

## テストマネジメント入門

### JSTQB Advanced Level に基づく 実践的テストマネジメント

2011年1月25日(火)

Japan Software Testing Qualifications Board

技術委員  
町田 欣史 (株式会社NTTデータ)  
佐々木 方規 (株式会社ベリサーブ)

© Japan Software Testing Qualifications Board



## アジェンダ

1. JSTQB認定テスト技術者資格 Advanced Level のご紹介
2. テストマネジメント入門
3. ケーススタディ

© Japan Software Testing Qualifications Board

1

## 1. JSTQB認定テスト技術者資格 Advanced Level のご紹介

© Japan Software Testing Qualifications Board

2

## 1.1 JSTQBの概要

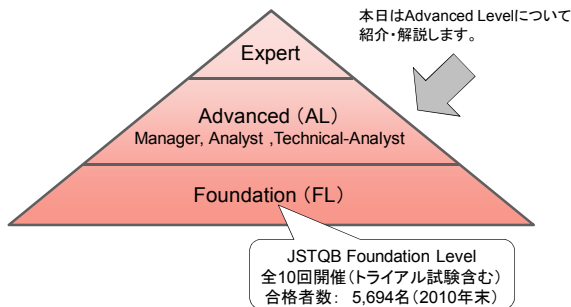
- 日本におけるソフトウェアテスト技術者資格認定の運営組織
  - JSTQB: Japan Software Testing Qualifications Board
    - <http://www.jstqb.jp/>
- ISTQBの加盟組織として2005年4月に認定
  - ISTQB: International Software Testing Qualifications Board
    - <http://istqb.org/>
  - 各国のテスト技術者認定組織が参加
    - 2010年末時点で47か国
  - 各国団体は資格および教育・訓練組織認証について相互認証を行う

© Japan Software Testing Qualifications Board

3

## 1.2 資格試験の概要

- 3段階のスキルレベルに応じた資格試験



© Japan Software Testing Qualifications Board

4

## 1.3 Advanced Level 資格認定の目的

- テストがきわめて重要で専門的なソフトウェアエンジニアリングの一領域であるという認識を得るため。
- テスト担当者のキャリア形成のための基準を提供するため。
- テストのプロフェッショナルとして資格認定を受けていることを、上層部、顧客、そして職場内に認知してもらうようにし、テスト担当者の評価を上げるため。
- 一貫して、良いテストのプラクティスをすべてのソフトウェアエンジニアリング領域へ広めるため。
- テスト関連事項が産業にて適切で価値のあるものだという認識を得るため。
- ソフトウェア開発業者が資格認定を受けたテスト担当者を雇用できるようにするため。またソフトウェア開発業者が、テスト有資格者の人材採用方針を広報することで、競合他社より商業的に有利に立てるようにするため。
- テスト担当者やテストに興味がある人に国際的に認められた資格を身につける機会を与えるため。

© Japan Software Testing Qualifications Board

5

## 1.4 Advanced Level 試験の対象者

- ソフトウェアテスト作業に関与する全ての人
  - テスト担当者
  - テストアナリスト
  - テストエンジニア
  - テストコンサルタント
  - テストマネージャ
  - ユーザ受け入れテスト担当者
  - ソフトウェア開発者
- ソフトウェアテストについて深い理解を望む以下の方々
  - プロジェクトマネージャ
  - 品質管理マネージャ
  - ソフトウェア開発マネージャ
  - ビジネスアナリスト
  - IT部門長
  - 経営コンサルタント

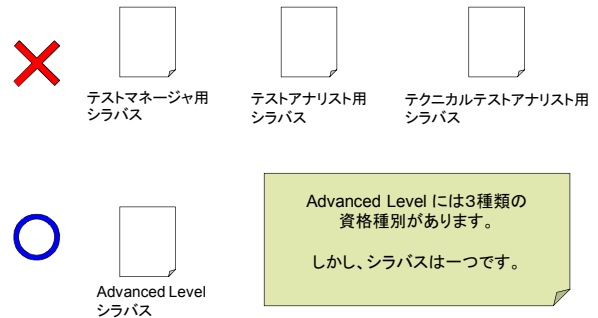
## 1.5 Advanced Levelの資格種別

- 3つの資格種別(タスク)
  - Advanced Level テストマネージャ ← 本日はテストマネージャについて紹介・解説します。
  - Advanced Level テストアナリスト
  - Advanced Level テクニカルテストアナリスト
- 3つの資格種別ごとに認定
  - 組織内におけるそれぞれの代表的な基本的責務と期待されることについて試験する
- すべての資格種別に合格すると、  
テスト技術者資格 Advanced Level 完全上級テスト技術者 (Full Advanced Level Testing Professional) の認定が受けられる

## 1.6 Advanced Level テストマネージャの責務

- 対象システムに対する包括的なテストのゴールおよびテスト戦略を定義する。
- 計画、スケジュールの立案とその追跡を行う。
- 必要となる活動を体系化し説明する。
- タスクに適したリソースを選択、調達し、割り当てる。
- テストチームの選択、編成、指導を行う。
- テストチームのメンバー間、テストチームと他のステークホルダ(利害関係者)とのコミュニケーションを体系化する。
- 妥当な状況判断をすると共に、適切な情報提供を行う。

## 1.7 資格種別とシラバスについて



## 1.8 テストマネージャの学習時間の割合

- K2、K3、K4レベルの学習時間 (K1レベルは全ての章が対象)

章タイトル	学習時間	割合
第1章: ソフトウェアテストの基本的側面	150分	7%
第2章: テストプロセス	120分	6%
第3章: テストマネジメント	1120分	55%
第4章: テスト技法	0分	0%
第5章: ソフトウェア特性のテスト	0分	0%
第6章: レビュー	120分	6%
第7章: インシデント管理	80分	4%
第8章: 標準およびテスト改善プロセス	120分	6%
第9章: テストツールおよび自動化	90分	4%
第10章: スタッパのスキル - チーム構成	240分	12%
※資格種別ごとに各章の重みが異なる	合計	2040分 100%

## 1.9 知識のレベル

- K1 (記憶レベル)
  - 用語または概念を認識し、記憶して、想起することができる。
- K2 (理解レベル)
  - 課題に関連する記述について理由または説明を選択することができる。
  - テスト概念、テスト手順(タスクの順番の説明)に関して要約、区別、分類することができる。
  - 事実を示す例を挙げる(たとえば用語を比較する)ことができる。
- K3 (適用レベル)
  - 概念または技法を正しく選択することができる。
  - 技法を特定の事例に適用することができる。
- K4 (分析レベル) ← Advanced Level から導入
  - 手順または技術に関連する情報を分類し、理解しやすくするため各部分に分割し構成することができる。
  - 事実と推論を区別することができる。
  - 典型的な適用例として、問題を解決するためやタスクを実行するために、ドキュメント、ソフトウェア、プロジェクトの状態を分析し、適した活動を提案できる。

## 1.10 Advanced Level 試験実施要項(予定)

- 実施形式
  - 試験時間 : 180分
  - 試験形式 : 複数の選択肢から正解を選ぶ形式
  - 問題数 : 65問
  - 合格ライン : 非公開
- 出題範囲
  - Advanced Level シラバス + Foundation Level シラバス
    - Advanced Level シラバスには、資格種別ごとに学習の目的が載っています。
      - ALシラバスのすべての箇所は、K1レベルで試験対象となります。
    - Foundation Level シラバスに書かれている概念や技法を用いる問題が、出題される可能性があります。

## 1.11 Advanced Level 受験資格

- 受験資格 : 次の1と2の条件をすべて満たす者とする。
  - 1. JSTQB認定テスト技術者資格 Foundation Level資格の合格者
    - ※受験申し込み書類に、認定書の写しの添付が必要。  
※認定書を無くした場合は合格証明書の発行を依頼して添付。  
※他国でFoundation Levelに合格している場合も受験可。  
(認定書の写し、または合格証明書は同様に必要)
  - 2. 業務経験3年以上(経歴書の提出あり)
    - 下記の業務に関して通算3年以上の経験が必要
      - ソフトウェアテストに関する業務
      - プロセス改善を含む品質保証に関する業務
      - ソフトウェア開発に関する業務
      - ソフトウェアの研究開発(R&D)に関する業務
    - ※ソフトウェア開発に関係する業務であれば、どのような業務でも可。  
※業務経験3年は経歴書の内容で判断。  
(ご本人またはご所属に対して経歴確認をする場合あり)

## 1.12 Advanced Level トライアル試験結果

- 実施概要
  - 資格種別 : テストマネージャ
  - 日時 : 2010年8月28日(土) 10:30~13:30
  - 開催地域 : 東京
  - 試験料 : 21,000円(税込)
  - 主催 : NPO法人ソフトウェアテスト技術振興協会(ASTER) JSTQB
  - 試験実施 : 財団法人日本科学技術連盟
- 実施結果
  - 受験者数 : 189名
  - 合格者数 : 19名
  - 合格率 : 10.05%

テストマネージャの次回試験、および他の資格種別の試験の開催は未定です。  
最新情報はJSTQBのWebサイト(<http://www.jstqb.jp/>)で確認してください。

## 2. テストマネジメント入門

## 本日解説するトピックス

- Advanced Level シラバスに基づいて、テストマネージャに必要な知識、考え方、技術等について解説します。
  - 本日はK4レベルのトピックスについて解説します。
    - 2.1 テストマネジメントドキュメント (3.2)
    - 2.2 テスト進捗のモニタリングおよびコントロール (3.6)
    - 2.3 リスクベーステスト (3.9)
    - 2.4 ソフトウェアライフサイクルにおけるテスト (1.2)
    - 2.5 レビューの導入 (6.4)
    - 2.6 インシデント管理 (7)
- ※トピックス名の後の括弧内はシラバスの章・節番号

## 2.1 テストマネジメントドキュメント

- 学習の目的
  - (K4) IEEE 829に準拠した、テスト計画、テスト設計仕様、テスト手順などのテストマネジメントドキュメントの概要を説明する。
  - (K2) テスト戦略/アプローチの4つの重要な要素を説明し、テスト戦略の要素が、IEEE829準拠のどのドキュメントに含まれるかを説明する。
  - (K2) テスト戦略からの逸脱を他のドキュメントでも扱う理由、またどのように扱うかを示す。

## 2.1 テストマネジメントドキュメント

共通的なテストマネジメントドキュメント

ドキュメント名	概要
テストポリシー	組織にとってのテストの理念。
テスト戦略	組織のテストの方法。 <ul style="list-style-type: none"> <li>□ プロダクトおよびプロジェクトのリスクマネジメント</li> <li>□ テストの実施順</li> <li>□ レベル、フェーズに分割すること</li> <li>□ テストにまつわる上位レベルの活動</li> </ul>
マスターテスト計画	実際のプロジェクトへのテスト戦略の適用内容。 <ul style="list-style-type: none"> <li>□ 実行するテストレベル</li> <li>□ テストレベル間の関係</li> </ul>
レベルテスト計画	各テストレベルで実行する実際の活動。 <ul style="list-style-type: none"> <li>□ マスターテスト計画を詳細化したもの</li> </ul>

## 2.1 テストマネジメントドキュメント

- 2.1.1 テストポリシー (3.2.1)
- 2.1.2 テスト戦略 (3.2.2)
- 2.1.3 マスターテスト計画 (3.2.3)
- 2.1.4 レベルテスト計画 (3.2.4)

※トピック名の後の括弧内はシラバスの章・節番号

### 2.1.1 テストポリシー

テストポリシーの主な記述内容

記述項目	例						
テストの定義	<ul style="list-style-type: none"> <li>□ システムが意図したとおり動作するという信頼感の確立</li> <li>□ 欠陥の欠如</li> </ul>						
基本的なテストプロセスのレイアウト	<table border="0"> <tr> <td>□ テスト計画とコントロール</td> <td>□ テスト終了基準の評価</td> </tr> <tr> <td>□ テスト分析と設計</td> <td>□ テストレポート</td> </tr> <tr> <td>□ テストの実装と実行</td> <td>□ テスト終了作業</td> </tr> </table>	□ テスト計画とコントロール	□ テスト終了基準の評価	□ テスト分析と設計	□ テストレポート	□ テストの実装と実行	□ テスト終了作業
□ テスト計画とコントロール	□ テスト終了基準の評価						
□ テスト分析と設計	□ テストレポート						
□ テストの実装と実行	□ テスト終了作業						
テストの有効性および効率性を評価する方法	<ul style="list-style-type: none"> <li>□ 検出される欠陥の割合</li> <li>□ テストで検出される欠陥とリリース後に検出される欠陥との相対コスト</li> </ul>						
実現したい品質対象	<ul style="list-style-type: none"> <li>□ 信頼性</li> <li>□ 使用性</li> </ul>						
テストプロセス改善のための活動	<ul style="list-style-type: none"> <li>□ テスト成熟度モデル/テストプロセス改善モデルの適用</li> <li>□ プロジェクトの「振り返し」からの提案の実践</li> </ul>						

### 2.1.2 テスト戦略

- テスト設計を開始するタイミングによる分類
  - 予防的戦略 …… 欠陥を予防するために早期にテスト設計を行う
  - 対処的戦略 …… ソフトウェアやシステムができた後にテスト設計を行う
- 代表的なテスト戦略

戦略	例
分析的戦略	リスクベーステスト
モデルベース戦略	運用プロファイル
方法的戦略	品質特性ベース
プロセス準拠/規格準拠戦略	IEEE829ベース
動的で経験則的な戦略	バグに基づく「攻撃」の使用
コンサルテーションベースの戦略	ユーザ主導のテスト
回帰テスト戦略	広範囲の自動化

### 2.1.2 テスト戦略

- 主な記述内容(1/2)
  - リスクマネジメント
    - プロジェクトリスクとプロダクトリスク
    - テストプロセスによるリスクマネジメントの方法
      - リスクとテストとの関連
  - 実行するテストレベル
    - 各テストレベルの開始基準、終了基準
    - テストレベル間での関連

### 2.1.2 テスト戦略

- 主な記述内容(2/2)
  - 統合手順
  - テスト仕様化技法
  - テストの独立性
  - 必須および任意の標準
  - テスト環境
  - テストの自動化
  - ソフトウェア成果物およびテスト成果物の再利用性
  - 再テストと回帰テスト
  - テストのコントロールおよびレポート作成
  - テストの測定およびメトリクス
  - インシデント管理
  - テストウェアの構成管理アプローチ

## 2.1.3 マスターテスト計画

- 主な記述内容
  - テストするアイテム / テストしないアイテム
  - テストする品質属性 / テストしない品質属性
  - テストスケジュール・予算
  - テスト実行サイクル、ソフトウェアリリース計画との関連
  - ビジネス上のテストの正当性、その価値
  - テストを行う人々・部署と他の人々・部署との関連性、提出書類
  - 各レベルで、どのテストアイテムが範囲内/範囲外かを示す定義
  - 各レベルの開始基準、継続(中止/再開)基準、終了基準
  - 各レベル間の関連性
  - テストプロジェクトリスク

## 2.1.4 レベルテスト計画

- 主な記述内容
  - マスターテスト計画を詳細化したもの
  - マスターテスト計画でカバーされていないもの
    - スケジュール
    - タスク
    - マイルストーン などの詳細

## 2.2 テスト進捗のモニタリングおよびコントロール

- 学習の目的
  - (K2)テスト進捗をコントロールするさまざまな手順を比較する。
  - (K2)テストプロセスに影響を与える原因をテスト進捗で見つける方法について5つ例を挙げて説明する。
  - (K4)現在のテストプロセスを改善するためのアクションプラン、または改善を提案するために、モニタリングとコントロール活動、および測定から見つかったテスト進捗に関連する事項を使用する。
  - (K4)テスト結果を分析し、テスト進捗状況を判断して、モニタリングレポートと4つのレポート要素をすべてカバーする最終テストサマリレポートを文書化する。

## 2.2 テスト進捗のモニタリングおよびコントロール

- テスト進捗のモニタリングの対象
  - プロダクトリスク
  - 欠陥
  - テスト
  - カバレッジ
  - 確信度合い …… 調査によって測定
- 測定結果の使用目的
  - テスト結果の分析により、プロダクト、プロジェクト、プロセスで何が起きているかを検出する。
  - レポートにより、テストの検出事項をプロジェクト参加者およびステークホルダーに通知する。
  - コントロールにより、一連のテストまたはプロジェクトを全体として見直し、その結果をモニタリングする。

## 2.2 テスト進捗のモニタリングおよびコントロール

テスト進捗のモニタリング対象と測定マトリクス

対象	マトリクス
プロダクトリスク	<ul style="list-style-type: none"> <li>□ 残存するリスクの数</li> <li>□ 軽減したリスクの数</li> </ul>
欠陥	<ul style="list-style-type: none"> <li>□ レポート済みの欠陥数と解決済みの欠陥数の比</li> <li>□ 平均故障間隔または故障率</li> <li>□ 欠陥数の分類               <ul style="list-style-type: none"> <li>□ テストアイテム別、原因別、導入/検出/除去されたフェーズ別、など</li> </ul> </li> <li>□ 欠陥のレポートから解決に至るまでのタイムラグの傾向</li> </ul>
テスト	<ul style="list-style-type: none"> <li>□ 計画、定義(開発)、実行、合格、不合格、ブロック、スキップされた総数</li> <li>□ 回帰テスト/確認テストのステータス</li> <li>□ 計画されている1日のテスト時間と実際のテスト時間の比</li> </ul>
カバレッジ	<ul style="list-style-type: none"> <li>□ 要件および設計要素のカバレッジ</li> <li>□ リスクのカバレッジ</li> <li>□ 環境/構成のカバレッジ</li> </ul>

## 2.2 テスト進捗のモニタリングおよびコントロール

- テストコントロールの目的
  - テスト計画からの逸脱を最小限に抑える
- 想定されるテストコントロール
  - テストケースの優先順位の再検討
  - 追加リソースの獲得
  - リリース日の延期
  - プロジェクトの範囲(機能)の変更
  - テスト終了基準の再検討

## 2.3 リスクベーステスト

- 2.3.1 リスクベーステストの紹介 (3.9.1)
- 2.3.2 リスクマネジメント (3.9.2)
  - 2.3.2.1 リスク識別 (3.9.2.1)
  - 2.3.2.2 リスク分析 (3.9.2.2)
  - 2.3.2.3 リスク軽減 (3.9.2.3)
- 2.3.3 ライフサイクルにおけるリスクマネジメント (3.9.3)

※トピックス名の後の括弧内はシラバスの章・節番号

### 2.3.1 リスクベーステストの紹介

- 学習の目的
  - (K2) リスクベーステストとリスクを対応させるさまざまな方法を説明する。
  - (K4) プロジェクトリスクとプロダクトリスクを識別し、これらのリスクに基づいて適切なテスト戦略およびテスト計画を決定する。

### 2.3.1 リスクベーステストの紹介

- リスクの度合いに影響する要因
  - 問題発生の可能性
  - 問題発生時の影響
- リスクの種類
  - プロダクトリスク
    - 潜在的な問題の主な影響がプロダクト品質におよぶもの
      - 例) 通常の運用時にシステムをクラッシュさせるような信頼性の欠陥(バグ)
  - プロジェクトリスク
    - 潜在的な問題の主な影響がプロジェクト成功におよぶもの
      - 例) プロジェクトの完了遅延を招くスタッフの不足

### 2.3.1 リスクベーステストの紹介

- リスクベーステストの方法(対応)
  - プロダクトリスクの度合いに応じた方法で、以下のような対応をする。
    - テスト工数の割り当て
    - 技法の選択
    - テスト活動の順序設定
    - 欠陥(バグ)の修復
  - プロジェクトリスクに対応する軽減策や、発生時対策が盛り込まれたテスト作業の計画と管理を行う。
  - 残存リスクに関するテスト結果およびプロジェクト情報の報告を行う。
    - 例) 未実行のテスト、修正/再テストが未実行の欠陥に基づくリスクなど

### 2.3.1 リスクベーステストの紹介

- テスト担当者がすべきこと
  - リスク分析
    - リスクを評価する(リスクの可能性や影響の増減を識別する)。
    - プロジェクトの進行とともに収集された追加情報に基づきリスクを分析する。
  - リスク軽減
    - (プロダクトリスクに対する)もっとも重要な欠陥を検出する。
    - テスト戦略/計画で規定した適切な軽減策を取り入れて、リスクを軽減させる。

### 2.3.2 リスクマネジメント

- 学習の目的
  - (K3) 個別のFMEAアプローチに従い、テスト担当者の視点からプロダクトのリスク分析を行う。
  - (K4) 主要なプロジェクトステークホルダーが抱えているリスクに対しさまざまな側面から結果をまとめ、その結果を基にして、テスト活動がリスクを軽減していることを示す。

## 2.3.2 リスクマネジメント

### ■ リスクマネジメントの活動

- リスク識別
- リスク分析
- リスク軽減

この順で実行されるが、継続的、反復的に実行する。

## 2.3.2.1 リスク識別

### ■ リスク識別のための技法

- エキスパートへのインタビュー
- 第三者によるアセスメント
- リスクテンプレートの使用
- 学習した教訓
- リスクに関するワークショップ(例: FMEA)
- ブレインストーミング
- チェックリスト
- 過去の経験の活用

## 2.3.2.2 リスク分析

### ■ リスク分析

- 識別されたリスクを調査する。
  - リスクを分類する。
  - リスクの可能性や影響を判定する。

### ■ リスクの種類

- 技術的なリスク
  - テスト中のシステムに潜在する問題を発生させる。
- ビジネスリスク
  - ユーザ、顧客、その他のステークホルダに影響する。

## 2.3.2.2 リスク分析

### ■ 技術的なリスクに影響する要因

- 技術およびチームの複雑性
- ビジネスアナリスト、設計者、プログラムの個人的な問題およびトレーニングの問題
- チーム内の衝突
- 供給者側との契約の問題
- 開発組織の地理的な分散
- レガシーアプローチ対新しいアプローチ
- ツールと技術
- 劣悪な管理または技術的なリーダーシップ
- 時間、リソース、管理のプレッシャー
- 初期品質保証の欠如
- 高い変更率
- 高い初期欠陥率
- インターフェースと統合の問題

## 2.3.2.2 リスク分析

### ■ ビジネスリスクに影響する要因

- 影響を受ける機能の使用頻度
- イメージの悪化
- 業務の喪失
- 財政的、経済的、社会的損失または責任の可能性
- 民事上または刑事上の法的拘束
- ライセンスの喪失
- 妥当な次善策の欠如
- 故障の判明による否定的な評判

## 2.3.2.2 リスク分析

### ■ リスクの度合いの決定

- 定量的アプローチ
  - 可能性 × 影響 = リスクに対応する損失の予測値
- 定性的アプローチ (通常のアプローチ)
  - 「可能性が非常に高い」「高い」「中程度」「低い」「非常に低い」など
  - 定量的アプローチに劣るわけではない

### 2.3.2.3 リスク軽減

- リスク軽減の方法
  - 予防手段でリスクを軽減し、可能性や影響を軽減させる。
  - 発生時対策を作成し、リスクが現実になった場合の影響を軽減させる。
  - リスクを他の処理担当部門に移転させる。
  - リスクを無視して、そのまま受け入れる。

### 2.3.2.3 リスク軽減

- プロジェクトリスク軽減
  - テストマネージャが軽減できるリスク
    - テスト環境およびツールの準備
    - テストスタッフの調達能力と資質
    - テストへの入力情報の質の低さ
    - テストへの入力情報の錯綜
    - テスト活動に対する標準類、ルール、技法の欠如
  - リスク軽減へのアプローチ
    - テストウェアの準備
    - テスト機器の事前テスト
    - プロダクトの以前のバージョンに対する事前テスト
    - テストの開始基準の厳格化
    - テスト容易性の要件
    - 以前のプロジェクト結果のレビューへの参加
    - 問題および変更管理への参加
    - プロジェクトの進捗および品質のモニタリング

### 2.3.2.3 リスク軽減

- プロダクトリスク軽減
  - テストの活動(欠陥を検出する範囲)
    - リリース前に欠陥やその処理機会を認識することでリスクを軽減させる。
  - テストの活動(欠陥を検出する以外の範囲)
    - 一定条件下でのシステムの動作を保証することでリスクを軽減させる。
  - テスト以外の活動
    - 正しく記述されていない要件に対してレビューを行う。
  - その他
    - 適切なテスト設計技法の選択
    - レビューおよびインスペクション
    - テスト設計のレビュー
    - 独立性のレベル
    - 経験豊富な担当者
    - 再テストを実行する方法
    - 回帰テスト

### 2.3.3 ライフサイクルにおけるリスクマネジメント

- 学習の目的
  - (K2) 反復のプロセスで必要となるリスクマネジメントの特徴について説明する。
  - (K3) リスクベーステストの戦略をテストの活動へ落とし込み、テスト期間中、状況をモニタリングする。
  - (K4) テスト結果を分析、レポートし、未解決のリスクを判定/提示して、プロジェクトマネージャが正しいリリース判断を行えるようにする。

### 2.3.3 ライフサイクルにおけるリスクマネジメント

- リスク識別・分析の開始
  - リスク軽減活動の計画
- } プロジェクトの初期フェーズで実施
- ↓
- リスクマネジメントの実施
    - 新しいリスクの識別
    - 既存リスクのレベルの再評価
    - リスク軽減作業の有効性の評価
- ↓
- 軽減したリスクの測定
    - テストケースを追跡し、検出した欠陥に対応するリスクを測定
- ↓
- リスクベーステストの実施
    - 残りのリスクから、正しいリリースのタイミングを決定する。

### 2.4 ソフトウェアライフサイクルにおけるテスト

- 学習の目的
  - (K2) テストがソフトウェア開発や保守の活動の一部であることを説明する。
  - (K4) ソフトウェアライフサイクルモデルを分析し、実行すべき最適なタスク/テストの活動を簡単に説明する(テスト活動と開発活動を区別する)。



## 2.4 ソフトウェアライフサイクルにおけるテスト

- テストは次の事項と連携して実行する
  - 要求エンジニアリング / 要求マネジメント
  - プロジェクトマネジメント
  - 変更管理 / 構成管理
  - ソフトウェア開発
  - ソフトウェア保守
  - テクニカルサポート
  - 技術ドキュメントの作成

## 2.4 ソフトウェアライフサイクルにおけるテスト

- テストレベル (Foundation Levelシラバスでの定義)
  - 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
- プロジェクトの状況に応じて、テストレベルを追加
  - ハードウェアソフトウェア統合テスト
  - システム統合テスト
  - フィーチャ統合テスト
  - 顧客プロダクト統合テスト

## 2.4 ソフトウェアライフサイクルにおけるテスト

- テストレベルの項目
  - テストのゴール
  - テストの範囲
  - テストベースに対するトレーサビリティ
  - 開始基準 / 終了基準
  - レポートを含むテスト提出物
  - テスト技法
  - 測定 / メトリクス
  - テストツール
  - 組織、その他の標準への準拠

## 2.4 ソフトウェアライフサイクルにおけるテスト

- Vモデルにおけるシステムテスト(1/2)
  - プロジェクト計画と同時にシステムテスト計画を開始する。
    - システムテストの実行と終了が完了するまで、テストコントロールを継続する。
  - 要求仕様、システムおよびアーキテクチャ設計仕様(高位レベル)、コンポーネント設計仕様(低位レベル)と同時期にシステムテスト分析および設計を行う。
  - システム設計の間にシステムテスト環境の構築を開始する。
    - コーディングやコンポーネントテストと同時に始めることが多い。

## 2.4 ソフトウェアライフサイクルにおけるテスト

- Vモデルにおけるシステムテスト(2/2)
  - システムテスト開始基準にすべて合致したらシステムテストを実行する。
    - 最低でもコンポーネントテストが完了し、コンポーネント結合テストも完了している。
    - システムテストの実行は、システムテスト終了基準に合致するまで継続する。
  - システムテスト終了基準の評価およびシステムテスト結果のレポートは、システムテスト実行を通じて行われる。
  - システムテスト終了基準に合致しシステムテスト実行の終了が宣言されたら、システムテスト終了作業を行う。
    - 場合によっては、受け入れテストが終わるまでのプロジェクト活動が終了するまで遅れることがある。

## 2.5 レビューの導入

- 学習の目的
  - (K2) レビュータイプを互いに比較し、相対的な強み、弱み、使用分野を示す。
  - (K3) 識別されている手順に従い行われる公式のレビューを通して、レビューチームを指導する。
  - (K4) プロジェクトの品質/テスト計画の一部としてのレビュー計画の概要を示す。このレビュー計画は、動的テストアプローチと検出すべき欠陥やスタッフのスキルを調整した上で選択したレビューの技法を考慮したものとなる。

## 2.5 レビューの導入

- レビューの導入のステップ
  - マネジメントからの支援の確保
  - コスト、利点、実装の問題についてのマネージャ教育
  - レビュー手順、形式、インフラストラクチャの選択と記述
  - レビュー技法および手順のトレーニング
  - レビューを行う人々、その作業をレビューする人々からの支援の確保
  - パイロットレビューの実行
  - コスト節約になるレビューの利点の証明
  - 重要なドキュメントへのレビューの適用

## 2.6 インシデント管理

- 学習の目的
  - (K3) IEEE1044-1993で提示されているインシデント管理ライフサイクル手順に従い、欠陥を処理する。
  - (K3) 品質を向上させるため、欠陥レポートをIEEE1044-1993および欠陥分類法と照らし合わせながら評価する。
  - (K4) これまで作成された欠陥レポートを分析し、欠陥分類法を更新する。

## 2.6 インシデント管理

- 欠陥ライフサイクルのステップ
  - ステップ1: 認識
  - ステップ2: 調査
  - ステップ3: 対応
  - ステップ4: 処理
- 各ステップでの情報収集活動
  - 記録
  - 分類
  - 影響の識別

## 2.6 インシデント管理

- 有用な欠陥レポート
  - 完全
  - 簡潔
  - 正確
  - 客観的
- 欠陥情報を用いて実施すること
  - 個別の欠陥の解決
  - テスト進捗のモニタリングとコントロール
  - 欠陥の正確な分類
  - リスク分析
  - プロセス改善

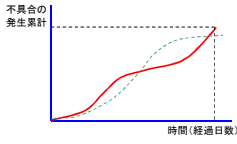
## 3. ケーススタディ

実際のプロジェクトにおいて直面する問題・課題に対し  
 前述の考え方や技術を用いることによって  
 テストマネージャがすべき判断、対策を紹介します。

### 3.1.1 ケーススタディ

#### ■ テスト進捗のモニタリングおよびコントロール

テスト実行の進捗を管理しています。次の不具合発生に関するグラフを見て、確認すべきマトリクスは何か検討して下さい。



グラフでは不具合が収束していません。  
このような状況下で、確認する例を示します。

- ①変化点の存在  
(コードの追加・変更(要因は様々)、組織など)
- ②時間当たりの消化テスト数との関係
- ③発生している不具合の種類  
(重要度、発生エリア、デグレードの有無)

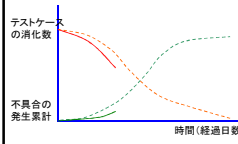
**ポイント！！**

発生している不具合の数にとらわれず、**不具合の質・内容**に着眼するようなマトリクスや要因を確認します

### 3.1.2 ケーススタディ

#### ■ テスト進捗のモニタリングおよびコントロール

テスト実行の進捗を管理しています。次のテストケースの消化状況を見て、状況を判断して下さい。



テストケースの消化数に比べて不具合の検出が少ない状況です。  
このような状況の判断する例を示します。

- ①テスト実行の進捗(消化数)  
実施されたテストエリアの確認が必要。既存機能部分のテスト中心の場合、進捗状況が良好とは判定し難い。今後、変更量の多い新機能部分のリリースで、テスト消化の遅延と重大な不具合の発生が予想される。
- ②不具合件数の状況(発生件数)  
実施されたテストエリアの確認が必要。既存機能部分のテスト中心の場合、テスト対象の品質が良いと判断するより、不具合が多く存在すると予想される新機能のテストが実施されていないことが要因と考えられる。

**ポイント！！**

計画数値的な進捗だけでなく、テストエリアや優先度なども考慮して状況を判断します。また、不具合に関しては、発生状況だけでなく、修正状況の確認も重要です。

### 3.2 ケーススタディ

#### ■ リスクベーステストの紹介

新しいテストプロジェクトを担当し、テストの計画を立案しようとしています。技術的なリスクに影響する要因を検討して下さい。

リスクの要因やリスク発生に影響する例を示します。

- ①開発実績(経験)の有無 → 完成度、変更規模、難易度
- ②開発の背景(社内システム/製品(OEM)/製品(自社)/受注開発) → 仕様変更の可能性、仕様の完成度
- ③開発の進捗状況 → 開発スケジュール、プレッシャー
- ④利用環境や利用者の想定 → ソフトウェア品質特性、非機能特性
- ⑤過去の不具合 → ソフトウェアの欠陥モード

仕様や環境が変化すると、テスト計画時のリスクも変化します。

**ポイント！！**

テストケースに対してビジネス影響度などを考慮した「リスク値」を付加し、テストケースの消化数だけでなく、リスク値の消化数も計画する。リスク要因等は、リスク値の消化(減少)に影響を与えます。

### 3.3 ケーススタディ

#### ■ ソフトウェアライフサイクルにおけるテスト

システムテストを実施中だが、システムテストのテストレベル以前の基本的な不具合が多く検出されたため、予定していたレベルテスト計画が進まない。対応方法を検討して下さい。

テストレベルをエスカレーションする例を示します。

- ①テスト開始基準(エントリークライテリア)を確認し、合致していない項目が無いかを確認する。
- ②システムテストのレベルに達しているか、テストを受け入れの可否を決める「テスト受け入れテスト」を実施する。「テスト受け入れテスト」は、システムテストの実施に最低限必要な基本テストセットを構築し、実施します。
- ③「①」、「②」でシステムテストのレベルに達していないと判断した場合、開発担当にテスト開始基準およびシステムテストの相違要因の不具合を報告し、改善依頼を実施する。原則、システムテストは一時保留にする。
- ④一時保留になった場合は、開発担当に(テスト開始基準の他に)再開始基準を通知し、確認依頼を実施します。

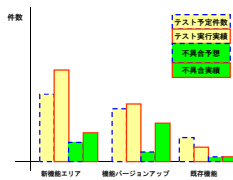
**ポイント！！**

各テストレベルは、その開始基準に合致していることを確認する。エスカレーションは、開発担当とテスト担当の意図を合わせる事が大切です。

### 3.4 ケーススタディ

#### ■ インシデント管理

インシデントレポートから欠陥の傾向を分析し、対策を行います。



左のテスト実行結果を分析し、対策を検討して下さい。

- ①新機能のテスト結果分析  
不具合数が予想より多い。そのために実行件数が増加していると考えられる。このような場合、同じテストケースを重複して実行されたこともあるので、実施したテストケースの分布も分析し、カバレッジを確認する。
- ②機能バージョンアップのテスト結果分析  
不具合数が予想より多い。この場合、デグレードが発生している可能性が非常に高く、回帰テストの実行量と実施分布を分析が必要。回帰テストの実施タイミングも分析のポイントとなる。
- ③既存機能  
不具合数が予想に近いが、テスト実行件数が予想より少ない。新機能および機能バージョンアップのテスト実行に工数が取られたと予想される。特に不具合の内容を分析し、今回の開発による影響が既存不具合を判断し、特に前者の場合は回帰テストを実施することを強く推奨する。

**ポイント！！**

インシデントレポートの数(予定件数)の差異は重要でなく、欠陥の傾向を分析する必要があります。特に欠陥数が多く報告されたプロジェクトでは、予定工数をオーバーしている可能性があり、テストケースの消化数・カバレッジや回帰テストの有無を合わせて分析する必要があります。

ご清聴ありがとうございました