

# **ソフトウェアテスト標準用語集 (日本語版)**

**Version 2.3.J02**

**International Software Testing Qualifications Board  
用語集作業班**

**編集者: Erik van Veenendaal (Bonaire)  
(翻訳: JSTQB® 技術委員会)**

## **Copyright Notice**

This document may be copied in its entirety, or extracts made, if the source is acknowledged.

Copyright © 2015, International Software Testing Qualifications Board (hereinafter called ISTQB®).

## 謝辞

このドキュメントは、Glossary working group of the International Software Testing Qualifications Board (ISTQB の Glossary 作業部会)が執筆した。

本チームは、全ての国の国際部会のメンバーによる提案と意見に感謝したい。

本用語集(Version 2.3)完成時の、Glossary working group (Glossary 作業部会)のメンバーは以下のとおりである(アルファベット順)。

Armin Beer, Armin Born, Mette Bruhn-Pedersen, Josie Crawford, Ernst Döring, George Fialkovitz, Matthias Hamburg, Bernard Homes, Ian Howles, Ozgur Kisir, Gustavo Marquez-Soza, Judy McKay (Vice-Chair) Avi Ofer, Ana Paiva, Andres Petterson, Juha Pomppu, Meile Posthuma. Lucjan Stapp, Erik van Veenendaal (Chair)

このドキュメントは、2014 年 3 月 28 日に開催された ISTQB®の総会で正式に発行された。

## 改訂履歴

### Version 2.3. 03-28-2014

この新版は、Foundation Extention Agile Tester シラバスをサポートするために開発されている。また、いくつかの変更要求も、ISTQB 用語集に反映されている。

#### New terms added:

- build verification test
- burndown chart
- BVT
- content reference model
- escaped defect
- MCDC
- n-wise testing
- PRISMA
- process reference model
- Product RiSk MAnagement
- QFD
- quality function deployment
- system under test
- TDD
- test analysis
- test reporting
- traceability matrix

#### Terms changed;

- acceptance criteria
- accuracy testing
- agile manifesto
- agile software development
- agile testing
- classification tree method
- combinatorial testing
- configuration item
- configuration management
- confirmation testing
- defect-based test design technique
- Defect Detection Percentage (DDP)
- defect taxonomy
- exploratory testing
- factory acceptance testing
- incremental development model
- iterative development model
- maintainability testing
- model-based testing
- orthogonal array testing
- pairwise testing
- performance testing
- product risk
- quality risk
- re-testing
- regression testing
- risk analysis
- risk assessment
- security testing
- smoke test
- software lifecycle
- test approach
- test automation
- test basis
- test charter
- test design
- test-driven development
- test estimation
- test execution automation
- test oracle
- test strategy
- unit test framework
- usability testing
- user story

## 改訂履歴 日本語版

バージョン	日付	摘要
Version 2.0. J01	2009-11-20	シラバスに合わせて修正 バージョン表記方法の変更
Version 2.0. J02	2011-04-19	※J02 では表記上の修正のみを行ない、用語の変更、削除、新規追加等を行なっていない <ul style="list-style-type: none"> <li>✓ 日本語としての揺れを一部修正</li> <li>✓ 誤字脱字を一部訂正</li> <li>✓ フォントなど一部表記を訂正</li> <li>✓ セキュリティ設定の変更</li> </ul>
Version 2.1. J01	2012-06-01	ISTQB 用語集 Version 2.1 (dd. April 1st, 2010) に対応 和訳の表現を全体的に見直し 以下用語削除 <ul style="list-style-type: none"> <li>✓ ユニットテストフレームワークツール</li> </ul> 以下用語名変更 <ul style="list-style-type: none"> <li>✓ インシデント管理ツール → インシデントマネジメントツール</li> <li>✓ インストーラビリティ → 設置性</li> <li>✓ インストーラビリティテスト → 設置性テスト</li> <li>✓ カバレッジ計測ツール → カバレッジ測定ツール</li> <li>✓ 擬似乱数 → 擬似ランダム</li> <li>✓ 構成管理委員会 → 構成コントロール委員会</li> <li>✓ 構造ベーステスト → 構造ベースドテスト</li> <li>✓ コードベーステスト → コードベースドテスト</li> <li>✓ 資源利用 → 資源効率性</li> <li>✓ 資源利用テスト → 資源効率性テスト</li> <li>✓ 実際の結果 → 実行結果</li> <li>✓ 実際の結果 → 実行結果</li> <li>✓ 仕様ベーステスト → 仕様ベースドテスト</li> <li>✓ 仕様ベーステスト設計技法 → 仕様ベースドテスト設計技法</li> <li>✓ 証明 → 認定</li> <li>✓ 信頼テスト → コンフィデンステスト</li> <li>✓ 遂行済み(の物) → 遂行済み(のもの)</li> <li>✓ 制御フロー分析 → 制御フロー解析</li> <li>✓ 成熟度 → 成熟性</li> <li>✓ 操作性 → 運用性</li> <li>✓ ソフトウェア有用性測定一覧表 → ソフトウェア使用性測定一覧表</li> <li>✓ タイムビヘイビア → 時間的振る舞い</li> <li>✓ 提供品 → 成果物</li> <li>✓ デヴィエーション → 逸脱</li> <li>✓ デヴィエーションレポート → デヴィエーションレポート</li> <li>✓ デザインベーステスト → デザインベースドテスト</li> <li>✓ テスト駆動型開発 → テスト駆動開発</li> <li>✓ テストデータジェネレータ → テストジェネレータ</li> <li>✓ テスト入力データ → テスト入力</li> <li>✓ テストの目的 → テスト目的</li> <li>✓ テスト見積 → テスト見積り</li> </ul>

		<ul style="list-style-type: none"> <li>✓ テスト容易性 → 試験性</li> <li>✓ テスト容易性レビュー → 試験性レビュー</li> <li>✓ ディフェクトマスキング → 欠陥マスキング</li> <li>✓ 独立性 → テストの独立性</li> <li>✓ バグ追跡ツール → バグトラッキングツール</li> <li>✓ ビジネスプロセステスト → ビジネスプロセスベースドテスト</li> <li>✓ フォールトトレランス → 障害許容性</li> <li>✓ 変異解析 → ミューテーション解析</li> <li>✓ 変更条件判定カバレッジ → 改良条件判定カバレッジ</li> <li>✓ 変更条件判定テスト → 改良条件判定テスト</li> <li>✓ 変更複合条件テスト → 改良複合条件テスト</li> <li>✓ 変更複合条件カバレッジ → 改良複合条件カバレッジ</li> <li>✓ 法令遵守 → 標準適合性</li> <li>✓ 法令遵守テスト → 標準適合性テスト</li> <li>✓ メトリクス → メトリック</li> <li>✓ ユーザビリティ → 使用性</li> <li>✓ ユーザビリティテスト → 使用性テスト</li> <li>✓ 要件ベーステスト → 要件ベースドテスト</li> <li>✓ リスクベーステスト → リスクベースドテスト</li> <li>✓ 論理網羅テスト → 論理カバレッジテスト</li> </ul>
Version 2.2. J01	2013-10-19	<p>ISTQB 用語集 Version 2.2 (dd. October 19th, 2012) に対応 誤記及び表現の揺れを全体的に修正(意味が異なるような対処 については以下に用語毎記載)</p> <p>以下の用語に対して用語説明を変更</p> <ul style="list-style-type: none"> <li>✓ ミューテーションテスト</li> </ul>
Version 2.2. J02	2014-04-25	<p>「3.構成」章の「並び順」の説明を一部変更</p> <p>以下の用語に対して用語説明を変更</p> <ul style="list-style-type: none"> <li>✓ アクセシビリティテスト</li> <li>✓ グラスボックステスト</li> <li>✓ 構造テスト設計技法</li> <li>✓ 構造テスト</li> <li>✓ 構造ベースのテスト設計技法</li> <li>✓ コードベースドテスト</li> <li>✓ チェックリストベースドテスト</li> <li>✓ バランスドスコアカード</li> <li>✓ 論理カバレッジテスト</li> <li>✓ 論理駆動テスト</li> </ul>
Version 2.2. J03	2014-05-12	<p>以下の用語に対して用語説明を変更</p> <ul style="list-style-type: none"> <li>✓ 頑健性〔堅牢性〕</li> <li>✓ 工場受け入れテスト</li> <li>✓ 故障モード影響解析</li> <li>✓ ボリュームテスト</li> </ul>
Version 2.3. J01	2014-10-29	<p>ISTQB 用語集 Version 2.3 (dd. March 28th, 2014) に対応 誤記及び表現の揺れを全体的に修正(意味が異なるような対処</p>

		及び用語名称の変更については以下に用語毎記載) 以下の用語に対して用語説明を変更 ✓ DD パス ✓ 品質の属性
Version 2.3. J02	2015-03-25	以下用語名変更 ✓ 論理テストケース → 論理的テストケース 以下の用語に対して用語説明を変更 ✓ N スイッチカバレッジ ✓ N スイッチテスト ✓ 受け入れテスト

## 目次

まえがき.....	8
1.序文.....	9
2.対象範囲.....	9
3.構成.....	9
4.商標.....	10
5.定義（あいうえお順）.....	11
あ.....	11
い.....	12
う.....	14
え.....	14
お.....	16
か.....	16
き.....	18
く.....	20
け.....	20
こ.....	22
さ.....	25
し.....	26
す.....	29
せ.....	30
そ.....	33
た.....	34
ち.....	35
て.....	35
と.....	43
に.....	45
の.....	45
は.....	45
ひ.....	47
ふ.....	49
へ.....	52
ほ.....	54
ま.....	54
み.....	55
め.....	55
も.....	55
ゆ.....	56
よ.....	57
ら.....	57
り.....	58
れ.....	59
ろ.....	59
わ.....	60
付録 A（参考資料）.....	61
索引（英語）.....	63

# まえがき

本用語集の編纂では、できるだけ広い分野で受け入れられる国際的テスト標準を作成するため、産業界、ビジネス分野、政府の諸団体・組織から可能な限り広範囲の見解やコメントを求めてきた。この類のドキュメントでは、全員が合意することはほとんどあり得ない。本ドキュメントの用語は、世界中のソフトウェアテスト団体から寄稿されたものである。

多数のソフトウェアテスト担当者は、BS7925-1 (British Standard *Glossary of Software Testing Terms*:ソフトウェアテスト用語の標準用語集)を 1998 年の初版以降使用してきた。同標準は、当初、コンポーネントテストを中心に編纂したものであった。しかし、同標準の出版以降、同標準がソフトウェアテストの広い範囲をカバーし、内容を改善するため、多数のコメントや提案が寄せられた。本用語集では、多数の提案が取り込まれている。本用語集は、国際ソフトウェアテスト資格認定委員会:International Software Testing Qualifications Board (ISTQB®)のソフトウェアテスト資格認定要綱の参考資料として使用する。

ISTQB 用語集には、主として次の 2 つの目的がある。

- さまざまシラバスで使用される用語を定義することにより、ISTQB シラバスの理解を促進する
- 標準テスト語彙を提供することにより、国際的なテストコミュニティ内のコミュニケーション及び関係者とのコミュニケーションを促進する

ISTQB に加盟している国又は地域の委員会は、ISTQB 用語集をその国又は地域で使用されている言語に翻訳できる。これらの委員会は、必要に応じて、ISTQB 用語集を特定の言語に適用できる。

Translation Copyright © 2005–2014, 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、又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。

本用語集は、Standard glossary of terms used in Software Testing Version 2.3 (dd. March 28th, 2014)を基に翻訳をしています。

本用語集は、予告なしに変更する可能性があります。



# 1.序文

テストの世界には、ステートメントカバレッジと判定条件カバレッジ、テストスイートとテスト仕様とテスト計画など、よく似た用語があり、これが、社会のさまざまな分野間で、インターフェースになっている。よく似た用語の違いを明確にできないと、産業界、ビジネス界、政府、技術や学術団体の内外部で、時間や労力の無駄になる。しかも、類似用語は、業界用語、技術用語として、別の意味を持つことも多い。

## 2.対象範囲

本ドキュメントは、ソフトウェアテストや、関連分野でのコミュニケーションを容易にするため、概念、用語、定義を記述したものである。さまざまな ISTQB のシラバスで使用される用語に対応している。

ただし、Foundation Agile extension シラバスで使用される一般的なアジャイル用語は、この用語集ではカバーしていない。これらの用語については、多くの著名なインターネットサイトで定義されており、これらの情報は Foundation Extension Agile Tester シラバスで紹介している。

## 3.構成

### 並び順

本用語集は、50 音順に並べた単一の定義部で構成する。用語の中には、他の同義語より優先させたものもある。その場合、優先した用語に定義を記述し、他の同義語は、優先度の高い用語を参照するようにした。たとえば、structural testing は、white-box testing を参照する。同義語には、「参照のこと」を表示した。

「～も参照のこと」というクロスリファレンスも使用した。これにより、目的の用語を高速で検索可能となる。「～も参照のこと」は、広義の用語と狭義の用語や、意味の重なりが分かるような構成になっている。

### キーワード

ISTQB 用語集は、さまざまな理由により、多くの用語を収容している。一部は、ISTQB シラバスの読者がシラバスを理解するのに支援するためだけに提供されている。一部は、用語が以前のバージョンのシラバスで使用されており、下位互換性の原則を適用するために提供されている。ただし、最も重要な用語は資格試験で使用されるキーワードで、さまざまな ISTQB シラバスで明示的に識別される。これらのキーワードの重要なユーザグループは、ISTQB 試験の準備をしているテスト専門家である。このユーザグループを支援するために、この用語集では、特定の試験用に理解する必要のあるキーワードに、識別子を付加している。継承の原則が適用される。つまり、ISTQB Advanced Level の受験者は、ISTQB Foundation Level のキーワードを理解する必要がある。キーワードには、次の識別子が付加されている。

F: ISTQB Foundation Level シラバスのキーワード

F-AT: ISTQB Foundation Extension Agile Tester シラバスのキーワード

ATM: ISTQB Advanced Level – Test Manager シラバスのキーワード

ATA: ISTQB Advanced Level – Test Analyst シラバスのキーワード

ATT: ISTQB Advanced Level – Technical Test Analyst シラバスのキーワード

EITP: ISTQB Expert Level – Improving the Testing Process シラバスのキーワード

ETM: ISTQB Expert Level – Test Management シラバスのキーワード

キーワードがシラバスで見つかるが、用語集では優先される用語とされていない場合、キーワードとそれが参照する用語（「参照のこと」）の両方に、適切なシラバス識別子が付いている。

### 参照

この用語集では、参照は 2 つの方法で使用される。

- 「after」が追記されていない大カッコ（たとえば、[ISO 9126]）は、参照として正確なテキストが使用されていることを意味する。

- 参照からの定義が小規模な変更により ISTQB 用語集に適用されている場合、「after」が追記されている(たとえば、[After ISO 9126])。

## 4.商標

本書では以下の商標を使用している。

- CMMI and IDEAL are registered trademarks of Carnegie Mellon University
- EFQM is a registered trademark of the EFQM Foundation
- Rational Unified Process (RUP) is a registered trademark of Rational Software Corporation
- STEP is a registered trademark of Software Quality Engineering
- TMap, TPA and TPI Next are registered trademarks of Sogeti Nederland BV
- TMMi is a registered trademark of the TMMi Foundation

## 5.定義(あいうえお順)

### あ

**アイソレーションテスト(isolation testing)：** 周辺コンポーネントとは独立に、個々のコンポーネントをテストすること。必要に応じて、スタブやドライバで、周辺コンポーネントをシミュレートする。

**EITP IDEAL モデル(IDEAL)：** 開始、計画、及び改善のアクションを実装するためのロードマップとして機能する組織的な改善モデル。IDEAL モデルは、開始、診断、確立、行動、及び学習の五つのフェーズにちなんで名付けられた。JSTQB 訳注)IDEAL のフェーズ名称は「CMMI V1.2 モデル - 開発のための - 公式日本語翻訳版」の定義を使用。

**アイテム送付レポート(item transmittal report)：** release note を参照のこと。

**アウトソーステスト(outsourced testing)：** プロジェクトとは異なる場所で、従業員ではない人々により実行されるテスト。

**アクションワード駆動テスト(action word driven testing)：** keyword-driven testing を参照のこと。

**ATA アクセシビリティテスト(accessibility testing)：** 身体的な制約を持つ人を含むユーザが、どの程度容易にコンポーネントやシステムを利用できるか判定するテスト。[Gerrard]

**アクター(actor)：** 特別な方法でテスト対象のシステムとやりとりするユーザ又は他の人、あるいはシステム。

**アークテスト(arc testing)：** branch testing を参照のこと。

**F-AT EITP アジャイルソフトウェア開発(agile software development)：** イテレーティブ-インクリメンタル開発に基づくソフトウェア開発。自己組織化された機能横断的役割を担うチーム間での共同作業によって要件と解決策を発展させていく。

**EITP アジャイルテスト(agile testing)：** エクストリームプログラミング(XP)のような技法や手法が取り込まれているアジャイルソフトウェア開発方法論を用いたり、開発をテストの一部とみなしたり、テストファースト設計パラダイムを重視するプロジェクトで実施するテスト。test-driven development を参照のこと。

**F-AT アジャイルマニフェスト(agile manifesto)：** アジャイルソフトウェア開発を支援する価値に関する宣言。ここでいう価値とは、以下のようなものである。

- ・プロセスやツールよりも個人との対話を重視する
- ・網羅的なドキュメントよりも動作するソフトウェアを重視する
- ・契約交渉よりも顧客との協調を重視する
- ・計画に従うことよりも変化に対応することを重視する

**EITP アセスメントレポート(assessment report)：** アセスメント結果、たとえば結論、提案、所見などをまとめたドキュメント。process assessment も参照のこと。

**EITP アセッサ(assessor)：** アセスメントを実行する人。アセスメントチームのメンバー。

**アドホックテスト(ad hoc testing)：** 非公式に実施するテスト。公式なテストの準備をせず、実績のあるテスト設計技法を用いず、テスト結果も予測せず、毎回、テスト方法が変わる。

**アドホックレビュー(ad hoc review)：** informal review を参照のこと。

**アナライザ(analyzer)：** static analyzer を参照のこと。

ATM **誤った合格結果 (FALSE-pass result)**: テスト対象に存在する欠陥を識別できなかったテスト結果。

ATM **誤った失敗結果 (FALSE-fail result)**: テスト対象には欠陥が存在しないにもかかわらず、欠陥として報告したテスト結果。

F **誤り (mistake)**: error を参照のこと。

**アルゴリズムテスト (algorithm test [TMap])**: branch testing を参照のこと。

F **アルファテスト (alpha testing)**: 潜在的なユーザ、顧客又は開発者のサイトではなく、開発組織の外部で独立したテストチームが、シミュレーションや実際のオペレーションにより実行するテスト。既成ソフトウェアでは、内部受け入れテストの一つとして実施することが多い。

**安全性 (safety)**: 利用者が指定された利用の状況で、人、事業、ソフトウェア、財産又は環境への害に対して、容認できるリスクの水準を達成するためのソフトウェア製品の能力。[ISO/IEC 9126] JSTQB 訳注) JIS X 0129-1:2003 より引用

**安全性テスト (safety testing)**: ソフトウェア製品の安全性を判定するテスト。

ATT **アンチパターン (anti-pattern)**: 最初は有益で一般的に使用できるように思われるが、実際には効果がない、又は逆効果である反復的な活動、プロセス、構造、又は再利用可能なソリューション。

ATT **安定性 (stability)**: ソフトウェアの修正による、予期せぬ影響を避けるソフトウェア製品の能力。[ISO/IEC 9126] maintainability も参照のこと。JSTQB 訳注) JIS X 0129-1:2003 より引用

## い

EITP **EFQM エクセレンスモデル (EFQM (European Foundation for Quality Management) excellence model)**: 組織の品質マネジメントシステムのための規範に縛られない (non-perspective) フレームワークであり、EFQM(ヨーロッパ品質マネジメント財団: European Foundation for Quality Management)によって定義され、所有している。優れた品質マネジメントシステム実現に向けた、五つの実現可能にするための基準(組織がすべきことを網羅)と四つの結果に関する基準(組織が達成すべきことを網羅)がある。JSTQB 訳注) <http://www.efqm.org/> 参照。

**移行テスト (migration testing)**: conversion testing を参照のこと。

ETM **石川ダイアグラム (Ishikawa diagram)**: cause-effect diagram を参照のこと。

**移植性 (portability)**: ある環境から他の環境に移すためのソフトウェア製品の能力。備考: 環境には組織、ハードウェア又はソフトウェアの環境を含めてもよい。[ISO/IEC 9126] JSTQB 訳注) JIS X 0129-1:2003 より引用

F **移植性テスト (portability testing)**: ソフトウェア製品の移植性を判定するテストのプロセス。

ATT

**一貫性 (consistency)**: 均一性、規格統一性、及び、コンポーネントやシステムのドキュメントや構成物間の無矛盾性の度合。[IEEE 610]

**逸脱 (deviation)**: incident を参照のこと。

F **インクリメンタル開発モデル (incremental development model)**: 開発ライフサイクルの一つ。プロジェクトで開発する要件全体を部分的な機能に分割し、サブプロジェクトにおいて、部分機能を積み上げながら連続して開発する方式。各要件に優先順位を付け、優先度の高い順に適切な大きさの機能をリリースする。このライフサイクルモデルには、各サブプロジェクトが、設計、コーディング、テストという

ミニ V 字モデルで進むものもある。

**インクリメンタルテスト(incremental testing)：** 全てのコンポーネント、システムを統合・テストするまで、コンポーネントやシステムを一度に一つ、又は、いくつかを統合・テストする方式。

F **インシデント(incident)：** 発生した事象の中で、調査が必要なもの。[After IEEE 1008]

F **インシデントマネジメント(incident management)：** インシデントを認識、調査、対策、解明するプロセス。インシデントの記録、分類、影響度の識別が必要。[After IEEE 1044]

F **インシデントマネジメントツール(incident management tool)：** インシデントの記録や、状態追跡を支援するツール。ワークフロー志向の機能を持っていることが多く、インシデントの場所特定、修正、再テストをトレース・コントロールしたり、レポートを出力したりすることができる。defect management tool も参照のこと。

F **インシデントレポート(incident report)：** 発生したあらゆるイベント(テストの最中に調査を必要とする事象など)を報告するドキュメント。[After IEEE 829]

F **インシデントログ(Incident logging)：** テスト中などに発生したインシデントに対する詳細な記録。

**インストルメンタ(instrumenter)：** インストルメンテーションを実行するソフトウェアツール。

**インストルメンテーション(instrumentation)：** 実行中にプログラムの振る舞いの情報(たとえば、コードカバレッジの測定)を集めるため、プログラムにコードを追加すること。

**インストールウィザード(installation wizard)：** インストールプロセス中、インストールを誘導するソフトウェア。いろいろな形態の媒体で提供される。インストールを実行し、インストール実施結果情報を出力し、オプション機能のガイダンスを表示するものが多い。

**インストールガイド(installation guide)：** インストールプロセス中、インストールをガイドする説明書で最適の媒体で提供される。マニュアルガイド、段階的な処理手順、インストールウィザード、その他類似の手順記述などの形式をとる。

F **インスペクション(inspection)：** ピアレビューの一種。ドキュメントの目視検査により、欠陥を検出する方法。これにより、たとえば、開発標準の違反や、上位レベルドキュメントへの準拠違反が見つかる。最も公式なレビュー技術なので、必ず、文書化された実施基準に従って進める。[After IEEE 610, IEEE 1028] peer review も参照のこと。

**インスペクションリーダー(inspection leader)：** moderator を参照のこと。

**インスペクタ(inspector)：** reviewer を参照のこと。

**インソーステスト(insourced testing)：** 同じ組織ではない人が、プロジェクトチームと同じ場所で行なうテスト。

**インターフェーステスト(interface testing)：** 統合テストの一種。コンポーネントやシステムのインターフェーステストを実施する。

**インテークテスト(intake test)：** スモークテストの特別な形態。コンポーネントやシステムが、詳細なテストやさらに詳細なテストを開始できるか判定するためのもので、テスト実行フェーズの開始時に行なうことが多い。smoke test も参照のこと。

**インバリッドテスト(invalid testing)：** コンポーネントやシステムが拒否する入力値を使うテスト。error tolerance, negative testing も参照のこと。

## う

**ATA** **Web サイト解析と測定一覧表(WAMMI) (Website Analysis and MeasureMent Inventory (WAMMI))**: エンドユーザの視点から Web サイトのソフトウェア品質を測定するために、アンケートを用いる使用性テストの技法。

**F** **ウォークスルー(walkthrough)**: ドキュメントの著者による段階的なドキュメント内容の説明。情報を集めて、内容の共通理解を確立するために行なう。[Freedman and Weinberg, IEEE 1028] peer review も参照のこと。

**受け入れ(acceptance)**: acceptance testing を参照のこと。

**F-AT** **受け入れ基準(acceptance criteria)**: ユーザ、顧客、その他の認可団体が、コンポーネントやシステムを受け入れる場合、満たさねばならない終了基準。[IEEE 610]

**受け入れテスト(acceptance testing)**: システムが、ユーザのニーズ、要件、ビジネスプロセスを満足するかをチェックするための公式なテスト。このテストにより、システムが受け入れ基準を満たしているかどうかを判定したり、ユーザ、顧客、その他の認可団体がシステムを受け入れるかどうかを判定したりすることができる。[After IEEE 610]

**埋め込み型イテレーティブ開発モデル(embedded iterative development model)**: 開発ライフサイクルサブモデルの一つ。シーケンシャルモデル全体の中の、詳細設計、コーディング、及びテストに対するイテレーティブなアプローチ。この場合、高位の設計ドキュメントはプロジェクト全体に対して準備及び承認されるが、詳細設計、コーディング、及びテストは、イテレーティブに行なわれる。

**ATT** **運用受け入れテスト(operational acceptance testing)**: 受け入れテストフェーズでの運用テスト。通常はオペレータやアドミニストレータスタッフが(シミュレートした)運用環境にて運用面に焦点を当てて行なう。この運用面とは、たとえば、回復性、リソースの振る舞い、設置性、技術的標準適合性などがある。operational testing も参照のこと。

**ATA** **運用性(operability)**: 利用者がソフトウェアの運用及び運用管理を行なうことができるソフトウェア製品的能力。[ISO/IEC 9126] usability も参照のこと。JSTQB 訳注) JIS X 0129-1:2003 より引用

**運用テスト(operational testing)**: 運用環境の中で、コンポーネントやシステムを評価するテスト。[IEEE 610]

**ETM** **運用プロファイリング(operational profiling)**: 運用プロファイルの開発及び実装を行なうプロセス。operational profile も参照のこと。

**ETM** **運用プロファイル(operational profile)**: 着目すべきシステムやコンポーネントを使って行なわれるタスクセットの描写。各タスクは、コンポーネント、システムと利用者のやりとりやそのときに起きる可能性のある事象に基づいて行なうことが多い。タスクは物理的というより論理的であり、複数のマシン上、ある一つのタイミングのものとして実行される。

**運用プロファイルテスト(operational profile testing)**: システム運用(短時間のタスク)モデルと、典型的操作パターンの使用確率を使う統計的なテスト。[Musa]

## え

**F** **影響度分析(impact analysis)**: 特定の要件を変更する前に、開発ドキュメント、テストドキュメント、コンポーネントの各階層が、変更によりどのような影響を受けるか評価すること。

**API テスト(API (Application Programming Interface) testing)：** プロセス、プログラム、又はシステムの間の通信を可能にするコードのテスト。API テストには、多くの場合、エラー処理のロバスト性の検証など否定テストが関連する。interface testing も参照のこと。

**EITP エクストリームプログラミング (XP)(extreme programming (XP))：** アジャイルソフトウェア開発の中で使用されるソフトウェアエンジニアリング方法論。中心となるプラクティスは、ペアプログラミング、徹底したコードレビュー、全てのコードの単体テスト、単純明快なコードなどである。agile software development も参照のこと。

**EITP SPI(SPI)：** Software Process Improvement を参照のこと。

**ATA SUMI(SUMI)：** Software Usability Measurement Inventory を参照のこと。

**N スイッチカバレッジ(N-switch coverage)：** テストスイートが遂行した N+1 遷移のシーケンスのパーセンテージ。[Chow]

**N スイッチテスト(N-switch testing)：** 状態遷移テストの一形式。N+1 遷移の全ての有効なシーケンスを実行するテストケースを設計する。[Chow] state transition testing も参照のこと。

**n ワイズテスト(n-wise testing)：** ブラックボックステスト設計技法の一つ。n 個の入力パラメータの任意のセットを、設定可能な個々の組み合わせの全てで実行するためのテストケースを設計する。combinatorial testing, orthogonal array testing, pairwise testing も参照のこと。

**エミュレータ(emulator)：** ある特定のシステムと同じ入力を受け入れ、同じ出力を作り出す装置、コンピュータプログラム、又はシステム。[IEEE 610] simulator も参照のこと。

**MC/DC(MCDC)：** modified condition decision coverage を参照のこと。

**EITP エモーションナルインテリジェンス(emotional intelligence)：** 自己や他者、グループの感情を理解し、評価し、管理する能力、技能のこと。

**F エラー(error)：** 間違った結果を生み出す人間の行為。[After IEEE 610]

**エラーシーディング(error seeding)：** fault seeding を参照のこと。

**エラーシーディングツール(error seeding tool)：** fault seeding tool を参照のこと。

**F エラー推測(error guessing)：** テスト設計技法の一つ。テスト担当者の経験を駆使し、エラーが起きた場合にどのような欠陥がテスト対象のコンポーネントやシステムの中に存在するかを予想して、その欠陥を検出するテストケースを設計すること。

**ATA**

**エラー耐性(error tolerance)：** 誤ったデータを入力しても、通常運用を続けられるコンポーネントやシステムの能力。[After IEEE 610]

**LCSAJ(LCSAJ)：** Linear Code Sequence And Jump (リニアコードシーケンスアンドジャンプ)の略。慣例として、ソースコードリスト中の行番号で識別する以下の3アイテムからなる。①実行ステートメントの線形シーケンスの開始行、②線形シーケンスの終了行、③線形シーケンスの最後にて制御フローを移された(ジャンプ先の)行。

**LCSAJ カバレッジ(LCSAJ coverage)：** テストスイートが遂行したコンポーネントの LCSAJ (リニアコードシーケンスアンドジャンプ) のパーセンテージ。100%の LCSAJ は、100%のデジジョンカバレッジを意味する。

**LCSAJ テスト(LCSAJ testing)：** ホワイトボックステスト設計技法の一つ。LCSAJ のテストケースを

実行するために設計する。

## お

**ATM** **オープンソースツール(open source tool)**: ソフトウェアツールの一つの種類。全ての潜在的なユーザに、通常はインターネットを介して、ソースコードの形式で提供される。一般的に、ユーザはライセンスの下で許可が与えられ、調査、変更、機能強化、場合によってはソフトウェア配布を行なうために使用する。

**オラクル(oracle)**: test oracle を参照のこと。

## か

**ETM** **回帰回避テスト(regression-averse testing)**: ソフトウェアが前の状態に戻るもののリスクをマネジメントするためにさまざまな技法を使用するテスト。たとえば、再利用可能なテストウェアの設計や一つ以上のテストレベルでのテストの大幅な自動化などがある。

**F** **回帰テスト(regression testing)**: 変更により、ソフトウェアの未変更部分に欠陥が新たに入り込んだり、発現したりしないことを確認するため、変更実施後、すでにテスト済みのプログラムに対して実行するテスト。ソフトウェアや、実行環境が変わる度に実施する。

**EITP** **開始 (IDEAL)(initiating (IDEAL))**: IDEAL モデルの 1 フェーズであり、改善活動が成功するための基盤を具体化する。開始フェーズは、活動背景の確認、支援体制及び活動体制の構築によって構成される。IDEAL も参照のこと。JSTQB 訳注) IDEAL のフェーズ名称は「CMMI V1.2 モデル - 開発のための - 公式日本語翻訳版」の定義を使用。

**F** **開始基準(entry criteria)**: 定義したタスク(たとえば、テストフェーズ)の開始を許可するための基準となる一般・特定の条件のセット。この目的は、開始基準を満足させるための労力に比べ、はるかに多くの(無駄な)労力を投入しないとタスクが始まらない状況を防ぐことである。[Gilb and Graham]

**開始点(entry point)**: プロセスの開始を意図した実行ステートメント、若しくはポイントとして定義したプロセスステップ。

**ATT** **解析性(analyzability)**: ソフトウェアにある欠陥の診断又は故障原因の追及、及びソフトウェアの修正箇所の識別を行なうためのソフトウェア製品の能力。[ISO/IEC 9126] maintainability を参照のこと。JSTQB 訳注) JIS X 0129-1:2003 より引用

**開発テスト(development testing)**: コンポーネントやシステムの実装中に実施する公式又は非公式のテスト。通常は、開発担当者の製品開発環境で実行する。[After IEEE 610]

**回復性(recoverability)**: 故障時に、指定された達成水準を再確立し、直接に影響を受けたデータを回復するソフトウェア製品の能力。[ISO/IEC 9126] reliability も参照のこと。JSTQB 訳注) JIS X 0129-1:2003 より引用

**ATT** **回復性テスト(recoverability testing)**: ソフトウェア製品の回復性を判定するテストのプロセス。reliability testing も参照のこと。

**回復テスト(recovery testing)**: recoverability testing を参照のこと。

**改良条件判定カバレッジ(MC/DC)(modified condition decision coverage)**: 判定結果に対して独立に影響する全単一条件結果のうち、テストスイートが遂行した条件結果のパーセンテージ。



100%の改良条件判定カバレッジ(MC/DC)は、100%の判定条件カバレッジを意味する。

**改良条件判定テスト(modified condition decision testing)：** ホワイトボックステスト設計技法の一つ。判定結果に対して独立に影響する単一条件結果を実行するテストケースを設計する。

**改良複合条件カバレッジ(modified multiple condition coverage)：** modified condition decision coverage を参照のこと。

**改良複合条件テスト(modified multiple condition testing)：** modified condition decision testing を参照のこと。

EITP **学習 (IDEAL)(learning (IDEAL))：** IDEAL モデルの 1 フェーズであり、経験から学習し、将来の新しいプロセスと技術を適用するために自らの能力を向上させる。学習フェーズは、分析、妥当性確認、及び以降の活動提案によって構成される。IDEAL も参照のこと。JSTQB 訳注) IDEAL のフェーズ名称は「CMMI V1.2 モデル - 開発のための - 公式日本語翻訳版」の定義を使用。

**拡張性(scalability)：** 増加する負荷に応じてソフトウェア製品をアップグレードできる能力。[After Gerrard]

**拡張性テスト(scalability testing)：** ソフトウェア製品の拡張性を判定するテスト。

F **確認テスト(confirmation testing)：** 修正が成功したかを検証するために、前回不合格に終わったテストケースを再実行するテスト。

EITP **確立 (IDEAL)(establishing (IDEAL))：** IDEAL モデルの 1 フェーズであり、組織が、計画された目標にどのように到達するかを特定する。確立フェーズは、優先順位の決定、アプローチの開発、行動計画の活動によって構成される。IDEAL も参照のこと。JSTQB 訳注) IDEAL のフェーズ名称は「CMMI V1.2 モデル - 開発のための - 公式日本語翻訳版」の定義を使用。

**カスタムソフトウェア(custom software)：** bespoke software を参照のこと。

ATM **カスタムツール(custom tool)：** 特定のユーザ又は顧客に特別に開発されたソフトウェアツール。

**カバレッジ(coverage)：** 特定のカバレッジアイテムをテストスイートが遂行した度合。パーセンテージで表す。

**カバレッジアイテム(coverage item)：** テストカバレッジの基礎となる実体や属性。たとえば、同値分割やステートメント。

**カバレッジ測定ツール(coverage measurement tool)：** coverage tool を参照のこと。

F **カバレッジツール(coverage tool)：** テストスイートが遂行したステートメントや分岐等の構成要素の客観的な測定結果を提供するツール。

**カバレッジ分析(coverage analysis)：** 特定のカバレッジアイテムのうち、どれだけ実行したかをテスト実行中に計測すること。追加テストが必要か、必要ならどのテストを追加するか判定用に事前に定めた判定基準に照らして実施する。

**可用性(availability)：** 使用する際にコンポーネントやシステムが稼動し、利用可能な度合。パーセンテージで表すことが多い。[IEEE 610]

ATT **環境適応性(adaptability)：** ソフトウェアにあらかじめ用意された以外の付加的な作業又は手段なしに、指定された異なる環境にソフトウェアを適応させるためのソフトウェア製品の能力。[ISO/IEC 9126] portability を参照のこと。JSTQB 訳注) JIS X 0129-1:2003 より引用

**ATT 頑健性[堅牢性](robustness)：** 不正な入力や過負荷の環境条件の中でも、コンポーネント又はシステムが正しく機能できる度合。[IEEE 610] error tolerance, fault tolerance も参照のこと。

**ATM 監査(audit)：** 標準、ガイドライン、仕様、客観的な基準に基づいた手続きに遵守することを確認し、以下の(1)～(3)を規定したドキュメント等を含む、ソフトウェアプロダクトや開発プロセスの独立した評価。

(1) 開発するプロダクトの形式や内容

(2) プロダクトの開発プロセス

(3) 標準やガイドライン遵守の測定方法 [IEEE 1028]

**監査証跡(audit trail)：** プロセスの出力をスタート地点とし、プロセスを遡って、プロセスへの元の入力(たとえば、データ)までたどることができるパス。これにより、欠陥分析が容易になり、プロセス監査を実施できる。[After TMap]

**完全テスト(complete testing)：** exhaustive testing を参照のこと。

**ETM 管理図(control chart)：** 統計に基づくプロセス管理ツール。プロセスを監視し、統計的に管理されているかどうかを確認する。プロセスの平均値と、上方管理限界及び下方管理限界(最大値及び最小値)を図示する。

## き

**ATM 偽陰性結果(FALSE-negative result)：** false-pass result を参照のこと。

**机上チェック(desk checking)：** 手動の実行シミュレーションによるソフトウェア又は仕様のテスト。static testing も参照のこと。

**擬似ランダム(pseudo-random)：** ランダムであるようにみえるが、実際にはいくつか事前に決められた順序に従って生成される一揃いのもの。

**既製ソフトウェア(off-the-shelf software)：** 一般の市場用(不特定多数のユーザ用)に開発したソフトウェア製品。全く同じものを多数の顧客に提供する。

**期待結果(expected outcome)：** expected result を参照のこと。

**期待結果(expected result)：** 特定の条件下で、仕様や他の情報から期待できるコンポーネントやシステムの振る舞い。

**規程テスト(regulation testing)：** compliance testing を参照のこと。

**機能性(functionality)：** ソフトウェアが、指定された条件の下で利用されるときに、明示的及び暗示的必要性に合致する機能を提供するソフトウェア製品の能力。[ISO/IEC 9126] JSTQB 訳注) JIS X 0129-1:2003 より引用

**機能性テスト(functionality testing)：** ソフトウェア製品の機能性を判定するためのテストのプロセス。

**F 機能テスト(functional testing)：** コンポーネントやシステムの機能仕様の分析に基づいて実施するテスト。black box testing も参照のこと。

**機能テスト設計技法(functional test design technique)：** コンポーネントやシステムの内部構造を参照せず、機能仕様の分析に基づいてテストケースを設計・選択すること。black box test design technique も参照のこと。

**機能統合(functional integration)：** システム統合法のアプローチの一つ。早い時期に、基本機能を動作させるために、コンポーネントやシステムを結合すること。integration testing も参照のこと。

F **機能要件(functional requirement)：** コンポーネントやシステムが実行すべき機能を特定した要件。[IEEE 610]

**キーパフォーマンスインジケータ(key performance indicator)：** performance indicator を参照のこと。

**基本比較テスト(elementary comparison testing)：** ブラックボックステスト設計技法の一つ。改良条件判定カバレッジ(MC/DC)の概念を活用した入力組み合わせを実行するテストケースを設計する。[TMap]

**キャスト(CAST)：** Computer Aided Software Testing (コンピュータ支援ソフトウェアテスト)の頭字語。test automation も参照のこと。

ATT **キャプチャ/プレイバックツール(capture/playback tool)：** テスト実行ツールの一つ。手動テスト中の入力を記録させ、自動テストのスクリプトを作成して、後に実行(再現)させるもの。自動回帰テストで利用することが多い。

**キャプチャ/リプレイツール(capture/replay tool)：** capture/playback tool を参照のこと。

**QFD(QFD)：** quality function deployment を参照のこと。

EITP **共依存的振る舞い(codependent behavior)：** 他人に対する過度の感情的及び精神的依存。特に、人が望ましくない振る舞いをしている場合にそれを変えさせようとせず、その支援しか行なわないという精神的状態。たとえばソフトウェアテストにおいては、テストへの移行が遅延することについてクレームを付けはするものの、足りないテスト時間を補うために超過勤務する(ある意味)「ヒーロー」的な仕事を楽しんでおり、その結果スケジュールの遅延を増長することがある。

**境界値(boundary value)：** 同値分割した領域の端、あるいは端のどちらか側で最小の増加的距離にある入力値又は出力値。たとえばある範囲の最小値又は最大値。

**境界値カバレッジ(boundary value coverage)：** テストスイートによって遂行された境界値のパーセンテージ。

**境界値テスト(boundary value testing)：** boundary value analysis を参照のこと。

F **境界値分析(boundary value analysis)：** ブラックボックステスト設計技法の一つ。境界値に基づいてテストケースを設計する。boundary value も参照のこと。

ATA **偽陽性結果(FALSE-positive result)：** false-fail result を参照のこと。

ATT **共存性(co-existence)：** 共通の資源を共有する共通の環境の中で、他の独立したソフトウェアと共存するためのソフトウェア製品の能力。[ISO/IEC 9126] portability も参照のこと。JSTQB 訳注) JIS X 0129-1:2003 より引用

ATT **記録再生ツール(record/playback tool)：** capture/playback tool を参照のこと。

**記録者(recorder)：** scribe を参照のこと。

F **キーワード駆動テスト(keyword-driven testing)：** テストスクリプト記述技術の一つ。テストデータと期待結果だけでなく、テスト対象アプリケーションに関係するキーワードを含んだデータファイルを使う。ATA **キーワードは、テストの制御スクリプトが呼び出す特別な補助スクリプトが解釈する。data-driven testing** ATT

も参照のこと。

ATT **近隣統合テスト(neighborhood integration testing)**: 統合テストの形態の一つ。特定のノードに接続される全てのノードが、テストの基底となる。

## く

ATA **具体的テストケース(concrete test case)**: low level test case を参照のこと。

ATA **組み合わせテスト(combinatorial testing)**: 事前に定義されたレベルのカバレッジを達成するために、適切に組み合わせたテストのサブセットを識別する手段。複数のパラメータを持つオブジェクトをテストする場合、それぞれのパラメータにも複数の値があり、予定された時間内で実行できるテスト数よりも多くの組み合わせが発生するため、組み合わせテストを使用する。classification tree method, n-wise testing, pairwise testing, orthogonal array testing も参照のこと。

**クラシフィケーションツリー(classification tree)**: 階層的な順番で同値分割を示したツリーであり、クラシフィケーションツリー法でテストケースを作成するときに使う。classification tree method も参照のこと。

ATA **クラシフィケーションツリー法(classification tree method)**: ブラックボックステスト設計技法の一つ。クラシフィケーションツリーを使って入力ドメインや出力ドメインの代表値の組み合わせを設計し、テストケースを記述する。[Grochtmann] combinatorial testing も参照のこと。

**ガラスボックステスト(glass box testing)**: white-box testing を参照のこと。

**クリアボックステスト(clear-box testing)**: white-box testing を参照のこと。

ATM  
EITP **クリティカルテストプロセス(Critical Testing Processes)**: 12 の重要なプロセスから構成されるテストプロセス改善のためのコンテンツベースドモデル。企業の利益や評価に影響を与えるようなミッションクリティカルなプロセスやコンピタンスの状況を同僚や経営陣が判断できるよう高度に可視化したプロセスを含む。content-based model も参照のこと。

## け

ATA **経験ベースの技法(experience-based technique)**: experience-based test design technique を参照のこと。

**経験ベースのテスト(experience-based testing)**: テスト担当者の経験・知識・直感をベースに行なうテスト。

F  
ATA **経験ベースのテスト設計技法(experience-based test design technique)**: テスト担当の経験・知識・直感をベースにテストケースを導き出したり選択したりする技法。

**ケース(CASE)**: Computer Aided Software Engineering (コンピュータ支援ソフトウェア開発)の頭字語。

**結果(outcome)**: result を参照のこと。

**結果(result)**: テスト実行後の成果。画面への出力、データの変化、レポート、外部へ送信するメッセージを含む。actual result, expected result も参照のこと。

F ATM	<p><b>欠陥(defect)：</b> コンポーネント又はシステムに要求された機能が実現できない原因となる、コンポーネント又はシステムに含まれる不備。たとえば、不正なステートメント又はデータ定義。実行中に欠陥に遭遇した場合、コンポーネント又はシステムの故障を引き起こす。</p> <p><b>欠陥カテゴリ(defect category)：</b> defect type を参照のこと。</p>
EITP	<p><b>欠陥検出率(Defect Detection Percentage (DDP))：</b> あるテストレベルで見つけた欠陥の数をそのテストレベル、及び、以降のテストレベルで見つけた欠陥の総数で除算した値。escaped defects も参照のこと。</p>
ATM	<p><b>欠陥選別委員会(defect triage committee)：</b> defect management committee を参照のこと。</p> <p><b>欠陥追跡ツール(defect tracking tool)：</b> defect management tool を参照のこと。</p> <p><b>欠陥のタイプ(defect type)：</b> 欠陥を分類する要素。欠陥の分類は、次を含むさまざまな検討事項に基づいて識別できる。(ただし、これだけではない)</p> <ul style="list-style-type: none"> <li>・欠陥が作り込まれたフェーズ又は開発活動。たとえば、仕様エラー又はコーディングエラー。</li> <li>・欠陥の特性。たとえば、「off-by-one」欠陥。</li> <li>・不正確性。たとえば、不正確な関係演算子、プログラム言語の構文エラー、無効な仮定。</li> <li>・性能問題。たとえば、過剰な実行時間、不十分な可用性。</li> </ul>
F-AT ATA	<p><b>欠陥分類法(defect taxonomy)：</b> 欠陥の分類を再現しやすくするように設計された(階層的)カテゴリの体系。</p>
ATA	<p><b>欠陥ベースの技法(defect-based technique)：</b> defect-based test design technique を参照のこと。</p>
ATA	<p><b>欠陥ベースのテスト設計技法(defect-based test design technique)：</b> 一つ以上の欠陥のタイプをターゲットにしたテストケースを導いたり、選び出す技法。テストケースは特定の欠陥のタイプから開発していく。defect taxonomy も参照のこと。</p> <p><b>欠陥マスキング(defect masking)：</b> 一つの欠陥が他の欠陥の検出を妨げる現象。[After IEEE 610]</p> <p><b>欠陥マネジメント(defect management)：</b> 認識、調査、行動、及び、欠陥の処置の手順。欠陥の記録、分類、及び、影響の識別を含む。[After IEEE 1044]</p>
ATM	<p><b>欠陥マネジメント委員会(defect management committee)：</b> クロスファンクショナルなステークホルダのチーム。報告された欠陥の初期検出から最終的な解決(欠陥除去、欠陥除去の延期、報告取り消し)に至るまでを管理する。構成管理委員会と同じチームとなることがある。configuration control board も参照のこと。</p> <p><b>欠陥マネジメントツール(defect management tool)：</b> 欠陥及び変更の記録と状態追跡を容易にするツール。多くの場合、欠陥の割り振り、訂正、再テストを追跡、コントロールするために、ワークフロー指向の機能を搭載している他に、レポート機能も持っている。incident management tool も参照のこと。</p>
F	<p><b>欠陥密度(defect density)：</b> コンポーネント又はシステムの中で識別された欠陥の数をコンポーネント又はシステムのサイズで割った値。(サイズを表す標準的な尺度には、コード行数、クラス数、又はファンクションポイント数がある)</p> <p><b>欠陥レポート(defect report)：</b> コンポーネント又はシステムに要求された機能が実現できない原因となる、コンポーネント又はシステムに含まれる不備を報告するドキュメント。[After IEEE 829]</p>

- EITP **原因結果グラフ(cause-effect graph)**: 入力、刺激(原因)、関連する出力(結果)を図式表現したもの。テストケースの設計で利用できる。
- EITP **原因結果グラフ(cause-effect diagram)**: 問題のさまざまな根本的原因の相互関係を整理し示すための図表現。(潜在的な)欠陥あるいは故障をルートノードとする水平的なツリー構造を用いて、その欠陥あるいは故障の原因をカテゴリ、サブカテゴリに整理する。[After Juran]
- ATA **原因結果グラフ法(cause-effect graphing)**: ブラックボックステスト設計技法の一つ。原因結果グラフからテストケースを設計する。[BS7925/2]
- 原因結果デシジョンテーブル(cause-effect decision table)**: decision table を参照のこと。
- 原因結果分析(cause-effect analysis)**: cause-effect graphing を参照のこと。
- EITP **原因分析(causal analysis)**: 欠陥の根本的原因を特定するための分析。[CMMI]
- F **検証(verification)**: 客観的証拠を提示することによって、規定要求事項が満たされていることを確認すること。[ISO 9000] JSTQB 訳注) JIS Q 9000:2006 より引用

## こ

- ATA **高位レベルテストケース(high level test case)**: 具体的な(実行レベルの)入力値や予測結果を使わないテストケース。論理演算子は使用するが、値のインスタンスは未定義や使用不可であるといった状態にある。low level test case も参照のこと。
- 合格/失敗基準(pass/fail criteria)**: テストアイテム(機能)やフィーチャが、テストに合格したか失敗したかを判定するための判定規則。[IEEE 829]
- F **攻撃(attack)**: 品質を評価するために、直接的に焦点を定めて行なう試み。特定の故障が発生するような強制を試みることによって、テスト対象の品質、特に信頼性を評価する。negative testing も参照のこと。
- 攻撃ベースドテスト(attack-based testing)**: 経験ベースのテスト技法で、ソフトウェア攻撃を使用して、故障、特にセキュリティ関連の故障を引き起こさせる。attack も参照のこと。
- F **公式レビュー(formal review)**: 文書化された手順と要件に特徴付けられるレビュー。たとえば、インスペクション。
- 工場受け入れテスト(factory acceptance testing)**: コンポーネント又はシステムが要件を満たすかどうかを確認するために、製品開発の場所で供給者組織の従業員により実行される受け入れテスト。要件には、ソフトウェア要件だけでなく、ハードウェア要件も含む。alpha testing も参照のこと。
- 構成(configuration)**: コンポーネントやシステムの構成要素の数、性質、相互連結によって定義される構成。
- F-AT **構成アイテム(configuration item)**: ハードウェア、ソフトウェア、又は、両方の集合体。構成管理の対象であり、構成管理プロセスでは、一つの実体として扱う。[IEEE 610]
- 構成監査(configuration auditing)**: 構成アイテムのライブラリの内容をチェックする役割。たとえば、標準適合性をチェックするなど。[IEEE 610]
- F  
F-AT **構成管理(configuration management)**: 技術的かつ管理的な指示と監視を適用する規範。この規範の目的は

- ・構成アイテムの特性を機能的、物理的に識別・文書化すること
- ・特性に対する変更をコントロールすること
- ・処理の変更と実装の状況を記録し、報告すること
- ・特定の要求への整合を実証すること

である。[IEEE 610]

F **構成管理ツール(configuration management tool):** 構成アイテムの識別やコントロール(変更やバージョンの状態や構成アイテムをまとめたベースラインのリリース)を支援するツール。

**構成コントロール(configuration control):** 構成管理の一つの要素。構成アイテムを公式に定義後、構成アイテムに対する変更を、評価、調整、承認・否認、実施すること。[IEEE 610]

**構成コントロール委員会(configuration control board (CCB)):** 構成アイテムに対する変更提案を評価、承認・否認することに責任を持つグループ。承認した変更の実施にも責任を負う。[IEEE 610]

**構成識別(configuration identification):** 構成管理の一つの要素。システムの構成アイテムを選択し、構成アイテムの機能的、物理的特性を技術ドキュメントに記録する。[IEEE 610]

**構成テスト(configuration testing):** portability testing を参照のこと。

**構造化ウォークスルー(structured walkthrough):** walkthrough を参照のこと。

**構造カバレッジ(structural coverage):** コンポーネント又はシステムの内部構造に基づいたカバレッジの測定値。

F **構造テスト(structural testing):** white-box testing を参照のこと。

**構造テスト設計技法(structural test design technique):** white-box test design technique を参照のこと。

F **構造ベーステスト(structure-based testing):** white-box testing を参照のこと。

ATT **構造ベースの技法(structure-based technique):** white-box test design technique を参照のこと。

**構造ベースのテスト設計技法(structure-based test design technique):** white-box test design technique を参照のこと。

EITP **行動 (IDEAL)(acting (IDEAL)):** IDEAL モデルの 1 フェーズであり、改善を展開し、実践し、組織全体に導入する。行動フェーズでは、解決策の創造、試行/試験的導入、改良、実行の活動によって構成される。IDEAL も参照のこと。JSTQB 訳注) IDEAL のフェーズ名称は「CMMI V1.2 モデル - 開発のための - 公式日本語翻訳版」の定義を使用。

**合目的性(suitability):** 指定された作業及び利用者の具体目標に対して適切な機能の集合を提供するソフトウェア製品の能力。[ISO/IEC 9126] functionality も参照のこと。JSTQB 訳注) JIS X 0129-1:2003 より引用

ATA **合目的性テスト(suitability testing):** ソフトウェア製品の合目的性を判定するためのテストのプロセス。

ATM **効率性(efficiency):**

ATT (1) 明示的な条件の下で、使用する資源の量に対比して適切な性能を提供するソフトウェア製品の能力。[ISO/IEC 9126] JSTQB 訳注) JIS X 0129-1:2003 より引用

(2) 使用するリソースの量に対比して、意図した結果を生成するプロセスの能力。

	<b>効率性テスト(efficiency testing)：</b> ソフトウェア製品の効率を測定するテストのプロセス。
EITP	<b>交流分析(transactional analysis)：</b> 人と人の心の内の交流分析。交流は刺激とその反応により定義される。交流は、人と人の間と、人の心の中にある自我の状態(個性の部分)の間に発生する。
	<b>互換性テスト(compatibility testing)：</b> interoperability testing を参照のこと。
F ATM	<b>故障(failure)：</b> コンポーネントやシステムが、期待した機能、サービス、結果から逸脱すること。[After Fenton]
	<b>故障モード(failure mode)：</b> 物理的又は機能的な故障の兆候。たとえば、故障モードのシステムは、遅い運用、間違った出力、又は実行の完全な打ち切りなどで特徴付けられる。[IEEE 610]
EITP	<b>故障モード影響解析(Failure Mode and Effect Analysis (FMEA))：</b> リスクの識別と分析(故障モードの識別と、発生防止策の試み)を行なう体系的なアプローチ。Failure Mode, Effects, and Criticality Analysis (FMECA)も参照のこと。
	<b>故障モード影響・致命度解析(Failure Mode, Effects, and Criticality Analysis (FMECA))：</b> 基本的な FMEA に加え、致命度解析を含んで拡張したもの。結果の重要性に対して、故障モードの発生確率を図で表現したものを使用する。これにより、他と比べて高い発生確率と、結果の重要性を持った故障モードが強調されるため、最大の価値を生み出す方向に向かう改善活動ができる。Failure Mode and Effect Analysis (FMEA)も参照のこと。
F	<b>故障率(failure rate)：</b> 測定単位に発生したあるカテゴリの故障数の率。たとえば、単位時間あたりの故障数、トランザクション数あたりの故障数、コンピュータの運用回数あたりの故障数。[IEEE 610]
F	<b>COTS(COTS)：</b> Commercial Off-The-Shelf software (市販ソフトウェア)の頭字語。off-the-shelf software を参照のこと。
	<b>コード(code)：</b> プログラミング言語、又は機械語、コンパイラやその他の変換装置によってアウトプットされたコンピュータ命令とデータ定義を表現したもの。[IEEE 610]
	<b>コードアナライザ(code analyzer)：</b> static code analyzer を参照のこと。
F	<b>コードカバレッジ(code coverage)：</b> テストスイートが、ソフトウェアのどの部分を実行(カバー)し、どの部分が未実行かを判定する分析手法。たとえば、ステートメントカバレッジ、デシジョンカバレッジ、条件カバレッジ。
	<b>コードベースドテスト(code-based testing)：</b> white-box testing を参照のこと。
EITP	<b>コーポレートダッシュボード(corporate dashboard)：</b> 企業業績の状況に関するダッシュボードスタイルの表現。balanced scorecard, dashboard も参照のこと。
EITP	<b>ゴールクエスチョンメトリック(Goal Question Metric)：</b> 三つのレベル、すなわち、概念的なレベル(ゴール)、運用上のレベル(クエスチョン)、定量的なレベル(メトリック)のモデルを用いたソフトウェア測定のアプローチ。
	<b>コールグラフ(call graph)：</b> プログラムにおけるサブルーチン間の呼び出し関係の抽象的表現。
ETM	<b>コンサルテーションベースのテスト(consultative testing)：</b> 外部テストチームの適切な専門家(テクノロジー専門家又はビジネスドメインの専門家)のアドバイス及びガイダンスに基づいて実行されるテスト。
	<b>コンテンツ参照モデル(content reference model)：</b> content-based model を参照のこと。
EITP	<b>コンテンツベースドモデル(content-based model)：</b> 優れたエンジニアリングプラクティス(たとえば、



テストプラクティス)に関する詳細な説明を提供するプロセスモデル。

F **コンパイラ(compiler)**: 高度な命令語で表現されたプログラムを、等価の機械語に翻訳するソフトウェアツール。[IEEE 610]

**コンフィデンステスト(confidence test)**: smoke test を参照のこと。

**コンポーネント(component)**: 独立してテストできるソフトウェアの最小単位。

**コンポーネント仕様(component specification)**: コンポーネントに関する機能(特定の条件下で、特定の入力値に対する出力値を観点としたもの)、及び、要求される非機能的な振る舞い(たとえば、資源効率性)の記述。

F **コンポーネントテスト(component testing)**: 個々のソフトウェアコンポーネントのテスト。[After IEEE 610]

**コンポーネント統合テスト(component integration testing)**: 統合したコンポーネント間のインターフェースや相互作用の欠陥を検出するためのテスト。

ATM **根本原因(root cause)**: 欠陥の発生源のことで、根本原因が除去されると、欠陥が削減又は除去される。[CMMI]

ATA **根本原因分析(root cause analysis)**: 欠陥の根本原因の識別を目的とした分析技法。根本原因に是正を行なうことで、欠陥再発を最小化することが期待できる。

## さ

**再開基準(resumption criteria)**: テストの中断後、テスト活動の全て又は一部を再開するために使用される基準。

**再開要件(resumption requirements)**: 中断後のテスト再開時に、繰り返す必要のある定義済みのテスト活動のセット。[After IEEE 829]

**サイクロマティック数(cyclomatic number)**: cyclomatic complexity を参照のこと。

ATT **サイクロマティック複雑度(cyclomatic complexity)**: プログラムの中を通る線形的で独立したパスの最大数。 $L - N + 2P$  で計算できる。

-L=グラフ中のエッジ/リンクの数

-N=グラフ中のノードの数

-P=グラフの繋がっていない部分の数(たとえば、呼ばれるグラフ又はサブルーチンの数 [After McCabe])

F **再テスト(re-testing)**: confirmation testing を参照のこと。

**サイト受け入れテスト(site acceptance testing)**: ユーザや顧客が自分のサイトで実施する受け入れテスト。コンポーネントやシステムが、ユーザや顧客のニーズを満たしているかどうか、ユーザや顧客のビジネスプロセスに適合するかを判定するために実施する。通常、ソフトウェアだけでなく、ハードウェアも含める。

**サニティテスト(sanity test)**: smoke test を参照のこと。

**サブパス(subpath)**: コンポーネント内の一連の実行可能なステートメント。

**3 点見積り(three point estimation)**: テスト見積り方法の一つ。見積り対象案件の「楽観値」、「悲観値」、及び「最頻値」の見積りを使用して、結果となる見積りの確実度合を定義する。

## し

ATM  
EITP

**CMMI(CMMI)**: Capability Maturity Model Integration を参照のこと。

**資格認定(qualification)**: 特定の要件を満たした能力を明らかにするプロセス。注)有資格(qualified)という用語は相当する能力を有する場合に使われる。[ISO 9000]

**時間的振る舞い(time behavior)**: performance を参照のこと。

EITP

**GQM(GQM)**: Goal Question Metric を参照のこと。

**資源効率性(resource utilization)**: 明示的な条件の下で、ソフトウェアの機能を実行する際の、資源の量及び資源の種類を適切に使用するソフトウェア製品の能力。たとえば、プログラムが使うメインメモリとセカンダリメモリ量、必要な一時ファイルやオーバーフローファイルの大きさ。[After ISO/IEC 9126] efficiency も参照のこと。JSTQB 訳注)JIS X 0129-1:2003 より引用

ATT

**資源効率性テスト(resource utilization testing)**: ソフトウェア製品の資源効率性を判定するためのテストのプロセス。efficiency testing も参照のこと。

ATT

**試験性(testability)**: 修正したソフトウェアの妥当性確認ができるソフトウェア製品の能力。[ISO/IEC 9126] maintainability も参照のこと。JSTQB 訳注)JIS X 0129-1:2003 より引用

**試験性レビュー(testability review)**: テストプロセスの入力ドキュメントとして十分な品質レベルにあるかどうかを判定するため、テストベースを詳細にチェックすること。[After TMap]

**事後条件(postcondition)**: テストやテスト手順の実行後に満足すべき、環境と状態の条件。

**システム(system)**: 特定の機能や、機能の組み合わせを実現するために組織化したコンポーネントのセット。[IEEE 610]

**システムオブシステムズ(system of systems)**: 相互接続したドメインと複数レベルのネットワークに組み込まれた複数の分散システムや異機種環境のこと。通常、共通的な管理構造を持たずに、広範囲に専門領域間をまたがる共通の問題や目的に対処する。

F

**システムテスト(system testing)**: 統合されたシステムが、特定の要件を満たすことを実証するためのテストのプロセス。[Hetzel]

**システム統合テスト(system integration testing)**: システムやパッケージを統合して行なうテスト。外部機構とのインターフェース(たとえば、電子データ交換やインターネット)をテストすること。

**事前条件(precondition)**: 特定のテストやテスト手順でコンポーネントやシステムを実行する前に満足すべき、環境と状態の条件。

**実行可能パス(feasible path)**: 実行するための、入力値のセットと事前条件が存在するパス。

**実行結果(actual outcome)**: actual result を参照のこと。

**実行結果(actual result)**: コンポーネントやシステムをテストしたときに、生じた／観察された振る舞い。

**実行後比較(post-execution comparison)**: ソフトウェアの実行後に、実行結果と期待結果を比較すること。

	<b>実行ステートメント(executable statement)：</b> コンパイル時にオブジェクトコードに翻訳され、プログラムが走るときに手順に沿って実行されて、データに対して動作を行なうステートメント。
	<b>実行不可能パス(infeasible path)：</b> 入力値をどのように組み合わせても実行できないパス。
	<b>失敗(fail)：</b> 実行結果が期待結果と一致しない状態。この場合、テストは「失敗」とみなす。
ATM EITP	<b>CTP(CTP)：</b> Critical Testing Processes を参照のこと。
	<b>自動化テストウェア(automated testware)：</b> ツールスクリプトのような自動化されたテストの中で使われるテストウェア。
	<b>シナリオテスト(scenario testing)：</b> use case testing を参照のこと。
F	<b>市販ソフトウェア(COTS)(Commercial Off-The-Shelf software)：</b> off-the-shelf software を参照のこと。
EITP	<b>指標(indicator)：</b> 他の測定値の見積り又は予測に使うことができる測定値。[ISO/IEC 14598]
	<b>シミュレーション(simulation)：</b> 物理的なシステム、あるいは、抽象的なシステムの代表的な動作特性を他のシステムで模倣すること。[ISO/IEC 2382/1]
	<b>シミュレータ(simulator)：</b> テストで使われる装置、コンピュータプログラム、システムで、ある入力の設定に対し、特定のシステムのような振る舞いや動作をするもの。[After IEEE 610, DO178b] emulator も参照のこと。
ETM	<b>収束メトリック(convergence metric)：</b> 定義した基準に向けて進捗していることを示すメトリック。たとえば、実行されたテストの合計数が、実行が計画されているテストの総数に収束することを示すメトリック。
ATA	<b>習得性(learnability)：</b> ソフトウェアの適用を利用者が習得できるソフトウェア製品の能力。[ISO/IEC 9126] usability も参照のこと。JSTQB 訳注) JIS X 0129-1:2003 より引用
ETM	<b>シューハート管理図(Shewhart chart)：</b> control chart を参照のこと。
EITP	<b>重要成功要因(critical success factor)：</b> 組織やプロジェクトが使命を全うするのに必要な要素。成功を確実なものにするのに不可欠な要因や活動。
ATM	<b>重要度(severity)：</b> コンポーネントやシステムの開発、また運用に対し欠陥が与える影響の度合。[After IEEE 610]
	<b>終了基準(completion criteria)：</b> exit criteria を参照のこと。
F ATM ATA	<b>終了基準(exit criteria)：</b> あるプロセスを公式に完了させるため、ステークホルダーが承認した一般・特定条件のセット。終了基準の目的は、未完了部分のあるタスクが、完了とみなされるのを防ぐことにある。あらかじめ計画していた終了基準を、テスト完了時の報告に利用する。[After Gilb and Graham]
	<b>終了点(exit point)：</b> プロセスを終了させることを意図した実行可能なステートメント、若しくはポイントとして定義したプロセスステップ。
	<b>出力(output)：</b> コンポーネントが書き込む変数(コンポーネントの内部、外部に格納される)。
	<b>出力値(output value)：</b> 出力のインスタンス。output も参照のこと。

**出力ドメイン(output domain)：** 選択できる有効な出力値のセット。domainも参照のこと。

**仕様(specification)：** コンポーネントやシステムの要件、設計、振る舞い、その他の特性を(理想としては完全で的確、かつ検証可能な方法で)詳細に記述したドキュメントであり、記述内容が満足できるものであることを明らかにする手順を示したものも多い。[After IEEE 610]

**障害許容性(fault tolerance)：** ソフトウェアの障害部分を実行した場合、又は仕様化されたインターフェース条件に違反が発生した場合に、指定された達成水準を維持するソフトウェア製品の能力。[ISO/IEC 9126] reliability, robustnessも参照のこと。JSTQB 訳注) JIS X 0129-1:2003 より引用

**小規模統合テスト(integration testing in the small)：** component integration testing を参照のこと。

**条件(condition)：** 真又は偽として評価することのできる論理的表現。たとえば、 $A > B$ 。condition testingも参照のこと。

**条件カバレッジ(condition coverage)：** テストスイートが遂行した条件結果のパーセンテージ。条件カバレッジを 100%にするには、各判定ステートメントの全ての単一条件に対し、真と偽をテストする必要がある。

**条件組み合わせカバレッジ(condition combination coverage)：** multiple condition coverage を参照のこと。

**条件組み合わせテスト(condition combination testing)：** multiple condition testing を参照のこと。

**条件結果(condition outcome)：** 真又は偽になる条件の評価。

ATT **条件テスト(condition testing)：** ホワイトボックステスト設計技法の一つ。条件結果を実行するテストケースを設計する。

**条件判定カバレッジ(condition determination coverage)：** modified condition decision coverage を参照のこと。

**条件判定テスト(condition determination testing)：** modified condition decision testing を参照のこと。

**使用性(usability)：** 指定された条件の下で利用するとき、理解、習得、利用でき、利用者にとって魅力的であるソフトウェア製品の能力。[ISO/IEC 9126] JSTQB 訳注) JIS X 0129-1:2003 より引用

F  
F-AT  
ATA **使用性テスト(usability testing)：** ソフトウェア製品が、特定の条件下で理解され、学びやすく、使用しやすく、ユーザにとって魅力的である能力を判定するためのテスト。[After ISO/IEC 9126]

**状態遷移(state transition)：** コンポーネントやシステムにおいて、二つの状態の間を遷移すること。

**状態遷移図(state diagram)：** コンポーネント又はシステムが取りうる状態を示し、ある状態から他への状態の変化の原因となる、(又は)その結果として生ずる、イベントや状況を表すダイアグラム。[IEEE 610]

F  
ATA **状態遷移テスト(state transition testing)：** ブラックボックステストの設計技法の一つ。無効と有効の状態遷移を実行するテストケースを設計する。N-switch testingも参照のこと。

**状態遷移表(state table)：** 発生する可能性のあるイベントと状態の組み合わせから、生じる結果を示す遷移のテーブル。無効な遷移と、有効な遷移の両方を示す。

**仕様ベースドテスト(specification-based testing)：** black box testing を参照のこと。

**仕様ベースドテスト設計技法(specification-based test design technique)：** black box test design technique を参照のこと。

ATA **仕様の技法(specification-based technique)：** black box test design technique を参照のこと。

F **書記(scribe)：** レビューミーティングで指摘された欠陥や、改善の提案を、所定の書式に記録する人。書記は、記録が読みやすく、理解しやすくなるように留意しなくてはならない。

**シンタックステスト[構文テスト](syntax testing)：** ブラックボックステスト設計技法の一つ。入力ドメインと出力ドメインの定義を基にテストケースを設計する。

EITP **診断 (IDEAL)(diagnosing (IDEAL))：** IDEAL モデルの 1 フェーズであり、現在の状態と改善後の状態を明らかにする。診断フェーズは、現在の状態と望ましい状態を明らかにしたり、改善すべき点を明らかにしたりする活動で構成される。IDEAL も参照のこと。JSTQB 訳注) IDEAL のフェーズ名称は「CMMI V1.2 モデル - 開発のための - 公式日本語翻訳版」の定義を使用。

ETM **信頼区間(confidence interval)：** プロジェクトリスクのマネジメントにおいて、リスクの影響を効率的に軽減するためにコンティンジェンシーアクションを実行する必要がある期間。

**信頼性(reliability)：** 指定された条件下で利用するとき、指定された達成水準を維持するソフトウェア製品の能力。[ISO/IEC 9126] JSTQB 訳注) JIS X 0129-1:2003 より引用

F **信頼性テスト(reliability testing)：** ソフトウェア製品の信頼性を判定するテストのプロセス。  
ATT

ATT **信頼度成長モデル(reliability growth model)：** コンポーネントやシステムのテストにおいて、信頼性に関する故障を発見し、その欠陥を取り除いていくことで時間と共に信頼性が成長することを示すモデル。

## す

**遂行済み(のもの)(exercised)：** 「遂行済み」とは、テストケースによって入力値がプログラム要素の実行を引き起こすことである。要素としては、ステートメント、判定、又は他の構造的要素がある。

**垂直トレーサビリティ[垂直追跡性](vertical traceability)：** 開発ドキュメントからコンポーネントまでの各階層を通して要件を追跡すること。

**水平トレーサビリティ(horizontal traceability)：** テスト関連ドキュメント(テスト計画書、テスト設計仕様書、テストケース仕様書、テスト手順仕様書、テストスクリプトなど)の階層を通して、あるテストレベルで対象となる要件を追跡すること。

EITP **スクラム(SCRUM)：** アジャイルソフトウェア開発におけるプロジェクト管理のためのイテレーティブインクリメンタル型フレームワーク。agile software development も参照のこと。

F **スクリプト言語(scripting language)：** 実行可能テストスクリプトとして書かれ、テスト実行ツール(たとえば、キャプチャ・プレイバックツール)で使われるプログラミング言語。

**スクリプトテスト(scripted testing)：** すでに記述されているテスト順序通りに実行するテスト方法。

**スコアカード(scorecard)：** 長期的なゴールに向けた実行の進捗状況を表す要約されたパフォーマンス

ンス測定の表現。スコアカードは、定義された間隔中、若しくはその間隔の終了時に静的なパフォーマンス測定を提供する。balanced scorecard, dashboard も参照のこと。

F **スタブ(stub)**: 特定のコンポーネント(仮に A と呼ぶ)をテストするため、A に呼び出される(特定目的のための最小限度の)コンポーネント。スタブがないと、実物ができるまで、開発やテストを待たねばならない。スタブは、最終的には、呼び出されるコンポーネントで置き換える。[After IEEE 610]

**ステータスアカウンティング(status accounting)**: 構成管理の要素で、構成を効果的に管理する場合に必要となる情報を記録、報告すること。必要情報には、承認済み構成識別表、承認済み構成に対する変更提案の状態表、承認済み変更の実施状態の一覧表などがある。[IEEE 610]

ATM **STEP(STEP)**: Systematic Test and Evaluation Process を参照のこと。  
EITP

**ステートメント(statement)**: プログラミング言語の実体。実行の最小単位。

F **ステートメントカバレッジ(statement coverage)**: テストスイートによって遂行されたステートメントのパーセンテージ。

ATT **ステートメントテスト(statement testing)**: ホワイトボックステスト設計技法の一つ。ステートメントを実行するテストケースを設計する。

**ストレージ(storage)**: resource utilization を参照のこと。

**ストレージテスト(storage testing)**: resource utilization testing を参照のこと。

F **ストレステスト(stress testing)**: 予測又は特定した負荷、若しくはメモリやサーバなどのリソースの可用性が低減したときの限界、又は、それを超えた条件でシステムやコンポーネントを評価するために行なわれる性能テストの一種。[After IEEE 610] performance testing, load testing も参照のこと。

F **ストレステストツール(stress testing tool)**: ストレステストを支援するツール。

**スペシファイドインプット(specified input)**: 仕様から結果を予測した入力。

ETM **S.M.A.R.T. ゴール方法論(S.M.A.R.T. goal methodology)**: 目標を、汎用的ではなく、非常に具体的に定義する方法論。SMART は、定義される目標の属性を表す Specific (具体的な)、Measurable (測定可能な)、Attainable (達成可能な)、Relevant (妥当な)、及び Timely (タイムリーな)からの頭字語である。

**スモークテスト(smoke test)**: 定義・計画した全テストケースのサブセット。プログラムの必須機能が正常に動作することを確認するのが目的で、コンポーネントやシステムの主要機能を網羅し、細かな点は無視する。日次のビルドやスモークテストは、(ソフトウェア)業界において最も効率的・効果的な実践方法の一つ。build verification test, intake test も参照のこと。

**スレッドテスト(thread testing)**: コンポーネント統合テストのアプローチの一つ。要件のサブセットの実装に合わせて、コンポーネントを一つずつ統合して行なう。階層レベルごとにコンポーネントを統合する方式とは対極に位置する。

## せ

**正確性(accuracy)**: 必要とされる精度で、正しい結果若しくは正しい効果、又は同意できる結果若しくは同意できる効果をもたらすソフトウェア製品の能力。[ISO/IEC 9126] functionality も参照のこと。JSTQB 訳注) JIS X 0129-1:2003 より引用

- ATA**      **正確性テスト(accuracy testing)：** ソフトウェア製品の正確性を判定するためのテストプロセス。accuracyも参照のこと。
- 成果物(deliverable)：** (業務用)プロダクトの開発担当者以外に提供する必要がある、あらゆる(業務用)プロダクト。
- F**      **制御フロー(control flow)：** コンポーネントやシステムの実行における一連のイベント(パス)。
- ATT**      **制御フロー解析(control flow analysis)：** コンポーネントやシステムを実行する際に表現される特定のパス(一連のイベント)に基づいた静的解析の一種。制御フロー解析では、閉ループやデッドコードなどの制御フローの異常な箇所を見つけ出すことによって、制御フロー構造の完全性を評価する。
- 制御フローグラフ(control flow graph)：** コンポーネントやシステムの実行における全てのイベント(パス)のシーケンスを抽象表現したもの。
- ATT**      **制御フローテスト(control flow testing)：** 構造ベースドテストのアプローチの一つ。テストケースはイベントを特定の順序で実行するように設計される。制御フローテストには、さまざまな技法が存在する。たとえば、デシジョンテスト、条件テスト、パステストなどがあり、それぞれが特定のアプローチと制御フローカバレッジのレベルを持つ。decision testing, condition testing, path testingも参照のこと。
- 制御フローパス(control flow path)：** pathを参照のこと。
- ATT**      **成熟性(maturity)：**  
(1)プロセスや作業の有効性と効率に関する組織の能力。Capability Maturity Model Integration, Test Maturity Model integrationも参照のこと。  
(2) ソフトウェアに潜在する障害の結果として生じる故障を回避するソフトウェア製品の能力。[ISO/IEC 9126] reliabilityも参照のこと。JSTQB 訳注)JIS X 0129-1:2003 より引用
- 成熟度モデル(maturity model)：** 組織の成熟度の特定側面を構造的に記述した集合体で、組織のプロセスの定義と理解を助ける。成熟度モデルは、多くの場合、共通言語、ビジョンの共有、及び改善活動の優先順位を付けるためのフレームワークを提供する。
- EITP**      **成熟度レベル(maturity level)：** 定義済みプロセス領域のセット全体において、セット内の全てのゴールが達成されているプロセス改善の度合。[TMMi]
- ATT**      **静的アナライザ(static analyzer)：** 静的解析を行なうツール。
- F**      **静的解析(static analysis)：** (たとえば、要件又はコードなどの)ソフトウェア開発の成果物を実行せずに解析すること。静的解析は通常、支援ツールを用いて実施する。
- ATT**
- F**      **静的解析ツール(static analysis tool)：** static analyzerを参照のこと。
- 静的コードアナライザ(static code analyzer)：** 静的コード解析を実施するツール。ソースコードをチェックし、コーディング基準や品質メトリクスへの準拠度合、データフローの不整合などを検出する。
- 静的コード解析(static code analysis)：** ソフトウェアを実行せずにソースコードを解析すること。
- F**      **静的テスト(static testing)：** ソフトウェア開発の成果物(要件、設計、又は、コードなど)の実行をせずに実施する成果物のテスト。たとえば、レビュー、静的解析など。
- 性能(performance)：** システムやコンポーネントが、処理時間やスループット率の制約内で、定義した機能を果たす度合。[After IEEE 610] efficiencyも参照のこと。
- F**      **性能テスト(performance testing)：** ソフトウェア製品の性能を判定するためのテストプロセス。

F-AT ATT	efficiency testing も参照のこと。
F ATT	<p><b>性能テストツール(performance testing tool)：</b> 性能テストを支援するツール。負荷生成機能とテストトランザクション測定機能の二つを備えることが多い。負荷生成機能は複数ユーザや大量の入力データをシミュレートする。テストの実行中、選択したトランザクションの応答時間を計測・記録する。性能テストツールは、テストの結果記録に基づいたレポートや、応答時間に対する負荷のグラフを提供することが多い。</p> <p><b>性能プロファイル(performance profiling)：</b> 分析を行なうタスク。たとえば、生成されたメトリクスに基づいて性能のボトルネックを特定したり、ツールを使用してソフトウェアコンポーネント又はシステムの性能を調整したりする。</p> <p><b>製品受け入れテスト(production acceptance testing)：</b> operational acceptance testing を参照のこと。</p>
ETM	<p><b>責任分担(RACI)マトリクス(RACI matrix)：</b> プロジェクトの参加者への割り当てを示すマトリクス。割り当てられるものには、タスクを完了するためのさまざまな役割、あるいはプロジェクト又はプロセスの成果物がある。役割と責任を明確にするのに特に役立つ。RACI は、ほとんどの場合に使用される 4 つの主要な役割である Responsible (実行責任者)、Accountable (説明責任者)、Consulted (協業先)、及び Informed (報告先)からの頭字語である。</p> <p><b>セキュリティ(security)：</b> 許可されていない人又はシステムが情報又はデータを読んだり、修正したりすることができないように、及び許可された人又はシステムが情報又はデータへのアクセスを拒否されないように、情報又はデータを保護するソフトウェア製品の能力。[ISO/IEC 9126] functionality も参照のこと。JSTQB 訳注) JIS X 0129-1:2003 より引用</p>
F	<b>セキュリティツール(security tool)：</b> セキュリティの運用を支援するツール。
F F-AT ATT	<p><b>セキュリティテスト(security testing)：</b> ソフトウェア製品のセキュリティを判定するテスト。functionality testing も参照のこと。</p> <p><b>セキュリティテストツール(security testing tool)：</b> セキュリティの特性と脆弱性をテストするための支援をするツール。</p> <p><b>セッションベースドテスト(session-based testing)：</b> テスト設計と実行を行なう連続するセッションとして計画されたテスト活動を行なうテストアプローチ。多くの場合、探索的テストにおいて使用される。</p> <p><b>セッションベースのテストマネジメント(session-based test management)：</b> セッションベースドテストの測定と管理の手法。たとえば、exploratory testing。</p>
ATT	<p><b>設置性(installability)：</b> 指定された環境に設置するためのソフトウェア製品の能力。[ISO/IEC 9126] portability も参照のこと。JSTQB 訳注) JIS X 0129-1:2003 より引用</p> <p><b>設置性テスト(installability testing)：</b> ソフトウェア製品の設置性をテストするプロセス。portability testing も参照のこと。</p> <p><b>セーフティクリティカルシステム(safety critical system)：</b> 故障や誤動作が人命や人々への深刻な損害、若しくは機器へのダメージや環境被害となる可能性のあるシステム。</p>
EITP	<b>先験ベースド品質(transcendent-based quality)：</b> 品質に対する考え方の一つ。品質は正確に定義することはできないが、みれば分かる、あるいは、欠落していることにも気づくものである、というもの。この品質は、個人やグループがプロダクトに対して持つ知見や思い入れに依存する。[After Garvin] manufacturing-based quality, product-based quality, user-based quality, value-based quality も



参照のこと。

**F** **全数テスト(exhaustive testing)：** テストのアプローチの一つ。テストスイートにより、入力値と事前条件の全組み合わせをテストすること。

## そ

**EITP** **総合的品質管理(Total Quality Management)：** 全てのメンバーが参加することを基本に、社会の利益と組織のメンバー全員の利益と顧客満足を通じた長期的な成功を目的とした品質中心の組織全体のマネジメントアプローチ。総合的品質管理は、計画、組織化、監督、コントロールと保証からなる。[After ISO 8402]

**相互運用性(interoperability)：** 一つ以上の指定されたシステムと相互作用するソフトウェア製品の能力。[After ISO/IEC 9126] functionality も参照のこと。JSTQB 訳注) JIS X 0129-1:2003 より引用

**F** **相互運用性テスト(interoperability testing)：** ソフトウェア製品の相互運用性を判定するテストのプロセス。functionality testing も参照のこと。

**ATA**

**操作環境(operational environment)：** ユーザや顧客のサイトにインストールしたハードウェアやソフトウェア製品。この環境で、テスト中のコンポーネントやシステムを動作させる。操作環境のソフトウェアには、オペレーティングシステム、データベースマネジメントシステム、その他のアプリケーションを含むこともある。

**測定(measurement)：** ある実体の特性を表すため、数字や種別を付加する手順。[ISO/IEC 14598]

**測定尺度(measurement scale)：** データ分析のタイプに合わせた尺度であり、その尺度でデータ分析を行なう。[ISO/IEC 14598]

**EITP** **測定値(measure)：** 測定することによって、ある実体の特性に付加した数字や種別。[ISO/IEC 14598]

**ソースステートメント(source statement)：** statement を参照のこと。

**ソフトウェア(software)：** コンピュータのプログラムやプロシージャのこと。コンピュータシステムの運用に関連したドキュメントやデータも含む場合もある。

**ソフトウェア完全性レベル(software integrity level)：** ステークホルダが選択したソフトウェアやソフトウェアベースのシステム特性にソフトウェアが準拠する、又は準拠する必要のある度合。一連のシステム特性は、たとえば、ソフトウェアの複雑性、リスクアセスメント、安全レベル、セキュリティレベル、期待される性能、信頼性、又はコストで構成され、ステークホルダにとってのソフトウェアの重要度を反映するように定義される。

**ソフトウェア攻撃(software attack)：** attack を参照のこと。

**ソフトウェア・故障モード影響解析(Software Failure Mode and Effect Analysis (SFMEA))：** Failure Mode and Effect Analysis (FMEA)を参照のこと。

**ソフトウェア・故障モード影響・致命度解析(Software Failure Mode, Effects, and Criticality Analysis (SFMECA))：** Failure Mode, Effects, and Criticality Analysis (FMECA) を参照のこと。

**ATA** **ソフトウェア使用性測定一覧表(Software Usability Measurement Inventory (SUMI))：** エンドユーザの視点からソフトウェア品質を測定するための、アンケートに基づいた使用性テストの技法。

[Veenendaal04]

**ソフトウェア製品特性(software product characteristic)：** quality attribute を参照のこと。

**ソフトウェアテストインシデント(software test incident)：** incident を参照のこと。

**ソフトウェアテストインシデントレポート(software test incident report)：** incident report を参照のこと。

**ソフトウェア品質(software quality)：** ソフトウェア製品の全体として、機能とフィーチャが、明示的、暗示的なニーズを満たしている能力。[After ISO/IEC 9126] quality も参照のこと。

**ソフトウェア品質特性(software quality characteristic)：** quality attribute を参照のこと。

**ソフトウェアフィーチャ(software feature)：** feature を参照のこと。

**ソフトウェア・フォールトツリー解析(software Fault Tree analysis (SFTA))：** Fault Tree Analysis (FTA) を参照のこと。

**EITP** **ソフトウェアプロセス改善(Software Process Improvement)：** 組織のソフトウェアプロセスとその結果のパフォーマンス及び成熟度を改善する活動プログラム。[After CMMI]

**F-AT**  
**EITP** **ソフトウェアライフサイクル(software lifecycle)：** ソフトウェアプロダクトの最初から最後、つまり企画段階から利用終了までの期間。ソフトウェアライフサイクルは通常、コンセプトフェーズ、要件フェーズ、設計フェーズ、実装フェーズ、テストフェーズ、インストールとチェックアウトフェーズ、運用と保守フェーズを含み、ときに除去フェーズを含むこともある。注)これらのフェーズは重複することもあるし、反復することもある。

た

**大規模統合テスト(integration testing in the large)：** system integration testing を参照のこと。

**ATM**  
**EITP** **体系的テストと評価プロセス(STEP) (Systematic Test and Evaluation Process)：** 体系的なテスト方法論。テストプロセス改善のためにコンテンツベースドモデルとしても使用される。STEP による改善は、順序には依存しない。content-based model も参照のこと。

**ETM** **対処テスト(reactive testing)：** テスト対象の実際のシステムの動作又は入手したテスト結果を基に、動的に対処するテスト。対処テストでは、ほとんどの場合、計画サイクルが短縮されており、テスト対象を受け取るまでテスト設計とテスト実装のフェーズが実施されない。

**ETM** **ダッシュボード(dashboard)：** 組織や活動の業務成績の動的な測定結果を、メタファによって表されたメトリクスを用いて表現したもの。ここでいうメタファとは、自動車のダッシュボードに似せた、目にみえるダイヤルやカウンターなどである。これによって、出来事や活動の結果が容易に把握でき、業務上の目標と関連付けることができる。corporate dashboard, scorecard も参照のこと。

**ダーティテスト(dirty testing)：** negative testing を参照のこと。

**F** **妥当性確認(validation)：** 検査、及び、特定の使用法や適用に対する要件が満たされていることを客観的な証拠で確認すること。[ISO 9000]

**ATA** **WAMMI(WAMMI)：** Website Analysis and MeasureMent Inventory を参照のこと。

**WBS(WBS)：** Work Breakdown Structure を参照のこと。

EITP	<b>段階表現(staged representation)</b> : 成熟度レベルとして確立したプロセスエリアのゴールに達したかを見るモデル構造であり、各レベルは次のレベルの土台となるよう組み立てられている。[CMMI]
F F-AT ATA	<b>探索的テスト(exploratory testing)</b> : 非公式なテスト設計技法の一つ。テストを実施する過程で、テスト担当者がテスト実施情報を活用しながらテスト設計をコントロールし、積極的に質の高い新しいテストケースを設計する。[After Bach]
ATT	<b>短絡評価(short-circuiting)</b> : 複合条件を評価するためのプログラミング言語/インタープリタの技法。論理演算子の片方が最終結果を決定するのに十分である場合に、他方は評価されないことがある。

## ち

**チェック担当(checker)**: reviewer を参照のこと。

ATA	<b>チェックリストベースドテスト(checklist-based testing)</b> : 経験ベースのテスト設計技法であり、経験を積んだテスト担当者が、気づき、チェックし、あるいは記憶すべき項目の高位レベルのリストやルール集、検証すべき基準を使用して実施する。
ATT	<b>置換性(replaceability)</b> : 同じ環境で、同じ目的のために、他の指定されたソフトウェア製品から置き換えて使用することができるソフトウェア製品の能力。[ISO/IEC 9126] portability も参照のこと。 JSTQB 訳注) JIS X 0129-1:2003 より引用

**チャータ(charter)**: test charter を参照のこと。

**中間者攻撃(man in the middle attack)**: ユーザに気づかれることなく、第三者により、通信(たとえば、クレジットカード決済情報)の傍受、模倣、改ざん、転送が行なわれる攻撃。

**中止基準(suspension criteria)**: テストアイテムに関わる全て、又は、一部のテストの活動を(一時的に)止める場合に参照する基準。[After IEEE 829]

**抽象的テストケース(abstract test case)**: high level test case を参照のこと。

**注文ソフトウェア(bespoke software)**: 特定ユーザや特定顧客専用に関連したソフトウェア。反対語は off-the-shelf software (既製ソフトウェア)。

**Chow のカバレッジメトリクス(Chow's coverage metrics)**: N-switch coverage を参照のこと。  
[Chow]

ATA	<b>直交表(orthogonal array)</b> : 特殊な数学的性質を使って構築した 2 次元の配列であり、配列の中から選択した二つの列から、その配列の中の各値に対して全てのペアの組み合わせを提供する。
ATA	<b>直交表テスト(orthogonal array testing)</b> : 直交表を使った変数のオールペア組み合わせテストの体系的な方法。変数を全て組み合わせたときの数を、オールペア組み合わせでテストできるまでに減らす。combinatorial testing, n-wise testing, pairwise testing も参照のこと。

## て

ATA	<b>低位レベルテストケース(low level test case)</b> : 入力データと期待結果が具体的(実装レベル)なテストケース。高位レベルのテストケースからの論理演算子は、論理演算子に相当する実際の値に置き
-----	--

換えられる。high level test case も参照のこと。

ATM  
EITP **TMMi(TMMi)**: Test Maturity Model integration を参照のこと。

ATT **定義使用ペア(definition-use pair)**: 変数の定義と、その変数の以降での使用とを結合したもの。変数の使用の例として、計算(たとえば、乗算)、パス実行の制御(述語的使用)がある。

EITP **TQM(TQM)**: Total Quality Management を参照のこと。

F  
F-AT  
ETM **TDD(TDD)**: test-driven development を参照のこと。

**DD パス(dd-path)**: アルゴリズムの二つの判定間のパス。又は他の判定を含まない対応するグラフの二つの判定ノード間のパス。path も参照のこと。

ATM  
EITP **TPI Next(TPI Next)**: テストプロセスを改善するための、継続するビジネス駆動のフレームワーク。効果的で効率的なテストプロセスの主要な要素を定義する。

EITP **TPG(TPG)**: Test Process Group を参照のこと。

**デイリービルド(daily build)**: システム全体を毎日(多くの場合、夜間)、コンパイル、リンクして作成する開発の活動。これにより、最新の変更を反映した一貫性のあるシステム、アプリケーションにいつでもアクセスできるようになる。

**デヴィエーションレポート(逸脱報告)(deviation report)**: incident report を参照のこと。

**適合テスト(conformance testing)**: compliance testing を参照のこと。

F  
ATM **テクニカルレビュー(technical review)**: どのような技術的アプローチをとるかで意見を一致させることを目的とした、ピアグループによるディスカッション。[Gilb and Graham], [IEEE 1028] peer review も参照のこと。

**デザインベースドテスト(design-based testing)**: テストに対するアプローチ法の一つ。コンポーネントやシステムの構造や詳細設計に基づいたテストケースを設計するもの(たとえば、コンポーネントやシステム間のインターフェースのテスト)。

F **デシジョンカバレッジ(decision coverage)**: テストスイートによって遂行された、判定結果のパーセンテージ。100%のデシジョンカバレッジは、100%の branch coverage(ブランチカバレッジ)と 100%の statement coverage(ステートメントカバレッジ)の両方を意味する。

**デシジョンテスト(decision testing)**: ホワイトボックステスト設計技法の一つ。判定を実行するテストケースを設計する。

**デシジョンテーブル(decision table)**: 入力と刺激(原因)、及び、対応する出力と処理(結果)の組み合わせを示す表。テストケースの設計に利用できる。

F  
ATA **デシジョンテーブルテスト(decision table testing)**: ブラックボックステスト設計技法の一つ。デシジョンテーブルにある入力と刺激(原因)の組み合わせを実行するテストケースを設計する。[Veenendaal04] decision table も参照のこと。

**テスト(test)**: 一つ以上のテストケースのセット。[IEEE 829]

F **テスト(testing)**: 全てのライフサイクルを通じて実施する静的、動的なプロセスにおいて、成果物が特定の要件を満足するかを判定し、目的に合致することを実証し、欠陥を見つけるため、ソフトウェア

ロダクトや関連成果物に対し、計画、準備、評価をすること。

**テストアイテム(test item)：** テストを実施する個々の要素。通常、一つのテスト対象に対し、多数のテストアイテムがある。test object も参照のこと。

**テストアイテム送付レポート(test item transmittal report)：** release note を参照のこと。

**ETM      テストアーキテクト(test architect)：**  
(1)テスト組織に対して、テスト組織と他の分野との関連について、ガイダンスと戦略的指針を提供する人。  
(2)特定のシステム用にテストを構造化する方法を定義する人。テストツールやテストデータのマネジメントなどのトピックも定義する。

**F      テストアプローチ(test approach)：** 特定のプロジェクトのためのテスト戦略を実現化したもの。この中には、(テスト実施)プロジェクトのゴールと評価済みリスクに基づいて決めた決定事項、テストプロセスの開始ポイント、適用するテスト設計技法、テスト終了基準、実施するテストタイプを含む。  
**F-AT**  
**ATM**

**テストインシデント(test incident)：** incident を参照のこと。

**テストインシデントレポート(test incident report)：** incident report を参照のこと。

**テストインフラ(test infrastructure)：** テスト環境、テストツール、オフィス環境、処理手続きからなるテストの実施に必要な構造的なもの。

**F      テストウェア(testware)：** テストプロセスを通じて作成される、テストの計画、設計、実行に不可欠なもの。たとえば、ドキュメント、スクリプト、入力、期待結果、セットアップとクリーンアップの処理手順、ファイル、データベース、環境、その他、テストで使用する付加的なソフトウェアやユーティリティなど。[After Fewster and Graham]

**F-AT      テストオラクル(test oracle)：** テスト対象のソフトウェアの実行結果と比較する期待結果のソース。オラクルは、実在する(ベンチマーク用の)システム、他のソフトウェア、ユーザマニュアル、個人の専門知識の場合があるが、コードであってはならない。[After Adrion]

**EITP      テスト改善計画(test improvement plan)：** 組織のテストプロセスとテストプロセスの資産について、長所と短所を徹底的に理解し、組織的にテストプロセス改善を達成することをベースとした計画。[After CMMI]

**F      テストカバレッジ(test coverage)：** coverage を参照のこと。

**F      テスト環境(test environment)：** テストの実行に必要なハードウェア、インスツルメンテーション、シミュレータ、ソフトウェアツール、その他の支援要素を含む環境。[After IEEE 610]

**テスト完了基準(test completion criteria)：** exit criteria を参照のこと。

**テスト技法(test technique)：** test design technique を参照のこと。

**F      テスト駆動開発(test driven development)：** 本体のソフトウェアを開発する前に、テストケースを開発(場合によると自動化)するソフトウェア開発手法。  
**F-AT**  
**ETM**

**F      テスト計画(test plan)：** 計画されたテスト活動の狙い、アプローチ、リソース、スケジュールを記述するドキュメント。テストアイテム、テストすべきフィーチャ、タスク、各タスク担当者、テスト担当者の独立の度合、テスト環境、用いるテスト設計技法と開始・終了基準、それらの選択の理論的根拠、それに代替計画を必要とするあらゆるリスクを識別する。これはテスト計画プロセスの記録である。[After IEEE 829]  
**ATM**

**ATM      テスト計画作業(test planning)：** テスト計画を策定し、更新すること。

ATA

F  
ATM  
**テストケース(test case)：** 入力値、実行事前条件、期待結果、そして、実行事後条件のセットで、特定のプログラムパスを用いることや特定の要件が満たされていることを検証することのような、特定の目的又はテスト条件のために開発されたもの。[After IEEE 610]

F  
**テストケース仕様(test case specification)：** テストアイテム用のテストケースのセット(目的、入力値、テスト実行、期待結果、実行事前条件)を規定するドキュメント。[After IEEE 829] test specificationも参照のこと。

**テストケーススイート(test case suite)：** test suite を参照のこと。

**テストケース設計技法(test case design technique)：** test design technique を参照のこと。

**テスト結果(test outcome)：** result を参照のこと。

**テスト結果(test result)：** result を参照のこと。

F  
ATM  
**テスト結果記録(test log)：** テスト実行の詳細を時系列的に記録したもの。[IEEE 829]

**テスト結果記録作業(test logging)：** テスト実行の情報を記録するプロセス。

**テスト結果比較(test comparison)：** テスト対象のコンポーネントやシステムの実行結果と期待結果の違いを識別するプロセス。テスト結果比較は、テスト実行中(動的比較)、又は、テスト実行後に実施することができる。

F  
ATM  
ATA  
**テストコントロール(test control)：** 監視中に計画から逸脱していることを検出した場合に、テストプロジェクトを軌道修正するための対策を考えたり適用したりするテストマネジメントタスクの一つ。test managementも参照のこと。

**テストサイクル(test cycle)：** 識別可能な単一のテスト対象のリリースに対し、テストプロセスを実行すること。

**テスト再現性(test reproducibility)：** テストを実行する度に、同じ結果を出力できるかどうかを示す特性。

F  
ATM  
**テストサマリレポート(test summary report)：** テスト活動と結果を要約したドキュメント。テストアイテムがテスト終了基準を満足しているかどうかの評価も含む。[After IEEE 829]

**テストジェネレータ(test generator)：** test data preparation tool を参照のこと。

**テストシチュエーション(test situation)：** test condition を参照のこと。

F  
ATM  
ATA  
**テスト実行(test execution)：** テスト対象のコンポーネントやシステムでテストを実行し、実行結果を出力するプロセス。

**テスト実行技法(test execution technique)：** 実際のテストを、手動又は自動で実行する技法。

F-AT  
**テスト実行自動化(test execution automation)：** たとえばキャプチャ/プレイバックツールのようなソフトウェアを使用して、テストの実行、実行結果と期待結果の比較、テスト条件の設定、その他のテストコントロールやレポート機能を(自動)制御すること。

F  
**テスト実行スケジュール(test execution schedule)：** テスト手順を実行していくための計画。注) テスト実行スケジュールには、テスト手順書の実際に実行する内容とその実行順を記述する。

F ATA ATT	<p><b>テスト実行ツール(test execution tool)：</b> テストツールの一種。キャプチャ/プレイバックのような自動化テストスクリプトを使い、他のソフトウェアを実行できる。[Fewster and Graham]</p> <p><b>テスト実行フェーズ(test execution phase)：</b> ソフトウェア開発ライフサイクル中で、ソフトウェアプロダクトのコンポーネントを実行し、要件を満たしているかどうか判定する期間。[IEEE 610]</p>
ATM ATA	<p><b>テスト実装(test implementation)：</b> テストデータを考え出し、テスト手順の開発及び優先度付けを行なうプロセス。テストハーネスの準備や自動テストスクリプト記述を含むこともある。</p> <p><b>テスト失敗(test fail)：</b> fail を参照のこと。</p>
F-AT	<p><b>テスト自動化(test automation)：</b> ソフトウェアを使って、テストマネジメント、テスト設計、テスト実行、結果チェックなどのテスト活動の実行や支援をすること。</p> <p><b>テストシナリオ(test scenario)：</b> test procedure specification を参照のこと。</p>
ETM	<p><b>テスト使命(test mission)：</b> 組織にとってのテストの目的で、多くの場合、テストポリシーの一部として文書化される。test policy も参照のこと。</p>
ATM	<p><b>テスト終了作業(test closure)：</b> テストプロセスに含まれるテスト終了作業フェーズの間、経験、テストウェア、事実、数字をまとめるために、データを完了した活動から収集する。テスト終了作業フェーズはテストウェアの仕上げ、保管とテスト評価レポートの準備を含むテストプロセスの評価からなる。test process も参照のこと。</p> <p><b>テスト仕様化技法(test specification technique)：</b> test design technique を参照のこと。</p>
F ATM	<p><b>テスト条件(test condition)：</b> コンポーネントやシステムのアイテムやイベントで、テストケースにより検証できるもの。たとえば、機能、トランザクション、フィーチャ、品質の属性、構造要素など。</p> <p><b>テスト仕様書(test specification)：</b> テスト設計仕様、テストケース仕様、テスト手順仕様からなるドキュメント。</p> <p><b>テスト進捗レポート(test progress report)：</b> 定期的に作成するテストの活動と結果をまとめたドキュメント。テスト活動の進捗がベースライン(当初のテスト計画など)に沿っているかどうか報告するため、かつリスクや決定が必要な代替案をマネジメント層へ伝えるために作成する。</p>
F	<p><b>テストスイート(test suite)：</b> テスト対象のコンポーネント又はシステムのためのいくつかのテストケースのセット。一つのテストの実行事後条件は、次のテストの実行事前条件としてよく利用される。</p>
F ATM	<p><b>テストスクリプト(test script)：</b> 一般的にテスト手順仕様を指して用いられる。特に自動化時のスクリプトを指す。</p> <p><b>テストスケジュール(test schedule)：</b> 活動、タスク、又はテストプロセスのイベントに関して、開始/終了日、時間、依存関係を識別できるリスト。</p> <p><b>テストステージ(test stage)：</b> test level を参照のこと。</p> <p><b>テスト成果物(test deliverable)：</b> テストプロダクト作成者以外に提供する必要のある、あらゆるテスト(作業)プロダクト。deliverable も参照のこと。</p>
ATM EITP	<p><b>テスト成熟度モデル統合(Test Maturity Model integration (TMMi))：</b> 能力成熟度モデル統合(CMMI)と関連させた5段階からなるテストプロセス改善のためのフレームワークであり、効果的なテストプロセスのための重要要素を記述している。</p>

F ATM ATA	<p><b>テスト設計(test design)：</b></p> <p>(1) test design specification を参照のこと。</p> <p>(2) 概略的なテスト目的を具体的なテスト条件とテストケースに変換するプロセス。</p>
F	<p><b>テスト設計技法(test design technique)：</b> テストケースを作成したり選択したりするための技法。</p> <p><b>テスト設計仕様(test design specification)：</b> テストアイテムのテスト条件(カバレッジアイテム)、詳細なテストアプローチ、及び、関連する高位レベルテストケースを記述したドキュメント。[After IEEE 829] test specification も参照のこと。</p>
F ATA	<p><b>テスト設計ツール(test design tool)：</b> テスト入力を生成することでテスト設計を支援するツール。テスト入力の生成は、CASE ツールのリポジトリ(たとえば要件マネジメントツール)に格納している仕様、ツールの中に保存してある特定のテスト条件、又はコードから行なわれる。</p> <p><b>テストセッション(test session)：</b> テスト実行中の連続した一区切りの時間。探索的テストでは、各テストセッションは一つのチャータに焦点を当ててテストを行なう。しかし、セッション中にテスト担当者は新しい気づきや問題に対してもまた探索することもある。テスト担当者はその場でテストケースを考えて実行し、進捗を記録する。exploratory testing も参照のこと。</p> <p><b>テストセット(test set)：</b> test suite を参照のこと。</p>
F F-AT ATM ATA ETM	<p><b>テスト戦略(test strategy)：</b> 組織や(一つ若しくは複数プロジェクトの)プログラムで実施するテストレベルと各テストレベルでのテスト内容を高位レベルで説明したもの。</p> <p><b>テスト対象(test object)：</b> テストすべきコンポーネント又はシステム。test item も参照のこと。</p> <p><b>テスト対象システム(system under test)：</b> test object を参照のこと。</p> <p><b>テストタイプ(test type)：</b> コンポーネント又はシステムをテストするためのテスト活動をまとめたものであり、たとえば機能テスト、使用性テスト、回帰テストなどのように特定のテスト目的に焦点を当てている。テストタイプは一つ又は複数のテストレベル又はテストフェーズで行なわれる。[After TMap]</p> <p><b>テストターゲット(test target)：</b> テスト終了基準のセット。</p>
F	<p><b>テスト担当者(tester)：</b> コンポーネントやシステムのテストを実施する熟練した専門家。</p>
F-AT ATA	<p><b>テストチャータ(test charter)：</b> テスト目的を明記したもの。テスト実施法のアイデアを含む場合もある。探索的テストにて使用する。exploratory testing も参照のこと。</p>
EITP	<p><b>テストツール(test tool)：</b> 一つ以上のテスト活動を支援するソフトウェア製品。たとえば、計画とコントロール、仕様化、初期ファイルやデータの構築、テスト実行とテスト分析を支援する。[TMap] CAST も参照のこと。</p>
ATM	<p><b>テストディレクタ(test director)：</b> テストマネージャをマネジメントする上級マネージャ。test manager も参照のこと。</p>
F ATM	<p><b>テスト手順(test procedure)：</b> test procedure specification を参照のこと。</p>
F ATM	<p><b>テスト手順仕様(test procedure specification)：</b> テストの実行のために、一連の手順を定めたドキュメント。テストスクリプト、又は、手動テストスクリプトとしても知られる。[After IEEE 829] test specification も参照のこと。</p>



- F** **テストデータ(test data):** テスト実行前に実在する(たとえば、データベースの中にある)データであり、テスト対象のコンポーネントやシステムに影響を与えたり、影響を受けたりするもの。
- F** **テストデータ準備ツール(test data preparation tool):** テストに使うデータを選択(データベース内の実データなど)、又は、作成、生成、操作、編集をするためのツール。
- ATA**
- テストデータマネジメント(test data management):** テストデータ要件の分析、テストデータ構造の設計、テストデータの作成と保守を行なうプロセス。
- テストドライバ(test driver):** driverを参照のこと。
- テスト入力(test input):** テスト実行中にテスト対象が外部ソースから受け取ったデータ。外部ソースは、ハードウェア、ソフトウェア、人の場合がある。
- F** **テストの独立性(independence of testing):** 責任を分離すること。これにより、客観的なテストを促進できる。[After DO-178b]
- ATM**
- テストパス(test pass):** passを参照のこと。
- F** **テストハーネス(test harness):** テスト実行に必要なスタブやドライバからなるテスト環境。
- テストパフォーマンスインジケータ(test performance indicator):** テストの開発をよい方向へ導きコントロールするために利用する有効性と効率の高位レベルなメトリック。たとえば、欠陥検出率(DDP)。
- F** **テスト比較ツール(test comparator):** テスト実行時の実行結果と期待結果との比較を自動的に実施するテストツール。
- テスト評価レポート(test evaluation report):** 全てのテスト活動と結果を要約した、テストプロセスの最後にまとめるドキュメント。テストプロセスの評価や、学んだ教訓を含むこともある。
- テストフェーズ(test phase):** テスト活動をプロジェクト中で管理(マネジメント)しやすいフェーズにまとめたセット。たとえば、あるテストレベルの実行活動。[After Gerrard]
- テストプロセス(test process):** 基本的なテストプロセスは、テストの計画とコントロール、テストの分析と設計、テストの実装と実行、終了基準の評価と報告、テスト終了作業によって構成される。
- EITP** **テストプロセス改善マニフェスト(test process improvement manifesto):** アジャイルマニフェストにより表明された声明で、テストプロセスを改善するための価値を定義している。ここでいう価値とは以下のようなものである。
- ・プロセスの詳細化よりも柔軟性
  - ・テンプレート(ひな形)よりもベストプラクティス
  - ・プロセス志向よりも展開志向
  - ・品質保証(部門)よりもピアレビュー
  - ・モデル駆動よりもビジネス駆動
- [Veenendaal08]
- EITP** **テストプロセスグループ(Test Process Group):** (テスト)専門家の集団。組織が使用するテストプロセスの定義、改善や保守を促進する。[After CMMI]
- EITP** **テストプロセス担当(test process improver):** テスト改善計画に基づきテストプロセス改善を実行する人。
- テスト分析(test analysis):** テストベースとテスト目的の定義を分析するプロセス。

F F-AT	<p><b>テストベース(test basis)：</b> コンポーネント要件やシステム要件を推測できる全てのドキュメント。これらのドキュメントがテストケースのベースとなる。公式な改訂手順を経ないとドキュメントの改訂ができない場合、そのテストベースを「凍結テストベース」と呼ぶ。[After TMap]</p> <p><b>テストベッド(test bed)：</b> test environment を参照のこと。</p> <p><b>テストポイント分析(TPA) (Test Point Analysis (TPA))：</b> ファンクションポイント分析に基づき、数式でテストの必要リソースを見積る手法。[TMap]</p>
F ATM EITP ETM	<p><b>テストポリシー(test policy)：</b> 組織にとってのテストに関わる原理原則、アプローチ、主要な目的を記述する高位レベルのドキュメント。</p>
ATM	<p><b>テストマネジメント(test management)：</b> テスト活動の計画、見積り、監視、コントロール。主としてテストマネージャによって実施される。</p>
F ATT	<p><b>テストマネジメントツール(test management tool)：</b> テストプロセスのマネジメントとコントロールを支援するツール。テストウェアマネジメント、テストスケジューリング、結果の記録、進捗管理、インシデントマネジメント、テスト報告等の能力を持つことが多い。</p>
F	<p><b>テストマネージャ(test manager)：</b> テストの活動とリソースのマネジメント、テスト対象の評価に責任を持つ個人。テストプロジェクトを指揮、コントロール、運営し、テスト対象の評価を計画し統制する。</p>
F-AT ATM	<p><b>テスト見積り(test estimation)：</b> テストのさまざまな局面に紐付けられた概算結果(たとえば、工数(effort spent), 完了日(completion date), コスト(costs involved), テストケース数(number of test cases), など)。ただし、入力情報が不完全、又は不確か、若しくは余計な情報を含んでいても実施できる。</p>
F	<p><b>テスト目的(test objective)：</b> テストを設計、実行する理由や目的。</p>
F ATM ATA	<p><b>テストモニタリング(test monitoring)：</b> テストプロジェクトの状態の周期的なチェックに関連した活動を扱うテスト管理タスク。実際の活動と計画した活動とを比較した報告が準備される。test management も参照のこと。</p> <p><b>テスト容易化要件(testable requirement)：</b> 要件を満たしているかどうかを判断できるテストの設計(後に続くテストケースの作成を含む)と実行を可能にする観点から記述されている要件。[After IEEE 610]</p> <p><b>テスト要件(test requirement)：</b> test condition を参照のこと。</p> <p><b>テストラン(test run)：</b> テスト対象の特定のバージョンでテストを実行すること。</p> <p><b>テストランログ(test run log)：</b> test log を参照のこと。</p> <p><b>テストリグ(test rig)：</b> test environment を参照のこと。</p>
F	<p><b>テストリーダー(test leader)：</b> test manager を参照のこと。</p> <p><b>テストレコーディング[テスト結果記録作業](test recording)：</b> test logging を参照のこと。</p> <p><b>テストレコード[テスト結果記録](test record)：</b> test log を参照のこと。</p>
F ATM	<p><b>テストレベル(test level)：</b> 系統的にまとめ、管理していくテストの活動のグループ。各テストレベルはプロジェクトの特定の責務と対応付けができる。テストレベルの例には、コンポーネントテスト、統合テスト、システムテスト、受け入れテストがある。[After TMap]</p>

**テストレポート(test report)：** test summary report, test progress report を参照のこと。

**テストレポート作業(test reporting)：** テスト活動からデータを収集して分析し、その後、データをレポートにまとめてステークホルダに情報を提供する作業。test process も参照のこと。

**データ完全性テスト(data integrity testing)：** database integrity testing を参照のこと。

F  
ATT

**データ駆動テスト(data-driven testing)：** スクリプト作成技術の一つ。テスト入力と期待結果をテーブルやスプレッドシートに格納し、一つの制御スクリプトでテーブル中の全テストを実行するもの。キャプチャ/プレイバックツールのような、テスト実行ツールのアプリケーションで使うことが多い。[Fewster and Graham] keyword-driven testing も参照のこと。

**データ定義(data definition)：** 変数に値を割り当てる実行ステートメント。

**データ品質(data quality)：** データの属性の一つ。事前に定義した基準(たとえば、ビジネス期待度、データ完全性の要件、データ一貫性など)に関する正確さを示す。

F

**データフロー(data flow)：** データオブジェクトの順序と、起こり得る状態の変化を抽象的に表現したもの。オブジェクトの状態は、発生、使用、消滅のいずれかになる。[Beizer]

ATT

**データフロー解析(data flow analysis)：** 変数の「定義と使用」に基づく静的解析の一つ。

**データフローカバレッジ(data flow coverage)：** 変数の「定義と使用」を1組とし、テストスイートによってどれだけ遂行されたかをパーセンテージで表したもの。

**データフローテスト(data flow testing)：** ホワイトボックステスト設計技法の一つ。変数の「定義と使用」の組を実行するようにテストケースを設計する。

**データベース完全性テスト(database integrity testing)：** データ(ベース)をアクセス・管理する方法や手順をテストすること。アクセス方法、手順、データルールが期待通りに動作すること、データベースへのアクセス中にデータが破壊されたり、期待に反して削除、更新、生成されたりしないことを確認する。

**手続きテスト(procedure testing)：** コンポーネントやシステムが、新規、若しくは現行利用者のビジネス手順、又は運用手順と共に運用できることの確認を目的にしたテスト。

**デッドコード(dead code)：** unreachable code を参照のこと。

**デバグガ(debugger)：** debugging tool を参照のこと。

F

**デバグ(debugging)：** ソフトウェアの故障の原因を見つけて、分析して、取り除くプロセス。

F  
ATT

**デバグツール(debugging tool)：** 故障を再現して、プログラムの状態を調査し、対応した欠陥を見つけるために、プログラマが使用するツール。デバグガは、プログラムの変数をセットし検査するために、どのプログラムステートメントの場所でもプログラムを停止したり、プログラムを1ステップずつ実行したりできる。

EITP

**デミングサイクル(Deming cycle)：** 反復的な四つのステップからなる問題解決のプロセス(計画-実施-評価-改善)。特にプロセス改善において用いられる。[After Deming]

と

**統計的テスト(statistical testing)：** 入力 of 統計的な分散モデルを使い、代表的なテストケースを作成するテスト設計技法。operational profile testing も参照のこと。

**凍結テストベース(frozen test basis)：** 公式な変更コントロールプロセスによってのみ、改訂が可能なテストベースドキュメント。baseline も参照のこと。

F **統合(integration)：** コンポーネントやシステムを組み合わせ、さらに大きな集合体を作るプロセス。

F **統合テスト(integration testing)：** 統合したコンポーネントやシステムのインターフェースや相互作用の欠陥を摘出するためのテスト。component integration testing, system integration testing も参照のこと。

**同時処理テスト(concurrency testing)：** 同じ時間幅内に発生した二つ以上の処理をコンポーネントやシステムがどう扱うか(処理を分割して交互に実行したり、並列に実行したりするなど)を判定するためのテスト。[After IEEE 610]

**到達不能コード(unreachable code)：** 到達できないため、実行不能なコード。

**同値クラス(equivalence class)：** equivalence partition を参照のこと。

**同値分割(equivalence partition)：** 仕様に基づき、コンポーネントやシステムの振る舞いが同じとみなせる入力ドメインや出力ドメインの部分。

**同値分割カバレッジ(equivalence partition coverage)：** テストスイートが遂行した、同値分割した領域のパーセンテージ。

F  
ATA **同値分割法(equivalence partitioning)：** ブラックボックステスト設計技法の一つ。同値分割した領域から代表値を実行するテストケースを設計する。原則として、最低 1 回各同値分割した領域を実行するように設計する。

ATT **動的解析(dynamic analysis)：** 実行中のシステムやコンポーネントの振る舞い(たとえば、メモリの使用効率、CPU の使用状況)を評価するプロセス。[After IEEE 610]

F **動的解析ツール(dynamic analysis tool)：** ソフトウェアコードの状態に関する実行時(ランタイム)情報を提供するツール。未割り当てのポインタの識別、ポインタ計算のチェック、メモリの割り当て・使用・解放のチェック、メモリリークの検出で使うことが多い。

F **動的テスト(dynamic testing)：** コンポーネントやシステムのソフトウェアを実行させて確認するテスト。

**動的比較(dynamic comparison)：** (たとえば、テスト実行ツールを使用し)ソフトウェアの実行中に、実行結果と期待結果とを比較すること。

**ドキュメンテーションテスト(documentation testing)：** ユーザガイドやインストールガイドのようなドキュメントの品質をテストすること。

**トップダウンテスト(top-down testing)：** 統合テストを段階的に実施する方法。コンポーネント階層の最上位にあるコンポーネントからテストを開始する。下位レベルのコンポーネントは、スタブでシミュレーションする。上位コンポーネントのテストが終了すると、それを使って、下位コンポーネントをテストする。このプロセスは、最下位レベルのコンポーネントをテストするまで繰り返す。integration testing も参照のこと。

**ドメイン(domain)：** データのセット。ここから、有効な入力値や出力値を選ぶ。

ATA **ドメイン分析(domain analysis)：** ブラックボックステスト設計技法の一つ。複数の変数を同時にテス

トできる、又はテストする必要がある場合に効率的かつ効果的なテストケースを識別するのに使用される。同値分割法と境界値分析に基づいて構築され、これらを汎用化する。boundary value analysis, equivalence partitioning も参照のこと。

F **ドライバ(driver):** コンポーネントやシステムをコントロールしたり呼び出したりする上位コンポーネントの代わりとなるソフトウェアコンポーネントやテストツール。[After TMap]

F **トレーサビリティ(traceability):** ドキュメントとソフトウェアの関連事項(たとえば、ある要件と、それを検証するテストケース)を識別する能力。horizontal traceability, vertical traceability も参照のこと。

**トレーサビリティマトリクス(traceability matrix):** 2 つのエンティティ(たとえば、要件とテストケース)を関係付ける 2 次元の表。この表を使用すると、エンティティ間の関係を前工程及び後工程の双方向に追跡できるので、達成したカバレッジを測定でき、変更点の影響を評価できる。

## に

**入力(input):** コンポーネントが参照する変数。コンポーネント内部又は外部に格納されている。

**入力値(input value):** 入力のインスタンス。input も参照のこと。

**入力ドメイン(input domain):** 入力値のセット。ここから、有効なテストの入力値を選択する。domain も参照のこと。

**認定(certification):** コンポーネント、システム、人が、適格要件に従っていることを確認するプロセス。たとえば、試験に合格するなど。

## の

ATM  
EITP **能力成熟度モデル統合 (CMMI)(Capability Maturity Model Integration (CMMI)):** 効率的にプロダクトを開発・保守するためのプロセスの重要要素を記述したフレームワーク。プロダクトの開発・保守における計画、エンジニアリング、マネジメントでのベストプラクティスをカバーする。[CMMI]

## は

**ハイパーリンク(hyperlink):** ウェブページで、他のウェブページへ導くためのポインタ。

ATT **ハイパーリンクテストツール(hyperlink test tool):** Web サイト上に切断されたハイパーリンクが存在しないことを確認するために使用されるツール。

F **バグ(bug):** defect を参照のこと。

**バグトラッキングツール(bug tracking tool):** defect management tool を参照のこと。

**バグ分類法(bug taxonomy):** defect taxonomy を参照のこと。

**バグレポート(bug report):** defect report を参照のこと。

**ハザード分析(hazard analysis):** リスクの要素を特徴付けるために使う技法。ハザード分析の結果はシステム開発やテストのために使う手法を牽引する。risk analysis も参照のこと。

F	<p><b>バージョンコントロール(version control)</b>： configuration control を参照のこと。</p> <p><b>パス(path)</b>： コンポーネントやシステムで、開始点から終了点へ至るイベント(たとえば、実行ステートメント)の順列。</p> <p><b>パス[合格](pass)</b>： 実行結果が期待結果と一致した場合、テストは「合格」とみなす。</p> <p><b>パスカバレッジ(path coverage)</b>： テストスイートが遂行したパスのパーセンテージ。100%のパスカバレッジは 100%の LCSAJ カバレッジを意味する。</p> <p><b>パスセンシタイジング(path sensitizing)</b>： あるパスを実行させるため、入力値のセットを選択すること。</p>
ATT	<p><b>パステスト(path testing)</b>： ホワイトボックステスト設計技法の一つ。パスを実行するようテストケースを設計する。</p> <p><b>バックツーバックテスト(back-to-back testing)</b>： 二つ又はそれ以上の異なるコンポーネント又はシステムにおいて、同じ入力で実行し、出力を比較し、相違がある場合はそれを分析する。[IEEE 610]</p> <p><b>バッファ(buffer)</b>： 一時的にデータを保管するために使うデバイス若しくはストレージの領域。データのやりとりや利用に関わるデバイス若しくはプロセスによるデータフローの割合の違いやイベントの発生率やタイミングの違い、データ量の違いに対処することができる。[IEEE 610]</p> <p><b>バッファオーバーフロー(buffer overflow)</b>： 固定長のバッファの境界を越えてデータを格納しようとするプロセスの試みによって、結果的に隣接するメモリ領域を上書きしたり、オーバーフロー例外が発生してしまったりするなどのメモリアクセスの障害。buffer も参照のこと。</p> <p><b>パーティションテスト(partition testing)</b>： equivalence partitioning を参照のこと。[Beizer]</p> <p><b>ハードウェアソフトウェア統合テスト(hardware-software integration testing)</b>： ハードウェアコンポーネントとソフトウェアコンポーネントとの間のインターフェース及び相互作用に存在する欠陥を検出するために実行されるテスト。integration testing も参照のこと。</p> <p><b>パフォーマンスインジケータ(performance indicator)</b>： 効果や効率を示す高位レベルのメトリック。開発をコントロールし、方向性を決めるために利用する。たとえば、ソフトウェア開発のためのリードタイムの遅れ。[CMMI]</p>
EITP	<p><b>バランスドスコアカード(balanced scorecard)</b>： 企業の営業活動がその目的と整合しているかどうかをビジネスの構想や戦略の観点から評価するための戦略的ツール。corporate dashboard, scorecard も参照のこと。</p>
EITP	<p><b>バリューベースド品質(value-based quality)</b>： 品質に対する考え方の一つ。品質は価格で定義することができる。プロダクトやサービスの品質は、許容できるコスト範囲内で望まれるパフォーマンスを提供するものである、というもの。この品質はステークホルダ間で、期間、労力とコストのトレードオフに基づく意思決定プロセスによって決められる。[After Garvin] manufacturing-based quality, product-based quality, transcendent-based quality, user-based quality も参照のこと。</p>
EITP	<p><b>パレート分析(Pareto analysis)</b>： 意思決定における統計的手法の一つで、全体的影響を与える特定の重要な要因を選択するために使用される。品質改善の面では、問題の大半(80%)は少数の主要な原因(20%)によって発生する。</p> <p><b>バーンダウンチャート(burndown chart)</b>： 一つのイテレーションにおける残作業と時間の関係を表すために公開されるチャート。このチャートは、イテレーションにおけるタスクの完了状況と傾向を示す。一般的に、X 軸はスプリントにおける日数を表し、Y 軸は残作業の量(通常、理想的なエンジニアリングの時間又はストーリーポイントのどちらか)を表す。</p>

**判定(decision)：** 制御フローとして、二つ以上の選択可能なルートがあるプログラムポイント。分岐を切り分けるため、二つ以上のリンクを持ったノード。

**判定結果(decision outcome)：** 判定の結果(これにより、どの分岐を選択するかが決まる)。

**判定条件カバレッジ(decision condition coverage)：** テストスイートが遂行した条件と判定のパーセンテージ。100%の判定条件カバレッジは、100%の条件カバレッジと 100%のデジジョンカバレッジを意味する。

ATT **判定条件テスト(decision condition testing)：** ホワイトボックステスト設計技法の一つ。条件と判定の結果を実行するようテストケースを設計する。

F  
F-AT **反復開発モデル(iterative development model)：** プロジェクトをいくつかの(通常は、多数の)反復部分に分割して開発するライフサイクルの一種。一つの反復部分は、実行可能なプロダクトを(内部、あるいは、外部へ)リリースするという結果をもたらす、完全な開発ループである。

## ひ

F **ピアレビュー(peer review)：** 欠陥を識別して修正するため、プロダクト開発を担当している同僚がソフトウェア成果物をレビューすること。たとえばインスペクション、テクニカルレビュー、ウォークスルー。

**比較ツール(comparator)：** test comparator を参照のこと。

**非機能テスト(non-functional testing)：** コンポーネントやシステムで、機能に関係しない属性のテスト。たとえば、信頼性、効率性、使用性、保守性、移植性のテスト。

**非機能テスト設計技法(non-functional test design technique)：** 非機能テストケースを設計し、選択するための技法。コンポーネントやシステムの内部構造を参照することなく、仕様の分析に基づき設計する。black box test design technique も参照のこと。

F **非機能要件(non-functional requirement)：** 機能性に関係しない要件で、信頼性、効率性、使用性、保守性、移植性などの属性に関係する。

F  
ATM **非公式レビュー(informal review)：** 公式な(文書化した)処理手続きに基づかないレビュー。

**ビジネスプロセスベースドテスト(business process-based testing)：** ビジネスプロセスの記述や知識に基づきテストケースを設計するテスト手法。

**ビッグバンテスト(big-bang testing)：** 統合テストアプローチの一つ。ソフトウェアの構成要素、ハードウェアの構成要素、又は両方を、段階的にではなく、一挙にコンポーネントやシステムに結合して実施する。[After IEEE 610] integration testing も参照のこと。

**否定テスト(negative testing)：** コンポーネントやシステムが、正しく機能しないことを証明するためのテスト。否定テストは、特定のテストアプローチやテスト設計技法ではなく、テスト担当者の意向を反映したもの。たとえば、無効入力値や例外値を用いたテスト。[After Beizer]

**ビーバグging(bugging)：** [Abbott]fault seeding を参照のこと。

**BVT(BVT)：** build verification test を参照のこと。

ATA **ヒューリスティック評価(heuristic evaluation)：** ユーザインターフェース又はユーザインターフェー

ス設計での使用性の問題を対象にして、使用性をレビューする技法。この技法では、レビューアがユーザインターフェースを分析して、認識された使用性の原則（いわゆる、ヒューリスティクス）との適合性を判断する。

**評価(evaluation)：** testing を参照のこと。

**EITP** **標準(standard)：** 公式であり、場合によっては必須となる要件のセットで、ガイドラインを提供するため、又は作業の方法に一貫性のあるアプローチを規定するために、開発、使用するもの。(たとえば、ISO/IEC 標準、IEEE 標準や団体による標準) [After CMMI]

**ETM** **標準準拠テスト(standard-compliant testing)：** 標準により定義されている一連の要件に準拠していることを確認するテスト。たとえば、業界のテスト標準又はセーフティクリティカルシステムのテスト標準などがある。process-compliant testing も参照のこと。

**標準ソフトウェア(standard software)：** off-the-shelf software を参照のこと。

**標準適合性(compliance)：** 規格、規約又は法律上及び類似の法規上の規則を遵守するソフトウェア製品の能力。[ISO/IEC 9126] JSTQB 訳注) JIS X 0129-1:2003 より引用

**標準適合性テスト(compliance testing)：** コンポーネントやシステムの標準適合性を判定するテストのプロセス。

**標準テスト(standards testing)：** compliance testing を参照のこと。

**F-AT** **ビルド検証テスト(build verification test)：** 新たに作成した各ビルドに対して、完全性を確認し、主要な機能性、安定性、及び試験性を検証する自動テストのセット。ビルドが頻繁にリリースされる場合(たとえば、アジャイルプロジェクト)に、業界で一般的に行なわれるテストである。新たに作成したビルドは、このテストの完了後に、次のテストに向けてリリースされる。regression testing, smoke test も参照のこと。

**F** **品質(quality)：** コンポーネント、システム、プロセスが、特定の要件、ユーザ、顧客のニーズ、期待を満たす度合。[After IEEE 610]

**品質管理(quality control)：** 品質マネジメントの一環としての運用技法及び活動。品質要件を満たすことに重点を置く。[after ISO 8402]

**品質機能展開(quality function deployment)：** ユーザ要求を設計品質に反映し、品質を構成する機能を展開し、さらには設計品質を達成するための方法をサブシステム及びコンポーネントの部品に展開し、最終的には製造プロセスの特定の要素に展開する方法。[Akao]

**品質ゲート(quality gate)：** プロジェクトにおける特別なマイルストーン。品質ゲートは、フェーズ間に存在し、前フェーズの結果に強く依存する。品質ゲートには、その前フェーズのドキュメントの正式な確認が含まれる。

**品質コスト(cost of quality)：** 品質にかかる活動や問題のトータルコスト。予防コストと評価コスト、内部失敗コスト、外部失敗コストというように分ける場合が多い。

**品質特性(quality characteristic)：** quality attribute を参照のこと。

**品質の属性(quality attribute)：** アイテムの品質に影響を与えるフィーチャや特性。[IEEE 610]

**品質保証(quality assurance)：** 品質マネジメントの一部。品質要件を満たしていることの確信度合に焦点を当てている。[ISO 9000]

**品質マネジメント(quality management)：** 品質に関して組織を指揮し、管理するための調整された



活動。

注記 品質に関する指揮及び管理には、通常、品質方針及び品質目標の設定、品質計画、品質管理、品質保証及び品質改善が含まれる。[ISO 9000] JSTQB 訳注) JIS Q 9000:2006 より引用

F-AT  
ATM

**品質リスク(quality risk)：** 品質の属性に関連するプロダクトリスク。quality attribute, product risk も参照のこと。

## ふ

**ファンクションポイント法(Function Point Analysis (FPA))：** 情報システムの機能規模を測定する手法。FPAでの測定は、対象技術に依存しない。生産性測定、必要リソースの見積り、プロジェクトコントロールのベースとして利用できる。

F

**V 字モデル(V-model)：** 要求仕様からメンテナンスまでのソフトウェア開発ライフサイクル活動を表現したフレームワーク。V 字モデルは、ソフトウェア開発ライフサイクルの各フェーズに、各テスト活動がどのように対応しているかどうかを示している。

**フィーチャ(feature)：** 要求仕様ドキュメントで、明示的、暗示的に規定したコンポーネントやシステムの属性となる機能群(たとえば、信頼性、使用性、設計上の制約など)。[After IEEE 1008]

ETM

**フィーチャ駆動開発(feature-driven development)：** クライアントにとっての機能性価値(フィーチャ)の観点で駆動される、イテレーティブかつインクリメンタルなソフトウェア開発プロセス。フィーチャ駆動開発は、ほとんどの場合、アジャイルソフトウェア開発で使用する。agile software development も参照のこと。

**フィッシュボーンダイアグラム(fishbone diagram)：** cause-effect diagram を参照のこと。

F

**フィールドテスト(field testing)：** beta testing を参照のこと。

**フェイルオーバーテスト(failover testing)：** 管理されている環境で、故障モードをシミュレーションするか、実際に故障を発生させて行なうテスト。故障の後、フェイルオーバー機能をテストし、データの損失及び破損が発生しておらず、合意されている全てのサービスレベル(機能の可用性、応答時間など)を維持していることを確認する。recoverability testing も参照のこと。

**フェーズテスト計画(phase test plan)：** 主に、一つのテストフェーズを扱うテスト計画。test plan も参照のこと。

ATA  
ATM

**フェーズ内阻止(phase containment)：** ソフトウェアライフサイクルの同一フェーズ内における、混入した欠陥数に対する除去された欠陥数のパーセンテージ。

F

**フォールト(fault)：** defect を参照のこと。

**フォールトインJECTION(fault injection)：** システムが欠陥を検出し、欠陥から復旧できることを確認する目的で、意図的に欠陥をシステムに追加するプロセス。フィールドで発生する可能性のある故障を模倣することを目的とする。fault tolerance も参照のこと。

**フォールト検出率(Fault Detection Percentage (FDP))：** Defect Detection Percentage(DDP)を参照のこと。

F

**フォールト攻撃(fault attack)：** attack を参照のこと。

**フォールトシーディング(fault seeding)：** 残存欠陥数の見積りや検出、除去の割合を監視する目的でシステムやコンポーネントに欠陥を意図的に加えるプロセス。フォールトシーディングは、通常、開発

(プレリリース)テストの一環として実施され、全てのテストレベル(コンポーネント、統合、又はシステム)で実行できる。[After IEEE 610]

**ATT** **フォールトシーディングツール(fault seeding tool)**: システムやコンポーネントにフォールトを埋め込む(つまり、意図的に挿入する)ためのツール。

**EITP** **フォールトツリー解析(Fault Tree Analysis (FTA))**: フォールト(欠陥)の原因分析で使用する手法。故障やヒューマンエラーと、(特定の顕在化したフォールトと結びつく)外部イベントとの関係を論理的に視覚化したモデルにする技法。

**フォールトマスキング(fault masking)**: defect masking を参照のこと。

**フォールト密度(fault density)**: defect density を参照のこと。

**ATT** **不可分条件(atomic condition)**: これ以上分解できない条件、すなわち、論理演算子(AND、OR、XOR)で結合されている2つ以上の単一条件を含まない条件。

**複合条件(compound condition)**: 論理演算子(AND、OR、又は XOR)によって二つ又はそれ以上の単一条件が結合されたもの。たとえば、「A>B AND C>1000」。

**複合条件(multiple condition)**: compound condition を参照のこと。

**複合条件カバレッジ(multiple condition coverage)**: テストスイートが遂行した一つのステートメントの中にある全ての単一条件結果の組み合わせのパーセンテージ。100%の複合条件カバレッジは、100%の改良条件判定カバレッジを意味する。

**ATT** **複合条件テスト(multiple condition testing)**: ホワイトボックステスト設計技法の一つ。(一つのステートメント中の)単一条件結果の組み合わせを実行するテストケースを設計する技法。

**F** **複雑度(complexity)**: コンポーネントやシステムの設計・内部構造において、理解、保守、検証することが難しい度合。cyclomatic complexity も参照のこと。

**ATM** **不正(anomaly)**: 要求仕様、設計ドキュメント、ユーザドキュメント、標準など、又は知見、経験から逸脱するあらゆる状態。レビュー、テスト、分析、コンパイルをする中で検出できるが、それだけにとどまらず、ソフトウェアプロダクトや該当するドキュメントを利用するときに検出できることもある。[IEEE 1044] bug, defect, deviation, error, fault, incident, problem も参照のこと。

**不適合(non-conformity)**: 特定の要求を満足しないこと。[ISO 9000]

**ブラックボックス技法(black box technique)**: black box test design technique を参照のこと。

**F** **ブラックボックステスト(black box testing)**: コンポーネント又はシステムの内部構造を参照せずに行なう、機能的又は非機能的なテスト。

**F** **ブラックボックステスト設計技法(black box test design technique)**: コンポーネントやシステムの内部構造を参照することなく、機能仕様や非機能仕様の分析に基づきテストケースを設計、選択する技法。

**ブランチ(branch)**: 基本ブロックの一つ。二つ以上の実行可能なパスから、一つのパスを実行するために、プログラム構造に基づき選択された基本ブロック。たとえば、case、jump、go to、if-then-else。

**ブランチカバレッジ(branch coverage)**: テストスイートによって遂行された分岐のパーセンテージ。100%のブランチカバレッジは 100%のデシジョンカバレッジと 100%のステートメントカバレッジの両方を意味する。

**ブランチコンデション(branch condition):** 分岐条件。condition を参照のこと。

**ブランチコンデションカバレッジ(branch condition coverage):** condition coverage を参照のこと。

**ブランチコンデション組み合わせカバレッジ(branch condition combination coverage):** multiple condition coverage を参照のこと。

**ブランチコンデション組み合わせテスト(branch condition combination testing):** multiple condition testing を参照のこと。

**ブランチテスト(branch testing):** ホワイトボックステスト設計技法の一つ。分岐を実行するためのテストケースを設計する。

**ETM プランニングポーカ(Planning poker):** 合意に基づく見積り技法の一つ。アジャイルソフトウェア開発で、ユーザストーリーの作業量又は相対規模を見積るのに使用される。ワイドバンドデルファイ方法の一つの種類で、チームが見積る単位を表す一組のカードを使用する。agile software development, Wide Band Delphi も参照のこと。

**EITP 振り返りミーティング(retrospective meeting):** プロジェクトチームのメンバーでプロジェクトを評価し、次のプロジェクトに適用することができる教訓を学ぶために行なうプロジェクト終了時のミーティング。

**PRISMA(PRISMA (Product RiSk MAnagement)):** リスクベースドテストの体系的なアプローチ。プロダクトリスクの識別と分析を行ない、発生可能性及び影響に基づくプロダクトリスクマトリクスを作成する。

**振る舞い(behavior):** 入力値や事前条件のセットに対する、コンポーネントやシステムの反応。

**プレディケート(predicate):** 真偽を評価できるステートメント。後続の判定ロジックの制御フローを決定する場合がある。decision も参照のこと。

**プログラムインストルメンタ(program instrumenter):** instrumenter を参照のこと。

**プログラムテスト(program testing):** component testing を参照のこと。

**プロジェクト(project):** 開始日及び終了日を持ち、調整され、管理された一連の活動からなり、時間、コスト及び資源の制約を含む特定の要求事項に適合する目標を達成するために実施される特有のプロセス。[ISO 9000] JSTQB 訳注) JIS Q 9000:2006 より引用

**プロジェクト後ミーティング(post-project meeting):** retrospective meeting を参照のこと。

**プロジェクトテスト計画(project test plan):** master test plan を参照のこと。

**EITP プロジェクトの振り返り(project retrospective):** 次のプロジェクト又は次のプロジェクトフェーズを改善するために、教訓を学び、具体的なアクションプランを作成する構造化された方法。

**F ATM プロジェクトリスク(project risk):** (テスト)プロジェクトのマネジメントとコントロールに関係するリスク。たとえば、スタッフの不足、厳しい締め切り、変更管理などがこれに当たる。risk も参照のこと。

**プロセス(process):** 相互関係のある活動のセット。入力を出力に変換する。[ISO/IEC 12207]

**EITP プロセスアセスメント(process assessment):** 参照モデルを用いた、組織におけるソフトウェアプロセスの統制度合の評価。[After ISO/IEC 15504]

**プロセス改善(process improvement)：** 組織のプロセスの成熟度とパフォーマンスを改善するために設計した活動プログラムとその結果。[CMMI]

**プロセスサイクルテスト(process cycle test)：** ブラックボックステスト設計技法の一つ。ビジネスの手順やプロセスを実行するためのテストケースを設計する。[TMap] procedure testing も参照のこと。

**プロセス参照モデル(process reference model)：** プロセスモデルの一つ。ベストプラクティスの汎用的に使用できる部分と、事前に定義された手順を順に実行してプロセスを改善する方法とを提供する。

ETM **プロセス準拠性テスト(process-compliant testing)：** 定義されているプロセスに従って実行するテスト。プロセスは、たとえば、標準化委員会など外部の団体によって定義される。standard-compliant testing も参照のこと

EITP **プロセスモデル(process model)：** 共通の性質を持つプロセスによって全体的に統一されたモデルで表したフレームワーク。たとえば、テスト改善モデル。

EITP **プロダクトベース品質(product-based quality)：** 品質に対する考え方の一つ。品質は明確に定義された品質の属性のセットに基づく、というもの。これらの属性は、客観的かつ定量的な方法で測定する必要がある。同じ種類のプロダクト品質の違いは、特定の品質の属性を実装した方法に由来する。[After Garvin] manufacturing-based quality, quality attribute, transcendent-based quality, user-based quality, value-based quality も参照のこと。

F **プロダクトリスク(product risk)：** テスト対象に直接関係するリスク。risk も参照のこと。

F-AT  
ATM  
ATA  
ATT

**プロダクトリスクマネジメント(Product RiSk MANagement)：** PRISMA を参照のこと。

**ブロックドテストケース(blocked test case)：** 事前条件を満足できないため、実行不能のテストケース。

F **プローブ効果(probe effect)：** 性能テストツールやモニタなどでコンポーネントやシステムを測定する場合、測定ツールによって埋め込まれる測定のためのコード(インスツルメント)がコンポーネントやシステムに及ぼす影響。たとえば、性能テストツールを使うことによって、性能は若干悪化する。

ETM **分析的テスト(analytical testing)：** たとえば、製品リスク又は要件の体系的な分析に基づくテスト。

へ

**ペアテスト(pair testing)：** ふたり(たとえば、テスト担当者2名、開発担当者とテスト担当者が1名ずつ、エンドユーザとテスト担当者が1名ずつ)が、共同で欠陥を見つけること。テスト期間中、ふたりで1台のコンピュータを共有し、交互に使用することが多い。

**ペアプログラミング(pair programming)：** ソフトウェア開発のアプローチの一つ。1台のコンピュータを共有するふたりのプログラマが、(製品用、テスト用の)コードを書く。これにより、コードレビューをリアルタイムに実施できることを期待している。

ATA **ペアワイズテスト(pairwise testing)：** ブラックボックステスト設計技法の一つ。入力パラメータの各ペアを、設定可能な個々の組み合わせの全てで実行するためのテストケースを設計する。combinatorial testing, n-wise testing, orthogonal array testing も参照のこと。

ATT **ペアワイズ統合テスト(pairwise integration testing)**: 統合テストの一つの形態。コールグラフで示される、連携して動作するコンポーネントペアを対象にしてテストを行なう。

**平均故障間隔(MTBF)(Mean Time Between Failures)**: システムにおける故障から次の故障までの平均時間。MTBF は、通常、欠陥修正プロセスの一部として、障害が発生したシステムを直ちに修復することを想定した信頼度成長モデルの一部である。reliability growth model も参照のこと。

**平均故障間隔(MTBF)**: Mean Time Between Failures を参照のこと。

**平均修復時間(MTTR)(Mean Time To Repair)**: システムが故障から回復するまでに要する平均時間。これには一般的に欠陥が解決されていることを保証するテストが含まれる。

**平均修復時間(MTTR)**: Mean Time To Repair を参照のこと。

**ベーシックブロック(basic block)**: 分岐を含まない一つ以上の連続した実行ステートメント。注)制御フローグラフにおけるノードが基本ブロックを表す。

**ベーステストセット(basis test set)**: コンポーネントの内部構造や仕様から設計したテストケースのセット。特定のカバレッジ基準の 100%カバールを目的とする。

**ベストプラクティス(best practice)**: 与えられた状況下で、組織の作業効率に寄与する優れた手法や革新的な実践法。他の同等な組織から、「最善」と認められることが多い。

**ベースライン(baseline)**: 公式のレビュー、承認を受けた仕様やソフトウェアプロダクト。以降の開発の基礎となることを意図しており、変更は公式の変更コントロールプロセスによってのみ可能。[After IEEE 610]

F **ベータテスト(beta testing)**: 製品開発環境以外の外部環境で、現在、あるいは、将来のユーザや顧客が実施するテスト。コンポーネントやシステムが、ユーザや顧客のニーズを満たし、ビジネスプロセスに適合するかを判定する。マーケットからのフィードバックを得るため、既成ソフトウェアの外部受け入れテストの一形式として用いることが多い。

**変換テスト(conversion testing)**: 現行システムからリプレースするシステム用にデータ変換するソフトウェアに対するテスト。

EITP **変更管理(change management)**:  
(1)個人やチーム、組織を現在の状態から望ましい状態へと移行させるための構造化されたアプローチ。  
(2)プロダクトやサービスに対して変更あるいは提案された変更を達成するためのコントロールされた方法。  
configuration management も参照のこと。

**変更コントロール(change control)**: configuration control を参照のこと。

**変更コントロール委員会(change control board)**: configuration control board を参照のこと。

ATT **変更性(changeability)**: 指定された修正を行なうことができるソフトウェア製品の能力。[ISO/IEC 9126] maintainability も参照のこと。JSTQB 訳注) JIS X 0129-1:2003 より引用

**変数(variable)**: ソフトウェアプログラムから名前を参照することでアクセスできるコンピュータ中のストレージの要素。

**ベンチマークテスト(benchmark test)**:

(1)測定や比較のベースとなる標準。

(2)コンポーネントやシステム同士の比較をするためのテスト、又は、(1)の標準と比較するためのテスト。

## ほ

**ポインタ(pointer)**: 他のデータアイテムのロケーションを指定するデータアイテム。たとえば、処理される次の従業員レコードアドレスを指定したデータアイテム。[IEEE 610]

ETM **方法的テスト(methodical testing)**: チェックリスト、品質基準、汎用テストケースなど、テストの基準セットに基づくテスト。

**保守性(maintainability)**: 修正のしやすさに関するソフトウェア製品の能力。修正は、是正若しくは向上、又は環境の変化、要求仕様の変更及び機能仕様の変更にソフトウェアを適応させることを含めてもよい。[ISO/IEC 9126] JSTQB 訳注) JIS X 0129-1:2003 より引用

F  
F-AT  
ATT **保守性テスト(maintainability testing)**: ソフトウェア製品の保守性を判定するテストのプロセス。

**保守性テスト(serviceability testing)**: maintainability testing を参照のこと。

F **保守テスト(maintenance testing)**: 運用システム自体の変更や、稼動環境の変更が運用システムに与える影響をテストすること。

**ボトムアップテスト(bottom-up testing)**: 段階的に統合テストを進める手法の一つ。最下位レベルのコンポーネントを最初にテストし、そのコンポーネントを利用して上位コンポーネントをテストする方式。最上位コンポーネントをテストするまで、この手順を繰り返す。integration testing も参照のこと。

**ボリュームテスト(volume testing)**: システムに大量のデータを与えるテスト。resource utilization testing を参照のこと。

**ホワイトボックス技法(white-box technique)**: white-box test design technique を参照のこと。

F **ホワイトボックステスト(white-box testing)**: コンポーネント又はシステムの内部構造の分析に基づいたテスト。

F  
ATT **ホワイトボックステスト設計技法(white-box test design technique)**: コンポーネントやシステムの内部構造の分析に基づきテストケースを設計、選択する技法。

## ま

ETM **マイヤーズ - ブリックスタイプ指標(MBTI) (Myers-Briggs Type Indicator (MBTI))**: 人々のさまざまな性格及びコミュニケーションスタイルを表す、個人の心理学的な傾向の指標。

**マイルストーン(milestone)**: (中間)成果物と結果の準備完了が定義されている、プロジェクトのある一時点。

EITP **マインドマップ(mind map)**: 中心となるキーワードやアイデアの周囲に、用語、アイデア、タスク、又は他の項目を表現し、関連付け、配置するために使われる図。マインドマップは、アイデアを作り出し、視覚化し、構造化し、分類するために記述する。学習、組織化、問題解決、意思決定、文書作成の補助として活用される。

ATM **マスターテスト計画(master test plan)**: 通常、複数のテストレベルを扱うテスト計画。test plan も参

照のこと。

**EITP** **マニファクチャリングベースド品質(manufacturing-based quality)**: 品質に対する考え方の一つ。品質はプロダクト又はサービスが、その意図した設計と要件に準拠する度合によって測定される、というもの。品質は、使用されるプロセス(群)に依存する。[After Garvin] product-based quality, transcendent-based quality, user-based quality, value-based quality も参照のこと。

**ATM** **マネジメントレビュー(management review)**: ソフトウェアの取得、供給、開発、運用、保守、といったプロセスを体系的に評価すること。管理者や管理者の代理者が、進捗を監視し、計画やスケジュールの状態を判定し、要件やシステム割り当てを確認し、目的遂行用に最適化されるようマネジメントアプローチの効率を評価する。[After IEEE 610, IEEE 1028]

## み

**見逃された欠陥(escaped defect)**: 以前のテストレベルで検出することを想定していたタイプの欠陥で、そのテストレベルでは検出されなかった欠陥。Defect Detection Percentage も参照のこと。

**ミューテーション解析(mutation analysis)**: テストスイートの完全性を判定する手法の一つ。プログラムのわずかな変形(変異)をテストスイートがどの程度識別できるかを測定する。

**ミューテーションテスト(mutation testing)**: back-to-back testing を参照のこと。

**ATA** **魅力性(attractiveness)**: 利用者にとって魅力的であるためのソフトウェア製品の能力。[ISO/IEC 9126] usability も参照のこと。JSTQB 訳注) JIS X 0129-1:2003 より引用

## め

**F** **EITP** **メトリック(metric)**: 測定尺度、及び、測定手法。[ISO/IEC 14598]

**ATT** **メモリリーク(memory leak)**: プログラムの動的領域割り当てロジックの欠陥によって起こるメモリアクセス異常。この欠陥により、使用メモリの解放ができなくなる。メモリ不足によって、最終的にプログラムや他の並行プロセスが失敗する。

**メンテナンス(maintenance)**: リリース後のソフトウェア製品を変更すること。欠陥の修正、性能や他の特性の改善、変更した環境への製品適合などを目的とする。[IEEE 1219]

## も

**モジュール(module)**: component を参照のこと。

**モジュールテスト(module testing)**: component testing を参照のこと。

**F** **モデリングツール(modeling tool)**: ソフトウェアやシステムのモデルの作成、修正及び妥当性確認を支援するツール。[Graham]。

**ETM** **モデルベースドテスト(model-based testing)**: テスト対象のコンポーネント又はシステムのモデルに基づくテスト。モデルとしては、たとえば、信頼度成長モデル、運用プロファイルなどの使用モデル、デシジョンテーブル又は状態遷移図などの振る舞いモデルなどが使用される。

F  
ATM **モデレータ(moderator)**: インспекションや、他のレビュープロセスに責任を持つ、リーダーや中心人物。

**モニタ(monitor)**: テスト時に、コンポーネントやシステムと同時に実行し、コンポーネントやシステムの振る舞いを監視、記録、分析するソフトウェアツールやハードウェア装置。[After IEEE 610]

F **モニタリングツール(monitoring tool)**: monitor を参照のこと。

**モンキーテスト(monkey testing)**: 製品の使用方法については全く考慮せず、広範囲の入力からランダムに選択し、ボタンをランダムに押すことでテストを行なう方法。

**問題(problem)**: defect を参照のこと。

**問題マネジメント(problem management)**: defect management を参照のこと。

**問題レポート(problem report)**: defect report を参照のこと。

## ゆ

**有限状態機械(finite state machine)**: 計算モデルの一つ。有限個数の状態と、それらの状態間遷移から構成される。状態遷移に対応する動作も記述できる。[IEEE 610]

**有限状態テスト(finite state testing)**: state transition testing を参照のこと。

ATM **有効性(effectiveness)**: 意図する結果を生成する能力。efficiency も参照のこと。

ATM **優先度(priority)**: あるアイテム(たとえば、欠陥)に割り当てた(ビジネス上の)重要さのレベル。

F **ユーザ受け入れテスト(user acceptance testing)**: acceptance testing を参照のこと。

**ユーザシナリオテスト(user scenario testing)**: use case testing を参照のこと。

F-AT **ユーザストーリー(user story)**: 高位のユーザ要件又はビジネス要件。主に、アジャイルソフトウェア開発で用いられる。ユーザが要求とする機能及びあらゆる非機能を獲得し、日常言語又はビジネス言語で表現される一つ以上の文で構成する。また、受け入れ基準も含む。agile software development, requirement も参照のこと。

ATA **ユーザストーリーテスト(user story testing)**: ブラックボックステスト設計技法の一つ。ユーザストーリーが正しく実装されていることを検証するために、テストケースをユーザストーリーに基づいて設計する。user story も参照のこと。

**ユーザテスト(user test)**: コンポーネント又はシステムの使用性を評価するために現実のユーザが参加するテスト。

EITP **ユーザベース品質(user-based quality)**: 品質に対する考え方の一つ。ユーザの必要性、要望、願望を、どこまで満たすか、というもの。要求を満たしていないプロダクトやサービスは、いかなるユーザの支持も得ることができない。この(ユーザベース)品質は、背景によって異なる流動的なアプローチである。なぜならば、プロダクトごとに異なる品質の側面によりビジネスに求められる特徴が変わってくるためである。[after Garvin] manufacturing-based quality, product-based quality, transcendent-based quality, value-based quality も参照のこと。

**ユースケース(use case)**: アクターとコンポーネント又はシステムとの間の対話における一連のトランザクション。視覚できる結果を伴う。アクターは、ユーザ又はシステムと情報交換するあらゆるものになり



うる。

F  
ATA **ユースケーステスト(use case testing)**: ブラックボックステスト設計技法の一つ。ユースケースのシナリオを実行するテストケースを設計する。

**ユニット(unit)**: component を参照のこと。

**ユニットテスト(unit testing)**: component testing を参照のこと。

F  
F-AT **ユニットテストフレームワーク(unit test framework)**: 適切なスタブやドライバを併用した状態、又は独立した状態でコンポーネントをテストできるユニットテスト又はコンポーネントテスト用の環境を提供するツール。デバッグ機能など、開発者を支援するその他の機能もある。[Graham]

## よ

F **要件(requirement)**: ユーザが問題解決や目的を達成するために必要な条件や能力。契約、標準、仕様、その他の公式ドキュメントを満足するために、システムやシステムコンポーネントが満たし、保持すべき条件や能力。[After IEEE 610]

**要件フェーズ(requirements phase)**: ソフトウェアライフサイクルの一区切りの期間。ソフトウェアプロダクトの要件を定義、文書化する。[IEEE 610]

ATA **要件ベースドテスト(requirements-based testing)**: テストのアプローチの一つ。要件から導き出したテスト目的とテスト条件を基にしてテストケースを設計する。たとえば、特定の機能を遂行するテスト、又は、信頼性や使用性のような非機能属性を調べるテスト。

F **要件マネジメントツール(requirements management tool)**: 要件、要件属性(たとえば、優先順位、信頼できる情報元)、注釈を記録し、要件の階層をたどる追跡や、要件変更管理を支援するツール。要件マネジメントツールの中には、あらかじめ定義した要件規約を基に、整合性や違反をチェックするような、静的解析をするものもある。

**予測結果(predicted outcome)**: expected result を参照のこと。

**予備テスト(pretest)**: intake test を参照のこと。

## ら

EITP **ライフサイクルモデル(lifecycle model)**: プロダクト又はプロジェクトの全期間をフェーズに分割したもの。[CMMI] software lifecycle も参照のこと。

EITP **ラショナル統一プロセス(Rational Unified Process)**: 独自の適応性を持つ反復ソフトウェア開発プロセスフレームワーク。方向付け、推敲、作成、及び移行の四つのプロジェクトライフサイクルフェーズで構成される。

EITP **RUP(RUP)**: Rational Unified Process を参照のこと。

**ランダムテスト(random testing)**: ブラックボックステスト設計技法の一つ。擬似ランダム生成アルゴリズム等を使い、運用プロファイルに合致するテストケースを設計する。信頼性、性能等、非機能属性のテストに利用できる。

## リ

ATA **理解性(understandability)**: ソフトウェアが特定の作業に特定の利用条件で適用できるかどうか、及びどのように利用できるかを利用者が理解できるソフトウェア製品の能力。[ISO/IEC 9126] usability も参照のこと。JSTQB 訳注)JIS X 0129-1:2003 より引用

F **リスク(risk)**: 将来、否定的な結果を生む要素。通常、影響度と発生可能性として表現する。  
ATM

ATM **リスクアセスメント(risk assessment)**: リスクのレベルを決定するために、特定のプロジェクトリスク  
ATT 又はプロダクトリスクを識別し、その後分析するプロセス。一般的に発生確率と影響度を割り当てること  
によって行なう。product risk, project risk, risk, risk impact, risk level, risk likelihood も参照のこと。

**リスクカテゴリ(risk category)**: risk type を参照のこと。

ATM **リスク軽減(risk mitigation)**: risk control を参照のこと。  
ATA  
ATT

ATM **リスクコントロール(risk control)**: 特定のレベルまでリスクを減らす(あるいは、リスクレベルを維持す  
ATA る)ために、判定を下したり、対策したりするプロセス。  
ATT

ATM **リスク識別(risk identification)**:ブレインストーミング、チェックリスト、故障履歴などを使ったリスクを  
ATT 識別するプロセス。

**リスクタイプ(risk type)**: 一つ若しくは複数の共通の要因(品質の属性、発生原因、ロケーション、又は  
リスクの潜在する影響)で分類されたリスクのセット。リスクタイプを軽減(コントロール)できるテストの  
タイプと関連付けられる特定のプロダクトリスクのタイプ。たとえば、誤認識のようなユーザの相互作用の  
リスクは、使用性テストによって軽減することができる。

**リスクの影響(risk impact)**: リスクが実際の結果又は事象となった場合に引き起こされる損害。

**リスク発生確率(risk likelihood)**: リスクが実際の結果又は事象となりえる推定確率。

ATM **リスク分析(risk analysis)**: リスクのレベルを決定するために、特定のプロジェクトリスク又はプロダク  
ATA トリスクを評価するプロセス。一般的に、それらのリスクの影響度と発生確率(可能性)を見積ることによ  
ATT って行なう。

F **リスクベースドテスト(risk-based testing)**: プロジェクトの初期段階からプロダクトリスクのレベルを  
ATM 低減させ、ステークホルダーにその状態を通知するテストの方法。プロダクトリスクの識別の他、テストプロ  
ATA セスをガイドする際のリスクレベルの活用もこれに含まれる。  
ATT

ATM **リスクマネジメント(risk management)**: リスクの識別、分析、優先順位付け、コントロールのタスクに  
ATA 手順や実施方法を体系的に適用すること。

ATM **リスクレベル(risk level)**: 影響度合と発生頻度からみた特徴で定義したリスクの重要性。リスクのレ  
ATA ベルはテストを行なうときの強弱を決定するときに使われる。リスクレベルはまた、定性的(高/中/低な  
ATT ど)又は定量的に表現できる。

EITP **リードアセッサー(lead assessor)**: アセスメントを主導する人。CMMi や TMMi などの公式なアセス  
メントを行なう場合は、リードアセッサーは、公式な訓練を受けて認定を受ける必要がある。

**リリースノート(release note)**: テストアイテム、その構成、現在の状態、その他のリリース情報を記述  
したドキュメント。テスト実行フェーズの開始前に、開発側からテスト関係者へ、場合によって他のステー

クホルダに提供する。[After IEEE 829]

**リンクテスト(link testing)**： component integration testing を参照のこと。

## れ

**例外処理(exception handling)**： 内部の故障に対する、又は、人間や他のコンポーネントやシステムからの誤入力に対するコンポーネントやシステムの振る舞い。

F  
ATM **レビュー(review)**： プロダクトやプロジェクトの状態を評価する手法。計画した結果との違いを分析し、改善を提案する。例として、マネジメントレビュー、非公式レビュー、テクニカルレビュー、インスペクション、ウォークスルーがある。[After IEEE 1028]

F  
ATM **レビューア(reviewer)**： レビューの中でレビュー対象のプロダクトやプロジェクトの不整合を識別し、指摘する人。レビューアは、レビュープロセスでさまざまな視点や役割を担った人達を選ばれる。

ATM **レビュー計画(review plan)**： 意図するレビュー活動のアプローチ、リソース、及びスケジュールを記述するドキュメント。特に、レビュー対象のドキュメントとコード、使用するレビューの種類、参加者、公式レビューの場合に適用される開始と終了の基準、それらを選択した理由などが識別される。レビュー計画プロセスの記録である。

F **レビューツール(review tool)**： レビュープロセスを支援するツール。レビューの計画、追跡の支援、コミュニケーションのサポート、協調的なレビューの推進、収集や報告用のメトリクスを保存する機能を備えていることが多い。

ATM **レベルテスト計画(level test plan)**： 通常、一つのテストレベルを扱うテスト計画。test plan も参照のこと。

EITP **連続表現(continuous representation)**： 特定のプロセスエリア群内でのプロセス改善アプローチのための推奨順番として提供された、能力レベルをみる成熟度モデル構造。[CMMI]

## ろ

F **ロードテスト(load testing)**： コンポーネントやシステムの振る舞いを測定する性能テストの一種。負荷(たとえば、同時実行ユーザ数やトランザクションの数)を増加させ、コンポーネントやシステムがどの程度の負荷に耐えられるか判定する。performance testing, stress testing も参照のこと。

F **ロードテストツール(load testing tool)**： ロードテストをサポートするツール。特定の時間内に、負荷(たとえば、同時実行ユーザ数やトランザクションの数)を増加させるシミュレーションを実行できる。performance testing tool も参照のこと。

**ロードプロファイル(load profile)**： テストされるコンポーネントやシステムが稼動中にユーザが行なうであろう活動内容を記した仕様書。ロードプロファイルは、特定の時間内に所定の運用プロファイルに準じ決められたトランザクションを処理する仮想ユーザを複数指定して構成する。operational profile も参照のこと。

F **ロバストネステスト(robustness testing)**： ソフトウェア製品の頑健性(堅牢性)を判定するテスト。

**論理カバレッジテスト(logic-coverage testing)**： white-box testing を参照のこと。[Myers]

**論理駆動テスト(logic-driven testing)**： white-box testing を参照のこと。

ATA      **論理的テストケース(logical test case)**: high level test case を参照のこと。

## わ

ATM      **ワイドバンドデルファイ(Wide Band Delphi)**: 専門家によるテスト見積り技法。チームメンバーから集めた知識を用い、正確な見積りをするもの。

ATT      **ワイルドポインタ(wild pointer)**: 範囲外、若しくは、存在しない位置を参照しているポインタ。pointer も参照のこと。

**ワークブレイクダウストラクチャ(Work Breakdown Structure)**: 作業の要素を整理し、それらを最終プロダクトと関連付けたもの。[CMMI]

## 付録 A(参考資料)

### 標準

- [DO-178b] DO-178B:1992.Software Considerations in Airborne Systems and Equipment Certification, Requirements and Technical Concepts for Aviation (RTCA SC167)
- [IEEE 610] IEEE 610.12:1990.Standard Glossary of Software Engineering Terminology
- [IEEE 829] IEEE 829:1998.Standard for Software Test Documentation
- [IEEE 1008] IEEE 1008:1993.Standard for Software Unit Testing
- [IEEE 1028] IEEE 1028:1997.Standard for Software Reviews and Audits
- [IEEE 1044] IEEE 1044:1993.Standard Classification for Software Anomalies
- [IEEE 1219] IEEE 1219:1998.Software Maintenance
- [ISO 2382/1] ISO/IEC 2382-1:1993.Data processing - Vocabulary - Part 1:Fundamental terms  
JSTQB 訳注) 日本では JIS X 0001 として発行されている。
- [ISO 8402] ISO 8402: 1994. Quality Management and Quality Assurance Vocabulary
- [ISO 9000] ISO 9000:2005.Quality Management Systems – Fundamentals and Vocabulary  
JSTQB 訳注) 日本では JIS Q 9000 として発行されている。
- [ISO 9126] ISO/IEC 9126-1:2001.Software Engineering – Software Product Quality – Part 1: Quality characteristics and sub-characteristics  
JSTQB 訳注) 日本では JIS X 0129-1 として発行されている。
- [ISO 12207] ISO/IEC 12207:1995.Information Technology – Software Lifecycle Processes  
JSTQB 訳注) 日本では JIS X 0160 として発行されている。
- [ISO 14598] ISO/IEC 14598-1:1999.Information Technology – Software Product Evaluation - Part 1:General Overview  
JSTQB 訳注) 日本では JIS X 0133-1 として発行されている。
- [ISO 15504] ISO/IEC 15504-9:1998.Information Technology – Software Process Assessment – Part 9:Vocabulary

### 引用文献

- [Abbott] J. Abbot (1986), *Software Testing Techniques*, NCC Publications
- [Adrian] W. Adrian, M. Branstad and J. Cherniabsky (1982), Validation, Verification and Testing of Computer Software, in: *Computing Surveys*, Vol. 14, No 2, June 1982
- [Akao] Akao, Yoji (1994), Development History of Quality Function Deployment – The Customer Driven Approach to Quality Planning and Deployment, Minato, Tokyo 107 Japan: Asian Productivity Organization, pp. 339, ISBN 92-833-1121-3
- [Bach] J. Bach (2004), Exploratory Testing, in: E. van Veenendaal, *The Testing Practitioner – 2<sup>nd</sup> edition*, UTN Publishing, ISBN 90-72194-65-9
- [Beizer] B. Beizer (1990), *Software Testing Techniques*, van Nostrand Reinhold, ISBN 0-442-20672-0
- [Chow] T. Chow (1978), Testing Software Design Modelled by Finite-State Machines, in: *IEEE Transactions on Software Engineering*, Vol. 4, No 3, May 1978
- [CMM] M. Paulk, C. Weber, B. Curtis and M.B.Chrissis (1995), *The Capability Maturity Model, Guidelines for Improving the Software Process*, Addison-Wesley, ISBN 0-201-54664-7
- [CMMI] M.B.Chrissis, M. Konrad and S. Shrum (2004), *CMMI, Guidelines for Process Integration and Product Improvement*, Addison Wesley, ISBN 0-321-15496-7
- [Deming] D. W. Edwards (1986), *Out of the Crisis*, MIT Center for Advanced Engineering Study, ISBN 0-911379-01-0
- [Fenton] N. Fenton (1991), *Software Metrics: a Rigorous Approach*, Chapman & Hall, ISBN 0-53249-425-1
- [Fewster and Graham] M. Fewster and D. Graham (1999), *Software Test Automation, Effective use of test execution tools*, Addison-Wesley, ISBN 0-201-33140-3

- [Freedman and Weinberg] D. Freedman and G. Weinberg (1990), *Walkthroughs, Inspections, and Technical Reviews*, Dorset House Publishing, ISBN 0-932633-19-6
- [Garvin] D.A. Garvin (1984), What does product quality really mean?, in: *Sloan Management Review*, Vol. 26, nr. 1 1984
- [Gerrard] P. Gerrard and N. Thompson (2002), *Risk-Based E-Business Testing*, Artech House Publishers, ISBN 1-58053-314-0
- [Gilb and Graham] T. Gilb and D. Graham (1993), *Software Inspection*, Addison-Wesley, ISBN 0-201-63181-4
- [Graham] D. Graham, E. van Veenendaal, I. Evans and R. Black (2007), *Foundations of Software Testing*, Thomson Learning, ISBN 978-1-84480-355-2
- [Grochtmann] M. Grochtmann (1994), Test Case Design Using Classification Trees, in: *Conference Proceedings STAR 1994*
- [Hetzel] W. Hetzel (1988), *The complete guide to software testing – 2<sup>nd</sup> edition*, QED Information Sciences, ISBN 0-89435-242-3
- [Juran] J.M. Juran (1979), *Quality Control Handbook*, McGraw-Hill
- [McCabe] T. McCabe (1976), A complexity measure, in: *IEEE Transactions on Software Engineering*, Vol. 2, pp. 308-320
- [Musa] J. Musa (1998), *Software Reliability Engineering Testing*, McGraw-Hill Education, ISBN 0-07913-271-5
- [Myers] G. Myers (1979), *The Art of Software Testing*, Wiley, ISBN 0-471-04328-1
- [TMap] M. Pol, R. Teunissen, E. van Veenendaal (2002), *Software Testing, A guide to the TMap Approach*, Addison Wesley, ISBN 0-201-745712
- [TMMi] E. van Veenendaal and J. Cannegieter (2011), *The Little TMMi*, UTN Publishing, ISBN 97-89490986-03-2
- [Veenendaal04] E. van Veenendaal (2004), *The Testing Practitioner – 2<sup>nd</sup> edition*, UTN Publishing, ISBN 90-72194-65-9
- [Veenendaal08] E. van Veenendaal (2008), Test Improvement Manifesto, in: *Testing Experience*, Issue 04/08, December 2008

## 索引(英語)

### A

abstract test case	35
acceptance	14
acceptance criteria	14
acceptance testing	14
accessibility testing	11
accuracy	30
accuracy testing	30
acting (IDEAL)	23
action word driven testing	11
actor	11
actual outcome	26
actual result	26
ad hoc review	11
ad hoc testing	11
adaptability	17
agile manifesto	11
agile software development	11
agile testing	11
algorithm test	12
alpha testing	12
analytical testing	52
analyzability	16
analyzer	11
anomaly	50
anti-pattern	12
API (Application Programming Interface) testing	15
arc testing	11
assessment report	11
assessor	11
atomic condition	50
attack	22
attack-based testing	22
attractiveness	55
audit	18
audit trail	18
automated testware	27
availability	17

### B

back-to-back testing	46
balanced scorecard	46
baseline	53
basic block	53
basis test set	53
bugging	47
behavior	51
benchmark test	53
bespoke software	35
best practice	53
beta testing	53
big-bang testing	47
black box technique	50
black box test design technique	50

black box testing	50
blocked test case	52
bottom-up testing	54
boundary value	19
boundary value analysis	19
boundary value coverage	19
boundary value testing	19
branch	50
branch condition	50
branch condition combination coverage	51
branch condition combination testing	51
branch condition coverage	50
branch coverage	50
branch testing	51
buffer	46
buffer overflow	46
bug	45
bug report	45
bug taxonomy	45
bug tracking tool	45
build verification test	48
burndown chart	46
business process-based testing	47
BVT	47

### C

call graph	24
Capability Maturity Model Integration (CMMI)	45
capture/playback tool	19
capture/replay tool	19
CASE	20
CAST	19
causal analysis	22
cause-effect analysis	22
cause-effect decision table	22
cause-effect diagram	22
cause-effect graph	21
cause-effect graphing	22
certification	45
change control	53
change control board	53
change management	53
changeability	53
charter	35
checker	35
checklist-based testing	35
Chow's coverage metrics	35
classification tree	20
classification tree method	20
clear-box testing	20
CMMI	26
code	24
code analyzer	24
code coverage	24
code-based testing	24
codependent behavior	19
co-existence	19

combinatorial testing	20
Commercial Off-The-Shelf software	27
comparator	47
compatibility testing	24
compiler	25
complete testing	18
completion criteria	27
complexity	50
compliance	48
compliance testing	48
component	25
component integration testing	25
component specification	25
component testing	25
compound condition	50
concrete test case	20
concurrency testing	44
condition	28
condition combination coverage	28
condition combination testing	28
condition coverage	28
condition determination coverage	28
condition determination testing	28
condition outcome	28
condition testing	28
confidence interval	29
confidence test	25
configuration	22
configuration auditing	22
configuration control	23
configuration control board (CCB)	23
configuration identification	23
configuration item	22
configuration management	22
configuration management tool	23
configuration testing	23
confirmation testing	17
conformance testing	36
consistency	12
consultative testing	24
content reference model	24
content-based model	24
continuous representation	59
control chart	18
control flow	31
control flow analysis	31
control flow graph	31
control flow path	31
control flow testing	31
convergence metric	27
conversion testing	53
corporate dashboard	24
cost of quality	48
COTS	24
coverage	17
coverage analysis	17
coverage item	17
coverage measurement tool	17
coverage tool	17
critical success factor	27
Critical Testing Processes	20
CTP	27

custom software	17
custom tool	17
cyclomatic complexity	25
cyclomatic number	25

---

## *D*

daily build	36
dashboard	34
data definition	43
data flow	43
data flow analysis	43
data flow coverage	43
data flow testing	43
data integrity testing	43
data quality	43
database integrity testing	43
data-driven testing	43
dd-path	36
dead code	43
debugger	43
debugging	43
debugging tool	43
decision	46
decision condition coverage	47
decision condition testing	47
decision coverage	36
decision outcome	47
decision table	36
decision table testing	36
decision testing	36
defect	20
defect category	21
defect density	21
Defect Detection Percentage (DDP)	21
defect management	21
defect management committee	21
defect management tool	21
defect masking	21
defect report	21
defect taxonomy	21
defect tracking tool	21
defect triage committee	21
defect type	21
defect-based technique	21
defect-based test design technique	21
definition-use pair	36
deliverable	31
Deming cycle	43
design-based testing	36
desk checking	18
development testing	16
deviation	12
deviation report	36
diagnosing (IDEAL)	29
dirty testing	34
documentation testing	44
domain	44
domain analysis	44
driver	45
dynamic analysis	44



dynamic analysis tool .....	44
dynamic comparison .....	44
dynamic testing .....	44

---

## *E*

effectiveness .....	56
efficiency .....	23
efficiency testing .....	23
EFQM (European Foundation for Quality Management) excellence model .....	12
elementary comparison testing .....	19
embedded iterative development model .....	14
emotional intelligence .....	15
emulator .....	15
entry criteria .....	16
entry point .....	16
equivalence class .....	44
equivalence partition .....	44
equivalence partition coverage .....	44
equivalence partitioning .....	44
error .....	15
error guessing .....	15
error seeding .....	15
error seeding tool .....	15
error tolerance .....	15
escaped defect .....	55
establishing (IDEAL) .....	17
evaluation .....	48
exception handling .....	59
executable statement .....	26
exercised .....	29
exhaustive testing .....	32
exit criteria .....	27
exit point .....	27
expected outcome .....	18
expected result .....	18
experience-based technique .....	20
experience-based test design technique .....	20
experience-based testing .....	20
exploratory testing .....	35
extreme programming (XP) .....	15

---

## *F*

factory acceptance testing .....	22
fail .....	27
failover testing .....	49
failure .....	24
failure mode .....	24
Failure Mode and Effect Analysis (FMEA) .....	24
Failure Mode, Effects, and Criticality Analysis (FMECA) .....	24
failure rate .....	24
FALSE-fail result .....	12
FALSE-negative result .....	18
FALSE-pass result .....	12
FALSE-positive result .....	19
fault .....	49
fault attack .....	49

fault density .....	50
Fault Detection Percentage (FDP) .....	49
fault injection .....	49
fault masking .....	50
fault seeding .....	49
fault seeding tool .....	49
fault tolerance .....	28
Fault Tree Analysis (FTA) .....	50
feasible path .....	26
feature .....	49
feature-driven development .....	49
field testing .....	49
finite state machine .....	56
finite state testing .....	56
fishbone diagram .....	49
formal review .....	22
frozen test basis .....	44
Function Point Analysis (FPA) .....	49
functional integration .....	18
functional requirement .....	19
functional test design technique .....	18
functional testing .....	18
functionality .....	18
functionality testing .....	18

---

## *G*

glass box testing .....	20
Goal Question Metric .....	24
GQM .....	26

---

## *H*

hardware-software integration testing .....	46
hazard analysis .....	45
heuristic evaluation .....	47
high level test case .....	22
horizontal traceability .....	29
hyperlink .....	45
hyperlink test tool .....	45

---

## *I*

IDEAL .....	11
impact analysis .....	14
incident .....	13
Incident logging .....	13
incident management .....	13
incident management tool .....	13
incident report .....	13
incremental development model .....	12
incremental testing .....	13
independence of testing .....	41
indicator .....	27
infeasible path .....	27
informal review .....	47
initiating (IDEAL) .....	16
input .....	45
input domain .....	45

input value	45
insourced testing	13
inspection	13
inspection leader	13
inspector	13
installability	32
installability testing	32
installation guide	13
installation wizard	13
instrumentation	13
instrumenter	13
intake test	13
integration	44
integration testing	44
integration testing in the large	34
integration testing in the small	28
interface testing	13
interoperability	33
interoperability testing	33
invalid testing	13
Ishikawa diagram	12
isolation testing	11
item transmittal report	11
iterative development model	47

---

## *K*

key performance indicator	19
keyword-driven testing	19

---

## *L*

LCSAJ	15
LCSAJ coverage	15
LCSAJ testing	15
lead assessor	58
learnability	27
learning (IDEAL)	17
level test plan	59
lifecycle model	57
link testing	58
load profile	59
load testing	59
load testing tool	59
logical test case	59
logic-coverage testing	59
logic-driven testing	59
low level test case	35

---

## *M*

maintainability	54
maintainability testing	54
maintenance	55
maintenance testing	54
man in the middle attack	35
management review	55
manufacturing-based quality	54
master test plan	54

maturity	31
maturity level	31
maturity model	31
MCDC	15
Mean Time Between Failures	52
Mean Time To Repair	53
measure	33
measurement	33
measurement scale	33
memory leak	55
methodical testing	54
metric	55
migration testing	12
milestone	54
mind map	54
mistake	12
model-based testing	55
modeling tool	55
moderator	55
modified condition decision testing	17
modified multiple condition coverage	17
modified multiple condition testing	17
module	55
module testing	55
monitor	55
monitoring tool	56
monkey testing	56
MTBF	53
MTRR	53
multiple condition	50
multiple condition coverage	50
multiple condition testing	50
mutation analysis	55
mutation testing	55
Myers-Briggs Type Indicator (MBTI)	54

---

## *N*

negative testing	47
neighborhood integration testing	20
non-conformity	50
non-functional requirement	47
non-functional test design technique	47
non-functional testing	47
N-switch coverage	15
N-switch testing	15
n-wise testing	15

---

## *O*

off-the-shelf software	18
open source tool	16
operability	14
operational acceptance testing	14
operational environment	33
operational profile	14
operational profile testing	14
operational profiling	14
operational testing	14
oracle	16

orthogonal array	35
orthogonal array testing	35
outcome	20
output	27
output domain	27
output value	27
outsourced testing	11

---

## *P*

pair programming	52
pair testing	52
pairwise integration testing	52
pairwise testing	52
Pareto analysis	46
partition testing	46
pass	46
pass/fail criteria	22
path	45
path coverage	46
path sensitizing	46
path testing	46
peer review	47
performance	31
performance indicator	46
performance profiling	32
performance testing	31
performance testing tool	32
phase containment	49
phase test plan	49
planning poker	51
pointer	54
portability	12
portability testing	12
postcondition	26
post-execution comparison	26
post-project meeting	51
precondition	26
predicate	51
predicted outcome	57
pretest	57
priority	56
PRISMA	51
probe effect	52
problem	56
problem management	56
problem report	56
procedure testing	43
process	51
process assessment	51
process cycle test	51
process improvement	51
process model	52
process reference model	52
process-compliant testing	52
product risk	52
Product RiSk Management	52
product-based quality	52
production acceptance testing	32
program instrumenter	51
program testing	51

project	51
project retrospective	51
project risk	51
project test plan	51
pseudo-random	18

---

## *Q*

QFD	19
qualification	26
quality	48
quality assurance	48
quality attribute	48
quality characteristic	48
quality control	48
quality function deployment	48
quality gate	48
quality management	48
quality risk	49

---

## *R*

RACI matrix	32
random testing	57
Rational Unified Process	57
reactive testing	34
record/playback tool	19
recorder	19
recoverability	16
recoverability testing	16
recovery testing	16
regression testing	16
regression-averse testing	16
regulation testing	18
release note	58
reliability	29
reliability growth model	29
reliability testing	29
replaceability	35
requirement	57
requirements management tool	57
requirements phase	57
requirements-based testing	57
resource utilization	26
resource utilization testing	26
result	20
resumption criteria	25
resumption requirements	25
re-testing	25
retrospective meeting	51
review	59
review plan	59
review tool	59
reviewer	59
risk	57
risk analysis	58
risk assessment	58
risk category	58
risk control	58
risk identification	58

risk impact	58
risk level	58
risk likelihood	58
risk management	58
risk mitigation	58
risk type	58
risk-based testing	58
robustness	17
robustness testing	59
root cause	25
root cause analysis	25
RUP	57

---

## *S*

S.M.A.R.T. goal methodology	30
safety	12
safety critical system	32
safety testing	12
sanity test	25
scalability	17
scalability testing	17
scenario testing	27
scorecard	29
scribe	29
scripted testing	29
scripting language	29
SCRUM	29
security testing	32
security testing tool	32
security tool	32
serviceability testing	54
session-based test management	32
session-based testing	32
severity	27
Shewhart chart	27
short-circuiting	35
simulation	27
simulator	27
site acceptance testing	25
smoke test	30
software	33
software attack	33
Software Failure Mode and Effect Analysis (SFMEA)	33
Software Failure Mode, Effects, and Criticality Analysis (SFMECA)	33
software Fault Tree analysis (SFTA)	34
software feature	34
software integrity level	33
software lifecycle	34
Software Process Improvement	34
software product characteristic	33
software quality	34
software quality characteristic	34
software test incident	34
software test incident report	34
Software Usability Measurement Inventory (SUMI)	33
source statement	33
specification	28

specification-based technique	29
specification-based test design technique	29
specification-based testing	28
specified input	30
SPI	15
stability	12
staged representation	34
standard	48
standard software	48
standard-compliant testing	48
standards testing	48
state diagram	28
state table	28
state transition	28
state transition testing	28
statement	30
statement coverage	30
statement testing	30
static analysis	31
static analysis tool	31
static analyzer	31
static code analysis	31
static code analyzer	31
static testing	31
statistical testing	43
status accounting	30
STEP	30
storage	30
storage testing	30
stress testing	30
stress testing tool	30
structural coverage	23
structural test design technique	23
structural testing	23
structure-based technique	23
structure-based test design technique	23
structure-based testing	23
structured walkthrough	23
stub	30
subpath	25
suitability	23
suitability testing	23
SUMI	15
suspension criteria	35
syntax testing	29
system	26
system integration testing	26
system of systems	26
system testing	26
system under test	40
Systematic Test and Evaluation Process	34

---

## *T*

TDD	36
technical review	36
test	36
test analysis	41
test approach	37
test architect	37
test automation	39

test basis	41
test bed	42
test case	37
test case design technique	38
test case specification	38
test case suite	38
test charter	40
test closure	39
test comparator	41
test comparison	38
test completion criteria	37
test condition	39
test control	38
test coverage	37
test cycle	38
test data	40
test data management	41
test data preparation tool	41
test deliverable	39
test design	39
test design specification	40
test design technique	40
test design tool	40
test director	40
test driven development	37
test driver	41
test environment	37
test estimation	42
test evaluation report	41
test execution	38
test execution automation	38
test execution phase	39
test execution schedule	38
test execution technique	38
test execution tool	38
test fail	39
test generator	38
test harness	41
test implementation	39
test improvement plan	37
test incident	37
test incident report	37
test infrastructure	37
test input	41
test item	36
test item transmittal report	37
test leader	42
test level	42
test log	38
test logging	38
test management	42
test management tool	42
test manager	42
Test Maturity Model integration (TMMi)	39
test mission	39
test monitoring	42
test object	40
test objective	42
test oracle	37
test outcome	38
test pass	41
test performance indicator	41

test phase	41
test plan	37
test planning	37
Test Point Analysis (TPA)	42
test policy	42
test procedure	40
test procedure specification	40
test process	41
Test Process Group	41
test process improvement manifesto	41
test process improver	41
test progress report	39
test record	42
test recording	42
test report	42
test reporting	43
test reproducibility	38
test requirement	42
test result	38
test rig	42
test run	42
test run log	42
test scenario	39
test schedule	39
test script	39
test session	40
test set	40
test situation	38
test specification	39
test specification technique	39
test stage	39
test strategy	40
test suite	39
test summary report	38
test target	40
test technique	37
test tool	40
test type	40
testability	26
testability review	26
testable requirement	42
tester	40
testing	36
testware	37
thread testing	30
three point estimation	25
time behavior	26
TMMi	35
top-down testing	44
Total Quality Management	33
TPG	36
TPI Next	36
TQM	36
traceability	45
traceability matrix	45
transactional analysis	24
transcendent-based quality	32

---

## *U*

understandability	57
-------------------	----

<b>unit</b> .....	56
<b>unit test framework</b> .....	57
<b>unit testing</b> .....	57
<b>unreachable code</b> .....	44
<b>usability</b> .....	28
<b>usability testing</b> .....	28
<b>use case</b> .....	56
<b>use case testing</b> .....	56
<b>user acceptance testing</b> .....	56
<b>user scenario testing</b> .....	56
<b>user story</b> .....	56
<b>user story testing</b> .....	56
<b>user test</b> .....	56
<b>user-based quality</b> .....	56

---

## *V*

<b>validation</b> .....	34
<b>value-based quality</b> .....	46
<b>variable</b> .....	53

<b>verification</b> .....	22
<b>version control</b> .....	45
<b>vertical traceability</b> .....	29
<b>V-model</b> .....	49
<b>volume testing</b> .....	54

---

## *W*

<b>walkthrough</b> .....	14
<b>WAMMI</b> .....	34
<b>WBS</b> .....	34
<b>Website Analysis and MeasureMent Inventory (WAMMI)</b> .....	14
<b>white-box technique</b> .....	54
<b>white-box test design technique</b> .....	54
<b>white-box testing</b> .....	54
<b>Wide Band Delphi</b> .....	59
<b>wild pointer</b> .....	60
<b>Work Breakdown Structure</b> .....	60