。本節では、テストマネージャーに影響する側面に焦点を当てる。テストチームのマネジメントとリーダーシップに関するさらなる情報は、[Craig02] に記載されている。特に第 10 章「テストマネージャー」が参考となる。

#### 3.4.1 情報共有とコミュニケーション

テストマネージャーは、テストチームが、組織全体における自チームの役割と、チーム内における個々の役割を正しく理解できるようにする責任を担っている。

テストマネージャーの責任として、プロジェクト関連の活動に関する必要な情報をスタッフと共有し、テストチームのメンバーが最新の正しい情報を持っていることを確認し、職務を成功裏に遂行できるようにする。

優れたテストマネージャーは、情報の共有とコミュニケーションは双方向でなければならないことを常に念頭に置かなければならない。テストマネージャーは情報のないままマネジメントはできないので、テストチームメンバーからの情報を収集し、定期的に検討し、情報に基づいた判定ができるようにすることが必須である。この種のコミュニケーションは、公式または非公式であり、以下を含む：

- 1 on 1 ミーティング
- 週次状況報告
- 非公式なコミュニケーション

効果的なリーダーになるためには、テストマネージャーは、テキストメッセージやインスタントメッセージなどの気が散るものに屈しないことを含め、積極的に傾聴する姿勢を強力に示さなければならない。傾聴スキルの高いテストマネージャーは、よりオープンで存在感があり、反応がよい。また、組織のテストマネジメントツールを通じて、ステークホルダーに現在のプロジェクトステータスを提供するような情報を利用できる状態にする必要がある。

#### 3.4.2 忠誠心と信頼の醸成

テストチーム内の忠誠心と信頼を育むために、テストマネージャーはチームに対してオープンで正直でなければならず、テストチームメンバーが悪い知らせを共有しても罰せられないような安全な環境を奨励すべきである。さらに、テストマネージャーはテストチームメンバーに敬意を持って接し、メンバー自身およびメンバーの意見が尊重されていると感じられるように努めなければならない。

#### 3.4.3 チームビルディング

テストマネージャーの役割の 1 つは、結束力のあるテストチームを作り、育成し、ポジティブなチームの人間関係作りを支援することである。チームビルディングは、公式および非公式のさまざまな方法で行うことができる。オープンなコミュニケーション、チームメンバー間の敬意ある行動、グループの役割と責任に関する共通の理解は、すべて、結束力のあるチームを作るための要素である。

強力なチーム作りに役立つ日常的な活動に加えて、チーム開発を支援するさまざまな演習や活動がある。以下は、テストマネージャーが使えるチームビルディング活動の例である：

- チームメンバーそれぞれが特別な料理を持ち寄り、グループランチを行う
- プロジェクト完了後に打ち上げを開く
- 誕生日、結婚式、出産など、テストチームメンバーの特別なイベントを祝う
- テストチームのニュースレターやインターネット(社内ポータルなど)を作成する

- ランチタイムに教育セッションを企画し、テストチームメンバーが、新しいテストツールや注目のテスト記事など、チームにとって関心のあることを発表する。
- チームビルディング活動を専門とする外部企業を招く

チームビルディング活動を予定し、開催する場合、テストマネージャーは、オフサイトやオフショアのチームメンバーも含め、チームメンバー全員が参加できるように努めるべきである。これが不可能な場合は、別の活動を予定することもできる。

より広く、プロジェクトにフォーカスした観点では、複数の情報共有イベントを調整することもできる。イベントでは、さまざま専門分野の人(分析、開発、テスト、技術文書作成など)が、自分たちは何を行い、何を必要とし、どのような製品を生み出すかについて紹介できる。これは、組織内の各専門分野の人に誇りを持たせ、プロジェクトチーム全体の相互理解を促し、チームの結束を強める助けとなる。

### 3.4.4 テストチームのモチベーション向上と挑戦の機会の提供

多くの人々は、仕事がうまくいったこと、給料、近づいてる特別なイベント、または次に担当する業務を考えて、自分のモチベーションを高めている。しかし、テストマネージャーは、チームのモチベーションを高める要素を念頭に置くことが重要である。先に説明したチームビルディングの演習に加えて、テストチームメンバーのモチベーションに役立ついくつかのアイデアを以下に示す:

- プロジェクトや企業に関する重要な情報をチームメンバーに伝える
- 率直で有益なフィードバックを与える
- 異なるプロジェクトやタスクに移動させることで、さまざまなことをする機会を与える
- トレーニングセッションに参加することでスキルを向上させる機会を与える
- 何をすることに興味があるかを尋ね、その興味に向かうようなゴールと目的を設定する
- 能力や育成計画に応じて仕事を割り振る。若手にとってはやりがいのあることでも、ベテランにとって退屈なことかもしれない

### 3.4.5 分散したチームのマネジメント

多くのテストマネージャーは、地理的に異なる場所にいるテスト担当者のグループをマネジメントする責任を負っている。チームメンバーは、同じ都市内の異なるフロアやビルに分かれているかもしれないし、世界中に散らばっているかもしれない。

分散したチームをマネジメントすることには独特の課題がある:

- タイムゾーンを越えてマネジメントする
- 文化の違いに対処する
- 多言語でチームをマネジメントする
- 異なるスキルレベルと期待を持つチームをマネジメントする

テストマネージャーは、これらの要素を認識し、チームメンバーやチームリーダーと協力し、各チームに明確なコミュニケーションと明確な期待値が設定されるようにしなければならない。

オフサイトのチームとのコミュニケーションは非常に重要であり、テストマネージャーはコミュニケーション計画書を作成し、定期的な電話会議やビデオ会議のスケジュールを組み、本社オフィスとオフサイトの間で情報が共有されるようにするとよい。

## 4. 対外関係のマネジメント

330 分

キーワード

なし

### 対外関係のマネジメントの学習の目的

#### 4.1 イントロダクション

本節の学習の目的はない。

#### 4.2 対外関係のタイプ

LO 4.2.1 (K3) 特定の契約について、テスト責任に関するサードパーティー関係を定義する。

#### 4.3 契約上の問題

LO 4.3.1 (K4) 外部ベンダーとのプロジェクトについて、適切な SLA を定義する。

#### 4.4 コミュニケーション戦略

LO 4.4.1 (K2) 特定のサードパーティーのエンゲージメントに関するコミュニケーション戦略の可能性について議論する。

#### 4.5 外部ソースからの統合

LO 4.5.1 (K4) 特定のプロジェクトのシナリオを解析し、開発/テストの一部にサードパーティーを起用した製品について、社内テストチームに要求される適切なレベルのテストを判断する。

#### 4.6 テスト戦略のマージ

LO 4.6.1 (K2) サードパーティー組織とテスト戦略をマージする際に考慮しなければならない項目をまとめる。

#### 4.7 品質の検証

LO 4.7.1 (K6) 特定のサードパーティープロジェクトのシナリオに対して、一連の開始基準と終了基準を作成する。

## 4.1 イントロダクション

テストマネージャーは、サードパーティーベンダーが製品の一部の作成、テスト、および/または納品に関与しているさまざまなプロジェクト状況に遭遇することは珍しくない。テストマネージャーは、このような状況がどこで発生するかを識別し、役割を理解し、さまざまな成果物に期待される品質レベルを理解できなければならない。さらに、品質の高い製品を円滑に納品するために必要なスケジュール、テストアプローチ、文書化、コミュニケーションについても考慮しなければならない。この外部との関係は、ベンダー(指定されたサービスを提供するために報酬を受ける者)または同一組織内の外部グループ(例えば、組織内の別の部門によるテスト)である場合がある。これらを合わせて「サードパーティ」と呼ぶ。[Black09]の第 10 章を参照のこと。

## 4.2 対外関係のタイプ

対外関係にはいくつかの基本的なタイプがある:

- サードパーティが完成品を納品する(ターンキー:すぐに使える形での納品のこと)
- サードパーティが製品を開発するが、テストは受け手側の組織の責任である
- サードパーティが受領側のコードと統合しなければならないコードの一部を開発する
- サードパーティが社内開発製品のすべてのテストに責任を負う
- サードパーティが社内開発製品のテスト責任を社内のテストチームと分担する

1 つのプロジェクトに複数のサードパーティが関わっている場合、サードパーティ間のコミュニケーションもマネジメントしなければならない。これらの関係は、コミュニケーション、文書化、調整に関してそれぞれ要件が異なる場合がある。テストマネージャーは、これらの違いだけでなく、契約上の制約も認識しておかなければならない。

## 4.3 契約上の問題

外部リソースと協働する場合、通常は契約書を使用する。契約管理は、通常、法務部を通じて処理されるが、テストマネージャーは、テストプロジェクトに影響する可能性のある契約上の問題を認識し、テストサービスの専門家として技術的/方法論的な意見を提供しなければならない。

テストプロジェクトに影響を与える場合がある具体的な契約の項目は以下の通りである:

- サービスレベルアグリーメント(SLA)-欠陥修正の所要時間(重要度/優先度スキームに関連する場合がある)、質問や情報要求に対する所要時間、サポート担当の対応可能時間および連絡方法
- 成果物 - 文書(ユーザーガイド、ヘルプファイル、仕様書)、ビルドスケジュール、既知の欠陥に関する情報、リリースノート(テストの準備が整っていない部分の通知を含む)、テストデータ(機密情報が削除された大容量データファイルなど)
- 品質レベル - コードレビュー、ユニットテストカバレッジ(コードカバレッジ)、統合テストやシステムテストの結果、静的解析テスト、残存欠陥の許容数(重要度/優先度)に対する期待度

契約内容、業界、開発される製品、およびサードパーティとの関係性によっては、追加の項目が求められる場合がある。例えば、医療機器の製造を請け負うサードパーティには、おそらく、(米国の FDA などの)政府の承認を取得するレベルが求められるであろう。同様に、契約企業内にあるサードパーティのテスト組織であれば、実施されている運用がある場合には、SLA や成果物の定義を簡略化することが可能な場合もある。

テストマネージャーがベンダーの選定プロセスに関与している場合、本節で議論したすべての項目について、契約で確実に網羅されていることが重要である。テストマネージャーがサードパーティの選定に関与しない場合、将来の契約において、選定チームがサードパーティのポテンシャルを評価するのに役立つメトリクスを

追跡することが賢明である。これらの評価基準は、契約の標準を改善するのに役立ち、また、どのサードパーティを再度使用すべきで、どのサードパーティを使用すべきではないかを決定するのにも役立つ。

理想的には、テストマネージャーは、契約内容に対して重要な役割を果たすべきであり、特に、スケジュール、テスト関連の事項、および品質の測定に関する項目には積極的に関与すべきである。契約の下で提供される成果物は、完成品の妥当性を確認できるよう、正確に記述されていなければならない。

## 4.4 コミュニケーション戦略

サードパーティとのコミュニケーションは、製品、サードパーティ、関係する地理的な場所、文化の違い、プロジェクトの重要度や複雑度、過去の仕事での関係性によって、その量、詳細度、頻度、スタイル、形式、フォーマルさが大きく異なる場合がある。テストマネージャーは、プロジェクトのコミュニケーション戦略を定義する前に、これらの各要因を認識しなければならない。例えば、オフショアで開発とユニットテストを行うプロジェクトであれば、週 1 回の詳細なステータスレポート、日次の欠陥レポート、週に 2 回の電話会議、さらには、文書によるコミュニケーションで使用する言語に関する合意を要件とすることが理にかなっているかもしれない。

欠陥トラッキングとテストマネジメントに標準的なツールを使用することで、複数の関係者間の効果的で一貫性のあるコミュニケーションを促進することができる。これは、これらのツールに適切なアクセス性と応答性の確保が要求されることを意味し、また、ツールの 24 時間 365 日のサポートも要求されるかもしれない。また、誤解を避けるために、ツール使用者全員に使用ガイドライン(重要度／優先度スキームなど)を公表する必要があるかもしれない。これは、サードパーティが開発を行い、ローカルチームがテストを行う場合に特に重要である。

外部とのコミュニケーションは、他のテストや開発リソースに限定されるものではない。エンドユーザー、規制当局、その他のステークホルダーもコミュニケーション戦略内で考慮しなければならない。リスクアセスメントワークショップは、これらの人々をテストプロセスに参加させ、継続的かつ効果的な情報交換のためのオープンなコミュニケーションパスを作るよい方法である。

ミーティングのスケジュールは、すべての該当するタイムゾーンに対応できるよう、プロジェクト開始時に設定すべきである。午前 3 時のミーティングに出席させるのは避けたいものだ！休日や休暇のスケジュールも考慮しなければならない(特に異なる文化や分散した拠点)。アジャイルプロジェクトでは、毎日の「スタンダップ」ミーティング、スプリントの引き継ぎミーティング、その他の頻繁で定期的なコミュニケーション方法が、チームが分散している場合でも、チームに情報を提供し続けるために使われる。

具体的なコミュニケーションスタイルの要約は、[Evans04]の付録に記載されている。

## 4.5 外部ソースからの統合

テストマネージャーは、最終製品のテストにおけるチームの関与の程度を定義しなければならない。期待される関与の程度は、期待される成果物によって以下のように異なる：

- サードパーティ(または複数のサードパーティ)が、開発しテストした製品を納品する(ターンキー)  
-この製品に対して、テストはすべて完了しているべきである。テストチームは、事前に定義された基準(契約による要件、仕様書などの検証)に従って受け入れテストを実施する計画書を作成すること。この製品が製品パッケージとしてリリースされる前に他の製品と統合される場合は、システム統合テストが必要になる場合がある。
- サードパーティが、開発済みで部分的にテストされた製品(ユニットテスト済み)を納品する - この時点で、製品が統合/システムテストに受け入れられる前に、コードカバレッジメトリクスをレビューすべきである。テストチームは、完全な統合テスト／システムテストの実施、および必要と思われる受け入れテ

ストの実施を見込むべきである。これは、外部チームがコードの一部を開発し、それが内部で開発されたコードと統合される場合と同じシナリオである。

- サードパーティーが、開発済みだがテストされていない製品を納品する - 理想的には、テストチームはユニットテストを実施する技術的能力を持っているはずだが、ユニットテストは通常、開発チームが実施した方がよい。ユニットテストが行われなかつたと仮定して、統合テスト/システムテストを進める方が好都合かもしれない。このケースでは、品質メトリクスにかかるコストを追跡することは、ユニットテストを回避した場合のプロジェクトにかかるコストを判断する上で特に重要である。
- サードパーティーが、社内開発製品のすべてのテストを実施する - これはテストマネジメントの課題であり、テストマネージャーは、このテストを実施するために使用されるテスト戦略を定義するか、定義されたテスト戦略に同意しなければならない。これには、テストアプローチ、テスト文書、テストツール、およびテストされた製品とともに開発・提供されるべき自動化について理解することが含まれる。このタイプのプロジェクトでは、テスト作業の進捗を綿密にマネジメントする必要があるため、開発のみをアウトソースする場合とはコミュニケーションニーズが異なる場合がある。

プロジェクトによっては、より広範なレベルの統合と引き継ぎの文書化が必要な場合もある。テストマネージャーは、その領域特有の要件を理解し、すべてのテスト要件の完了の妥当性を確認するための引き継ぎチェックポイント(マイルストーン)があることを確認することが重要である。

契約は、プロジェクトの唯一の強制可能な基準であるかもしれない。上述したように、契約仕様書が受け入れ可能な品質レベルを保証するには不十分である場合、テストマネージャーは、品質レベルを上げて組織内部のニーズを満たすために、将来の契約に利用できるメトリクスを追跡するべきである。理想的には、テストマネージャーは、製品の品質基準の決定と実施に大きな影響力を持っているものである。もしこの権限が契約に盛り込まれていない場合、テストマネージャーは、今後の契約にこの権限を含めなければならないことを証明するために、必要なメトリクスを収集することが不可欠である。テストからの欠陥の取りこぼし、欠陥の傾向、ユニットテストのカバレッジなどのメトリクスは、この要件を設定する手助けになる。

## 4.6 テスト戦略のマージ

テストマネージャーにとって、最もマネジメントが難しいプロジェクトの 1 つは、テストの一部をサードパーティーが実施するものである。これは、複数の場所、複数の組織、そして/または複数の内部/外部グループにまたがる可能性がある。このようなシナリオには、多くの複雑なレベルがある。以下のようなものがある:

- 異なる欠陥追跡ツールを統合する
- 異なるテストマネジメントツールを統合する
- 自動化戦略/ツールを定義し、統一する
- テストレベルとそのレベルに対する責任を定義する
- テストおよび品質に関する共通ゴールを定義する
- プロジェクトのさまざまなメンバーによって採用される可能性のある、さまざまなテストアプローチを理解する
- 受け入れ可能な開始基準と終了基準を定義する
- 技術用語やテスト用語を含む、共通の用語集を定義する
- プロジェクトのコントロールに使用する共通のメトリクスと報告頻度を定義する
- セキュリティ/機密性の問題に対処する
- チーム間の責任範囲と連携ポイントを特定する(重複が定義されていないと抜け漏れが発生し、重複が多いと非効率になる)

プロジェクトが進むにつれて、これらの問題はより重要になってくるため、プロジェクト開始前にアプローチやツールに関する疑問を解決しておくことが重要である。これには、スケジュールやプロジェクト計画に含まれていないかもしれない計画や調整が必要となる。いずれにせよ、これはやらなければならない作業であり、時間をか

けることで、いずれ成果として返ってくる投資である。これらの項目について合意を得る際に生じがちな別の困難として、他組織のテスト対応責任者の日程確保がある。テストリソースは、プロジェクトの進行に伴って割り当てられるので、計画を立てるべきタイミングで、計画立案や調整に必要なリソースが確保できないことがある。そのため、リソースが確保できた際に効率的に活用できるよう、計画作業の時間をプロジェクトスケジュール、場合によっては契約書にも明記しておくことが重要である。

複数の組織が製品のテストに関わる場合、テストアプローチの整合性が必要である。これは、全員が同じテストアプローチ、テスト技法、またはテスト自動化戦略を使わなければならないという意味ではないが、テスト結果の全体が、プロジェクトのテスト戦略で定義されたニーズを満たすことを意味する。通常、テスト戦略のオーナーは1人であり、それは通常、他の組織のサービスを調達している組織のテストマネージャーである。戦略を統一するための1つの方法として、さまざまなリスク要因を評価するためのリスクアセスメントプロセスを確実に実施し、優先順位付けされたリスク要因を全体的なテスト戦略で扱うようにすることである。リスクの理解を統一し、リスク軽減計画に合意することは、戦略の共通実施に役立つ。

## 4.7 品質の検証

テストマネージャーにとっての課題の1つは、受け取ったものが期待していたもの／要求していたものと一致していることを、どのように確認するかを決定することである。受け取るべき成果物を予想可能な形で定義するには、プロジェクトの開始時点で、できる限りプロジェクト文書や契約書に明確に記載しなければならない。すべての受領ソフトウェアの客観的な品質測定を、ソフトウェアが受け入れられる前に定義し、測定しなければならない。これは、プロジェクトの各フェーズについて、測定可能な開始基準と終了基準を定義することで達成できる。これらの基準は、プロジェクトのマイルストーンと整合させることもあるが、マイルストーンは、品質指向（品質ゲート）よりもスケジュール指向（スケジュールマイルストーン）の方が多いかもしれない。

以下は、テストサイクルの各レベルへのソフトウェアの受け入れを決定するために使用される開始基準と終了基準の例である：

- ユニットテスト終了 - ステートメントカバレッジが85%以上である
- 統合テスト開始 - コードの静的解析が完了し、未解決のエラーがない
- 統合テスト終了 - 機能領域の全コンポーネントが統合された（インターフェースが正確に動作することが確認された）
- システムテスト開始 - 未解決でブロックとなる欠陥はない
- システムテスト開始 - 既知の欠陥をすべて文書化する
- システムテスト終了 - すべての性能要件を満たす
- システム統合テスト開始 - 未解決の優先度/重要度の高い欠陥がない
- 受け入れテスト開始 - テストグループにより予定されていたすべてのテストが完了し、テスト結果が所定の基準を満たす
- 受け入れテスト終了 - 受け入れ当事者による承認

実際のプロジェクトのリストは、より広範なものになるだろうし、基準の厳しさは、製品分野、サードパーティとの業務経験、契約／合意書に記載された要件によって異なるだろう。重要なことは、基準が客観的な方法で測定可能であり、プロジェクト開始時に合意されることである。基準が有用であるためには、基準は強制可能でなければならない。そのためには、プロジェクトを受け入れまたは拒否する権限を持っていることを確実にしなければならない。

マイルストーンは、プロジェクトを追跡し、スケジュール通りに進めるために重要である。基準と同様に、マイルストーンは、マイルストーンの完了が実証できるように定義されなければならない。このため、マイルストーンは上記で定義した基準と関連付けられることが多い。マイルストーンのトラッキングは、プロジェクトマネージャーの責

任であることが多いが、マイルストーンがテストスケジュールや品質関連の問題に関連する可能性がある場合は、テストマネージャーもトラッキングすべきである。

**5. 組織横断マネジメント****780 分**

キーワード

なし

**組織横断マネジメントの学習の目的****5.1 イントロダクション**

本節の学習の目的はない。

**5.2 テストチームの価値の提示**

- LO 5.2.1 (K4) テストチームの価値を提示し、宣伝するために取るべき適切なステップを定義する。
- LO 5.2.2 (K6) 特定の状況において、テストの定量的および定性的な利点を定義し、それらの利点をプロジェクトのステークホルダーに効果的に伝える。
- LO 5.2.3 (K3) テストマネージャーがテストチームを擁護する必要がある状況の例を挙げる。

**5.3 テストチームの配置**

- LO 5.3.1 (K5) 組織内のテストチームの適切な配置の選択肢を決定するために、組織の構造、ミッション、製品、顧客、ユーザー、および優先度を評価し、それらの選択肢の意味を評価し、それらの選択肢の上層マネジメントのための分析を作成する。

**5.4 ステークホルダーとのコミュニケーション**

- LO 5.4.1 (K3) テストに関連する重要な問題について、テスト担当者以外を含むテストのステークホルダーと効果的にコミュニケーションを取る。

**5.5 関係の形成と発展**

- LO 5.5.1 (K3) 特定の状況において、他のマネージャーやチームとどのように関係を築き、構築するかを示す。
- LO 5.5.2 (K3) 複雑度の高いプロジェクト環境において、テストマネージャーが構築すべき重要な関係について説明し、例を挙げる。

**5.6 組織全体にわたる品質活動の推進**

- LO 5.6.1 (K2) 品質に関連する活動に関与している、組織内の他の部門を識別する。

**5.7 組織全体にわたるツールの統合**

- LO 5.7.1 (K2) マルチユースツールを扱う際に考慮すべき事項を定義する。
- LO 5.7.2 (K4) マルチユースツールの変更案を分析し、テスト組織への影響を評価する。
- LO 5.7.3 (K6) ツールのライフサイクルの全フェーズを考慮して、新しいマルチユースツールを展開するための計画を作成する。

**5.8 倫理的問題への対処**

- LO 5.8.1 (K5) 特定の状況下における一連の行動を評価し、起こり得る倫理的問題を特定する。

## 5.1 イントロダクション

テスト組織内で優れたマネジメントを行えるだけでは不十分である。成功するテストマネージャーは、対外的なマネジメントも効果的に行わなければならない。これには、上司のマネジメント、他の開発チームや連携が必要なチームのマネジメント、効果的なコミュニケーションとデータの共有が含まれる。これは、一般的な組織構造や報告構造の違いに関係なく当てはまる。

## 5.2 テストチームの価値の提示

テストマネージャーの仕事の重要な部分は、テストチームと組織への貢献を支持し、擁護し、宣伝することである。

### 5.2.1 テスト組織の宣伝と価値の提示

テストマネージャーは、テストチームとテストの利点を宣伝しなければならない。これには、チームが達成した成功や、チームが生み出した利点、将来的に改善される可能性のある現実的な評価などを明示することが含まれる。テストチームの成果は正直に提示されるべきであり、改善すべき点を伴うべきである。ミスがあった場合は、そのようなミスを率直に認め、対処した是正措置、あるいは予定している是正措置とともに報告することが、通常、ごまかしや責任転嫁よりもテストチームのためになる。

テストマネージャーは、テストチームを支持し、継続的に上層マネジメントにもテストチームを支持するように働きかけるべきである。テストマネージャーは、チームに十分な情報を提供し、適切な人員を配置し、十分な訓練を受けさせ、適切なリソースを確保し、組織全体から尊敬されるようにしなければならない。

テストマネージャーは、敬意を持って効果的かつ一貫性のあるコミュニケーションを行い、チームのために機会を確保し、チームが相応の尊敬を受けるようにすることで、この擁護者の役割を果たす。テストチームは、ほとんどの技術チームと同様に、尊敬を得なければならない。テストマネージャーは、他のグループと敬意を持って接することで、チームが敬意を持って扱われる道を開く手助けをする。テストマネージャーは、チームのトレーニングと昇進の機会に常に注意を払っていなければならない。場合によっては、テストマネージャーは、改善の余地や新たな機会を察知し、チームがそれを活用できるように素早く配置することを意味する。例えば、テストチームのメンバーの中に、テスト対象のテクノロジーに対するスキルレベルで遅れを取っている者がいる場合、テストマネージャーは、そのような人たちのためにトレーニングの機会を確保し、新たなスキルが実務で発揮できるようにしなければならない。

客観的なメトリクスでは、品質コストの削減と、テストチームとテスト活動によって促進される効率性と生産性向上の両面から、組織全体に対するテストチームの重要性を裏付けるために使用するべきである。さらに、テストマネージャーは、組織が取り得るさまざまな選択肢の結果を予測するためにメトリクスを活用できる。これはテストチームから得られる付加価値の1つである。

### 5.2.2 テストの価値の売り込み

エキスパートテストマネージャーは、テストの価値を効果的に伝え、ときには売り込むことができなければならない。テストは、定量的および定性的な方法で組織に価値を提供する。定量的には、品質コスト分析を用いて、テストによる利点を、節約された費用や時間の観点から判断することができる。定性的には、テストの便益は、顧客満足度の向上、リスクの低減、好意の増大、および同様の有益な見返りの観点から評価することができる。テストの利点は、業界や組織によって異なる。テストマネージャーは、テストの利益とコストを、ステークホルダーが理解し納得できる言葉で正確に提示できなければならない。

テストの利点に関する情報は、テストマネージャーが、必要なテスト実施に必要なリソースを調達するため、また、将来の要件に対応したテスト組織を構築するために利用されるべきである。テストの価値については、[Black03]を参照のこと。

### 5.2.3 説明責任を果たせるチームの構築

テストマネージャーとしてのキャリアの中では、プロジェクトが計画通りに進まないことがあるだろう（極端な例として[Yourdon03]を参照）。その際、テストチームが問題の原因の 1 つとして名指しされる可能性がある。こうした状況において、テストマネージャーがどのように対応し、チームを擁護するかを理解しておくことが重要である。

説明責任を果たせるテストチームを作る鍵は、主に次のような予防的な取り組みである。

- オープンなコミュニケーション
- 適切なドキュメントの整備
- 強固なプロセスの構築

会社全体とプロジェクトのステークホルダーは、テスト組織の役割を理解する必要がある。テストマネージャーは、特定のプロジェクト内での役割だけでなく、グループの全体的な責任に対する期待を確立する必要がある。これは、テストチームの役割と責任のマトリクスを提供したり、特定のプロジェクトのテストアプローチやテスト計画書を提供することによって行うことができる。

テスト組織は、構造化されたテストプロセス (JSTQB Foundation のシラバスで紹介されているテストプロセスのようなもの) に従うべきである。このプロセスには、適切な文書化とフィードバックの仕組みが含まれており、問題や課題を早期に特定、報告し、問題なく解決できるようにする。強力なプロセスを持つことで、プロジェクトのステークホルダーは、テスト組織がどのように機能するかを理解することができ、チームに対する信頼を得ることができ、プロジェクトのライフサイクル中に何を期待すべきかを理解することができる。

テストマネージャーは、チームから継続的に情報を収集し、プロジェクトのステークホルダーに状況を伝えなければならない。テスト文書、ステータスレポート、欠陥情報を通して、適切なレベルでテスト組織の活動を可視化すべきである。突然の報告で関係者を驚かすべきではない。

プロジェクトの問題が発生した場合、テストマネージャーは迅速かつ効率的に対処すべきである。テストマネージャーは、言い訳をするのではなく、客観的な手段によって、何が発生し、プロジェクト問題に関連するリスクを軽減するために何ができるかを示すべきである。

アジャイルチームにおいては、テスト担当者は（アナリストやユーザー／顧客を含むことが多い）統合された開発チームの一員である。そのため、テストチームとして「説明責任を果たすような形で自らを弁明する」必要はない。むしろ、チームメンバーがテスト活動の意義を理解し、主体的に関与することが求められる。これを実現するには、オープンなコミュニケーションの継続、適切なプロセス、テストの努力が適切に認識されるようにすることが必要である。

### 5.2.4 チームの保護と支援

テストチームは、善意であろうとなかろうと、外部からの干渉を受けやすい。これは通常、以下のいずれかの理由で発生する：

- テストタスクが理解されておらず、見積りが甘い。
- テストが過小評価されている
- テストのスキルが過小評価されている。特に、テスト担当者でないスタッフが、誰でもテストができると信じている場合である。
- テスト担当者がテストチーム内で効果的に活用されていない。

- 開発部門はテストへの取り組みに不満を抱いており、テストを自分のものにしたいと考えている。
- テストマネジメントは、プロジェクト全体にとって邪魔で、非協力的で、非効率的だと思われている。
- テストはスピードが遅すぎる、あるいは時期が遅すぎると思われている。
- テストの責任と境界が、明確かつ測定可能な形で定義されていない。

テストマネージャーの仕事は、これらの状態が存在しないこと、あるいは悪化して持続しないことを確認することである。これらの状態のいずれか、またはそれらの組み合わせは、以下のようなタイプのリソースと尊重の問題につながる可能性がある。

#### 5.2.4.1 外部からの干渉

テストチームは、他のプロジェクトステークホルダーや他の組織から干渉を受けることがある。例えば、テストチームがテスト完遂に時間がかかりすぎると見なされた場合、開発組織は、何をテストすべきで、何をテストすべきでないか、また、どの程度の深さまでテストすべきかについて指示しようとすることがある。テストチームがテスト範囲について提案を受けることは珍しくないが、テストチームがテスト戦略やアプローチに責任を持たなくなり、品質重視の姿勢を保てなくなると問題になる。

外部からの干渉は、多くの場合、スケジュールのコントロールという形を取るが、テストケースのフォーマット、自動化戦略、テスト文書化、および常に品質を重視すべき他の領域に及ぶこともある。プロジェクトのステークホルダーやインターフェース組織からのインプットは歓迎すべきであるが、強力なテストマネージャーは、他の組織や個人がテストチームのミッションやそのミッションを達成する能力を妨害することを許すべきではない。

#### 5.2.4.2 マイクロマネジメント

外部からの干渉と同様に、テストタスクとテストリソースのマイクロマネジメントは、信頼の欠如を示している。マイクロマネジメントは、テストタスクや全体的なミッション、戦略、アプローチに対する可視性が欠如しているために発生することがある。外部の組織や個人が、テストチームやテストチームメンバーをマイクロマネジメントしようとする場合、テストマネージャーは、マイクロマネジメントの原因を特定することによって、問題を修正しなければならない。例えば、プロジェクトマネージャーがテストチームメンバー（そしてテストチームメンバーだけ）に対して、1時間単位で時間を記録することを求めていた場合、不信感があり、テスト情報へのアクセスが不十分であるとプロジェクトマネージャーが感じている可能性が高い。

マイクロマネジメントシナリオを避けるために、テストマネージャーは、全体的なスケジュールの計画に参加しならなければならない。テスト計画書と進捗状況をプロジェクトチームに明確に伝え、追加情報の要求に応えなければならない。テストアプローチ、テスト計画書、テスト見積りについて、ステークホルダーによるレビューと承認を含めた、計画活動中の明確なコミュニケーションが有効である。テストチームの活動とテスト計画書が見えれば見えるほど、チームがマイクロマネジメントにさらされる可能性は低くなる。

#### 5.2.4.3 敬意の欠如

テストマネージャーは、直接的であれ間接的であれ、テストチームに対する敬意を欠いた言動を決して容認すべきではない。こうした敬意の欠如は、しばしばテストチームの貢献価値に対する理解不足に起因する。例えば、テスト担当者のスキルに対する皮肉めいた発言、心無い態度、不適切なタスクの割り当て、時間やリソースの不十分な配分、トレーニングや知識共有へのアクセスが不十分、その他これに類する行為が挙げられる。

#### 5.2.4.4 組織の再編成とリソースの再配置

テストチームは、組織再編やリソースの再配置によって、コミットされた作業に対してテストリソースの確保ができず、不当に影響を受ける可能性がある。上記の問題と同様に、これは通常、組織の他のメンバーがテストタスクを理解しておらず、行われるべき作業を過小評価している場合に発生する。

人員削減の極端なケースでは、テストチームは開発組織よりも深刻な影響を受けるかもしれない。これは、開発がより簡単に定量化された結果を出すのに対して、テストの価値が人々に理解されていない（テストマネージャ

一が測定していない場合に発生する可能性がある。テストマネージャーは、テスト組織の価値と貢献が、組織の他のメンバーによく理解され、評価されるようにしなければならない。

### 5.3 テストチームの配置

すべての組織は違うものであり、組織構造も多種多様である。さらに、多くの組織が、プロジェクトごとに異なる構造を採用している。すべての組織やプロジェクトに通用する完璧な構造は存在しない。しかし、成功するテスト組織には、以下に述べるような共通の特徴がある。

要求されたスケジュールと予算の範囲内で作業する一方で、テストチームの第一の焦点は品質に置かれるべきである。品質とは、完璧な製品だけをリリースすることを意味するわけではない。テスト中のソフトウェアの品質を正確かつ誠実に評価し、報告できることを意味する。特にマネジメント層からの不当な扱いに恐れることなく行うことを意味する。品質の高い製品を作り上げるためには、品質への取り組みに十分な時間と資金を割かなければならぬことを理解した上で、スケジュールと予算を作成すべきである。

理想的には、品質重視のテストチームは、組織の最上層に至るまで品質を重視するマネジメント層のレポートラインを持つ。テストマネージャーの仕事は、組織のミッションと目的を達成しながら、品質への注力を促進するために、テストチームが最も適切なポジションを確保できるようにすることである。

品質を重視することに加え、テストチームの配置には他にも考慮すべき点がある。テストチームの成功には、情報へのアクセスしやすさが重要である。情報は通常、開発チーム、アナリストチーム、ビジネスステークホルダー、プロジェクトマネージャー、その他多くのプロジェクトステークホルダーからもたらされる。テストチームは、形式的(仕様書や要件の形式)にも非形式的(電子メール、会話、ホワイトボードディスカッションなど)にも、これらの情報源から得られる情報に適切にアクセスできなければならない。

テストチームが持つ全体的なスキルレベルと知識のタイプも重要である。ある組織では、ドメイン知識が各テスト担当者の重要要件であるかもしれない。また、テストに関する知識が最も評価される組織もある。システムコンフィギュレーション、ネットワーキング知識などの技術的スキルが高く評価される場合もある。組織において、テストチームが提供するスキルの組み合わせは、チームが受ける敬意と情報へのアクセスしやすさに寄与する。必要なスキルの組み合わせは組織固有ではなく、プロジェクト固有な場合がある。例えば、アジャイルチームでは、テストチームは開発チームの一部であり、プログラミングの専門知識が求められるかもしれない。アジャイルチームはより自己組織化する傾向があるため、個人は自分に最も適した役割に進化する傾向がある。テストマネージャーとして重要なのは、外部および内部のステークホルダーからの要件や期待の変化を把握し、それに応じてテストチームの人員配置やトレーニングが必要に応じて行われていることを確認することである。

### 5.4 ステークホルダーコミュニケーション

成功するテストチームは、必要とする情報に適切にアクセスできる。また、さまざまなステークホルダーに対しては、正確かつタイムリーで理解しやすい形で情報を提供しなければならない。その際には、各ステークホルダーの情報ニーズに合った形式や内容で伝えることが求められる。このコミュニケーションには、テストプロジェクト全般に関する情報、特定のテストステータス、特定の欠陥ステータス、トレンド情報、テストプロセスの有効性と効率性指標、その他のメトリクスが含まれ、定期的、周期的にコミュニケーションされる。成功するテストマネージャーは、他のステークホルダーとのコミュニケーションが双方向であることを理解し、情報の提供と収集の両方を効果的に行うことができる。

「いつ」「どのように」伝えるかを考えるとき、テストマネージャーは情報の緊急性と情報の明確性を評価する必要がある。例えば、スケジュールにおける緊急の課題は、関係するステークホルダーに積極的に伝えるべきである。一方、関心を持ちそうな人には、週次のステータスレポートを社内ポータルに掲載し、そのリンクを公開す

れば十分かもしれない。もしその情報が、メッセージを明確に理解するために、ある程度の解釈を必要とするものであれば(例えば、システムテスト中の品質問題は、ユニットテストの不足を示している)、その情報は、ミーティングで提示するのがよいだろう。その場合は、発生する可能性のある質問にもテストマネージャーが答えることができるためである。

テストマネージャーは、1人以上のステークホルダーに実際に必要とされ、かつ付加価値をもたらす情報の収集と解釈に焦点を当てるべきである。多すぎるコミュニケーションや、ステークホルダーにとって関係のない情報のコミュニケーションは、情報を受ける側に負担を与え、提供する側の時間を無駄にする。コミュニケーションの動機は理解されなければならない。もし、コミュニケーションの目的が経営陣の支援、またはコミットメントを得ることであるならば、受ける側の人たちが何を求められているかを理解し、予想されるコストと利点を理解し、求められている行動を理解するようにコミュニケーションを方向づけなければならない。

効果的なコミュニケーションとは、メッセージを送った人の意図と、メッセージを受け取った人の理解が同じである場合に発生すると定義できる。もし、送った人が、コミュニケーションの結果として受けた人が行動を起こすことを意図していた場合、その行動が起こればコミュニケーションは効果的だといえる。

## 5.5 関係の形成と発展

他のマネージャーや他のチームと関係を形成し、発展させ、維持する能力は、テストマネージャーの成功の決定的な要素である。成功するテストマネージャーは、このような関係を開拓し、相互の利点を判断し、その関係を維持できなければならない。例えば、テストマネージャーは、開発マネージャーと、プロジェクトの開発フェーズのステータス、期待される品質、期待されるメトリクス、特定された成果物について、正直に話すことができなければならない。また、契約上の制限の範囲内で、テストマネージャーは、プロジェクトに成果物をインプットする可能性のある、あるいはプロジェクトからのアウトプットの利用者である可能性のある、ベンダーや他のサードパーティーグループと効果的なコミュニケーションができなければならない。状況はそれぞれ異なるが、テストマネージャーは、公式なコミュニケーションの機会だけでなく、非公式なコミュニケーションの機会も考慮し、関係構築に役立てるべきである。開発マネージャーをランチに誘う、コーヒーブレイクの時間を共有する、新しい技術の議論にステークホルダーを参加させる、他のプロジェクトでの成功を認めて話し合う、などはすべて関係構築に役立つ方法である。

しかし、関係を構築するだけでは十分ではない。テストマネージャーは、組織の環境が変わり、人材が変わり、責任や期待さえも変わっても、それらの関係を維持する努力もしなければならない。成功するテストマネージャーは、このような組織の変動に適応することで、尊敬され、生産性の高いテストチームを率いることができる。

必要な人間関係とは、同じ仲間に限定されるものではなく、個々の貢献者、上層部、プロジェクトマネジメント、その他、プロジェクトやテスト組織に貴重な貢献をしてくれそうなステークホルダーも含まれるべきである。テストマネージャーは、組織図に示された上下関係が特定の個人によって行使される権力と影響力の1つの指標に過ぎないことを忘れてはならない。本当の権力や影響力は、特に平等主義的な組織文化や民族的文化においては、尊敬からも生まれるものである。テストマネージャーは、目に見える貢献者であり、テストチームを代表する強力で尊敬される存在であるべきである。

社外との関係構築に加え、テストマネージャーは、テストチーム内のリーダーや個人貢献者との社内関係にも、同様の丁寧さと配慮を払わなければならない。対外的な関係と同じレベルの敬意、コミュニケーション、プロフェッショナリズムを、対内的な関係にも適用しなければならない。

テストマネージャーは、プロジェクトチーム内だけで仕事をしている間は、すべての参加者を知っている可能性が高く、それらの人々と個々の関係を構築することができる。スコープが大きくなるにつれて、人間関係はより遠くなり、会ったことがないさまざまなプロジェクトチームのメンバーも出てくる。この場合、テストマネージャーは、関係を構築する代表的な人物を選ばなければならない。最低限、テストマネージャーは、さまざまなプロジェクト

チーム内の同僚と良好な関係を持たなければならない。また、テストマネージャーは、実際に業務を遂行する人たち（マネージャーではないこともある）とも関係を形成しなければならないし、テストチームが必要とする情報やリソースにアクセスできる人たちとも関係を持たなければならない。

多くのマネジメント業務と同様に、関係構築もうまく委任できる場合がある。マネージャー自身が、プロジェクトチームの全メンバーと個人的な関係を持つ必要はないが、テストマネージャーは、必要に応じて各メンバーにアクセスできる人物を知っている必要がある。このような「間接的」なマネジメントを身につけるのは難しいかもしれないが、大規模または分散環境では必須である。

テストマネージャーがマネジメントチェーンを昇り、責任範囲を拡大するにつれて、政治的スキルとステークホルダーマネジメントスキルがより重要になる。これらのソフトスキルは、テストグループの仕事を可能にする人間関係を構築し、維持するために不可欠である。主な関係者には、プロジェクトマネージャー、開発マネージャー、主要なステークホルダーの代表者、ビジネスアナリスト、要件エンジニア、ビジネス担当者、マーケティング、営業、テクニカルサポート、技術文書の作成者、ベンダー、およびテスト作業や品質評価に貢献または影響を与えるその他の関係が含まれる。

## 5.6 組織全体にわたる品質活動の推進

テストと品質活動は、テストグループに限定すべきではない。製品の定義、作成、サポートに貢献するすべてのグループは、その製品の品質にも貢献し、テスト責任を負っていることを認識すべきである。例えば、次のようなことがある：

- プロジェクトステークホルダーは、要件、設計文書、アーキテクチャ文書、データベース設計文書、テスト文書関連、ユーザー文書全般などの作業成果物のレビューに参加すべきである。
- ビジネスアナリストは、構築される製品が顧客のニーズを満たす製品であることを保証するために、妥当性をテストすることを頻繁に求められる。
- 開発者はユニットテスト、統合テストを実施する他、システムテストやシステム統合テストでテストチームが使用する自動化テストを作成する場合もある。
- 顧客担当者がユーザー受け入れテストを行う。
- 運用担当者は運用受け入れテストを行う。

このテスト結果は、テストマネージャーが収集し、テストチームが実施するテストのインプットとして、またテストチームが実施したテストの品質評価として使用するべきである。例えば、ユーザー受け入れテストで発見された事項は、要件レビューが不十分であったこと、システムテストが完了していなかったこと、あるいはサンプルテストデータが本番データを反映していないかったことを示す可能性がある。

他のテスト業務に関連するもう1つの課題は、他のグループが、テストグループで通常使用されるような厳密なプロセスに従って、欠陥の追跡とマネジメントを行っていないことが多いことである。テストマネージャーは、これらの他のグループと協力して、適切な解決を確実にするために、そのような欠陥について十分な情報を収集しなければならない。欠陥やその他のテスト関連情報を、テストグループで使用するのと同じマネジメントツールを使用して欠陥やその他のテスト関連情報を取り扱うと、助けになる。

テストマネージャーが、他のテスト作業を把握し、それが行われたかどうか、どの程度行われたかを検証する方法がない限り、テストチームは、すでに十分にテストされた領域を再テストしたり、他のプロジェクト関係者のスキルを十分に活用できなかつたりして、非効率になるリスクがある。

もちろん、他のグループがテストタスクを実行すると期待されているからといって、実際に実行するとは限らない。また、テストチームは、適切な検証なしに、他のグループがこれらの条件をカバーしていると思い込んだ場合、重要な条件のテストを見落とすリスクもある。これらの他のグループも、テストグループと同様に、タイトなスケジ

ユール、スコープの肥大化、リソースの問題に対処している。成功するテストマネージャーは、ある組織内のあるプロジェクトの品質、スケジュール、予算、機能の関係を理解している。トレードオフは想定しておくべきであり、テストチームは、プロジェクトのさまざまな側面の間で起きる優先度の変化に対処する準備すべきである。アジャイルチームでは、テストマネージャーは、プロジェクト全体を通して品質活動を提唱することで、品質コーチの役割を引き受けことがある。

## 5.7 組織全体にわたるツールの統合

ツールは、組織内の複数の職種間で共有されることが多いが、ツールの使用方法はさまざまである。例えば、欠陥追跡ツールを、テスト組織は欠陥の追跡に、開発組織は実装タスクの追跡に、ビジネスアナリストは要件の追跡に使用することができる。このツールに格納されたデータは、個々のグループには有用であるが、各データのグループ化(欠陥、タスクなど)を適切に処理できるように、データ中の識別子を慎重に計画しなければ、集計された形で使用することはできない。例えば、トレンドチャートは欠陥については意味があるが、開発者のタスクについては意味がない。

理想的には、さまざまな機能のためにグループを越えて使用されるツール、マルチユースツールは、すべての関係者にとっての使用性を確保しつつ、ツールの可能性を最大化するよう、あらかじめ定義された計画に従って取得、導入、使用されるべきである。より多くの場合、ツールの使用法は時間とともに成長し、ほとんど文書化されていない。その結果、保守性、更新、データ変換、ツールの移行に問題が生じる。

テストマネージャーは、マルチユースツールの所有者であることもあれば、ツールによって保守されるデータのアクセス容易性や正確性に重大な関心を持つステークホルダーであることもある。そのため、テストマネージャーは通常、このタイプのツールの購入、更新、移行、廃棄に関する問題に関与し、ときには多くの時間と労力を割くこともある。これは、テストマネージャーがすべての利用状況をコントロールできるツールを扱う場合よりも、はるかに複雑である。

マルチユースツールには、シングルユースツールとは異なる問題がある。ツールのライフサイクルの各段階には、特有の考慮事項がある:

- 購入／選択／取得
  - ツールの購入、使用、メンテナンスに関わるすべてのステークホルダーを含む選定委員会が必要となる。
  - 統合は、単一ツールレベルでも重要だが、マルチユースツールにおいては不可欠である。すべての統合に必要なポイントを定義し、データタイプを特定し、データ転送の仕組みを理解し、カスタマイズしたコードの要件(および誰がそのコードを書き、保守するか)を明確にしなければならない。
  - ツールに備わっている、ユーザーおよびアドミニストレーターのさまざまな権限定義をサポートする能力を明確に理解しなければならない。また、それらのユーザーをどうマネジメントするかについても同様に理解しておかなければならない。
  - ツールのオーナーシップは、誰がその費用を負担するのか、誰が導入作業を行うのか、誰が保守性を維持するのか、誰が導入の判定権限を持つのかなど、明確に示されなければならない。
- 更新／保守／サポート
  - ツールを単一の多用途ツールに初期統合するには、プログラミング、テスト、およびサポートリソースが必要となる場合がある。「最善の」ツールの選択と、その後のツールの統合には、かなりの計画と技術的専門知識が必要となる。
  - 誰が何を変更できるのか、変更のために必要な承認について、すべての関係者が定義されたプロセスに合意しなければならない。
  - ユーザーベースのサポート(ヘルプデスク)の責任を割り当て、資金を調達しなければならない。

- 各利用グループによるツールのカスタマイズに関するルールを定義し、明確に伝え、モニタリングしなければならない。
- データ移行(必要な場合)は、文書化され、ツールのサポートにより、入力されるデータの品質や既存のデータのインテグリティを損なうことなく、データが新しいツールに流れるように促進されなければならない。
- 影響度分析は、ツールの他の利用者に影響があるかどうかを判断するために、ツールセットに対して検討された修正について実施されなければならない。
- バックアップとリストアの要件と責任、およびサポート関連機能のSLAを定義しなければならない。
- 移行
  - ツールは決して永続的なものではなく、ビジネスニーズの変化、ベンダーの問題、強制的なアップグレード、その他の外的要因によって、現在のツールセットからの移行を余儀なくされることもある。
  - ツールへの移行およびツールからの移行のいずれについても、データの完全性、サービスの継続、一貫したデータアクセス、および許容可能なレベルの使用性を確保するために、慎重に計画されなければならない。
- 廃棄
  - ツールが時代遅れにならない限り、代替手段が用意されていなければ、通常はそのツールを廃棄できない。
  - ツールの切り替えと同様、ツールの廃棄後はビジネス機能の継続を可能にする別のツールや手法へ切り替えることが要件となる。
  - アーカイブされたデータは保存され、セキュリティで保護され、場合によってはアクセシビリティを維持しなければならない。

ツールの統合が容易であることは稀であり、かなりの時間、労力、外交力、思慮深さが必要となる。すべてのステークホルダーにとって受け入れ可能な解決策に到達するためには、通常、妥協、交渉、創造性が要件となる。ツールを 1 つのステークホルダーグループ内で所有させるのではなく、組織全体のビジネスニーズをサポートしながら、すべてのステークホルダーのために独立した客観的な使用ガイドラインを提供するために、別のツールグループがしばしば形成される。

## 5.8 倫理的問題への対処

倫理的な問題は、プロジェクトのさまざまな段階、さまざまなやりとりの中で遭遇する可能性がある。このような状況に対処するテストマネージャーの役割は、以下の項で議論する。

### 5.8.1 チームにおける行動規範のマネジメント

テストマネージャーは、ISTQB 行動規範に従わなければならない。行動規範で述べられている倫理は組織内外でのあらゆるやりとりで適用する。テストマネージャーの倫理は、チームメンバー個々の倫理観に対する手本となる。テストマネージャーは、チームがこれらの倫理を徹底するように働きかけ、定期的にトレーニング、レビュー、状況分析などを通じて行動規範の理解や遵守を強化すべきである。行動規範の順守は、人事評価の際に報奨の対象とすべきであり、規範に反する行為は容認すべきではない。

倫理は明確で当然のことのように思えるかもしれないが、決してそう思い込んではならない。行動規範を明文化し、かつ、組織内の他の個人の行動がどうであれ、行動規範で述べられている倫理に従うことが求められることを、チームが理解していることが重要である。テストチームは頻繁に対立状況に直面するため、プロフェッショナルな行動が重要である。一瞬の過ちが、個人とテストグループ全体にとって、深刻な長期的結果を招くこともある。

### 5.8.2 テストのステークホルダーとのやりとり

テストマネージャーは、さまざまなテストのステークホルダーと接するすべての場面で、プロフェッショナルとしての振る舞い、事実に基づいた客観的なレポート作業、ISTQB 倫理規定の順守が必要となる。テストマネージャーは、客観的で正確な情報の提示者となることで、テスト組織に対する尊敬と価値を築くことに貢献する。テストマネージャーは、ステークホルダーから信頼されなければならない。

例えば、テストチームが時間に追われているときや、他の組織が品質ゴールに対応するためではなく、スケジュールに合わせるためにテストを調整しようとしているときなど、テストの進捗状況を報告する際に感情を排除するのは難しいかもしれない。テストマネージャーは、さまざまなステークホルダーの視点や動機を理解しなければならないが、客観的なテストステータスの情報を、どんなに不評な情報であっても、明確かつ正直に提示しなければならない。

### 5.8.3 レポート結果

テストマネージャーと社外のテストのステークホルダーとのコミュニケーションの大部分は、文書レポート、チャート、メトリクス、プレゼンテーションの形で行われる。すべてのレポートは、データが正確であること、正しく明確に表示されていること、個人をターゲットにしていないことを確認するためにレビューしなければならない。例えば、発見した欠陥のレポートは、決して開発者別に報告すべきではなく、ソフトウェアの分野別、機能別、品質特性別、またはその他の個人を特定しない区分けを使用して報告すべきである。

テスト結果の報告に加え、欠陥レポート、電子メール、状況報告、その他の文書や口頭によるコミュニケーションなどの内部コミュニケーションは、常に客観的で公正であり、文脈が理解できるように適切な背景情報を含むべきである。テストマネージャーは、テストチームの全メンバーが、すべてのコミュニケーションにおいてこれらの要件に準拠することを保証しなければならない。テストチームは、たった 1 つのプロらしくない内容で書かれた欠陥レポートが上層マネジメントの手に渡っただけで、信頼を失うこともある。

### 5.8.4 テストマネジメントの倫理

チームの倫理と自分自身の個人的な倫理をマネジメントすることに加えて、テストマネージャーは、一般的にマネージャーに要求される倫理も考慮しなければならない。マネージャーは、倫理規定によって、従業員に対する信頼を損なう行動を禁じられている。倫理規定には、パフォーマンス問題の議論、チームメンバー間の対人問題の解決、プライバシーに関するデータの使用など潜在的な法的問題の処理、およびその他の人事関連状況に関するガイダンスが含まれている。特にテストマネージャーは、プレッシャーの高い環境で働いており、多くの場合、スケジュール、予算、品質、技術的な問題を両立させている。テストマネージャーは、このようなプレッシャーのかかる環境で発生する個々の人事問題にも対処しなければならず、それらの問題を迅速かつプロフェッショナルに処理しなければならない。

テストマネージャーは、マネージャーとして、他のマネージャーとオープンで誠実な関係を持たなければならぬ。テストマネージャーが、他部門の社員のパフォーマンス問題を最初に耳にすることは珍しくない。例えば、テストチームは、常に悪いコードを作っている開発者を最初に発見することがよくある。そのため、テストマネージャーは、開発マネージャーに適切な行動を取るために必要な情報を提供する立場になる。このデリケートで歓迎されにくい情報を伝える際は、すべてのコミュニケーションと同様に、専門的かつ事実に基づいて行わなければならない。他のマネージャーとこのような関係を築くことが特に重要になるのは、人事問題などが、グループ間の連携領域で顕在化する場合である。人間関係の構築は難しく、また簡単に損なわれる。それぞれの状況において、最も適切な行動を決定するための判断が必要である。

## 6. プロジェクトマネジメントエッセンシャルズ

615 分

### キーワード

コンフィデンスインターバル、プランニングポーカー

### プロジェクトマネジメントエッセンシャルズの学習の目的

#### 6.1 イントロダクション

本節の学習の目的はない。

#### 6.2 プロジェクトマネジメントタスク

- LO 6.2.1 (K6) あるプロジェクトについて、所定の見積り方法のうち少なくとも 2 つを使用してテスト工数を見積る。
- LO 6.2.2 (K6) 類似プロジェクトの過去データを使用して、現在のプロジェクトで発見、解決、およびリリース後に見つかる欠陥数を推定するモデルを作成する。
- LO 6.2.3 (K5) プロジェクト期間中、テストコントロールの一環として現状を評価し、テスト計画書からの逸脱を特定し、逸脱を解消する効果的な方法を提案するなど、テスト工数を継続的にマネジメント、追跡、調整する。
- LO 6.2.4 (K5) プロジェクト全体の変更(スコープ、予算、ゴール、スケジュールなど)の影響を評価し、その変更がテスト見積りに与える影響を識別する。
- LO 6.2.5 (K6) 過去プロジェクトの履歴情報と、プロジェクトステークホルダーから伝達された優先度を使用して、品質、スケジュール、予算、およびプロジェクトで利用可能な機能との間の適切なトレードオフを判断する。
- LO 6.2.6 (K2) 変更マネジメントプロセスにおけるテストマネージャーの役割を定義する。

#### 6.3 プロジェクトリスクマネジメント

- LO 6.3.1 (K4) リスクアセスメントワークショップを実施し、テスト作業に影響を及ぼす可能性のあるプロジェクトリスクを特定し、テスト関連のプロジェクトリスクに対して適切なコントロールとレポートメカニズムを実装する。

#### 6.4 品質マネジメント

- LO 6.4.1 (K4) 組織全体の品質マネジメントプログラムにテストがどのように適合するかを定義する。

## 6.1 イントロダクション

テストマネージャーのプロジェクトへの関与範囲は、多くの領域にまたがることが多い。プロジェクトマネジメントのように、これらすべての領域に直接的な責任があるとは限らないが、成功するテストマネージャーは、プロジェクトマネジメント、リスクマネジメント、レビューとアセスメント、テストチームが作成しなければならない適切な文書レベルの決定に注意を払い、関与することは間違いない。

ソフトウェアマネージャーに関連するプロジェクトマネジメントエッセンシャルズと関連する概念については、[Drucker06]、[Rothman07]、[McConnell97]、[Brooks95]、[Glass02]、[PMI08]を参照のこと。

## 6.2 プロジェクトマネジメントタスク

テストマネージャーは、主にテストプロジェクトをマネジメントするが、テストマネージャーは、全体的なプロジェクトマネジメントのタスクも意識し、関わらなければならない。これらの各分野については、以下で詳しく説明する。テストマネージャーにとって重要なことは、テストプロジェクトは、プロジェクトの他の部分から切り離されて発生するものではないということを覚えておくことである。

### 6.2.1 テスト見積り

テスト見積りには、テストプロセスのすべての活動を確実に実施するために必要な時間と労力を含める。テスト実行とレポート作業のテスト見積りは、テスト対象ソフトウェアの複雑度と品質に大きく影響されるが、実際のところ、すべてのテスト作業は、ソフトウェアの複雑度と、ソフトウェアとドキュメント両方の受け入れた時点での品質にある程度影響される。以下は、テスト実装と実行に必要な時間と労力を決定するために使用できる技法のリストである：

- テスト見積りについてプロジェクトチームとブレーンストーミングを行う。見積りの共通点を発見し、外れた見積りについて議論し、共通の前提を使用していることを確認する。アジャイルプロジェクトでは、プランニングポーカーセッションを実施し、ユーザーストーリーの工数レベルを決定するために、ストーリーポイントの見積りをまとめる。
- 各テストケースを 1 回実行するのに必要な時間を見積り、その時間に各テストケースの反復回数の見積りを乗じる。
- 品質リスク分析を用いて、プロジェクトに必要なテストケース数を見積る。例えば、技術リスク／ビジネスリスクの高いアイテムに必要なテスト数は、リスクの低いアイテムに必要なテスト数の数倍になる可能性が高い。そこから、各テストケースの作成／保守に要する平均時間と、各テストケースの実行に要する時間を見積る。
- ファンクションポイント分析 (FPA)、またはコード行数をファンクションポイントに変換する計算(例: コード 50 行 = 1 ファンクションポイント)を用いて、プロジェクトの規模を決定する。ファンクションポイントに基づいて、必要なテストケースの数を見積り、それらのテストケースの作成と実行に必要な時間を決定する。
- 開発者の工数を用いて、テスト工数を決定する。これは通常、類似の組織における類似プロジェクトの履歴データを用いて行われる。
- テストポイント分析 (TPA) を実施し、プロジェクトに必要な総テスト時間を決定する。
- 同じ、または類似の組織やプロジェクトの過去の経験則を使用する。
- ワークブレイクダウンストラクチャー (WBS) やプロダクトブレイクダウンストラクチャー (PBS) などのプロジェクトマネジメント手法を適用し、プロジェクトのスコープを設定し、ハイレベルのスケジュールを決定する。

すべてのテスト見積りには以下を含めること：

- テスト文書(テスト計画書、テストケースなど)の作成/整備にかかる時間。

- テストデータ準備に要する時間
- テスト対象になるシステムの調達、設定、検証に要する時間
- テスト実行時間
- 結果のレビュー、再テスト、リグレッションテストを行う時間
- テスト実行情報の収集とレポートにかかる時間

その他のプロジェクト時間の考慮事項としては、ミーティング、トレーニング、管理間接費、計画的な休暇/休日、その他主要リソースのスケジュールや稼働に影響を与える可能性のある時間的制約が考えられる。また、チームが利用できるテスト時間数を明記することも重要であろう。プロジェクト報告書を作成する際、これらの見積りは、実際に費やされた時間と比較される。

見積りプロセスで考慮すべき追加要因としては、(他に誰がテストを行っているかに基づく)ソフトウェアの受け入れ品質の見積り、変更コントロールと構成管理プロセス、テストプロセス自体の成熟度、プロジェクトチームの成熟度、ソフトウェア開発ライフサイクルのタイプ、欠陥修正の信頼性、テスト環境のサポートリソース、ビジネスリソースのコスト、正確な文書と訓練された要員を利用できるかなどがある。加えて、テストシステムの信頼性、テストデータの信頼性、テストオラクルの可用性は、すべてテストチームが効率的に実行できることに寄与する。

テストマネージャーは、テスト実施に必要な時間に加えて、発見される欠陥数も見積る必要がある。これは、欠陥の調査、再テスト(確認テストとリグレッションテストの両方)、および欠陥の文書化作業に必要な時間を決定するために使用される。テストマネージャーは、問題の重要度や複雑度に基づいて、必要となるさまざまな時間の割り当てを決定することができる。

欠陥数を見積ることは、テストプロジェクトの進捗をモニタリングし、品質がよいのか悪いのか、あるいは期待通りなのかを評価するのにも役立つ。欠陥の見積りは、上記のテスト見積りと同様の技法を用いて行うことができる。作成するソフトウェアの規模(コード行数、FPA、TPA、開発工数)や複雑度の尺度を用いて、予想される欠陥数を決定することができる。例えば、テストマネージャーは、開発者の 2 時間のコーディングにつき、1 つの欠陥が発生すると予想するのが妥当であると判断するかもしれない。同様に、テストマネージャーの調査によれば、ファンクションポイントごとに 5 つの欠陥が予想される。

これらの数値は、プロジェクトチームの類似プロジェクトでの作業履歴、業界平均、または類似開発者や類似組織での経験から導き出しができる。一旦欠陥数が見積られたら、テストマネージャーは、平均的な欠陥に対してテストチームが調査、文書化、再テストのために必要とする時間単位を割り当てることができる。これにより、テスト見積りは、実際に発見された欠陥数と見積り欠陥数に柔軟に対応することができる。欠陥見積りがテスト見積りの中に埋もれてしまうと、ソフトウェアの品質が予想よりよかつたり悪かつたりした場合に、見積りを調整することが難しくなる。業界平均については、[Jones03]を参照のこと。

アジャイルプロジェクトでは、プランニングポーカーのようなブレーンストーミング手法が、イテレーション/スプリントの工数を見積るために頻繁に使用される。このようなテクニックを使うことで、小規模で比較的明確に定義された一連の機能に対する計画であるため、チームは不確実性やリスクに迅速に対処することができる。潜在的リスクは、短い時間間隔の中で実現されるか、破棄されるかのいずれかであり、チームはそれに応じて次のイテレーションを調整することができる。

どのように見積りを導き出したかに関わらず、テストマネージャーは、見積りの正確さを検証するために、メトリクスを注意深く追跡しなければならない。差異が大きい場合は、リソース、計画した機能、および/またはスケジュールの調整が求められる場合がある。

## 6.2.2 テストスケジュールの決定

見積りが確定すると、テストマネージャーがテストスケジュールを定義することができる。このスケジュールには、測定可能なマイルストーン、連携ポイント、依存関係、グループ間の作業成果物の受け渡しを含め、プロジェクト全体の計画の中で作業しなければならない。テストスケジュールは、合意された期待事項がすべて満たされた状態での円滑な引き継ぎに強く依存するため、各依存関係と作業成果物の受け渡しに関する客観的に検証可能な基準をリストにして定義すべきである。これによって、テストチームは、依存関係や受け渡しに対する要件を確実に満たせるとともに、また、指定された基準を満たさない受け渡しが及ぼす影響を評価することができる。

例えば、テストスケジュールが、開発者が行ったユニットテストが 100% のステートメントカバレッジを達成したという仮定に基づいている場合、その仮定を明確に記載し、測定方法を定義すべきである。コードが納品され、テストがカバレッジ 20% しか達成できなかつた場合、コードと一緒に納品される未検出の欠陥数が増えるため、テストスケジュールを調整してテスト時間を増やす必要がある。あるいは、テストマネージャーは、当初の見積りを維持し、開始基準を満たさないためにコードの納品を拒否することを決定するかもしれない。

テストスケジュールの定義と追跡には、通常プロジェクトマネジメントツールが使われる。これらのツールは、クリティカルパス項目の識別だけでなく、チャート作成機能を提供する。定義された各依存関係と受け渡しポイントは、明確に識別して後続のすべてのタスクが、直前の依存関係および/または受け渡しに依存していることを示すべきである。

スケジュールで定義する必要があるのは、プロジェクトチームメンバー間の成果物だけではない。テストを進める前に、他のステークホルダーの意見や参加が要件となる場合もある。例えば、機能テスト中に発見された使用性の問題に対処するために、最初の機能テストの後、かつ、非機能テストを開始する前に、使用性分析が必要となる場合がある。これは、別のグループとの依存関係であり、正式な受け渡しはないが、ユーザビリティチームへの成果物（機能が完全に揃っている、動作するソフトウェア）、およびユーザビリティチームからの成果物（欠陥レポート、機能強化の提案）が期待される。これらはすべて、プロジェクト計画で明確に定義し、スケジュールに記載すべきである。

環境によっては、テストスケジュールは常に変更される可能性がある。例えば、アジャイル環境では、スケジュールは、イテレーションで計画された内容に基づいているが、イテレーションでの作業中に、項目が追加されたり、または削除されたりして、スケジュールを調整する必要があるかもしれない。プロジェクト開始時に設定されたスケジュールが、プロジェクトの全期間を通じて変更されることはほとんどない。テストマネージャーは、スケジュールの柔軟性を現実的にとらえ、いつ変更調整が必要になるかを察知できるようにモニタリングしなければならない。

## 6.2.3 予算編成とリソース割り当て

予算やリソースの要件は、プロジェクトや組織によって大きく異なる。テストチームが、決められたリソースを備えたラボで運用するケースもある。この場合、予算とリソースの割り当ては、テストマネージャーの仕事の重要な部分ではないかもしれないが、予算とリソースの制限は、テストスケジュールに大きく影響するかもしれない。決められたリソースがあっても、予算はプロジェクトベースで行われることがある。つまり、テストマネージャーは、さまざまなプロジェクトの予算制限を守りつつ、複数のプロジェクトにリソースを割り当てなければならないことがある。

予算を扱う場合、テストマネージャーは以下のニーズを考慮すべきである：

- 給与、手当、定期昇給
- 外部スタッフまたは臨時スタッフの費用
- 休暇の割り当て
- テスト対象機器（テストされる機器自体）

- テストに使用する機器(サーバー、ネットワーク、プリンターなど)
- テストに使用するソフトウェア(オペレーティングシステム、インターフェースソフトウェア、データベースなど)
- ソフトウェアツール(データベースツール、レポートツール、分析ツール、欠陥追跡ツール、テスト実行自動化ツールなど)
- 設備
- 書籍、トレーニング、旅費交通費
- 長期的な効率性向上への投資(テスト自動化、ツール調達など)。

組織がどのように構成されているか、また、組織全体でリソースがどのように使用され得るかに応じて、他の考慮事項があるかもしれない。チームの構成と能力は、重要な要素である。すなわち、正社員、請負業者、オフショア、アウトソーシングされたチームメンバーの具体的な構成や、チームの各メンバーのスキルレベルである。テストマネージャーは、どのリソースを他の組織から活用でき、どのリソースを活用すべきでないかを認識しておく必要がある。例えば、テストチームが開発者と同じシステムを共有することは、テスト環境を不安定にする結果になりかねない。

テストマネージャーにとって重要なことは、予算策定は継続的なプロセスであることを忘れないことである。例えば、プロジェクトにリソースを追加したり、オンショアからオフショアにリソースを切り替えたり、予想以上に欠陥が発見されたり、プロジェクトのリスクを再評価したりすることは、すべて予算要件に影響を与える可能性がある。テストマネージャーは、予算支出を追跡するために、プロジェクトを注意深くモニタリングし、差異が見つかった場合には速やかに報告、解決できるようにしなければならない。

#### 6.2.4 プロジェクトのマネジメントと追跡

プロジェクトを効果的にマネジメント、追跡する鍵は、見積りを作成し、その見積りに対する順守状況をモニタリングすることである。予算とスケジュールは通常この見積りに基づいているため、見積りへの追跡は、差異を素早く発見し、見積りプロセスを改善するための賢明な方法である。例えば、テストプロジェクトが見積りよりも時間がかかっている場合、テストマネージャーはその原因を突き止めるべきである。テストケースの実行回数が予想より多いのか？欠陥は見積りより多いか？テストシステムの問題によって、テストが妨げられ、または、非効率になっていないか？テストマネージャーは、スケジュールと見積りを把握することで、差異を発見しやすくなり、その差異に迅速に対処することができる。

テストチームが差異の原因を解決できないケースもある。例えば、ソフトウェアの品質が悪く、予想以上に欠陥が多かった場合、テストマネージャーは、さまざまな引き継ぎをレビューして、どこで問題を検出すべきだったかを確認すべきである。例えば、ユニットテストが不十分だったか？計画した統合テストが実施されていなかったのか？この情報は、プロセスの改善点を提案するのに役立つだけでなく、プロジェクトチームが問題を理解し、問題を解決するために協力するのにも役立つ。

アジャイルなチームでは、差異は通常チーム内の状況に起因するものであり、多くの場合、チーム内で解決する。スクラムプロジェクトでは、すぐに解決できない問題(阻害要因:インピディメントと呼ばれる)に起因する予期せぬ差異を特定し、最終的に解決することが、スクラムマスターの主な仕事の 1 つである。スクラムは、特定のチームの役割と用語(例えば、スクラムマスターはプロジェクトマネージャーの役割に取って代わる)を特徴とするアジャイル開発方法論である。チームメンバーは、コアチームメンバーまたは補助的チームメンバーとして区分されることがある。さらなる情報は[P&C-Web]から入手できる。

プロジェクトをマネジメント、追跡するには、測定可能なメトリクスを備えた計画があり、プロジェクトのさまざまな時点で評価できることが必要となる。計画のないプロジェクトは、すぐに軌道から外れてしまうが、スケジュールのその時点でプロジェクトがどこにあるべきかを誰も知らなければ、問題は発見されないままになってしまう。テストマネージャーは、プロジェクトマネージャーと協力して、測定可能な品質ゴールを定義し、そのゴールに向

けた進捗を追跡するべきである。プロジェクトをマネジメントする上で、トレードオフの判断をしなければならないことがよくある。例えば、ビジネスが強く望んでいる機能の納期が遅れるのを受け入れる方がよいか？、それともプロジェクトの品質のリスクを受け入れるのか？プロジェクトチームが最善の判断を下せるように、リスクを定量化し説明できることが、テストマネージャーの仕事である。

よくマネジメントされたプロジェクトには、タスクに対する明確なオーナーシップがある。テストチームには、明確に割り当てられたタスクがあり、チーム内の各メンバーは、自分が何をいつまでに行うべきかを知っているべきである。例えば、あるテストチームメンバーに性能など特定の品質特性のテストが割り当てられた場合、そのチームメンバーは性能テストをいつ開始し、いつテスト完了し、そのテスト結果としてどのようなメトリクスを出すべきかを知っているなければならない。

プロジェクト追跡のための有用なメトリクスには、テストケース設計完了率、品質コスト、欠陥検出率、発見した欠陥 **vs** 予想した欠陥、テストケース実行結果などがある。根本原因、欠陥除去の有効性、機能完了、ユニットテストカバレッジなどの開発者メトリクスも、テストチームが利用できるようにすべきである。

プロジェクトのマネジメントと追跡については、[Ensworth01]を参照のこと。

### 6.2.5 トレードオフへの対処

完璧なプロジェクトは稀である。ほとんどのプロジェクトでは、品質、スケジュール、予算、機能のトレードオフが必要となる。これらのトレードオフは、プロジェクトの開始時点でわかっている場合もあれば、テストフェーズで明らかになる場合もある。テストマネージャーは、プロジェクトの構成要素間の相互依存関係を認識し、トレードオフの判断に参加する準備をすべきである。テストマネージャーは常に品質を守るべきであるが、品質がプロジェクトの最も重要な側面でない場合もある。テストマネージャーは、スケジュール、予算、機能セットのどのような変更が、よくも悪くもプロジェクト全体の品質に影響を与える可能性があるかを客観的に分析できなければならぬ。例えば、プロジェクトに予定外の機能を追加すると、テストリソースがより薄く分散されるため、全体的な品質が低下するかもしれないが、その新機能はプロジェクトの受け入れにとって重要かもしれない。

テストマネージャーは、プロジェクトのトレードオフ、特に納品される製品の全体的な品質を低下させる可能性のあるトレードオフに同意しないかもしれないが、全員によって変更の予想される影響を明確に理解できるように、ステークホルダーに情報を提供できなければならない。例えば、ある提案による変更の実質的な結果として、すべてのコンポーネントに対するテスト時間が 3 分の 1 に削減となる場合、その結果得られる品質はどのような意味を持つのか。テストマネージャーは、テスト時間を削減する影響によるプロダクト品質リスクの増加を明確に説明できるべきである。また、追加された機能の影響によって、どの領域のテストが減るのか、あるいはテストがされなくなるのかを明確に説明すべきである。

トレードオフの取り扱いについては、[Galen04]を参照されたい。

### 6.2.6 変更マネジメント

前述のように、プロジェクトスコープの変更は、プロジェクト全体のスケジュール、予算、機能、品質ゴールに影響を与える可能性がある。また、変更を追跡し、スケジューリングし、影響について評価するために、変更をマネジメントしなければならない。優れた変更マネジメントシステムは、よくマネジメントされたプロジェクトの要件である。特にテストマネージャーは、当初の見積りやスケジュールに影響するすべての変更を追跡する必要がある。このような変更は、機能の追加や削除に限定されないことを覚えておくことが重要である。プロジェクトのテスト面に影響するすべての変更を考慮しなければならない。例えば、開発チームが互換性テストに必要なサーバーを受け取れなかった場合、テストチームがその影響を受け、追加のテストを行うことになるかもしれない。

テストマネージャーは、プロジェクトのテスト側面に影響を与える可能性のある、プロジェクトに対するあらゆる変更に関して、用心深くなければならない。すべての変更に対して適切な影響度分析が行われるのであれば、テ

ストマネージャーはその分析に関与しなければならない。影響度分析がチームによってほとんど行われていないか、全く行われていない場合、テストマネージャーは、影響度分析プロセスを推進し、そのプロセスの順守を保証しなければならない。

アジャイルプロジェクトでは、変化は当然の前提として受け入れられ、むしろ歓迎されるものである。アジャイルプロジェクトに固有の絶え間ない変化に対処するために、テストマネージャーは、軽量で素早く適応性のあるソリューションを持たなければならない。確立されたアジャイルの実践例は、シンプルかつ効率的なソリューションを通して、柔軟な対応を可能にする。

### 6.2.7 タイムマネジメント

多くの場合、時間はリリースの品質レベルを決定する要因である。テストマネージャーは、テストチームの各メンバーが時間を効率的に使えるようにしなければならない。効率的な時間の使い方には、次のようなものがある：

- 電子メールなど、他のコミュニケーション手段でより効果的に情報を伝えられる場合には、会議に費やす時間を最小限に抑える。
- レビューに参加し、納品するソフトウェアに関する知識を構築する。
- テスト設計に十分な時間を確保し、要件に関する問題を特定し、解決する。
- テスト開始前に、すべてのテストシステムが正しく動作していることを確認する。
- 複数のテスト担当者にコードを配布する前に、送られてくるリリースを効果的にスマートテストし、開始基準が満たされていることを検証する。
- リグレッションテスト戦略を含む効果的なテスト戦略を順守する。
- テスト実行自動化を反復的なテストに使用する。
- 将来のためのチームへの投資として、トレーニングの時間をスケジュールに組み込む。
- 適切な構成管理プロセスが守られていることを確認する。
- 特に地理的な境界や時差を超えて、チーム全体に効果的なコミュニケーションを提供する。
- アジャイルプロセスで、イテレーションまたはスプリントに設定されたタイムボックスに従う（テストされていない、またはテストが不十分な項目は、後のイテレーションに移動される）。

テストチームは、大半の時間をテストに費やし、テスト活動を効果的に進めるべきである。これを確実にするのがテストマネージャーの仕事である。

## 6.3 プロジェクトリスクマネジメント

プロジェクトリスクは、どんなプロジェクトにも現実に存在する。リスクの可能性や影響は、プロジェクトや組織によって異なる。テストマネージャーは、これらのリスクを特定し、各リスクの軽減計画を作成できなければならない。ソフトウェアプロジェクトのリスクについては、[Jones93]や[DeMarco03]を参照されたい。

### 6.3.1 プロジェクトリスクのマネジメント

プロジェクトリスクマネジメントの目的は、プロジェクトを危険にさらす可能性のある、あるいは危険をもたらし得るリスクを追跡し、コントロールすることである。すべてのリスクを排除する必要はないし、多くのケースでは不可能ですらある。プロジェクトマネージャーは、潜在的なリスクを認識しておかなければならない。そうすることで、リスクが顕在化し、プロジェクトの脅威となつた場合に、適切なアクションを取ることができる。テストマネージャーとプロジェクトマネージャーは、テスト活動をプロジェクト内の他のリスク対策と整合させなければならない。

プロジェクトを通して、テストチームおよび/またはテストマネージャーが、テスト関連のプロジェクトリスクを識別することがある。もちろん、最も重要なプロジェクトリスクは、計画活動中に識別されるのが理想的である。品質リスクと同様に、識別したプロジェクトリスクは、そのリスクの可能性と影響、そして重要性のレベルを決定するために分析すべきである。また、トリガー日（コンフィデンスインターバルと呼ばれることがある）を決定することも重

要である。トリガー日とは、リスクの影響を低減するために効果的なコンティンジェンシーを実施しなければならない期日である。

### 6.3.2 プロジェクト全体のリスクマネジメントへの参加

リスクは、プロジェクトのテストフェーズに限定されるものではなく、ソフトウェア開発ライフサイクルの中で、いつでも発生する可能性がある。テストマネージャーは、以下の領域で発生し得るリスクの可能性と影響を考慮しなければならない。

- 要件プロセス中のリスク - 潜在顧客への不十分なアクセシビリティ、不十分な要件文書、要件収集のための不十分な時間、要件が完成する前にコーディングが始まる、など。
- 設計プロセスにおけるリスク - 不十分な要件、不適切な設計の選択、誤った、あるいは望ましくない実装の選択(例:誤ったデータベース、誤った言語)、性能、保守性などの設計上の考慮事項の見落としなどである。
- 実装、コーディング、ユニットテスト、コードレビュープロセスにおけるリスク - ユニットテスト不足、不適切なコーディング実践例、不適切なレビュープロセス、コードの文書化不足、静的解析の欠如、スコープクリープなど。
- 統合プロセスにおけるリスク - 統合が系統的に行われていない、統合プラットフォームがテスト環境を代表していない、統合がエンドツーエンドではなく機能間でしか行われていない、テスト時に現実的なデータや現実的なシナリオがない、など。

テストマネージャーは、これらのリスクをコントロールすることはできないかもしれないが、他の分野のリスクが顕在化したときに、テストマネージャーはそれを認識し、そのリスクがテスト組織に与える影響を評価できるべきである。予測されるリスクに対処するためのコントロール活動を整備すべきであり、メトリクスやその他のモニタリング技術によって早期発見を可能にするべきである。テストマネージャーは、特定されたリスクに対して積極的なアプローチを取り、プロジェクトチーム内でリスクを軽減するために十分な努力が払われていることを確認するべきである。また、テストマネージャーは、現実に発生したリスクに効果的に対処できるように、テストチームを配置し、準備し、これらの発生に基づいて、見積りやスケジュールを迅速に調整できるようにすべきである。テストマネージャーは、プロジェクトマネージャーと協力して、誰がこのようなプロジェクト全体のリスクをマネジメントすべきかを決定すべきである。

## 6.4 品質マネジメント

エキスパートテストマネージャーは、ソフトウェアテストが、顧客やユーザーが使用するのに適した品質のソフトウェア製品やサービスを、求められるフィーチャーを備えた上で、期限通り、予算内で提供するために必要な、より大きな一連の品質マネジメント活動の一部であることを認識すべきである。品質マネジメントは、分析成果物(要件文書など)、開発成果物(ユニットテスト文書、コードレビュー情報など)、テスト成果物(テストケース、欠陥レポート)など、開発ライフサイクル中に作成され使用されるさまざまな成果物にも及ぶ。より大きな一連の品質マネジメント活動の知識の体系化と実務への適用は、ソフトウェア品質保証(SQA)または単に品質保証(QA)と呼ばれることもある。

真の品質マネジメントには、ソフトウェア品質保証(SQA)の活動を、ソフトウェア開発の各プロセス、およびライフサイクル全体に統合することが求められる。これには、レビュー、妥当性確認、プロセスの標準化、ソフトウェアプロセスの成熟性といった重要な実践例が含まれる。品質マネジメントシステムを組織で成功裏に実施するためには、人的な課題やその他の障壁を取り除く必要がある。また、品質マネジメントの矛盾を探し、それを解消する方法を模索する。これらの課題は、テストマネジメントの範囲を超えており、密接に関連している。

テストチームと品質保証チームの両方を持つ組織もあれば、テストチームに両方の役割を任せる組織もある。

テストチームが両方の役割を果たしている場合、テストチームに対してどのような品質保証および品質マネジメントの活動と成果を期待するのかを、組織のマネジメント層と明確にしておくことが不可欠である。テストは品質関連活動の一部であるとした上で、テストポリシーとテスト戦略は、包括的な品質ポリシーと品質戦略へと拡張すべきである。

テストチームが SQA チームを兼ねている場合でも、独立した SQA チームが存在する場合でも、品質マネジメントと品質保証に関する一連のプロセス、活動、メトリクスは統合され一貫性のある形で存在すべきである。存在しない場合、テストマネージャーは、SQA マネージャー(その役割が存在する場合)および組織マネジメントと協力して、それを作成する必要がある。SQA プロセス、活動、メトリクスが存在する場合、テストマネージャーはそれらを評価し、矛盾を識別し、SQA マネージャーと協働してそれらの矛盾を解決すべきである。効果的な品質マネジメントには、マネジメントの賛同が必要である。

品質マネジメントと業界におけるソフトウェア品質の現状については、[Jones08]と[Kan02]を参照されたい。

## 7. テストプロジェクトの評価とレポート 510 分

### キーワード

収束メトリクス、ダッシュボード、石川（フィッシュボーン）ダイアグラム、シーアハートチャート

### テストプロジェクトの評価とレポートの学習の目的

#### 7.1 イントロダクション

本節の学習の目的はない。

#### 7.2 追跡する情報

LO 7.2.1 (K4) あるテストレベルについて、プロジェクトをコントロールするために使用する最も効果的なメトリクスを決定する。

#### 7.3 情報の評価と活用 - 内部報告

LO 7.3.1 (K6) 指定されたプロジェクトについて、社内用に効果的なテスト結果レポート一式を作成する。

#### 7.4 情報の共有 - 外部報告

LO 7.4.1 (K6) 正確で理解しやすく、レポートの読み手に適したテスト状況ダッシュボードを作成する。

LO 7.4.2 (K4) 特定のプロジェクトとプロジェクトチームを想定し、週次テストレポート作業の適切なレベルと媒体を決定する。

#### 7.5 テスト結果のレポート作業と解釈

LO 7.5.1 (K6) 特定のライフサイクル、プロジェクト、製品を想定し、さまざまなステークホルダーの情報ニーズと習熟度を考慮に入れて、メトリクスを含む適切なテスト結果レポートプロセスおよびダッシュボードを設計する。

LO 7.5.2 (K4) 特定のテストダッシュボードと一連のメトリクスに基づいて、テストプロセスのどの部分においても、適切な定性的な解釈を含め、特定のテスト状況を解析する。

#### 7.6 統計的品質コントロール技法

LO 7.6.1 (K2) テストマネージャーが基本的な品質コントロール技法を理解する必要がある理由を説明する。

### 7.1 イントロダクション

テストマネージャーは、進行中のプロジェクトを客観的に追跡し、データを評価し、必要な是正措置を取ることができなければならない。また、社内外のステークホルダーに適切な情報を報告できなければならない。テスト評価とレポート作業の詳細については、[Black03]を参照のこと。

### 7.2 追跡する情報

プロジェクトをコントロールするためには、どのようなデータを追跡すれば、プロジェクトの進捗状況や品質を指標化できるかを知ることが重要である。追跡する情報は、有用であるために、客観的で、有益で、理解しやすいものでなければならない。また、対象者から提起される可能性のある具体的なゴールや疑問に対応したもので

なければならない。必要な情報を収集し、報告するために、ツールを活用すべきである。報告プロセスを自動化できるツールは、この取り組みにおいて価値がある。

プロジェクトで追跡すべきデータには、以下のようなものがある:

- 設計されたテストケース(計画対実績など)
- 欠陥メトリクス(例:ステータス別、重篤度/優先度別、領域別、傾向、収束、解決ターンアラウンド、再オープン率、拒否率)
- テスト実行メトリクス(例:ステータス別、自動実行対手動実行、実行率、収束メトリクス、計画対実績、テスト合格数)
- カバレッジメトリクス(コード、要件、リスクなど)
- マネジメントメトリクス(例:リソース時間、セットアップ時間、発生したダウンタイム、計画対実績)

傾向分析は、追跡された情報の重要な活用方法である。「ある時点の情報」はスナップショット的な視点を提供するが、スケジュールやリソース、品質に関する問題を示す全体的な傾向を見落とす可能性がある。傾向を導き出すためには、詳細情報を追跡しなければならない。

データを追跡することの一部には、報告されるデータが正しいことを確認するためにデータを「クリーニング」することも含まれる。これは、情報収集に使用されるツールの使用ガイドラインにも通じる。もしテストケースの完了状況について不正確なデータを報告する人がいれば、サマリーレポートもまた不正確となり、プロジェクトのステータスについて誤った、あるいは不完全なイメージを提示する結果になりかねない。

### 7.3 情報の評価と活用 - 内部報告

収集した情報を有用なものにするためには、加工しなければならない。これには、データの妥当性を確認し、実際の値と期待値を比較するために、情報を慎重に評価することが含まれる。情報がプロジェクトの期待値と一致しない場合、その差異を理解し、是正措置を講じるために調査が必要となるだろう。例えば、期待された欠陥発見率が週に 100 件であったのに、実際の欠陥発見率が週に 50 件であった場合、これはソフトウェアの品質が期待よりもかなりよいことを意味するか、テスト担当者が病欠していることを意味するか、テスト環境がダウンしているためにテストができなかったことを意味する。

情報の評価には専門知識が必要となる。評価者は収集された情報を理解するだけでなく、それを効果的に抽出し、解釈し、差異を調査できなければならない。生のデータを報告することは危険であり、データに欠陥があることが後に判明した場合、信用を失うことになりかねない。

データが評価され、正確性が確認されると、内部報告および行動計画に使用することができる。内部レポートはプロジェクトチーム内で回覧される。例えば、マネジメントメトリクスは、テスト担当者の効率性を向上させるための是正措置が実施される間、テストチーム内部にとどめておく場合がある。詳細な欠陥メトリクスは、通常開発チームと共有されるが、要約されるまではプロジェクトマネージャーや外部チームメンバーには回覧されないかもしれない。

内部レポート作業は、テストマネージャーがテストプロジェクトをマネジメントしコントロールするために使用する。テストマネージャーが役に立つと思う内部レポートのサンプルには、以下のようなものがある。

- 日次のテスト設計/実行ステータス
- 日次の欠陥ステータス
- 週次の欠陥傾向
- 毎日のテスト環境レポートとステータス
- リソースの空き状態

内部報告書は、外部報告書よりも形式ばらないかもしれないが、データの品質は、特にこれらの報告書の情報に基づいてマネジメントの判定が行われる場合には、同様に重要である。優れたレポートティングツールと手法の開発に費やす時間は、プロジェクト全体の時間節約につながる。

## 7.4 情報の共有 - 外部への報告

外部レポート作業の目的は、レポートの読み手に理解できる言葉で、プロジェクトのテストステータスを正確に描写することである。テストステータスが、プロジェクトの他の側面から報告されるステータスと食い違うことは珍しくない。これは、他の報告が必ずしも正しくないという意味ではなく、異なる視点から報告しているという意味である。テストマネージャーが、これが問題領域であると認識した場合、レポート対象者を混乱させるリスクよりも、他のレポートマネージャーと協力して、データを統一することを実践するのが望ましい。

外部レポート作業における最大の課題の 1 つは、情報の提示方法をレポートの読み手に合わせることである。例えば、開発チームにとっては詳細な欠陥レポートが効果的かもしれないが、経営層にとっては、高レベルのスナップショットを提示するダッシュボードの方がより有益であると感じるだろう。一般的に、組織の上位にいる人ほど、詳細なレポートよりも一般的なトレンドダイアグラムの方が適切かもしれない。図表や視覚的に魅力的なグラフは、読み手の技術的専門性が低下するにつれて、より重要になる。経営層は、レポートのレビューに割く時間がほとんどないかもしれないため、簡潔なビジュアル、解釈しやすいチャート、傾向図は、ポイントを迅速かつ正確に伝える上で特に有用である。

外部レポートを作成する際のもう 1 つの考慮点は、提供する情報の量である。例えば、レポートが会議で発表される場合、発表者は必要に応じて追加の詳細情報を提供する。一方、レポートがインターネット上でダッシュボードとして公開される場合、読者が興味を持ったときに詳細を表示できるドリルダウン機能を提供すると有用であろう。ハイレベルな情報のみが提供され、それを裏付ける詳細な情報がない場合、信頼性が失われる可能性がある。

外部レポート作業は、プロジェクトの重大さ、製品ドメイン、レポートの読み手の専門性によって異なる。よく使われる外部レポートの例としては、以下のようなものがある：

- 現在のテストと欠陥のステータスを表示するダッシュボード
- リソースと予算の報告
- リスク軽減チャート
- テストカバレッジチャート(要件ベースまたはリスクベース)
- 欠陥の傾向
- マイルストーンの達成状況
- プロジェクトの健全性指標(赤、黄、緑)
- 予測メトリクス
- 「Go／No Go(実行可否)」に関する推奨

外部レポートは定期的に、通常は毎週行うべきである。こうすることで、情報を受け取ることを期待し、その情報を自らの計画や判定に利用するよう、レポートの読み手を育成することができる。この情報を効果的に公表するには、以下のようにさまざまな方法がある：

- リアルタイムダッシュボード
- イントラネット上の既知のサイトに公開する。
- 配布リストに電子メールで送信する
- 定期的なプロジェクトステータスマーティングに含める
- オフィスの窓に物理的に掲示する

外部レポート作業は、それが対象の読み手に受け取られ、理解されて初めて効果的なものとなる。レポートは、特に読み手に合わせて、適切なレベルの詳細を含めなければならない。レポートの読み手が理解できない、または関心のない詳細情報を過剰に盛り込みすぎると、レポート全体を無視してしまうことになりかねない。

テストマネージャーとしての成功は、テストの状況をテスト組織の外部に正確に伝える能力と密接に関係している。テストマネージャーは、レポートの読み手とメッセージを考慮し、情報を適切に調整しなければならない。

## 7.5 テスト結果のレポート作業と解釈

Advanced シラバスでは、ソフトウェア開発ライフサイクルのさまざまなフェーズ、特にテストフェーズの間に追跡すべき、多くの異なるタイプのメトリクスについて議論している。この節では、収集された情報をどのように効果的に報告するか、またそれらの結果をどのように解釈するかに焦点を当てる。これは、内部使用だけでなく、その情報の他の利用者にとっても対象となる。自動テストと手動テストは、この議論の中でまとめて扱われる。

この節では、基本テストプロセスの各フェーズに注目し、追跡すべきメトリクスを簡単に確認し、それらのメトリクスをどのように解釈すべきかを議論する。

### 7.5.1 計画、モニタリング、コントロール

Advanced シラバスによると、テスト計画とは、テスト戦略で識別したミッションと目的を満たすために必要なすべての活動とリソースを識別し、実装することである。テストモニタリングとコントロールは、継続的な活動である。これらには、実際の進捗を計画と比較すること、ステータスを報告すること、そして必要に応じて情報に対応し、テストを変更し導くことを含む。

- テスト計画、モニタリング、コントロールプロセスのためのメトリクス
  - リスクカバレッジ
  - テストケースの開発と実行状況
  - 欠陥発見の傾向
  - テスト活動における計画時間対実績時間
- データの解釈
  - すべてのテストプロジェクトにおいて、見積りスケジュールと予算を守ることは非常に重要である。これらのメトリクスは、テストプロセスの計画対実際のステータスを迅速に判断するために使用できる。
  - ソフトウェアの品質は、欠陥発見の予想と実績の傾向に基づいて判断できる。欠陥の発見が予想より多かったり、重大な欠陥の割合が予想より多かったりする場合は、テスト実行フェーズに入った時点での品質に重大な問題が生じている可能性がある。
  - 適切な順序でリスク軽減が行われているかどうかは、チームが最もリスクの高いアイテムのテストを、よりリスクの低いアイテムよりも先に作成し実行しているかどうかをチェックすることで判断できる。低リスクアイテムが先に対処されている場合、テストアプローチをレビューし、修正する必要がある。
  - テストの進捗が不十分である原因は、実質的なテスト作業時間の不足による場合がある。計画対実際のテスト工数が期待に達していない場合、テストカバレッジは不十分となり、プロジェクトはすぐにテストスケジュールから遅れ始める。テストマネージャーは、進捗不足の原因を注意深く調査し、どのような要因（テストシステムの準備が整っていない、テストプロセスに入ってきた時点での品質が不十分、テスト文書が準備されていない、リソースが利用できない状態、など）が発生しているかを判断し、これらの問題を是正するための計画を作成しなければならない。

### 7.5.2 分析と設計

テスト計画段階を通じて、一連のテスト目的が定義される。テスト分析フェーズとテスト設計フェーズにおいて、テストチームはこれらのテスト目的からテスト条件を識別し、識別したテスト条件を確認するテストケースを作成する。

- 分析と設計のためのメトリクス
  - カバレッジメトリクス
  - テスト設計中に発見された欠陥
  - レビューサイクルで発見された欠陥
- データの解釈
  - カバレッジメトリクスには、テストケースによる要件のカバレッジ、テストケースによるリスクアイテムのカバレッジ、品質特性のカバレッジ、その他測定可能なカバレッジタイプが含まれる。カバレッジがいずれかの領域で不十分な場合、未対処のリスクが存在する。そのリスクは、リスクの影響とリスクの可能性に基づいて、それに対処する最善の方法を決定するために分析しければならない。
  - レビューサイクル中やテスト設計中に発見された欠陥は、解決されれば、その欠陥が次のフェーズに逃されるのを防ぐことができる。欠陥が発見されたが解決されなかつた場合、その欠陥の存在に起因する追加作業(文書化、ステータスレポート、可能な回避策など)が発生するが、既知の欠陥に対処するために費やす時間は、通常、以前に発見されなかつた欠陥を発見するために費やす時間よりも短い。コスト(金銭または時間)は、これらの欠陥にしばしば関連付けられ、そのコストは、通常、一般的なテストレポート作業の一部として、コスト削減の観点から報告される。
  - 欠陥の傾向もこの段階で重要である。レビューサイクルで欠陥が多数(計画よりも多く)発見された場合、これは作業成果物の品質が予想よりも低いことを示している可能性がある。その場合、許容できる品質に到達するために、作業成果物がライフサイクルの次のフェーズに移行する前に複数のレビューサイクルが必要となる可能性がある。

### 7.5.3 実装と実行

テストの実装と実行には、テストケースの整理、テストデータとテスト環境の確定、実行スケジュールの作成、実際のテストの実行などが含まれる。

- 実装と実行のためのメトリクス
  - 環境、システム、およびデータが構成され、テスト準備が整っている割合
  - テストデータの作成率と使用率
  - テスト実行によってカバーされたテスト条件とケースの割合とその実行結果
  - 自動化されたテストケースの割合と自動化の実行結果
  - リグレーションテストスイートに含めるようマークされたテストケースの割合
  - 自動化テストケースと手動テストケースの保守に費やす時間
  - テストのタイプ(例: ブラックボックス、ホワイトボックス、経験ベースのテスト)の内訳
- データの解釈
  - 100%のテストシステムとテスト環境が準備できていない状態でテストを開始することは、受け入れられるかもしれないし、望ましいテスト戦略であるかもしれない。しかし、不足している環境が最も高いリスクアイテムに必要なものである場合、よりリスクの高いテストをテストサイクルの最後に回すことを強制することになり、一般的には受け入れがたい。テストマネージャーは、パーセンテージの数字だけを見るのではなく、スケジュールへの影響や、構成上の問題に対応するためにテスト計画書全体に変更が必要かどうかを理解しなければならない。
  - テスト環境の問題と同様に、テストデータの可用性と正確性もテスト実行の順番に影響し、テストチームの効率を著しく低下させる可能性がある。例えば、必要な量のテストデータが入手できず、再利用を繰り返さざるを得ない場合、一部の領域を同時にテストすることができないかもしれない

い。自動テストでは、データを再利用できるように変更し、データ使用後にテスト前の安定した状態に戻す必要があるかもしれない。

- テストカバレッジを注意深くモニタリングしなければならない理由はいくつかある。カバレッジが計画通りに進まないと、計画外のところでリスクが顕在化する可能性がある。例えば、ロックされているテスト領域には、システムのアーキテクチャー変更につながる重大な欠陥が含まれている可能性があり、その結果、性能テストが遅延する。テストマネージャーはまた、合格／失敗情報をモニタリングして、期待される合格／失敗率が達成されているかどうかを判断しなければならない。予想よりも多くのテストケースが失敗している場合、テストサイクルの追加が求められる可能性がある。さらに、品質レベルが予想より低い理由についても調査が必要である。テストケースの合格数が予想より多い場合、テストマネージャーは、テストが十分であったかどうか、スケジュールの余分な時間をどこに集中させるべきか、将来のテストのために計画を調整すべきかどうかを判断する必要がある。
- テスト自動化の達成率と自動化の結果を、テストマネージャーはモニタリングし、自動化の取り組みがゴールを達成していること、テスト自動化の結果が全体のテスト結果に統合されていることを確認する必要がある。テスト自動化がテストの5%しかカバーしていないが、欠陥の50%を発見している場合、発見した欠陥のタイプを調査し、自動化による検証が正しく妥当なのか確認し、手動テストで問題がほとんど発見されない理由を調査する必要がある。
- テスト文書のメンテナンスに費やす時間は、プロジェクトにおいて重要な要素となる。テストマネージャーは、テストケースの作成に使われた要件が、テストケースの作成後に変更された場合、メンテナンス時間がスケジュールを超えて増えるという事態に直面することがある。また、納品されたコードが要件と一致しない場合、環境やデータが変更された場合、インターフェースシステムが変更された場合、その他無数の理由でメンテナンス時間が増加することもある。テストケースのメンテナンスに費やす時間は、実際のテストに費やされていない時間であるため、このメンテナンス作業はテストプロジェクト全体に大きな影響を与える可能性がある。
- テストタイプの内訳は、いくつかの観点から興味深い。テストマネージャーは、テスト時間がどのように配分されているかを知る必要がある。また、テストマネージャーは、欠陥検出、リスクカバレッジ、テスト担当者スキルの使用に関して、さまざまなテストタイプがどの程度生産的であるかを知る必要がある。もしテストマネージャーが、欠陥の60%が探索的テストで発見されていると判断した場合、テスト配分を変更し、より探索的テストに重点を置く必要があるかもしれない。また、テストマネージャーは、スクリプトテストで発見されるバグが予想よりも少ない理由を理解したいかもしれない。もしかすると、ソフトウェアがその領域で安定し、スクリプトテストの生産性が以前より低下しているのかもしれない。ホワイトボックステストで欠陥が発見されない場合、テストマネージャーは、そのテストが達成しているコードカバレッジレベルを理解したいと思うだろう。

#### 7.5.4 終了基準と報告の評価

テストマネージャーは、テストプロジェクト全体を通じて終了基準を評価し、着実に終了基準を満たすよう進捗していることを確認する。終了プロセスのレポート作業は、上記の項で説明したテスト活動とメトリクスの要約であるが、プロジェクト全体の視点で行う。テストプロジェクトのこの時点で、新しいメトリクスを導入すべきではない。ここで使用されるメトリクスは、上述したように、プロジェクト全体を通して計画され、追跡されたものでなければならない。

- 現段階で確定しているメトリクス
  - テスト条件、テストケースまたはテスト仕様書の実行対計画、およびそれぞれの最終ステータス
  - 実施したテストサイクル数対計画
  - 導入、実装、テストされたソフトウェアの変更の数と複雑度
  - スケジュールと予算の順守と差異
  - リスク軽減
  - 実テスト時間の計画対実績

- データの解釈
  - この時点では、これらのメトリクスを使って現行プロジェクトに変更を加えることはできない（テスト時間を持たせたり再作業を計画している場合を除く）。この時点のメトリクスは、将来のプロジェクトのプロセス改善に使用される。
  - この段階で収集されたメトリクスは、テスト作業の終了時に作成されるサマリーレポートで報告される。

### 7.5.5 テスト終了作業

テスト実行が完了しても、テストマネージャーはテスト終了作業を実施しなければならない。これらは、すべてのテスト作業が実際に完了したことを確認し、最終作業成果物を納品し、ふりかえりミーティングに参加し、プロジェクトのデータ、テストケース、システム構成などをアーカイブすることからなる。

- テスト終了作業の一環として完了したメトリクス
  - 将来のプロジェクトで再利用可能なテストケースの割合
  - リグレッションテストスイートに移行すべきテストケースの割合
  - ライフサイクルが終了した欠陥の割合（修正されない、処置不要、強化要求など）
- メトリクスの解釈
  - 前ステージと同様、これらのメトリクスは、テストプロセス全体を通して追跡されるべきであったが、レビューするよい機会である。再利用可能なテストケースは、一度使用され、もはや使用できないテストケースよりも、一般的によりよい時間の投資と見なされる。アジャイルライフサイクルに従うような変化の激しい環境では、初期のイテレーションで開発されたテストケースは、後のイテレーションまでに陳腐化することが多い。このような「使い捨てる」テストケースに投資した時間は、有効に使われたかどうか、あるいは別のテスト設計手法の方が効率的であったかどうか（例えば、スクリプトテストケースではなくチェックリストを使うなど）、レビューするべきである。
  - テスト対象の機能が変わらないことが予想されるため、リグレッションテストスイートに移されたテストケースは、自動化チームが自動化の可能性をレビューするのに適した候補となる。
  - 解決されない欠陥は、（いつものように）レビューして、テストチームがこれらの項目の特定と文書化においてよい実践例に従っているかどうかを判断すべきである。欠陥が再現不可能であるため「never fix」とマークされている場合、テストマネージャーは、これらの欠陥レポートにおいて、追加のテストが必要だったか、より適切な文書化が求められていたかを調査すべきである。

テスト終了作業では、追加でテストプロセス改善が必要な領域を識別する場合があり、また、以降のプロジェクトにてより詳細に追跡する必要がある領域を識別する場合もある。

## 7.6 統計的品質管理手法

テストマネージャーは、品質管理の役割を担う立場として、管理図や原因/結果グラフなどの基本的な統計的品質管理手法を適用できるべきである。例えば、以下のようなものがある：

- 石川（フィッシュボーン）ダイアグラムを使用して、自動テストの信頼性問題の原因を特定する。
- シューハートチャートを使用して、テストプロセスがコントロールされているかどうかを判断する。

統計的品質管理手法に関する追加情報は、他の品質保証向けの文書に見つけられる。統計的品質管理の詳細については、[Ishikawa91] および [Juran10] を参照のこと。

## 8. ドメインとプロジェクト要因に関するテスト上の考慮事項 270 分

### キーワード

フィーチャー駆動開発、テスト駆動開発

### ドメインとプロジェクト要因に関するテスト上の考慮事項の学習の目的

#### 8.1 イントロダクション

本節の学習の目的はない。

#### 8.2 ライフサイクルモデルのテストマネジメントに関する考慮事項

- LO 8.2.1 (K6) 特定の状況において、各ライフサイクルモデルの利点と欠点を評価し、報告する。  
LO 8.2.2 (K2) アジャイルプロジェクトにおいて通常見られる、テストアプローチに影響を与える可能性のあるコンセプトを説明する。

#### 8.3 部分的なライフサイクルプロジェクトのマネジメント

- LO 8.3.1 (K6) 開発ライフサイクルのさまざまな部分を対象にしたプロジェクトのタイプを比較し、それらのプロジェクトと純粋にすべての開発ライフサイクルを対象にしたプロジェクトとの違いについて議論する。

#### 8.4 リリースに関する考慮事項

- LO 8.4.1 (K4) デプロイメント、インストール、リリースマネジメント、および/または製品ロードマップに関するビジネスコンテキストを分析し、テストへの影響を判断する。

## 8.1 イントロダクション

テストマネージャーは、異なるプロジェクトライフサイクルモデルや、異なる統合レベルのプロジェクトを経験する。テストマネージャーにとって、これらの違いを認識することは、適切な関与のタイミング、最良のテストアプローチ、テストチームのトレーニングニーズを決定するのに役立つ。

## 8.2 ライフサイクルモデルのテストマネジメントに関する考慮事項

ライフサイクルモデルは、あるプロジェクトに対するテストマネージャーのアプローチに強く影響する。ライフサイクルによって、関与のタイミング、関与の程度、ドキュメンテーションの量(利用可能なものと作成されたものの両方)、およびテストのゴールが決まる。

### 8.2.1 ライフサイクルモデルの比較

次の表は、ライフサイクルモデルの比較と、それに伴うテストへの影響を示している:

	関与のタイミング	提供される文書	作成する文書	関与の程度	テストのゴール
ウォーター フォール	コード完成後	完全な仕様	完全なテスト文書	コード完成まで なし	システムテ スト、一部受 け入れテス ト
V字モデル	要件定義段階 でのレビュー、 コード開発段 階での計画、 コード完成時 のテスト	完全な仕様	完全なテスト文書	早期の計画、 要件文書のレ ビュー、コード 完成時のテスト	統合テス ト、シス テムテ スト、受け 入れテス ト
イテレーティ ブ/インクリ メンタル	テスト可能なコ ードの最初の イテレーション に関与する。	イテレーションごと の仕様文書	イテレーションテス ト計画書、イテレー ション関連のテス トケース、若干の自 動化	リグレッショ ンテストに重点を 置いた、イテレ ーションごとの 「ジャストインタ イム」計画とテ スト	統合テス ト、シス テムテ スト、受け 入れテス ト
アジャイル	プロジェクト開 始	あつたとしてもごく わずか	テスト自動化、テス トチェックリスト	プロジェク ト開始時からプロ ジェクトチーム に組み込まれ る	ユニットテ ストを含むす べてのレベ ルのテス ト

テストチームが関与するタイミングと度合いは、プロジェクトのライフサイクルによってかなり異なる。ライフサイクルは開発チームによって決断されることが多いので、テストマネージャーはこれらの決定をモニタリングし、適切なタイミングと関与の程度を決断しなければならない。

テストマネージャーは、Advanced レベルのシラバスすでに述べたように、最も一般的なライフサイクルモデルに精通していなければならない。それに加えて、エキスパートテストマネージャーは、アジャイルアプローチとアジャイルプロジェクトに効果的に参加する方法にも精通していなければならない。

実際の運用では、多くの組織が複数のライフサイクルモデルの要素を組み込んだライフサイクルを実施している。このアプローチについては、[Boehm03]を参照のこと。

### 8.2.2 アジャイル手法

アジャイル手法は、V字モデルのような伝統的なモデルとは根本的に異なる、ソフトウェア開発ライフサイクルに対するアプローチのグループである。アジャイルアプローチの中で最も一般的なものは次の通りである：

- スクラム - イテレーティブなプロジェクトのリリースにおいて、役割やプロジェクトマネジメント上の課題に焦点を当てる。
- エクストリームプログラミング (XP) - ソフトウェア開発チームが使用する技術的実践例に焦点を当てる。
- カンバン - 自己組織化チームを使い、進行中の作業を積極的に制限することによって、価値の流れを増大させることに重点を置く。

これらのアプローチは並行して開発されたが、相補的な側面を活用するために、プロジェクトではしばしば組み合わされる。

アジャイル手法に関連する特定の実践例や価値観のセットがいくつかある。フィーチャー駆動開発 (**feature-driven development**) のような開発実践例や、テスト駆動開発 (**test-driven development**) のようなテスト実践例は、アジャイルプロジェクトでよくみられる。テストマネージャーはアジャイル手法に適応する必要があるが、他のモデルに共通する多くの優れたテスト実践は引き続き有効であることを覚えておくことが重要である。

アジャイルに精通するためには、アジャイル方法論の 12 の原則を定義しているアジャイルマニフェスト [Agile Manifesto-Web] に精通することが役に立つ。これらは以下の通りである：

1. 顧客満足を最優先し、価値のあるソフトウェアを早く継続的に提供します。
2. 要求の変更はたとえ開発の後期であっても歓迎します。変化を味方に付けることによって、お客様の競争力を引き上げます。
3. 動くソフトウェアを、2-3 週間から 2-3 ヶ月というできるだけ短い時間間隔でリリースします。
4. ビジネス側の人と開発者は、プロジェクトを通して日々一緒に働くなければなりません。
5. 意欲に満ちた人々を集めてプロジェクトを構成します。環境と支援を与え仕事が無事終わるまで彼らを信頼します。
6. 情報を伝える最も効率的で効果的な方法はフェイス・トゥ・フェイスで話することです。
7. 動くソフトウェアこそが進歩の最も重要な尺度です。
8. アジャイル・プロセスは持続可能な開発を促進します。一定のペースを継続的に維持できるようにしなければなりません。
9. 技術的卓越性と優れた設計に対する不断の注意が機敏さを高めます。
10. シンプルさ(ムダなく作れる量を最大限にすること)が本質です。
11. 最良のアーキテクチャー・要求・設計は、自己組織的なチームから生み出されます。
12. チームがもっと効率を高めることができるかを定期的にふりかえり、それに基づいて自分たちのやり方を最適に調整します。

アジャイル手法は、ライフサイクル全体にわたり品質に強く注力することによって成り立っている。熟練したテスト担当者やテストマネージャーはアジャイルプロジェクトの成功に不可欠だが、その役割は多少変化する。例えば、テストマネージャーという役割は存在しない場合があり、スクラムマスター、プロダクトオーナー、あるいは個人貢献者の役割に再編されることもある。それでもテストマネージャーは、チーム全員が高品質な製品作りに貢献していることを確実にすること、品質およびテストのコーチとしての役割を維持しなければならない。このようにして、テストマネージャーはテスト組織そのものためではなく、プロジェクトのテスト活動の推進者となる。

アジャイルプロジェクトにおけるテストアプローチは、従来のライフサイクルモデルとは異なる傾向がある。特に、アジャイルプロジェクトでは、次のようなコンセプトが通常見られる：

- チーム全体が品質に責任を持つ。熟練したテスト担当者とテストマネージャーは、設計や開発などの他のプロセスについてより深く学び、他のチームメンバーがテストの必要性を理解するのを助ける。各個人が品質に対する責任を負わなければならない。
- プロセスやツールの軽量なソリューションが好まれる。ソリューションはプロジェクト内で頻繁に変更される可能性がある。シンプルなソリューションは効率性の促進に役立つ。
- リグレッションテストのコストを最小化するためには、テスト自動化を効果的に活用しなければならない。テストは頻繁に実行すべきである。開発は、自動化の開発と保守において積極的な役割を果たすべきである。
- “done” や “complete” といった「完了」を表す用語の定義を明確にしなければならない。テスト担当者はその定義に貢献し、それが測定可能で理解できるものであることを確認する必要がある。
- 変更が頻繁に行われ、文書化が少ないため、良好なコミュニケーションが不可欠である。イテレーションにおける全タスクを最も効率的にカバーするためには、グループ内でのクロストレーニングが必要である。
- テストはイテレーション全体を通して実施され、イテレーション終了時に完成した「製品」を提供するために、あらゆる側面をカバーしなければならない。
- ソフトウェアをより大きなシステムに統合することは、早期に、そしてしばしば段階的に行わなければならない。
- 継続的なプロセス改善を実施し、実際のプロジェクトに対応するために、計画されたアプローチを必要に応じて変更しなければならない。

一般的に、テストの分析と設計の技法は、アジャイルプロジェクトにおいても、他のライフサイクルと同じである。ただし、テスト担当者とテストマネージャーの役割は適応しなければならず、テスト作業はイテレーションのニーズに合わせて調整しなければならない。テスト条件の定義や適切なテスト技法の選択など、テストに関する最良の実践例は依然として適用可能である。

アジャイルプロジェクトの標準的なアプローチに加えて、テストマネージャーは、いくつかの独自の課題に直面することもある。アジャイルプロジェクトの前提の 1 つは、チームが緊密に統合され、通常は同じ場所に配置され(仕切りがなく、1 つの巨大なブースのような環境)、アイデア、計画書、スケジュール、要件、テスト結果を伝えるために、形式的な文書ではなく、人ととのコミュニケーションに依存することである。グローバルな環境においては、同拠点での作業は特に困難である。チームが同じ場所にいられない場合は、簡単で迅速なコミュニケーション方法を確立しなければならない。例えば、チーム全体の同期を取るために毎日電話でステータスマーティングを行ったり、作業中にインスタントメッセージングツールを利用したりするだけでも、十分にコミュニケーションを取ることができる。コラボレーションソフトを使えば、ユーザーストーリー、設計メモ、実装情報などのプロジェクトドキュメントを分割して更新編集できる。アジャイルプロジェクトは、リアルタイムコミュニケーションを実現するチームの能力によって推進ができる。分散チームが、まるで 1 つの場所に集まっているかのような気持ちになるよう、そのニーズに応えるツールが盛んに開発されている。

アジャイルプロジェクトでは、コミュニケーションに遅れが生じるという点で、タイムゾーンの問題が依然としてつきまとう。繰り返しになるが、チームは、昼夜の特定の時間帯にチームメンバー全員がオンラインでアクセス可能となる時間を設定することで、この問題を効果的に回避することができる。これは最適な作業環境を提供しないかもしれないが(特に深夜または早朝のミーティング)、チームのコミュニケーションを促進するためには必要である。

アジャイルプロジェクトにおけるテスト文書化も、進化するソフトウェアの詳細な文書化で発生するメンテナンス負担を考慮し、最小限にとどめる。チェックリストはテストのガイドとしてよく使われ、探索的テストはソフトウェアの新しいイテレーションごとに多用される。チェックリストは、実装対象機能に基づいて作成更新され、実際に実装された機能に基づいて修正される。

現在のテスト実践例をアジャイルプロジェクトに適応させることは困難な場合がある。場合によっては、変更されたプロジェクトアプローチに対応するために、新しい実践例を開発しなければならないこともある。テストマネージャーは、既存の優れた実践例を活用しつつ、新しいモデルに適応することが重要である。

## 8.3 部分的なライフサイクルプロジェクトのマネジメント

テストマネージャーは、開発の全部または一部が外部チームによって行われるプロジェクトにしばしば関与する。このようなタイプのプロジェクトでは、テストマネジメントの課題が発生する。この課題は、関与の遅れ、ドキュメンテーションの不足、複数のオーナーシップによって引き起こされることがある。この節では、このような部分的なライフサイクルプロジェクトについて詳しく見ていく。

### 8.3.1 統合プロジェクト

サードパーティと協働する場合のテストマネージャーの役割については、前の章で述べた。本節では、統合プロジェクトを扱うときに必要となるテストライフサイクルの変更に焦点を当てる。外部開発プロジェクト(または部分的に外部開発されたプロジェクト)への関与のタイミングは、多くの場合、自社開発プロジェクトより遅い。通常、コードは完全に開発され、統合のために受け取った時点で、少なくともユニットテストは完了していると想定される。これは、テストチームが要件やデザインレビューに関与していない可能性があり、プロジェクトの計画、分析、設計の時間が大幅に短縮されることを意味する。

「期待される」受け入れ品質レベルを明確に定義することが重要である。その製品は別のテストグループによってテストされたのか？一般的なリリースの準備ができていると考えられているのか、それともさらなる開発と統合の準備ができているだけなのか？このような情報は、利用可能であってもなくてもよい(そして正確であってもなくてもよい)。テストマネージャーは、文書化された機能の機能性を検証するスマートテストから始めるのが賢明だ。それが確認できたら、既知の、そして期待される詳細機能のスポットチェックを進めるべきである。これには、テストをサポートするために、既存のドライバーやスタブの開発または再利用が要件となるかもしれない。これは探索的テストや、チェックリストが利用可能であればチェックリストテストで行うこともできる。この時点で、さらにテストが必要かどうか、どのようなタイプのテストが必要かを判断するために、品質評価を行うことができる。このテストの時点では、すでに重大な欠陥を示しているソフトウェアは、追加のシステムテストが必要である。未知の欠陥がないソフトウェアは、使用目的に応じて保証する非機能テスト(性能、使用性など)の準備ができている。

市販ソフトウェア(COTS)は、一般的に品質が高く、一般リリースに適していると思われている。もちろん、これは予備テスト後に誤った仮定であることが判明する可能性はあるが、そうでないことが証明されるまでは、一般的にこれが作業上の仮定である。この分野の追加テストは、統合のレベル、テストの責任、ベンダーの経験に依存する。例えば、チームがプリンタードライバーを開発していて、ベンダーのパッケージで生成された画像が印刷可能であることを確認したい場合、ゴールはベンダーのパッケージをテストすることではなく、生成された画像(またはそれらのさまざまな画像)が印刷可能であることを確認することである。もしチームが、社内で開発したアプリケーションに統合され、ユーザーには単一のインターフェースとして提供される「サードパーティが開発したコンポーネント」をテストしている場合、そのサードパーティーコンポーネント自体の機能を徹底的にテストする必要がある(あるいは、別のテストグループが実施したテストを検証する必要がある)。同様に、内部アプリケーションとの統合ポイントについてもテストが必要となる。

場合によっては、サードパーティから受領した製品に、正式な受け入れサインが必要となる。この場合、機能テストと非機能テストが不要であれば、テストマネージャーは受け入れ基準を作成し、テストを実施し、結果をレポートする。受け入れテストが成功すれば、テストチームの関与は終了する。受け入れテストが不成功の場合は、機能／非機能テストとテストチームのさらなる関与の必要性を示すことがある。

### 8.3.2 メンテナンスプロジェクト

本番稼動中のソフトウェアは、定期的にメンテナンスリリースを必要とする。これは、システムを新しい環境に移行したり、システムの一部をリタイアさせたり、あるいはフィーチャーを追加したり、または修正を提供したりする必要があるためである。メンテナンスリリースは、定義されたフィーチャーセットとともに事前にスケジュールされることもある。メンテナンスリリースは、(もしかつたとしても)ほとんど計画を立てることなく、「緊急の」修正という形を取ることもある。メンテナンステストの最も重要な側面は、修正/新フィーチャーによってリグレッションが発生しないことを確実にすることである。このように、既存のリグレッションテストスイートは、通常のメンテナンスリリースの直前のスケジューリングと予期しないスコープに対処する上で貴重である。自動化されたリグレッションテストスイートは、このケースでは非常に貴重であり、修正と新フィーチャーを検証するための手動テストリソースを使わないで済む。

修正の緊急度によっては、ユニットテストや統合テストなど、開発グループによって通常のプロセスがバイパスされるかもしれない。その場合、テストグループは、そのコードが事実上テストされていないと仮定する必要があるかもしれない。そのため、システムテストの段階で、より徹底的なテストが要求され、テストに必要な時間が長くなる。影響度分析は、機能テストとリグレッションテストの両方をターゲットにする際に役立つ。

どのようなメンテナンスプロジェクトでも、ある程度の変更マネジメントが必要となる。ほとんどの組織では、計画されたリリースが「便利な入れ物」となり、追加の修正/変更を加える傾向がある。追加の変更には追加のリスクが発生するため、テストマネージャーは、予定していたメンテナンスへの追加で発生するリスクについて、プロジェクトチームに通知しなければならない。

### 8.3.3 ハードウェア/ソフトウェアおよび組込みシステム

ハードウェア/ソフトウェアを組み合わせたプロジェクトや、組込みシステムのプロジェクトには、特有の計画ステップがいくつか必要となる。特に、テスト構成がより複雑になる傾向があるため、テスト構成のセットアップとテストのために、より多くの時間をテストスケジュールに割り当てるべきである。このような環境では、試作機器、試作ファームウェア、および試作のためのツールが関わることが多く、その結果、機器の故障や交換の問題が頻繁に発生し、予期せぬダウンタイムを引き起こすことがある。テストスケジュールは、機器の停止を許容し、システムダウンタイム中にテストリソースを利用する方法を提供すべきである。

試作機器や、診断ツールなどの試作用ツールを扱うと、テストがさらに複雑になり得る。試作機器は、量産モデルにはない挙動を示すことがあり、ソフトウェアはその異常な挙動を扱うことを想定しない可能性がある。この場合、欠陥をレポートすることは適切ではないかもしれない。量産前モデルをテストに使用する場合、テストチームはソフトウェア開発者だけでなく、ハードウェア/ファームウェア開発者と緊密に連携し、何が変化しているのかを理解しなければならない。しかし、1つ注意点がある。逆のケースもあり得るからだ。試作機器が正しくない挙動を示し、ソフトウェアがエラーを出すべきなのに、その正しくない挙動を受け入れてしまうことがある。これは、正しく動作するハードウェアにのみ現れる問題を覆い隠してしまうことになる。

このようなタイプのシステムのテストには、ときとして、テストチームに一定のハードウェア/ファームウェアの専門知識が求められる。問題のトラブルシューティングは困難で時間がかかることがあり、また、ハードウェアとファームウェアに多くのリリース/変更が発生すると、再テスト時間が膨大になることがある。緊密な協力関係と、他のチームが何をしているのかの認識がなければ、継続的なハードウェア/ファームウェア/ソフトウェアの変更は、統合テストの繰り返しにより、テストチームの前進を阻害することになりかねない。

このようなタイプのシステムをテストする際には、統合戦略が重要である。統合を阻害する可能性のある問題を予測し、計画することは、プロジェクトの遅延を避けるのに役立つ。統合テストラボは、未テストの機器/ファームウェアをソフトウェアテスト環境から分離するためによく使用される。機器/ファームウェアは、システム統合テストフェーズに入る前に、統合ラボを「卒業」しなければならない。

テストマネージャーは、効率的なテストが行えるようにコンポーネントのスケジュールを調整する上で重要な役割を果たす。これには通常、論理的な開発スケジュールをサポートしながらテストを可能にするために、複数回のスケジュール会議、頻繁なステータス更新、機能提供の順序変更などが要件となる。

### 8.3.4 セーフティクリティカルシステム

セーフティクリティカルなシステムには、規制や公表されている規格に適合するための追加テストの必要性など、いくつかの特別な課題がある。テストチームは、ライフサイクルの全フェーズに関与することを想定し、要件レビュー、設計レビュー、コードレビュー、ユニットテストレビュー、統合テスト、システム統合テスト、および受け入れテストのために時間を割くべきである。プロジェクトのライフサイクルに継続的に関与することに加えて、テストチームは、トレーサビリティマトリクスを保守するだけでなく、テスト文書を含む文書の作成、検証、レビューに多大な時間を費やすことになる。ライフサイクルを通じて指摘された欠陥はすべて文書化されなければならず、テスト文書を含め、すべての文書はバージョン管理されなければならない。ドキュメントの内部的な一貫性をチェックし、すべての変更を検証し、署名しなければならない。バージョン記述文書は、各文書の各バージョンを追跡するために一般的に使用され、バージョンごとに発生したすべての変更を記す。

業界固有の規制や標準への準拠によっては、文書化のレベル、使用できる／できないツール、自動テストのカバレッジ、コードや要件のカバレッジ、終了基準、受け入れ基準、欠陥分類など、製品や業界に応じて必要とされる具体的な事項が規定されることがある。

セーフティクリティカルシステムについては、[Leveson95]を参照のこと。

## 8.4 リリースに関する考慮事項

テストマネージャーは、リリースに関する考慮事項について発言できるかもしれないし、できないかもしれない。これは多くの場合、ビジネス、マーケティング、プロダクトマネージャーの領域である。しかし、テストマネージャーは、異なるリリース方法、製品、顧客に対する考慮事項を知っておく必要がある。

### 8.4.1 市場の需要

市場の需要によっては、フィーチャーが完成する前に製品をリリースすることが意味を持つ場合がある。これは、競合製品がすでに市場に出ていたり、問題のあった製品の置換であったり、新しいコンセプトに対するフィードバックを早期に得たいためであったりする。どのような理由であれ、テストマネージャーは、このタイプの部分リリースをサポートするために、テストフェーズの終了基準を調整しなければならないかもしれない。終了基準の調整は、品質やサポートに影響を及ぼす可能性があるため、プロジェクトチームの承認を得てから行うべきである。

### 8.4.2 メンテナンスの容易さ

リリース判断では、顧客への修正プログラムの提供のしやすさも考慮する。製品によっては、修正プログラムは自動的にダウンロードされる場合もあれば、メディアの提供を要件とする場合もあり、また、顧客が必要に応じてダウンロードできるようにする場合もある。修正版の入手と設置の両方が容易であるかどうかが、最終テストの承認前に製品をリリースすることの受け入れ度合いを決定する場合がある。

悪評や顧客の不満、プロダクト故障のリスクがある場合、修正計画を立ててプレリリースすることは危険なアプローチとなり得る。クリティカルではないソフトウェアは、ユーザー層がより確実なリリースを待つよりも、すぐに解決されるであろう欠陥に対処することを望むと仮定して、この方法でリリースされるかもしれない。繰り返しになるが、これは通常、ビジネス、マーケティング、またはプロダクトマネージャーによって判定される。テストマネージャーは、テストのステータス、残存リスク、既知の問題点、修正プログラムのおおよそのスケジュールに関する正確な情報を提供できる必要がある。

頻繁に修正プログラムを提供する際のもう 1 つの考慮点は、顧客がその修正プログラムを採用する際の難しさである。例えば、チームが顧客に使用するために API を公開しているソフトウェアを扱っている場合、その API を変更することは、顧客側で追加のコーディングを必要とする可能性があり、顧客は変更や修正を迅速に採用しにくくなる。採用されなかった複数の修正は、複数の異なるバージョンのソフトウェアを実行しているユーザー層の存在によって、サポートを困難にする可能性がある。

#### 8.4.3 インストールの容易さ

メンテナントリースを検討する場合、テストチームは、修正がベースソフトウェアに与える影響を理解しなければならない。データコンバージョンは必要か？コーディングは必要か？ユーザーインターフェースの変更により、ユーザーを再教育する必要があるか。修正のインストールと実装の間、顧客にダウントIMEが発生するか？

インストールが簡単な修正もあれば、難しい修正もある。もし、顧客が一連の修正プログラムの中から選んでインストールできるのであれば、テストとサポートの両面から、コンバージョンとインストールの要件が非常に複雑になる可能性がある。5 つの修正プログラムがリリースされ、3 つ目の修正プログラムでコンバージョンが必要になった場合、4 つ目と 5 つ目の修正プログラムでは、コンバージョンが完了していることが前提になる。もし顧客が 3 つ目の修正を行わず、5 つ目の修正を望んだ場合、そのシステムでは動作しない可能性がある。このような事態を防ぐため、修正プログラムはバンドルされることが多いが、これにはより多くのテスト工数が必要となる。

デプロイの仕組み／能力もテストしなければならない。これは、適切な修正プログラムが適切な形式で適切なシステムに配布されることを保証するためである。これには、インストール手順、自動化ウィザード、さまざまなアップグレードパス、アンインストール手順、実際にユーザーに修正プログラムを配布する際に使用する仕組みなどのテストが含まれる。初期インストールテストと同様に、アップグレード機能も製品の一部として徹底的にテストされなければならない。

## 9. 有効性と効率性の評価 195 分

キーワード

有効性、効率性

有効性と効率性の評価の学習の目的

### 9.1 イントロダクション

本節の学習の目的はない。

### 9.2 テストプロセスの有効性、効率性、および満足度メトリクス

LO 9.2.1 (K2) 有効性、効率性、満足度のメトリクスを追跡する意義を説明し、テストプロジェクトに適用される例を挙げる。

### 9.3 テストポリシーの目的に対する有効性、効率性、満足度のメトリクス

LO 9.3.1 (K6) あるテストポリシーについて、有効性、効率性、満足度のメトリクスを定義、評価、レポートする。

### 9.4 プロジェクトのふりかえり

LO 9.4.1 (K6) ある完了したプロジェクトについて、プロジェクトのふりかえりをマネジメントし、プロジェクトのふりかえり中に収集したデータを分析し、改善計画を立案する。

## 9.1 イントロダクション

テストマネージャーは、改善の機会を探すだけでなく、進行中のテスト労力をモニタリングし、継続的にテストグループの有効性と効率性を評価しなければならない。

## 9.2 テストプロセスの有効性、効率性および満足度メトリクス

有効性メトリクスは、テストプロセスがリスク軽減を含む適切な作業成果物や成果をどの程度生み出しているか(すなわち、テストがどの程度うまく行われたか)を調べる。効率性メトリクスは、それらの作業成果物や成果を達成するために必要な労力、期間、そして/またはリソース投入量(すなわち、価値をどの程度うまくリリースできたか)を調べる。満足度メトリクスは、テストプロセスが、テストステークホルダーに実際の価値と認識される価値の両方をどの程度洗練した形でリリースできたか(すなわち、どのように受け止められたか)を調べる。

有効性メトリクス:

- コントロール活動のうち、特定された新規タスクの割合
- 初期分析後に特定された新規および変更されたテスト条件の割合
- 特定されたテスト条件によってカバーされる要件の割合
- テスト条件を網羅するために設計すべきテストについての、テストアナリストの理解度調査
- テストがカバーするテスト条件の割合
- テストカバレッジを検証するための、テスト技法の適切な使用
- プロジェクト終了後の終了基準逃れの評価(例えば、終了前に実行されなかった、あるいは意図的にスキップされたテスト群など)
- 見逃された欠陥率
- 以前のプロジェクトから学んだ教訓に基づく、現在のプロジェクトにおけるテスト目的達成度の向上

効率性メトリクス:

- テスト計画に対するマネジメントされていない変更の工数、期間、リソースへの影響
- 不測の阻害要因による損失時間の割合
- テスト文書における粒度の一貫性
- テスト設計あたりの平均工数
- テストの自動化率
- テスト実装あたりの平均工数
- 過去のプロジェクトから学んだ教訓に基づき、現在のプロジェクトで達成された労力、期間、および/またはリソースの節約
- 結果報告に関するプロジェクトオーバーヘッド
- 以前のプロジェクトからのテスト作業成果物の再利用の結果、現在のプロジェクトで達成された労力、期間、および/またはリソースの節約

満足度メトリクス:

- プロジェクトステークホルダーの調査結果
- ビジネスゴールに対する計画とスケジュールの一致度
- テストレポートの受領者に対する調査
- テスト終了作業の有用性に関するテスト担当者への調査

テストマネージャーは、テスト担当者やその他のプロジェクトチームメンバーが、プロセスマトリクスに関して満足のいく結果を得るために行動を変えることを忘れてはならない。経営陣による明確なゴールが定められていない場合、メトリクスは、個人貢献者やマネージャーの行動や成果物に、不適切な偏りを生じさせる可能性がある。

従って、テストマネージャーは、プロセスマトリクスの導入を注意深く計画し、意図しない副作用がないか常にモニタリングする必要がある。

テストマネージャーは、これらのプロセスマトリクスは、プロジェクトや組織全体の行動、活動、属性の影響を強く受けたことも忘れてはならない。システム要件仕様書、設計、コーディング、ユニットテスト、リリースエンジニアリングのような上流プロセスの実行が成功すると、テストプロセスの有効性、効率性、満足度に大きなプラスの影響を与える可能性がある。上流プロセスが失敗すると、テストプロセスに逆の影響を与える可能性がある。テストマネージャーは、全プロセスを俯瞰している傾向があり、その情報を活用して、全体的なプロセスフローに影響を与えるのを助けるべきである。

### 9.3 テストポリシーの目的に対する有効性、効率性、満足度のメトリクス

テスト目的を特定し、テストポリシーに文書化した後、有効性、効率性、満足度を測定するメトリクスを定義する必要がある。この節では、典型的なテスト目的に対するそのようなメトリクスの例を示す。テストプロジェクトはそれぞれ異なるので、特定のプロジェクトに適したメトリクスは異なるかもしれない。

次のような典型的なテスト目的を考えてみよう:顧客満足度またはユーザー満足度に影響する可能性のある重要な欠陥事項を発見する。そして、開発者(または他の作業成果物作成者)がリリース前に修正できるように、これらの欠陥に関する十分な情報を作成する。この目的に対する有効性、効率性、および満足度のメトリクスには、次のようなものがある:

- 欠陥検出率(有効性)
- 重大欠陥の検出率(有効性)
- 開発で却下された欠陥の割合(効率性)
- 本番環境に流出した重大欠陥の割合(有効性)
- 提供された情報に対する欠陥レポート受領者の満足度(満足度)
- テスト中に発見された欠陥 1 件あたりのコスト(場合によっては、本番で発見された欠陥 1 件あたりのコストに対する割合)(効率性)

次のような典型的なテスト目的を考えてみよう:主要な品質リスクに関する重要なテストをリスク順に実行することでリスクをマネジメントし、リリース前にシステムの品質に関するリスクを既知の許容レベルまで低減する。この目的に対する有効性、効率性、満足度メトリクスには、次のようなものがある:

- テスト実行の初期段階で発見された重大欠陥の割合(有効性)
- テスト実行の初期段階で実行された重要なテストの割合(有効性)
- リリース前に受容されたリスクに対するステークホルダーの認識。場合によっては、本番稼動後一定期間経過時のリスク認識との比較(満足度)
- テスト(実行)でカバーした、特定したリスクの割合(有効性)。
- テスト中にカバーしたリスクアイテムあたりの平均コスト(効率性)

次のような典型的なテスト目的を考えてみよう:品質、テスト、リリース準備に関する重要な情報をプロジェクトチームに提供する。この目的に対する有効性、効率性、満足度メトリクスには、次のようなものがある:

- テストによって提供される情報の十分性、重要性、適時性についてのプロジェクトチームへの調査(有効性および満足度)
- テストレポート作成に関連する間接費(効率性)

次のような典型的なテスト目的を考えてみよう:リリース前に欠陥が修正されない場合、カスタマーサポートやヘルプデスクが、これらの欠陥に関するユーザーの問題を解決できるように、重要性の低い欠陥や、それらに

に対する回避策を見つけ出すこと。この目的に対する有効性、効率性、満足度メトリクスには、次のようなものがある:

- テストによって提供された情報の十分性、重要性、適時性に関して、カスタマーサポートまたはヘルプデスクを対象とした調査(有効性および満足度)
- 欠陥情報には、適切な延期／修正の判定に必要な十分な情報が含まれているというステークホルダーの認識(有効性)
- カスタマーサポートまたはヘルプデスクスタッフへの情報提供に関連するコスト(効率性)

テスト目的に対するメトリクスの導出に関する詳細な議論については、[Riley&Goucher09]を参照のこと。

前節で述べた、意図しない副作用や上流工程の影響に関する注意点は、本節で取り上げるメトリクスにも当てはまる。

## 9.4 プロジェクトのふりかえり

プロジェクトのふりかえりミーティングとは、チームが、直近で完了したプロジェクトまたは活動の目的を有効的かつ効率的に達成した度合い、およびそれらの目的が達成された方法に対するチームの満足度を評価するチームミーティングのことである。アジャイルプロジェクトでは、ふりかえりはチームの全体的な成功にとって極めて重要である。各イテレーションの終わりにふりかえりを行うことで、アジャイルチームはプロセスをレビューし、変化する状況に適応し、人間関係の課題を識別して解決するのに役立つ方法で対話する機会を得る。

テストマネージャーは、効果的なふりかえりミーティングを開催し、進行できるようになるべきである。このようなミーティングでは、ときに対立を招きやすい論点を話し合うため、以下のルールを適用すべきである:

- オープンな意見交換を奨励するが、個人的な批判は控える。
- 参加者には、自分の主張を説明するためのデータを提示できるように準備させ、意見や感情に頼らないようにする。
- 今後のプロジェクトにおいて、単にうまくいかなかったという不満ではなく、改善のためのアイデアを持ってきてもらう。
- うまくいったこと、今後も継続すべきことの例を挙げてもらう。

[Derby&Larsen06]によれば、ふりかえりは以下に示す概略のような構造に従うべきである:

- 場を整える
- データを集める
- 気づきを出す
- 次にやることを決める
- ふりかえりを締めくくる

もちろん、ふりかえりの目的は、実際の変化をもたらす成果を出すことである。ふりかえりを成功させるには、以下のことが必要である:

- データ、報告書、ふりかえりミーティングの議事録を蓄積し、これらの情報を、将来のプロジェクトのためのリソースとしてすぐに利用できるようにしておくこと。
- 根本原因分析(RCA)セッションでは、発生した問題の原因を識別し、将来的にその原因を取り除くための解決策を提案することに集中すること。
- ふりかえりミーティングから得られた優れたアイデアを実装するためにマネジメントのコミットメントを得ること。改善は話し合うだけでなく、実際に実現されるようにすること。

テストマネージャーは、プロジェクト全体のふりかえりに参加する場合がある。また、テストチームだけでテストふりかえりを行い、テストチームがコントロールできることや改善できることに焦点を当てることにも価値がある。

## 10. 参考文献

### 10.1 ISTQB ドキュメント

ID	参考文献
[ISTQB-ALTM]	ISTQB Certified Tester, Advanced Level Syllabus, Test Manager, Version 2007, [ISTQB-Web]から入手可能。(日本訳注:2025年現在、最新バージョンである ALTM 3.0 が入手可能)
[ISTQB-EL-Modules]	ISTQB Expert Level Modules Overview, Version 1.0, March 11th 2011
[ISTQB-EL-CEP]	ISTQB Certified Tester Expert Level, Certification Extension Process, Version 1.0, June 17 <sup>th</sup> 2008. [ISTQB-Web] から入手可能。
[ISTQB-Glossary]	ISTQB Glossary of Terms Used in Software Testing, Version 2.1, [ISTQB-Web]から入手可能。(日本訳注:2025年現在、ISTQB Glossary は専用のサイトがある。 <a href="https://glossary.istqb.org/ja_JP/search">https://glossary.istqb.org/ja_JP/search</a> )

### 10.2 商標

本書では、以下の登録商標およびサービスマークを使用している：

ISTQB® , MBTI®

ISTQB® は International Software Testing Qualifications Board の登録商標である。

マイヤーズ=ブリッグス・タイプ指標 (MBTI®) は、マイヤーズ&ブリッグス財団の登録商標である。

### 10.3 書籍（日本語翻訳が出版されているものは追記している）

ID	書籍名
[Black03]	Black, Rex, Critical Testing Processes: Plan, Prepare, Perform, Perfect. Addison-Wesley, 2003.「ソフトウェアテスト 12 の必勝プロセス, 日経 BP 社」
[Black09]	Rex Black, Managing the Testing Process, Wiley, 2009, ISBN: 0-470-40415-9.「基本から学ぶテストプロセス管理, 日経 BP 社」
[Boehm03]	Boehm, Barry, and Richard Turner, Balancing Agility and Discipline: A Guide for the Perplexed, Addison-Wesley, 2003.
[Brooks95]	Brooks, Fred, The Mythical Man-Month: Essays on Software Engineering, 2e. Addison-Wesley, 1995.「人月の神話, 丸善出版」
[Craig02]	Craig, Rick David and Stefan P. Jaskiel, Systematic Software Testing, Artech House, 2002, ISBN: 1-580-53508-9.「体系的ソフトウェアテスト入門, 日経 BP 社」
[DeMarco99]	DeMarco, Tom, and Timothy Lister, Peopleware: Productive Projects and Teams, 2e. Dorset House, 1999.「ピープルウェア-やる気こそプロジェクト成功の鍵, 日経 BP 社」
[DeMarco03]	DeMarco, Tom, and Timothy Lister, Waltzing With Bears: Managing Risk on Software Projects. Dorset House, 2003.「熊とワルツを - リスクを愉しむプロジェクト管理, 日経 BP 社」
[Derby&Larsen 06]	Derby, Esther and Diany Larsen, Agile Retrospectives - Making Good Teams Great, The Pragmatic Bookshelf, 2006, ISBN 0-9776166-4-9.「アジャイルレトロスペクティブズ — 強いチームを育てる「ふりかえり」の手引き, オーム社」
[Drucker06]	Drucker, Peter, The Practice of Management, Harper, 2006.「現代の経営, ダイアモンド社」
[Ensworth01]	Ensworth, Patricia, The Accidental Project Manager: Surviving the Transition from Techie to Manager, Wiley, 2001.
[Evans04]	Evans, Isabel, Achieving Software Quality through Teamwork, Artech House, 2004, ISBN: 978-1580536622.
[Galen04]	Galen, Robert, Software Endgames: Eliminating Defects, Controlling Change, And The Countdown To On-time Delivery, Dorset House, 2004.
[Glass02]	Glass, Robert, Facts and Fallacies of Software Engineering, Addison-Wesley, 2002.「ソフトウェア開発 55 の真実と 10 のウソ, 日経 BP」
[Graham99]	Graham, Dorothy and Mark Fewster, Software Test Automation, Addison-Wesley, 1999.「システムテスト自動化 標準ガイド, 翔泳社」
[Ishikawa91]	What Is Total Quality Control? The Japanese Way, Prentice Hall Business Classics, ISBN:013952441X「日本の品質管理: TQC とは何か, 日科技連出版」
[Jones03]	Jones, Capers, Assessment and Control of Software Risks, Prentice Hall, 1993.「ソフトウェア病理学, 共立出版」
[Jones07]	Jones, Capers, Estimating Software Costs: Bringing Realism to Estimating, McGraw-Hill, 2007.「ソフトウェア見積りのすべて 第 2 版 — 現実に即した規模・品質・工数・工期の予測, 共立出版」
[Jones08]	Jones, Capers, Applied Software Measurement: Global Analysis of Productivity and Quality, 3e, McGraw-Hill Osborne Media, 2008.「ソフトウェア開発の定量化手法 第 3 版, 共立出版」
[Kan02]	Kan, Stephen, Metrics and Models in Software Quality Engineering, 2e, Addison-Wesley, 2002.「ソフトウェア品質工学の尺度とモデル, 構造計画研究所」

[Leveson95]	Leveson, Nancy, <i>Safeware: System Safety and Computers</i> , Addison-Wesley, 1995. 「セーフウェア：安全・安心なシステムとソフトウェアを目指して」, 翔泳社
[McConnell97]	McConnell, Steve, <i>Software Project Survival Guide</i> , Microsoft Press, 1997. 「新訳 ソフトウェアプロジェクトサバイバルガイド」, 日経 BP
[McConnell99]	McConnell, Steve, <i>After the Gold Rush: Creating a True Profession of Software Engineering</i> , Microsoft Press, 1999.
[McKay 07]	McKay, Judy, <i>Managing the Test People</i> , Rocky Nook, 2007, ISBN: 1-93395-212-1.
[Myers&Briggs 95]	Myers, Isabel Briggs (1980). <i>Understanding Personality Type</i> . Davies-Black Publishing, reprint edition, 1995, ISBN: 0-89106-074-X.
[PMI08]	Project Management Institute, <i>A Guide to the Project Management Body of Knowledge (PMBOK® Guide)</i> , 4e, Project Management Institute, 2008. 「プロジェクトマネジメント知識体系ガイド (PMBOK® Guide) 第4版」, Project Management Inst
[Riley 09]	Riley, Tim and Adam Goucher, <i>Beautiful Testing</i> , O'Reilly, 2009, ISBN: 0-596-15981-1. 「ビューティフルテスティング —ソフトウェアテストの美しい実践」, オライリージャパン
[Rothman04]	Rothman, Johanna, <i>Hiring The Best Knowledge Workers, Techies and Nerds: The Secrets &amp; Science Of Hiring Technical People</i> , Dorset House, 2004. 「ザ・ベスト 有能な人材をいかに確保するか」, 翔泳社
[Rothman07]	Rothman, Johanna, <i>Manage It!: Your Guide to Modern, Pragmatic Project Management</i> , Pragmatic Bookshelf, 2007. 「Manage It」, オーム社
[Uutting06]	Uutting, Mark and Bruno Legeard, <i>Practical Model-Based Testing: A Tools Approach</i> , Morgan Kaufmann, 2006.
[Yourdon03]	Yourdon, Edward, <i>Death March</i> , 2e. Prentice Hall, 2003. 「デスマーチ 第2版」, ソフトウェア開発プロジェクトはなぜ混乱するのか, 日経 BP
[Whittaker02]	Whittaker, James, <i>How to Break Software: A Practical Guide to Testing</i> , Addison Wesley, 2002.
[Wiegers96]	Wiegers, Karl, <i>Creating a Software Engineering Culture</i> , Dorset House, 1996. 「ソフトウェア開発の持つべき文化」, 翔泳社

## 10.4 論文／記事

<u>ID</u>	<u>論文／記事名</u>
[Copeland Paper 01]	Lee Copeland, "When helping doesn't help", SQE Magazine, January 2001

## 10.5 Web 参考文献

これらの参考文献は、Expert Level シラバスの発行時点で確認されているが、その後に利用できなくなっている場合については、ISTQB は責任を負わない。

<u>ID</u>	<u>Web サイト名</u>	<u>リンク</u>
[Agile Manifesto-Web]	Website of the Agile Manifesto	<a href="http://www.agilemanifesto.org">www.agilemanifesto.org</a>
[ISTQB-Web]	Website of the International Software Testing Qualifications Board.	<a href="http://www.istqb.org">www.istqb.org</a>

**ID**  
[P&C-Web]

**Web サイト名**  
Roles in agile projects

**リンク**  
[http://en.wikipedia.org/wiki/The\\_Chicken\\_and\\_the\\_Pig](http://en.wikipedia.org/wiki/The_Chicken_and_the_Pig)

## 11. 付録 A – 教育機関への通知

### 11.1 研修時間

シラバスの各章には、分単位で講義時間が割り当てられている。この目的は、認定コースの各節に割り当てられる相対的な時間の割合に関するガイダンスを与えることと、各節の講義にかかるおよその最低時間表示することである。

研修提供者は指標よりも多くの時間を費やすかもしれないし、受験者は読書や調査にまた多くの時間を費やすかもしれない。コースのカリキュラムは、シラバスと同じ順序に従う必要はない。

連続した 1 ブロックの時間でコースを実施する要件はない。

下表は、各章の講義時間と演習時間のガイドラインであり、職場で実施する演習のタイミングを別に示している（時間はすべて分単位で表示）。職場での演習は、研修提供者によるコースの一環として実施される場合もあることに留意されたい（下記参照）。

Nr.	章	講義と演習	実践演習	合計(分)
1	イントロダクション	60	0	60
2	テストミッション、ポリシー、戦略、ゴール	285	270	555
3	テストチームのマネジメント	690	165	855
4	対外関係のマネジメント	330	0	330
5	組織横断マネジメント	525	255	780
6	プロジェクトマネジメントエッセンシャルズ	255	360	615
7	テストプロジェクトの評価とレポート	510	0	510
8	ドメインとプロジェクト要因に関するテストの考慮事項	270	0	270
9	有効性と効率性の評価	195	0	195
合計		<b>3120</b>	<b>1050</b>	<b>4170</b>

以下の表は、1日の平均労働時間を7時間として、日数で合計した受講時間である。

コース内容	日数	時間	分
コースの講義と演習	7	3	0
職場での実践演習	2	3	30
トータル:	9	6	30

### 11.2 職場における実践演習のガイドライン

受験者が知識を応用することが期待されるすべての側面（K3 以上の学習の目的）について、実践を含めること。講義と演習は、シラバスの学習の目的とトピックの記述に基づくべきである。

特定の学習の目的は、職場で実践演習を実施することによって達成することができる。ISTQB の Web サイト [ISTQB-Web] の Expert Level には、職場で実施する際の教育機関に適用されるルールが記載されている。

次の表は、職場での実習によってカバーされる可能性のある学習の目的を示している：

対象領域	関連する学習の目的(イタリック体は LO に加えた調整を示す)
2.2 ミッション、ポリシー、成功のメトリクス	LO 2.2.2 (K6) 組織内のソフトウェア、品質、依存関係(組織的、経済的、技術的)に長期的な効果をもたらす可能性のあるテストゴール、優先度、テストの目的を定義し、説明し、評価する。
2.3 テスト戦略	LO 2.3.3 (K6) 重要な要素を含み、成功要因を可能にし、故障の原因を軽減する、組織のテスト戦略文書を作成する。
2.4 テストポリシーとテスト戦略の組織との整合	LO 2.4.1 (K6) 組織および/または組織内のテストチーム(テストのプロセスと組織、人、ツール、システム、技法を含む)のテストポリシーおよび戦略を長期的および短期的に定義、記述、評価、改善する。
3.3 テストチームの育成	LO 3.3.2 (K4) 個々のテスト担当者について、S.M.A.R.T. ゴールのセットを作成し、そのゴールがその人の履歴書と組織のテスト戦略に基づいて適切かどうかを判断する。
3.4 テストチームのリード	LO 3.4.1 (K5) 組織内のテストチームが必要とする情報とガイダンスを評価し、最も効果的なコミュニケーション方法を決定する。
5.2 テストチームの価値の提示	LO 5.2.1 (K4) 組織内でテストチームを促進または支持するために取るべき適切な手順を定義する。
5.2 テストチームの価値の提示	LO 5.2.2 (K6) 組織におけるテストの定量的・定性的な有効性を定義し、その有効性をプロジェクトのステークホルダーに効果的に伝える。
5.3 テストチームの配置	LO 5.3.1 (K5) 組織の構造、ミッション、製品、顧客、ユーザー、および優先度を評価し、テストチームを組織内に適切に配置するための選択肢を決定し、それらの選択肢の意味を評価し、それらの選択肢の上層マネジメント向けの分析を作成する。
6.2 プロジェクトマネジメントタスク	LO 6.2.1 (K6) 組織内のあるプロジェクトについて、所定の見積り方法のうち少なくとも 2 つを使用してテスト工数を見積る。
6.2 プロジェクトマネジメントタスク	LO 6.2.2 (K6) 組織内の類似プロジェクトの過去のデータを使用して、現プロジェクトで発見、解決、納品される欠陥数を見積るモデルを作成する。
6.2 プロジェクトマネジメントタスク	LO 6.2.3 (K5) 組織内のプロジェクトにおいて、テストコントロールの一環として現状を評価し、テスト計画書からの逸脱を特定し、逸脱を解消するための効果的な尺度を提案するなど、テスト工数を長期的にマネジメント、追跡、調整する。
6.2 プロジェクトマネジメントタスク	LO 6.2.5 (K6) 組織内の過去プロジェクトの履歴情報と、プロジェクトステークホルダーから伝達された優先度を使用して、プロジェクトで利用可能な品質、スケジュール、予算、機能の間の適切なトレードオフを決定する。

### 11.3 コース計画の例

以下の図は、研修提供者が採用する可能性のあるコース計画を示している。これらは情報提供のみを目的としている。

### 11.3.1 コース計画例 - 職場での学習の目的なし

Example course plan	Duration (Hours)	Course Part 1					Course Part 2				
		1	2	3	4	5	6	7	8	9	10
Chapter		Class	1 2 3 4 5 6 7	1 2 3 4 5 6 7	1 2 3 4 5 6 7	1 2 3 4 5 6 7	1 2 3 4 5 6 7	1 2 3 4 5 6 7	1 2 3 4 5 6 7	1 2 3 4 5 6 7	1 2 3 4 5 6 7
1. Introduction	1,00										
2. Test Missions, Policies, Strategies and Goals	9,25										
3. Managing the Test Team	14,25										
4. Project Management Essentials	10,25										
5. Managing External Relationships	5,50										
6. Managing Across the Organization	13,00										
7. Test Project Evaluation and Reporting	8,50										
8. Testing Considerations for Domain&Project	4,50										
9. Evaluating Effectiveness and Efficiency	3,25										
Total	69,50	Hours									

このコース計画例では、すべての学習の目的を講義で教える。この計画では、コースを 2 つのパートに分け、それぞれを 5 日間とすることを提案している。講義に費やす時間は 10 日間(69.5 時間)である。

### 11.3.2 コース計画例 - 職場での学習の目的を含む

Example course plan	Duration (Hours)	Course Part 1					Workplace	Course Part 2			
		1	2	3	4	5		1	2	3	4
Chapter	Total	Work	Class	1 2 3 4 5 6 7	1 2 3 4 5 6 7	1 2 3 4 5 6 7	1 2 3 4 5 6 7	1 2 3 4 5 6 7	1 2 3 4 5 6 7	1 2 3 4 5 6 7	1 2 3 4 5 6 7
		Place									
1. Introduction	1,00	0,00	1,00								
2. Test Missions, Policies, Strategies and Goals	9,25	3,00	6,25								
3. Managing the Test Team	14,25	2,00	12,25								
5. Managing Across the Organization	13,00	3,00	10,00								
6. Project Management Essentials	10,25	4,00	6,25								
			7,00								
7. Test Project Evaluation and Reporting	8,50	0,00	8,50								
4. Managing External Relationships	5,50	0,00	5,50								
8. Testing Considerations for Domain&Project	4,50	0,00	4,50								
9. Evaluating Effectiveness and Efficiency	3,25	0,00	3,25								
Total	69,50	12,00	64,50	Hours							

N days in  
the  
workplace  
(WP)

Feedback day

このコース計画例では、職場で実施される可能性のあるすべての学習の目的が、コースパート 1 に続く 1 つのブロックに割り当てられている。コースパート 2 の開始時には、丸 1 日(7 時間)が職場からのフィードバックに割り当てられている。総講義時間は 9 日間(64.5 時間)である。

## 11.4 e ラーニングのルール

本シラバスの以下の部分は、e-ラーニングとして実施するには適切でない。

節	理由
3.2 テストチームの構築	面接のスキルは遠隔では効果的に練習できない。
3.3 テストチームの育成	パフォーマンス・レビューの実施は、遠隔では効果的な実践ができない。

## 11.5 標準の利用

このシラバスには、研修資料を作成する際に使用しなければならない、確立された標準への言及が含まれている。使用される各標準は、本シラバスの最新版に引用されているものでなければならない。このシラバスで引用されていない他の出版物、テンプレートまたは標準を使用し、参照することもできるが、審査は行わない。