

IT 検証技術者 教育研修シラバス

ミドルレベル（レベル3、レベル4）

Ver.1.0.1 (2009/05)



一般社団法人 IT 検証産業協会
教育・研修部会

IT 検証技術者教育研修シラバス

はじめに

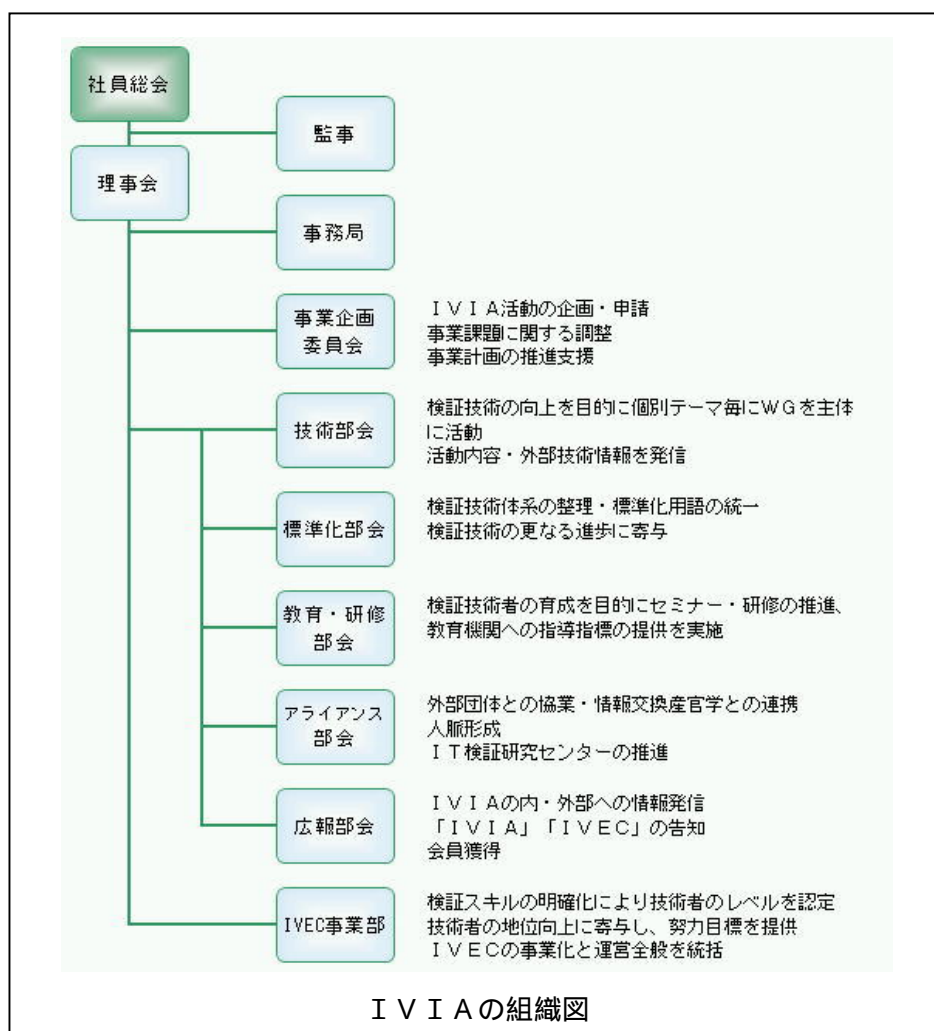
IT 製品のソフトウェアは、製品の競争力を高めるため高機能 / 多機能でありながら高品質 / 高信頼性を維持することを求められています。一方では、これらの製品に対応できる専門性の高いテストを実施できる技術者の不足が、業界全体の問題となっています。

これらの課題に対応すべく 2005 年の 10 月に設立された「IT 検証産業協会（以後、IVIA と表記する場合もあり）」（IT Verification Industry Association：略称 IVIA アイビア <http://www.ivia.or.jp/>）では、人材育成や IT 検証技術者の地位向上を目指し、テスト技術の向上やテストの標準化の部会活動を行なっています。

IVIA の教育・研修部会では、IT 検証技術者の育成支援に取り組んでいます。その中で本書は、技術者の育成に留まらず、教育機関への支援や技術提供も目的としています。

高品質な製品の開発を支援する「IT 検証技術者」の教育体系として、本書を広く公開して行きます。

* IVIA は 2009 年 4 月に一般社団法人化しました。



著作に関する注意点

本書の著作権は、IVIA（一般社団法人 IT 検証産業協会）が管理、所有します。本書の全部ないし一部を承諾なしに複製し利用することは、著作権法の制限事項に基づき、禁じられています。

改定

バージョン	改定内容	改定日	備考
Ver.0.9.0	初版（IVIA 内部公開）	2007/7/31	
Ver.0.9.1	初版（一般公開）	2007/9/4	
Ver.1.0.0	正式版	2008/8/13	重要項目の解説
Ver.1.0.1	誤記修正、一般社団法人化に伴う修正	2009/5/1	文言修正

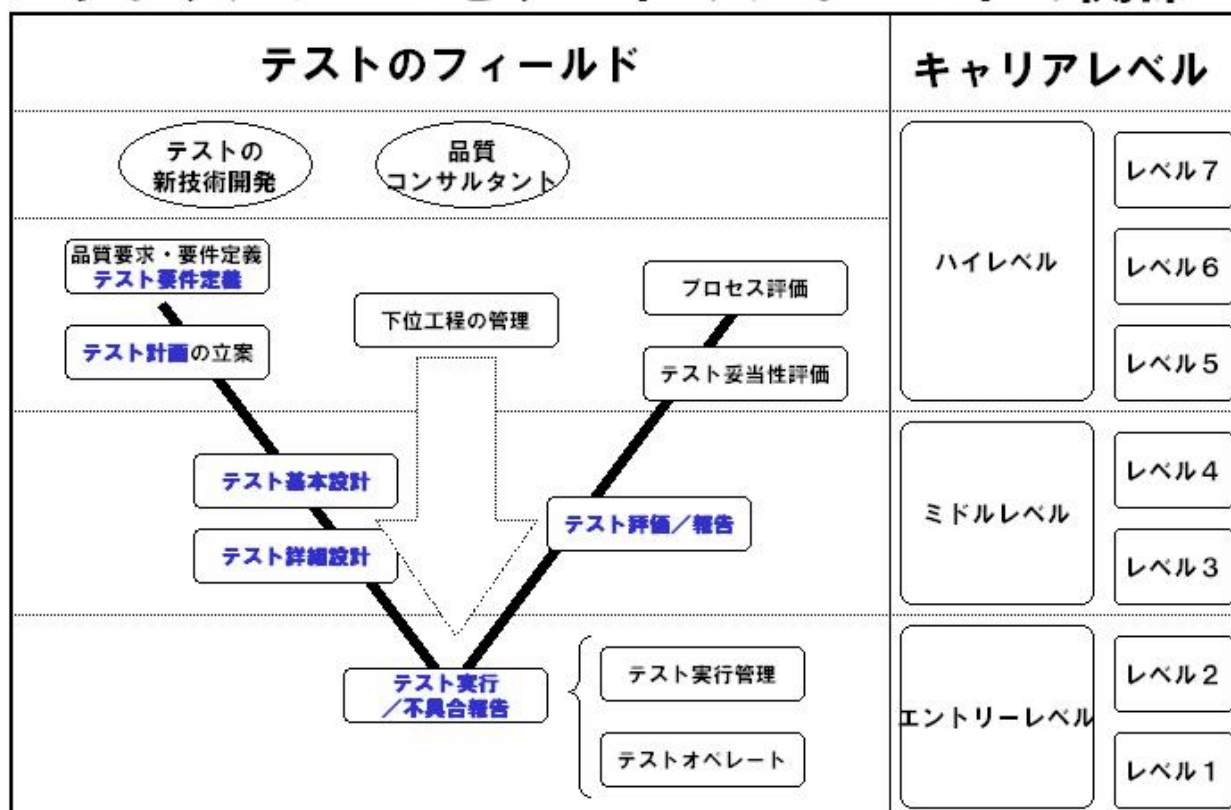
目 次

IT 検証技術者教育研修シラバス	2
はじめに	2
著作に関する注意点	3
改定	4
目 次	5
1 教育・研修シラバスの体系	6
1-1 目標となるキャリアレベル	7
2 スキルトレーニング編	8
2-1 スキル体系	9
2-2 仕様の把握	11
2-3 仕様管理（トレーサビリティ）	14
2-4 網羅性設計	15
2-4-1 網羅性設計（テスト要素抽出）	15
2-4-2 網羅性設計（テスト変数抽出）	18
2-5 テストケース設計	22
2-5-1 テストケース設計（基本フロー）	22
2-5-2 テストケース設計（組み合わせ）	23
2-5-3 テストケース設計（期待値設計）	24
2-6 モニタ設計	25
2-7 テスト環境設計	26
2-8 テストの種類	27
2-9 プロジェクト管理（個人進捗管理も含む）	29
3 ソフトウェアテスト用語集	31
4 付録 A 本書について	36

1 教育・研修シラバスの体系

IVIAでは、IT検証技術者の習得すべき技術をテスト工程（テストのフレームワーク）に合わせて規定しています。下記にIVIAが規定するキャリアレベルとテストのフィールドの関係の概要を示します。

キャリアレベルとテストのフィールドの関係



IVIAでは、ソフトウェアテストを実施できる技術者に「スキル」と「知識」を要求しています。「スキル」は、実際にプロジェクトに必要な能力を示します。「スキル」に対して「知識」は、ソフトウェアテスト中級レベルを想定して必要な用語を体系化しています。

「スキル」と「知識」の学ぶべきレベルを“認知”・“引用”・“実用”の3段階の習得レベル^(注1)で定義しています。

本書では、ミドルレベルの「スキル」に関する習得を「2. スキルトレーニング編」で、「知識」に関する習得を「3. ソフトウェアテスト用語集」で説明しています。

IVIAの認定教育機関または教育ベンダーは、学習要項に従って必要な「スキル」と「知識」の教育とトレーニングを実施して下さい。

(注1) 本書について(付録A)の「習得のレベルについて」を参照。

1-1 目標となるキャリアレベル

本書は、ミドルレベル（レベル3、レベル4）のIT検証技術者に求める「スキル」と「知識」の習得を目標としています。具体的には、ミドルレベルにはIT検証の実務で「テスト基本設計/テスト詳細設計」の工程を実施できることが求められます。

本書では動的テストのテスト設計が実施できる技術者を対象としています。

ソフトウェアテストは静的解析と動的テストに分類することができます。

静的解析の領域では、ウォークスルーやインスペクションなどのレビュー、モデル検査、形式検証、パターン分析、コード解析などソフトウェア品質に影響を与える重要な手法やプロセスが存在します。

静的解析の領域は「知識」レベルのみを習得するものとし、動的テストとのシームレスなテストプロセスが実現された場合に対応するものとしします。

各レベルのキャリア像は、下記の通りです。

ミドルレベル レベル3

テスト基本設計書（テスト基本設計の出力）から、テスト実行が可能なテストケース（テスト手順書、テストデータ（およびデータ作成手順）、判定基準（期待値）、テストスクリプト等を示す）を作成できる技術者を目標としています。これらには、テスト実行で必要なテストケースの分類や予想テスト実行時間の情報、およびテストの実行者へテストケースの目的を伝達する事が含まれます。また、テスト詳細設計書（テスト詳細設計の出力）は、テスト対象の仕様変更に対応するためにメンテナンス（保守性）やポータビリティ（再利用性）を考慮します。

ミドルレベル レベル4

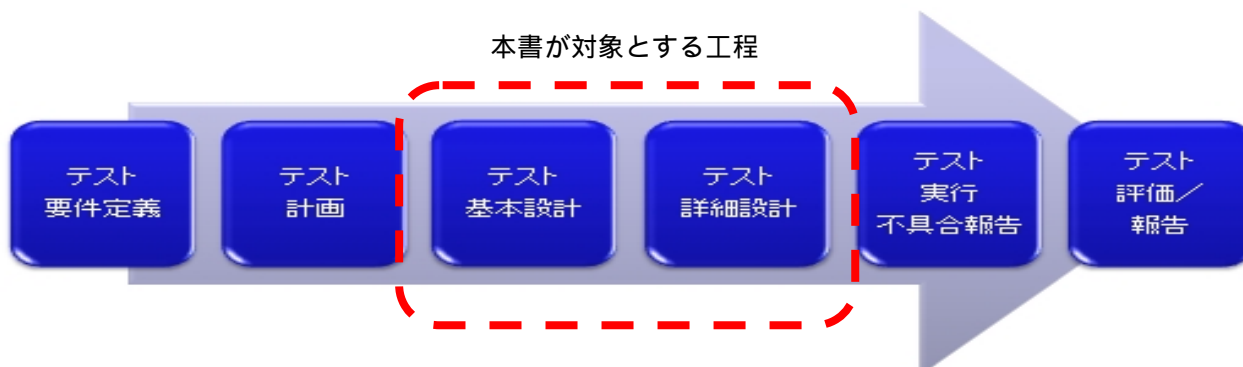
テスト計画やテストの要求から、テストのアプローチやテストの目的に合ったテストの設計ができる技術者を目標としています。テスト基本設計のテストカバレッジ（テスト対象の網羅性）設計や組み合わせられるコンディション（条件）やコンフィギュレーション（環境）などの考慮も含まれます。また、テスト基本設計書は、テスト対象の仕様とのトレーサビリティも考慮します。テストの種類（機能テストや性能テスト、セキュリティテスト等、テスト目的を有するテストの種類）はテストの目的によって分類されますが、個別（専門的）なスキルを有するテストの種類設計は本書では対象外とします。

尚、本書がテスト手順（方法）を示す場合は、機能テスト（Functional Testing）を取り扱っているものとします。

その他のキャリアレベルは、IVIAが定義する「IT検証技術者 キャリアレベル」を参照して下さい。

2 スキルトレーニング編

スキルトレーニングは、ソフトウェアテストの工程に合わせて学習していきます。ミドルレベルのスキルトレーニングは、ソフトウェアテスト工程の「テスト基本設計/テスト詳細設計」を対象とします。



本書が対象とする「テスト基本設計/テスト詳細設計」では、テスト要件定義の要求に対し、正確および効果的に「テスト実行/不具合報告」で利用されるテストウェアを設計しなくてはなりません。

したがって、「テスト基本設計/テスト詳細設計」を学習する前提として、エントリーレベルの「テスト実行/不具合報告」を習得しておく必要があります。

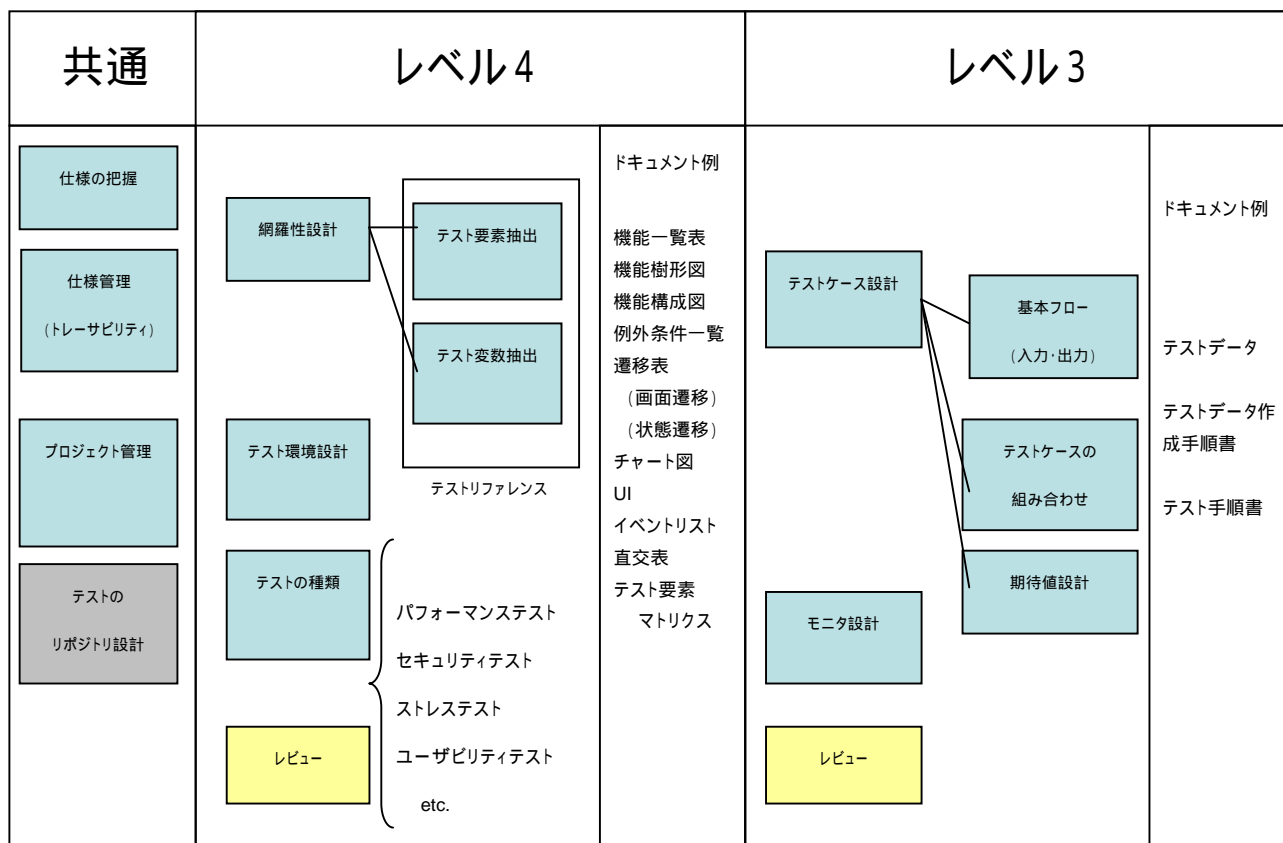
スキルトレーニングは、I V I A が認定した教育機関および教育ベンダーが、本書に準拠した研修教材を利用して実施されることを想定しています。

I T 検証技術者は、I V I A が公開した教育機関や教育ベンダーが提供する教育・研修コースを参照の上、「スキル」の習得を進めて下さい。

2-1 スキル体系

IVIAのスキル体系では、教育機関や教育ベンダーが実施する「スキルトレーニング」の履修の体系とおおよその学習時間を規定します。

下記に、ミドルレベルのスキル体系の概要を示します。



学習時間の詳細は、本書が記載された学習の目的を満足されていると認定された場合は、変更可能とします。

本文中に「テスト設計」の記述がある場合は、「テスト基本設計」と「テスト詳細設計」の双方を示すものとします。

テスト基本設計工程 テスト詳細設計工程	工程概要	学習時間(参考) 時間(m)	
		レベル3	レベル4
仕様の把握	テスト設計では、テスト対象の仕様（機能要求/非機能要求）を把握する。仕様の把握の作業中に必要なドキュメントを調査し、入手する。	30	45
仕様管理 (トレーサビリティ)	テスト設計では、テスト対象の仕様とテスト設計のトレーサビリティ（追跡性）が求められる。 テスト対象の仕様変更に対する整合性（テスト対象の仕様とテスト設計の整合性）やテストカバレッジ（テスト対象機能やテスト非対象機能の明確化、テスト対象機能の網羅性）を確保する。	30	45
網羅性設計- (テスト要素抽出)	テスト基本設計では、テスト対象の機能（仕様ベースのテストの場合は仕様書）から確認すべきテスト要素（テスト因子）を抽出する。 テストの要素はテスト基本設計のベースとなり、機能性テストの網羅性設計にも利用する。	30	90
網羅性設計- (テスト変数抽出)	テスト要素に関係するコンディション（条件）とコンフィギュレーション（環境）などの入力情報を抽出する。	30	90
テストケース設計- (基本フロー)	テスト基本設計で抽出したテスト要素とテストコンディション（条件）/テストコンフィギュレーション（環境）を最小単位で組み合わせてテストケースを作成する。	90	30
テストケース設計- (組み合わせ)	テスト基本設計で抽出したテスト要素とテストコンディション（条件）/テストコンフィギュレーション（環境）を複合条件で組み合わせてテストケースを作成する。	90	30
テストケース設計- (期待値設計)	テストケース設計時には必ず期待値を設計する。 期待値がテストケースに対して一律でない場合は、特に期待値の確認方法とあわせて設計を実施する。	60	30
モニタ設計	テスト実行結果（期待値）を確認するためのモニタ方法を設計する。	15	60
テスト環境設計	テスト設計を反映したテスト環境の設計のポイントを習得する。	15	60
テストの種類	本書で説明した機能テスト（Functional Testing）以外のテストの種類を習得する。本章では、代表的なテストの種類を習得する。	30	90
プロジェクト管理	テスト設計の工程全般でのプロジェクト管理のポイントを習得する。	30	90

2-2 仕様の把握

学習概要

テスト基本設計 / テスト詳細設計では、テスト対象の仕様（機能要求 / 非機能要求）を把握する。仕様の把握の作業中に必要なドキュメントを調査し、入手する。

本章に関連する用語

リファレンス、非機能要求、ブレインストーミング

学習の目的		習得のレベル	
		レベル3	レベル4
1) テスト対象の仕様（リファレンス）の調査と入手			
<p>テストの目的やテストアプローチを確認して、テスト設計するテストの種類に必要なテスト対象のドキュメントや情報を入手する。</p> <p>学習のポイント！ テスト種類によって、入手すべき仕様が異なることを理解する。</p> <p>(解説)</p> <p>テスト設計をする前にテスト計画書から、テスト設計の目的を確認する。</p> <p>テストの目的によって複数のテスト設計が必要な場合があり、それぞれのテスト設計に必要な情報を仕様などから得る。</p> <p>テスト設計に利用する資料は仕様書のみではなく、開発者からのヒアリングなども有効となる。</p>	B	C	
<p>テスト設計で必要となる(利用する)ドキュメントの一覧を作成する。</p> <p>学習のポイント！ トレーサビリティを維持する為、ドキュメントの入手元、バージョンおよびリビジョンなどの管理を習得する。</p> <p>(解説)</p> <p>テスト設計で利用するドキュメントは仕様変更等により改定されることが多い。</p> <p>テスト設計では、どの情報を利用して設計したかを明確にしておくことで、仕様変更が生じても全体を見直すことなく、差分のみを最低限の変更で対処できるように心がける。</p>	C	C	

学習の目的		習得のレベル	
		レベル3	レベル4
<p>テスト対象の機能要求を調査しドキュメントを入手する。</p> <p>学習のポイント！ 調査すべきテスト対象の機能要求には、機能（要求される振る舞い）のみでなくコンディション（条件）やコンフィグレーション（環境）なども調査することを理解する。</p> <p>（解説）</p> <p>テスト設計では、機能（要求の振る舞い）にさまざまな動作条件（環境や引数、状態など）を組み合わせる。機能は仕様書に定義されていることが多いが、動作条件は仕様書外の外部条件も考慮する必要があるので、テスト対象の特性にあわせて調査しておく。</p>	B	C	
<p>テスト対象に関わるコンプライアンス（法規、標準（業界標準なども含む）、規格等）に関する情報を調査し入手する。</p> <p>学習のポイント！ 設計仕様以外でテスト対象に影響を与える情報の調査と入手の必要性を理解する。</p> <p>（解説）</p> <p>テスト対象が公共性の特性を持つ場合などは、法的な規制や基準が設けられていることがある。これらは、仕様より優先すべき内容が多く、十分に考慮しておく。</p>	B	C	
<p>テスト対象の非機能要求を調査する。</p> <p>学習のポイント！ テスト対象の設計仕様に明示的に記述されていない品質要求の取得方法を習得する。</p> <p>（解説）</p> <p>非機能要求は、テスト対象の（全体的な）応答時間や使用性など機能仕様として定義が困難な要求である。これらは、テスト対象の運用（利用）環境を調査し、テスト設計の情報として用いる。</p>	A	B	
<p>テスト対象が運用（利用）される環境やリテラシー情報を調査する。</p> <p>学習のポイント！ 利用環境やリテラシー情報がテスト設計条件に大きく影響することを理解する。</p> <p>（解説）</p> <p>「テスト対象の非機能要求を調査する」と同様。</p>	C	C	
<p>過去のテストドキュメント（テスト仕様、テスト設計書（テストケースを含む）、テスト報告書等）を入手し調査する。</p> <p>学習のポイント！ 仕様を把握すると同時に、過去のテスト資産の再利用の重要性を理解する。</p> <p>（解説）</p> <p>過去のテストドキュメントには、テスト対象の概要や機能構成を示す機能樹形図が記述されているために、テスト対象の仕様を把握し易い。</p> <p>また、過去のテスト資産を再利用することで、テスト設計を構造化することになり、仕様の変更部分の特定を容易にすることができる。</p>	A	B	

学習の目的		習得のレベル	
		レベル3	レベル4
i) 仕様の把握と管理			
<p>調査した仕様をテスト設計文書として記録する。</p> <p>学習のポイント! 仕様をテスト設計文書として記録する方法（文書体系、文書フォーマット等）を理解する。</p> <p>（解説）</p> <p>IEEE829-1998. などには、テストの文書体系が示されている。</p> <p>テスト設計の文書体系を定義することで、他者にテスト設計構造の理解を容易にし、さらに文書フォーマットによりテスト設計の項目をテスト設計者に示唆すること（項目漏れの防止など）も目的にしている。</p>	C	C	
i) テスト対象の仕様の把握と漏れの防止対策			
<p>ブレインストーミングを実施して、仕様の把握とテスト観点の漏れを防止する。</p> <p>学習のポイント! 仕様の把握のため仕様調査・検討する担当でブレインストーミングを試みる理由を理解する。</p> <p>（解説）</p> <p>テスト対象の有識者を集めてブレインストーミングを進めることで、テスト設計者以外の視点や観点を知ることができる。</p> <p>また、ブレインストーミングの内容をまとめることで、テスト設計時のチェックリストに利用できる。</p>	B	C	

2-3 仕様管理（トレーサビリティ）

学習概要

テスト設計では、テスト対象の仕様とテスト設計のトレーサビリティ（追跡性）が求められる。テスト対象の仕様変更に対する整合性（テスト対象の仕様とテスト設計の整合性）やテストカバレッジ（テスト対象機能やテスト非対象機能の明確化、テスト対象機能の網羅性）を確保する。

本章に関連する用語

トレーサビリティ（追跡性）

学習の目的		習得のレベル	
		レベル3	レベル4
i) 仕様の構成管理			
<p>テスト設計物とテスト対象の仕様（リファレンス）とのトレーサビリティ（追跡性）の必要性を理解する。</p> <p>学習のポイント！ 仕様の追加、変更がテスト設計に影響することを理解する。</p> <p>(解説)</p> <p>テスト設計中に仕様変更が発生すると、変更作業を実施する必要がある。</p> <p>仕様変更の対応も、テスト設計の作業が下流工程に進むほど変更量が多くなる。</p> <p>仕様変更に対応するテスト設計部分を容易に特定することは、テスト設計作業の影響を最低限に抑えることを可能にする。</p>	C	C	
<p>テスト構成管理を計画する。</p> <p>学習のポイント！ トレーサビリティ（追跡性）を考慮したテスト構成管理計画の立案方法を習得する。</p> <p>(解説)</p> <p>テスト設計では、テスト設計を構造化することでテスト構成の管理を実施する。</p> <p>これらは、一部のテスト設計が変更になっても、影響範囲を最低限に抑えることができる。</p>	A	B	
<p>テスト構成管理の方法を習得する。</p> <p>学習のポイント！ テスト対象の仕様（リファレンス）とのリレーション方法（TOOL等の利用も含む）やテスト設計物の変更手順・変更承認の方法を理解する。</p> <p>(解説)</p> <p>仕様とテスト設計の関連付けや仕様変更の対応方法など仕様変更の対応に関連深いルールを決めておくと、仕様変更への対応がスムーズに行える。</p> <p>特に仕様変更の連絡ミスなどコミュニケーションの問題は、大きな手戻りになることもあるので注意する。</p>	C	C	

2-4 網羅性設計

2-4-1 網羅性設計（テスト要素抽出）

学習概要

テスト基本設計では、テスト対象の機能から確認すべきテスト要素（テスト因子）を抽出する。テストの要素はテスト基本設計のベースとなり、機能性の網羅性設計にも利用する。

本章に関連する用語

テスト要素、機能樹形図（Function Tree）、テストカバレッジ

学習の目的		習得のレベル	
		レベル3	レベル4
1) テスト要素の抽出			
<p>テスト対象の仕様（リファレンス）をベースにテスト要素を抽出し、テスト要素の一覧を作成する。</p> <p>学習のポイント！ 仕様からテスト要素を抽出する。特にテスト対象の仕様（リファレンス）がユースケース記述などの場合はテスト要素の抽出を容易にするが、製品仕様などからはテスト要素の切り出し方が重要であることを理解する。</p> <p>（解説）</p> <p>開発設計書によっては、テスト要素となる機能（テスト対象が動作する振る舞い）が曖昧に記述されている場合がある。ブラックボックステストの場合にテスト要素は、与える条件（入力）と期待結果（出力）との入出力の関係になることを理解しておく、抽出を容易にする。</p>	B	C	
<p>機能樹形図（Function Tree）を利用して、機能構成を構造化する。</p> <p>学習のポイント！ 機能樹形図（Function Tree）を用いることで、機能間の関係（結びつき）と機能を階層化することを理解する。</p> <p>（解説）</p> <p>機能を階層構造にすると、機能システムが異なっても類似する機能の存在などが理解し易い。</p> <p>例えば、タイマー処理などの機能が、テスト対象が持つそれぞれの機能システムに存在する場合など、類似機能による影響も検討し易くなる。また、一部の機能が変更になった場合にも、影響する部分（類似機能や変更機能の下位層の機能）を特定し易い。</p>	B	C	
<p>機能樹形図（Function Tree）の最下層が、最小単位のテスト要素と一意になることを確認する。</p> <p>学習のポイント！ 機能樹形図（Function Tree）では、テスト要素がトップダウンで作成される。最下層の項目が最小単位のテスト要素になっていることを確認する。</p> <p>（解説）</p> <p>機能テストの設計で、テスト要素とその条件（テスト要素に与える条件など）を混同する場合がある。テスト要素に与える条件などはテスト設計に重要な項目であるが、条件などは多数の機能に関連していることが多いため、機能樹形図に記述すると機能間の関係を見る場合に混乱する可能性がある。</p>	B	C	

学習の目的		習得のレベル	
		レベル3	レベル4
<p>テスト要素にコンディション(条件)が含まれていないことを確認する。コンディション(条件)にあたる場合は、一覧や表形式(マトリクス)にまとめて、テスト変数(テスト水準)とする。</p> <p>学習のポイント! 最小単位のテスト要素は、テストコンディションと混同されることが多い。</p> <p>テストコンディションの場合は、他のテスト要素にも関係する場合が多いためテスト要素と組み合わせて利用できるようにする。</p> <p>(解説)</p> <p>特に機能テストを設計する場合は、機能間で共通のテスト条件になる変数や利用データなどを別表にしておくが良い。</p> <p>テスト要素となる機能とこれらの条件を結びつけると、機能変更や条件の変更の際に影響範囲を特定し易くなる。</p>	C	C	
<p>機能樹形図(Function Tree)で別系統の同一機能を確認する。</p> <p>学習のポイント! 同じ機能が別の系統に体系化されている場合、系統間の結び付きが強いことを理解する。</p> <p>(解説)</p> <p>別系統で同一機能がある場合は、影響しあうことが多い。</p> <p>これらは、外部仕様からでは特定できないが、コードがクローン化(コピーによる作成)されていたりすると、機能変更があった場合に類似する機能も影響している場合が多い。</p> <p>これらのことを考慮して、テスト設計を系統だてすることを推奨する。</p>	B	C	

学習の目的		習得のレベル	
		レベル3	レベル4
l) テストカバレッジ設計			
<p>抽出したテスト要素の一覧とテスト対象の仕様（リファレンス）との関係付けを実施して、双方の未対応な項目を洗い出す。</p> <p>学習のポイント！ 関連付けが“1対多”になることも想定して、関連付けを確認する。</p> <p>（解説）</p> <p>仕様ベースのテストでは、テスト対象の仕様（リファレンス）に記載のテスト要素が網羅的に抽出されていることが重要である。テスト要素の抽出漏れを防止するために、テストカバレッジをとる。</p>	B	C	
;) レビュー&承認			
<p>抽出したテスト要素は、レビューを実施して漏れの防止に努める。</p> <p>学習のポイント！ 公式レビューや非公式レビュー、インスペクション、ウォークスルーなどレビューの種類と適切なレビューを実施する。</p> <p>（解説）</p> <p>テスト設計は、テスト対象が正しく動作することを確認するために行なうが、テスト設計が妥当であること、およびテスト設計の視点に問題がないことを第三者レビューで確認する。</p> <p>特にレビューアは、テスト対象の有識者が望ましい。</p>	B	C	
<p>テスト基本設計の結果（テスト要素、コンディション、コンフィギュレーションなど）は、要求元の確認を受けテスト設計内容の承認を得る。</p> <p>学習のポイント！ テストの性質上、テスト対象の品質に責任が取れない。テストプロジェクトの品質の確保のため、レビュー&承認のプロセスが必要なことを理解する。</p> <p>（解説）</p> <p>テストでは、製品の品質（特に不具合が無くなった）を保証することができない。</p> <p>その為に、機能テストを含む動的テストでは、テスト実行を通じて「正しく動作した」ことを確認している。</p> <p>網羅的に動的テストを実施することは電源実的ではなく、無限数になるテストを、リスクを考慮することで有限化している。</p> <p>テストの承認プロセスは、テスト対象の責任者にリスクがあることを通知する意味も持つ。</p>	B	C	

2-4-2 網羅性設計（テスト変数抽出）

学習概要

テスト要素に関するコンディション（条件）とコンフィギュレーション（環境）などの入力情報を抽出する。

本章に関連する用語

テストコンディション（条件） テストコンフィギュレーション（環境）

学習の目的		習得のレベル	
		レベル3	レベル4
1) テストコンディション（条件）の抽出			
<p>テスト要素に与える入力情報として、テストコンディション（条件）を抽出する。テストコンディションは、属性により単一／複合／排他など変化する。</p> <p>学習のポイント！ テストコンディションの抽出方法は、属性により一覧表やマトリクス（組み合わせ）デシジョンテーブル（排他）などを利用して行う。</p> <p>（解説）</p> <p>テストコンディション（条件）は、テスト要素に対して“正常”、“正常でない”、“異常”（これらは、仕様上では全て正常とされる）などを考慮する。</p> <p>特に変数化されたデータは、“正常”な値だけでなく他の値も抽出しておく。</p>	B	C	
<p>テストコンディションが複雑に影響する場合は原因結果グラフやマトリクスなどの手法を用いる。</p> <p>学習のポイント！ 原因結果グラフ、マトリクスなどの有効性や利用方法を理解する。</p> <p>（解説）</p> <p>テストコンディションによっては、これらの条件が排他的であったり、複合条件で期待結果が変わる場合があり、一覧では確認しにくい。</p> <p>これらには、マトリクスやデシジョンテーブルなどを取り入れる。</p>	B	C	
<p>イベントを抽出し、一覧表やイベント関係図を作成する</p> <p>学習のポイント！ テスト要素の入力（起動される場合が多い）条件として、イベント（割り込み、プロセス等）を洗い出す</p> <p>（解説）</p> <p>動的テストで機能を確認するには、イベントを発生させる必要がある。</p> <p>これらのイベントは、テスト要素やテスト要素への引数条件に関連することがある。</p> <p>例えば、ある機能はプログラム処理から呼び出される場合と H/W 割り込みで呼び出された場合とでは、処理の優先順位が異なったりする場合を考慮する必要がある。</p> <p>動的テストで、機能を呼び出すイベントを体系図にしておくとう理解し易い。</p>	A	C	

学習の目的		習得のレベル	
		レベル3	レベル4
l) テストコンフィギュレーション（環境）の抽出			
<p>テスト対象を構成するH/Wコンフィギュレーション（環境）を抽出する。</p> <p>学習のポイント！ H/Wには、テスト対象のオプション製品やインターオペラビリティの対象なども考慮する。</p> <p>（解説）</p> <p>ソフトウェアテストでは、テスト対象を動作させるH/Wプラットフォームが必要となる。</p> <p>テスト対象が、複数のプラットフォームで動作することができる場合や構成要素を変更できる場合は、互換性のテストも実施するようにする。</p>	A	C	
<p>テスト対象を構成するS/Wコンフィギュレーション（環境）を抽出する。</p> <p>学習のポイント！ S/Wには、OSなどのプラットフォーム以外に、テストデータ（印刷データや出力パターン）なども考慮する。</p> <p>（解説）</p> <p>OSなどのS/Wプラットフォームを利用する場合は、OSのバージョンやバージョンUPも環境として考慮すべきである。</p> <p>テスト対象に与えるテストデータも、時間軸のようにテスト対象に影響を与える要因が含まれていることを理解しておく必要がある。</p>	B	C	

学習の目的		習得のレベル	
		レベル3	レベル4
;) テストカバレッジ設計			
抽出したテストコンディション（条件）と仕様（リファレンス）との関係付けを実施して、双方の未対応な項目を洗い出す。 学習のポイント！ 関連付けが“1対多”になることも想定して、関連付けを確認する。 (解説) 仕様ベースのテストでは、テスト対象の仕様（リファレンス）に記載のテスト要素が網羅的に抽出されていることが重要である。 テスト要素の抽出漏れを防止するために、テストカバレッジをとる。	B	C	
テストコンディション（条件）が有効値/無効値の属性を有する場合は、適正值が抽出されていることを確認する。 学習のポイント！ テストコンディション（条件）が範囲を持つ場合に、有効値のみでなく無効値にも着眼する。	B	C	
抽出したテストコンフィグレーション（環境）と仕様（リファレンス）との関係付けを実施して、双方の未対応な項目を洗い出す。 学習のポイント！ テストコンフィグレーション（環境）には、プラットフォームの他にテスト対象に影響する構成物も検討する。	B	C	
テストコンフィグレーション（環境）での保証範囲外でも検討されていることを確認する。 学習のポイント！ テストコンフィグレーション（環境）によっては、動作を保証しないことも想定する。 (解説) テスト対象の動作保証範囲外の状況でも、テスト対象に致命的な障害を与えないことも考慮する。 例えば、テスト対象の動作中に電源断があった場合には、テスト対象の動作は保証しないが、テスト対象の破損や故障を引き起こさないことなどは確認しておきたい。 物理的な衝撃や過度の状態異常は、この場合には該当しない。	B	C	
直交表（実験計画法）を用いて、コンディションの組み合わせの網羅性を高める。（任意） 学習のポイント！ 直交表のメリット・デメリットを理解し、有効な適用を実施する。 (解説) 直交表だけでなく各種のテスト手法には、それぞれメリット・デメリットが存在する。 個々の手法のデメリットは、ここでは言及はしないが、用途に合った利用方法は理解しておく必要がある。	B	C	

学習の目的		習得のレベル	
		レベル3	レベル4
1) レビュー&承認			
<p>抽出したテスト変数（コンディション、コンフィギュレーション）は、レビューを実施して漏れの防止に努める。</p> <p>学習のポイント！ 公式レビューや非公式レビュー、インスペクション、ウォークスルーなどレビューの種類と適切なレビューを実施する。</p> <p>（解説）</p> <p>テスト設計は、テスト対象が正しく動作することを確認するために行なうが、テスト設計が妥当であること、およびテスト設計の視点に問題がないことを第三者レビューで確認する。</p> <p>特にレビューアは、テスト対象の有識者が望ましい。</p>	B	C	
<p>テスト基本設計の結果（テスト要素、コンディション、コンフィギュレーションなど）は、要求元の確認を受けテスト設計内容の承認を得る。</p> <p>学習のポイント！ テストの性質上、テスト対象の品質に責任が取れない。テストプロジェクトの品質の確保のため、レビュー&承認のプロセスが必要なことを理解する。</p> <p>（解説）</p> <p>テストでは、製品の品質（特に不具合が無くなった）を保証することができない。</p> <p>その為に、機能テストを含む動的テストでは、テスト実行を通じて「正しく動作した」ことを確認している。</p> <p>動的テストを網羅的に実施することは現実的ではなく、無限数になるテスト項目を、リスクを考慮することで有限化している。</p> <p>テストの承認プロセスは、テスト対象の責任者にリスクがあることを通知する意味も持つ。</p>	B	C	

2-5 テストケース設計

2-5-1 テストケース設計（基本フロー）

学習概要

テスト基本設計で抽出したテスト要素とテストコンディション（条件）/テストコンフィギュレーション（環境）を最小単位で組み合わせてテストケースを作成する。

本章に関連する用語

マトリクス

学習の目的		習得のレベル	
		レベル3	レベル4
い) 基本フローを構成する			
	テスト要素と関係するテストコンディション（条件）を組み合わせる。 <i>学習のポイント！</i> マトリクスなどを利用して、組み合わせの漏れを防ぐ。	C	C
	テスト要素と関係するテストコンフィギュレーション（環境）を組み合わせる。 <i>学習のポイント！</i> マトリクスなどを利用して、組み合わせの漏れを防ぐ。	C	C
	テスト要素と例外のテストコンディション（条件）を組み合わせる。 <i>学習のポイント！</i> 仕様の範囲内で通常状態でない条件を抽出し、組み合わせる。	C	C
	テスト要素と異常（障害）のテストコンディション（条件）を組み合わせる。 <i>学習のポイント！</i> 仕様の範囲内外で異常（障害）の条件を抽出し、組み合わせる。	C	C
い) その他			
	テストケースの作成レベルの決定。テストケースは、テスト実行者のスキルや状況（工数や納期）を把握して作成する。 <i>学習のポイント！</i> テストケースの資産化（再利用、流用性）を考慮した場合は、テストケース作成者以外の第三者がテスト実行を出来る程度に作成する。 テストプロジェクトの要求により、作成レベルを検討する。	C	C
	テストケース作成者以外の第三者にテストケースの目的が理解できるように作成する。 <i>学習のポイント！</i> テストケースの作成意図がテスト実行者に理解されない場合は、テストケースに記述以外の問題の発見が期待できないことを理解する。	C	C

2-5-2 テストケース設計（組み合わせ）

学習概要

テスト基本設計で抽出したテスト要素とテストコンディション（条件）/テストコンフィギュレーション（環境）を複合条件で組み合わせることでテストケースを作成する。

本章に関連する用語

コンフリクト（衝突）、コンカレント（競合）

学習の目的		習得のレベル	
		レベル3	レベル4
i) 組み合わせの検討			
有効なコンフリクト（衝突）の条件を抽出し、文書化する。 学習のポイント！ コンディション（条件）やイベント、テスト要素など、全ての項目がコンフリクト（衝突）の対象になることを理解する。	C	C	
有効なコンカレント（競合）の条件を抽出し、文書化する。 学習のポイント！ コンカレント（競合）には、競合するリソースが存在する。有効な競合リソースを抽出する。	C	C	
前提事項となるテストコンディション（条件）/テストコンフィギュレーション（環境）を抽出し、関連を文書（図）にまとめる。 学習のポイント！ 組み合わせるテスト要素の前提条件を検討する。	C	C	
ii) その他			
テストケースの作成レベルの決定。テストケースは、テスト実行者のスキルや状況（工数や納期）を把握して作成する。 学習のポイント！ テストケースの資産化（再利用、流用性）を考慮した場合は、テストケース作成者以外の第三者がテスト実行を出来る程度に作成する。 テストプロジェクトへの要求により、作成レベルを検討する。	C	C	
テストケース作成者以外の第三者にテストケースの目的が理解できるように作成する。 学習のポイント！ テストケースの作成意図がテスト実行者に理解されない場合は、テストケースに記述以外の問題の発見が期待できないことを理解する。	C	C	

2-5-3 テストケース設計（期待値設計）

学習概要

テストケース設計時には必ず期待値を設計する。期待値がテストケースに対して一律でない場合は、特に期待値の確認方法とあわせて設計を実施する。

本章に関連する用語

テストケース、テスト手順書、チェックリスト

学習の目的		習得のレベル	
		レベル3	レベル4
l) 有効な期待値の設計			
<p>テストケースの期待値設計では、テストケースの対象となるテスト要素の期待値のみでなく、期待値以外の出力結果の確認を促す。</p> <p>学習のポイント！ 期待値のみを重点的に確認するテストケースでは、テストケースの設計外の問題を発見し難くなる。</p> <p>テストケースでは、可能な限り広範囲な設計を実施する。</p>	C	C	
<p>リソース状況やプロトコルフレームのように動的に結果が変化する期待値を設計する場合には、期待値の部分にマスク（変化するフレームなど）や、期待値の有効範囲を設定する。</p> <p>学習のポイント！ 期待値に部分的なマスクをした場合でも、マスク範囲が正常な有効値であることを確認する。</p>	C	C	
l) 期待値のリファレンスの準備			
<p>テストケースの文書（テスト手順書、チェックリストなど）に期待値を記述できない場合には、期待値と比較できるリファレンス（テスト基準）を用意する。</p> <p>学習のポイント！ テスト結果がアナログ的（NCなどの工作機械で加工したものなど）なものは、物理的なリファレンスを用意する。</p> <p>また、互換確認などは、基準となるリファレンス品を規定する。</p>	C	C	
<p>自動テストツールなどの利用を前提とした場合には、テストケースのスクリプトに対応するコンペアデータを作成する。</p> <p>学習のポイント！ コンペアデータは、適用するテストツールの機能を利用しても良い。</p>	C	C	

2-6 モニタ設計

学習概要

テスト実行結果（期待値確認）を確認するためのモニタ方法を設計する。テスト対象には、HMIの機器等に結果を出力できない時やモニタによる監視が必要な場合がある。期待値を確認するためのテスト実施環境（またはテスト実施方式）の設計を習得する。

本章に関連する用語

学習の目的		習得のレベル	
		レベル3	レベル4
i) ツールによるモニタ設計			
専用ツールによるモニタ設計（または、ツールの調達）を実施する。 学習のポイント！ 擬似データなどをテスト対象に出力するエミュレータの設計（または、専用ツールの準備）を実施してテスト結果（期待値）の確認方法を確立することを習得する。	C	C	
リソース（ネットワーク、ネットワークプロトコル、H/W資源など）を監視するツールを利用したモニタ監視の方法を決定する。 学習のポイント！ 出力結果（期待値）が動的に変化するテスト対象の期待値の確認方法を理解する。テスト対象にパフォーマンスの影響を与える可能性を理解しておく。	C	C	
専用ツールによる出力分析ツール設計（または、ツールの調達）を実施する。 学習のポイント！ クロマトグラフィーや色素分析などの専門のアナライザーや分析ツールを用いることで、出力結果（期待値）を数値結果で確認する方法を理解する。	C	C	

2-7 テスト環境設計

学習概要

テスト設計を反映したテスト環境の設計のポイントを習得する。

本章に関連する用語

学習の目的		習得のレベル	
		レベル3	レベル4
i) テスト（設計）支援ツールの検討			
	テスト（設計）支援ツールの導入目的、効果、導入手順（アプローチ）を検討する。 学習のポイント！ ツールの導入が生産性を一時的に低下させる可能性があることを理解しておく。	A	C
ii) テスト環境の設計			
	テスト設計に基づき、テスト対象のテスト実行を実現する環境を設計する。 学習のポイント！ 実機を用いないシミュレーションや実機を駆動するエミュレータ環境など、テスト設計にテスト環境が影響することを理解する。	B	C

2-8 テストの種類

学習概要

本書で説明した機能テスト (Functional Testing) 以外のテストの種類を習得する。本章では、代表的なテストの種類を習得する。

本章に関連する用語

学習の目的		習得のレベル	
		レベル3	レベル4
1) テストの種類 (機能テスト)			
機能テスト：仕様ベースのテスト。テスト対象の正確性を確認する。 学習のポイント! 機能には、明示的・暗示的な要件があることを理解しておく。	C	C	
状態 (および画面) 遷移テスト：仕様を基準にイベント (操作) に対して、テスト対象が持つ状態 (HMI の画面) が正しく遷移することを確認する。 学習のポイント! テスト対象の状態とイベントの関係は状態遷移図と状態遷移表を利用することを理解する。	C	C	
構成 (コンフィギュレーション) テスト：仕様ベースにテスト対象を構成する環境に影響しないことを確認する。 学習のポイント! テスト対象の構成物の抽出方法を理解しておく。	C	C	
互換性 (コンパチビリティ) テスト：仕様ベースにインターオペラビリティ (相互運用性) を含む、相互互換を確認する。 学習のポイント! 仕様を基準に互換の範囲を理解しておく。	C	C	
セキュリティ機能テスト (セキュリティテスト)：仕様を基準にしたテスト対象がセキュリティ状態を維持することを確認する。 学習のポイント! テスト対象が不特定多数の利用 (運用) が想定されるとき、仕様でセキュリティの有無を検討されていることを理解しておく。	B	C	

学習の目的		習得のレベル	
		レベル3	レベル4
l) テストの種類（非機能テスト）			
	パフォーマンステスト（性能テスト）：性能要件に従い、効率性（時間、資源）テストを実施する。 学習のポイント！ ビジネスモデル（実利用（実運用）想定）に基づいたテストを実施することを理解する。	B	C
	ストレステスト（性能テスト）：過負荷状態で正しい動作を確認するテストを実施する。 学習のポイント！ 過負荷状態をテスト目的に合致してテスト対象に与えている（過負荷量、適用場所）ことを理解する。	B	C
	セキュリティテスト（脆弱性テスト）：アプリケーションやインフラの脆弱性をテストする。 学習のポイント！ 脆弱性確認のセキュリティパターンおよび確認方法の妥当性を確認する。	B	C
	ユーザビリティテスト：仕様性基準に遵守していることを確認するテストを実施する。 学習のポイント！ フィールドモニタ等のベンチマーク（評価）とユーザビリティテストの違いを理解する。	B	C
	障害対応テスト：テスト対象が運用を継続し続けることを確認するテストを実施する。 学習のポイント！ 想定する障害状態にあっても、運用の継続を維持（または回復手段の有無）できることを確認する。 運用を継続する方法、および回復手段が妥当であるかを判断することを習得する。	B	C

2-9 プロジェクト管理（個人進捗管理も含む）

学習概要

テスト設計の工程全般でのプロジェクト管理のポイントを習得する。

本章に関連する用語

学習の目的		習得のレベル	
		レベル3	レベル4
イ) チーム管理			
スコープ管理：テスト設計目的（テストの種類やターゲット）とテスト設計目標（カバレッジや抽出件数）の管理点を理解する。 学習のポイント！ テストの目的に合致したテストの種類的设计およびテスト対象が設計要求範囲に差異が無いこと、およびテストカバレッジや目標件数（妥当性を含む）の管理を理解する。	B	C	
スケジュール管理：テスト設計のスケジュールの管理点を理解する。 学習のポイント！ テスト設計のスケジュールに影響する要因（リスク管理とも関係）を把握し、スケジュールを維持する方法（進捗管理、定例会議、日報提出）を理解する。	B	C	
コスト管理：テストの設計工数の管理点を理解する。 学習のポイント！ テスト設計の見積りと実績の差異分析を理解する。	B	C	
品質管理：テスト設計に要求される品質の要素と管理点を理解する。 学習のポイント！ テスト設計では、テスト対象の仕様変更や仕様の不明（曖昧、未定義など）に対応し、仕様とテスト設計のトレーサビリティ（テストの構成管理）や不明な部分の仕様の確認、状況を管理することを理解する。	B	C	
要員管理：要員のスキル・作業の状況の管理点を理解する。 学習のポイント！ テスト設計の要員のスキル（テスト対象の理解度）習得度の把握と作業への取り組みに関する阻害要素（モチベーションの低下など）を理解する。	B	C	
コミュニケーション管理：テスト設計中でのコミュニケーションの管理点を理解する。 学習のポイント！ 仕様変更管理や質問表などのツールやチーム内・外の情報交換を阻害する要素と対策を理解する。	B	C	

学習の目的		習得のレベル	
		レベル3	レベル4
<p>リスク管理：テスト設計中に発生する想定・想定外のリスクの把握と対策方法を理解する。</p> <p>学習のポイント！</p> <p>仕様の不明（曖昧、未定義など）の問題の対策方法を理解する。</p> <p>想定外のリスクが発生したときの対応（トラブルの解析や対応する担当者の事前配置など）手順の立案方法を理解する。</p>	B	C	
<p>調達管理：テスト設計に用いるドキュメント（仕様やプロジェクト計画書など）およびリソース（機器、テスト対象など）の管理を理解する。</p> <p>学習のポイント！ テスト設計で調達したドキュメント管理点（機密、変更など）を理解する。</p>	C	C	
い) 個人管理			
<p>役割の確認：テスト設計での役割を理解する。</p> <p>学習のポイント！ テスト設計のチーム内で割り当てられる役割を理解する。</p>	C	C	
<p>テスト設計の生産性の管理：テスト設計の工数の予測と実績の管理を理解する。</p> <p>学習のポイント！ テスト設計工数の管理はテストスケジュールに深く関係することを認識し、その管理方法を理解する。</p>	C	C	
<p>テスト設計管理：テスト設計の予実管理を理解する。</p> <p>学習のポイント！ テスト設計エリア、件数などのテスト目標やテスト設計スケジュール（個人のスケジュール）などの管理を理解する。</p>	C	C	
<p>不明点管理：テスト対象の仕様の不明（曖昧、未定義など）の問い合わせや状況管理を理解する。</p> <p>学習のポイント！ テスト対象の不明点の確認方法（コミュニケーション）や問い合わせをしている状況の管理を理解する。</p>	C	C	
<p>報告・連絡・相談・質問：テスト設計中のコミュニケーションの重要性と方法を理解する。</p> <p>学習のポイント！ テスト設計における、適切な報告とその方法を理解する。</p>	C	C	

3 ソフトウェアテスト用語集

ソフトウェアテスト用語集は、本書を学ぶ為に必要な用語と対象のIT検証技術者に必要なソフトウェアテストの用語をまとめています。ソフトウェアテストの用語を学ぶには、IVIAの標準化部会が提供する「用語集」と併用することをお勧めします。

用語	説明	レベル 3	レベル 4
テストの種類			
単体テスト(Unit Test)	テスト対象の構造を理解した上で、それらのプログラムの一つ一つが意図した通りに動作しているかを確認する	B	B
結合テスト(Component Test)	ユニットテストが完了したクラス同士を結合させてテストを行い、主にクラス間のメッセージングやインタフェースの正当性をチェックする	B	B
統合テスト(Integration Test)	全てのソフトウェア部品を統合して、全てのクラスの動作の連携を確認する	B	B
機能確認テスト(Functional Test)	システム内部の構造を意識せずに、入力と出力だけに着目し、様々な入力に対して仕様通りの出力が得られることを確認して、外部から見た機能を検証する	C	C
性能テスト(Performance Test)	テスト対象が要求通りの性能(パフォーマンス)を状態が変化しても維持できることを検証する。目的に合わせたツールやブルーブを利用して、性能を測定する	B	B
回帰テスト(Regression Test)	仕様の変更や改修でテスト対象に変更を加えることで、変更した箇所以外に影響がないかを確認するテスト	C	C
検収・受入テスト(User Acceptance Test)	ユーザがテスト対象の受入(検収)を確認するテスト。一般的に、ユーザの要求の視点で動作が問題ないことを確認する	C	C
テスト受入テスト(Test Acceptance Test)	テスト対象がテスト実行に耐えうる品質であるかを判定するテスト。一般的に、テスト対象の主機能や重要な機能のテストケースを抽出して実施することが多い	C	C
不具合修正確認テスト(Verify Test)	不具合が期待値通りに修正されている事を確認するテスト。一般的に、ベリファイテストは不具合報告をした本人が実施する	C	C

用語	説明	レベル 3	レベル 4
テスト技法			
構造テスト(White-Box Test)	テスト対象の内部構造を理解して、プログラムが正しく記述されている事を確認するテスト	B	B
ブラックボックステスト (Black-Box Test)	テスト対象の内部構造は意識せず、外部から見た機能を検証するテスト。入力（操作や条件、パラメータ等）と出力（結果や動作）だけに着目し、様々な入力条件を考慮して仕様書通りの結果が得られるかどうかを確認する	C	C
同値分割	処理を正常に通過するデータ群と正常に通過しないデータ群をそれぞれ集合として捕らえ、集合の中の任意のデータをテストデータとして抽出する技法	C	C
限界値分析	処理を正常に通過するデータと正常に通過しないデータの境界値を用いてテストデータを導出する技法	C	C
原因結果グラフ	入力と出力の関係を表す表を作成し、テストを行う技法	C	C

用語	説明	レベル 3	レベル 4
テストの文書			
リファレンス	テスト対象の設計書やマニュアルなどソフトウェアテストの入力になるソース	C	C
テスト設計書	リファレンスをもとにテストを実現するために設計された文書	C	C
テストアイテム	テスト対象のテスト実行で確認すべき項目	C	C
テストケース	テスト設計書（テストアイテム）に基づいて作成された、テスト実行を実現するためのドキュメント群。テストケースには、複数のテストアイテムが含まれる場合がある	C	C
テスト手順書	テスト実行の操作手順を記述した文書。通常、テストの操作を Step by Step で記述し、確認が必要な操作には「期待結果」を併記する	C	C
チェックシート	テストの実施や準備など、漏れを防止するための確認項目が記述された文書	C	C
テストエビデンス	テスト結果の合否に関わらず、テストを実施した結果を示す証拠	C	C
テスト記録	テストの実施結果(実施した日時やテストケース数、テスト時間および合否結果など)を記録した文書	C	C
不具合報告書	テスト対象から検出された不具合を報告するための文書	C	C
不具合番号	不具合報告書の識別番号。一般的には、テストフェーズ毎に識別できるように割り当てられる	C	C
不具合ステータス	<p>通常は、プロジェクト毎に規定される。下記は、参考例</p> <p>OPEN：新しく報告された不具合で、対応する担当者が未アサインの状態</p> <p>ACCEPT：報告された不具合に対応する担当者がアサインされた状態</p> <p>FIX：報告された不具合を担当者が対応し、完了した状態。修正確認の実施待ち</p> <p>REOPENED：修正確認を実施した結果、対応に問題がある状態。この後、再度 ACCEPT される</p> <p>VERIFIED：修正確認を実施した結果、問題が解決したことを確認した状態。</p> <p>CLOSED：全ての処理が完了して、テスト責任者が該当する不具合の処理を確認した状態</p>	C	C
優先度	<p>通常は、プロジェクト毎に規定される。下記は、参考例</p> <p>P1 緊急度：大 該当の不具合によって、テストの実施が出来ないカテゴリがある</p> <p>P2 該当の不具合によって、テストの実施が出来ない多数の機能がある</p> <p>P3 緊急度：中 該当の不具合によって、テストの実施が出来ない重要な機能がある</p> <p>P4 該当の不具合によって、テストの実施が出来ない機能がある</p> <p>P5 緊急度：小 該当の不具合は、限定されている</p>	C	C

用語	説明	レベル 3	レベル 4
重要度	通常は、プロジェクト毎に規定される。下記は、参考例 A テスト対象の停止や破壊を伴う不具合 B 他の機能にも影響する不具合 C 機能単独の不具合 D 機能の不具合としては認められないが、修正をした方が良い問題	C	C
影響 / 影響範囲	影響や影響する範囲（他の機能やテスト対象、およびその他）を記入する	C	C
処理結果	不具合を対応した方法。下記は、参考例（bugzillaの参考例） FIXED (開発担当者が)修正済み INVALID 不具合ではない（仕様やテストオペレーションミスが考えられる） WONTFIX 修正しない不具合 LATER 次期バージョンで対応 REMIND 制限事項などコメントで対応 DUPLICATE 重複している不具合 WORKSFORME 再現しない、または問題が確認できない	C	C
不具合件名（サマリ）	不具合の要約。一般的には、不具合を一覧管理するときに利用されるが、文字数制限がある上、不具合の重複登録を防ぐ為に適格な表現が求められる	C	C

用語	説明	レベル 3	レベル 4
その他の用語			
インシデント	テスト実行時に検出された期待値と異なる全ての事象。インシデントの段階では、不具合とは断定されない	C	C
不具合	インシデントを解析した結果、修正すべき問題と確認された事象	C	C
テストケースエラー	インシデントを解析した結果、テストケースの問題と確認された事象	C	C
オペレーションミス	インシデントを解析した結果、テストオペレーションの問題と確認された事象	C	C
バグ	通常、ソフトウェアの欠陥を表わす	C	C
欠陥	不具合の原因	C	C
不具合ステータス管理	不具合の検出から修正対応および修正対応確認までの流れを管理する。テスト実行中は、随時状況を管理し優先度や重要度の高い不具合の効率良い対応を実施する	C	C
テストフェーズ	テストの工程の単位や期間	C	C
テストサイクル	テスト実行を複数の期間に分けて実行する単位。一般的には、テスト対象のリリースのタイミングやテスト実行の視点を考慮して分割される	C	C
UI	ユーザ・インタフェースの略。テスト対象に対する操作などを表わす	C	C
画面遷移	複数の画面のインタフェースを持つテスト対象の画面のフロー（流れ）を表わす。画面遷移図は、（操作）画面などのフローが定義されている	B	C
状態遷移	複数の状態を持つテスト対象の状態のフロー（流れ）を表わす。状態遷移図は、テスト対象が持つ状態のフローが定義されている	B	C

4 付録 A 本書について

本書は、I V I A が I T 製品を対象としたソフトウェアテスト技術者の育成に用いるために、ソフトウェアテストの技術・知識を習得が容易な形式で体系的にまとめました。本書は、下記の目的で利用することを前提としています。

ソフトウェアテスト技術者のスキルトレーニング
ソフトウェアテスト技術者の知識体系の習得
I T 検証技術者試験

および は、ソフトウェアテスト技術者のトレーニングを実施する教育機関や教育ベンダーを I V I A が認定する基準として利用されます。

は、I V I A が実施する I T 検証技術者試験の試験レベルと出題範囲を示します。I T 検証技術者試験については、別途 I T 検証技術者試験の要項を参照して下さい。

- 習得のレベルについて

本書では、習得のレベルを 3 レベルで表示している。各レベルは、以下の通りです。

レベル	レベルの説明	備考
A	用語を認知しており、会話で利用できる	認知レベル
B	用語・内容を認知しており、会話・文書作成で利用できる	引用レベル
C	用語・内容を理解しており、実務で利用できる	実用レベル

通常実務では、B および C レベルが求められます。本シラバスでは、I V I A が定めた各キャリアレベル（ミドル レベル 3、レベル 4）に合わせた習得のレベルを学習課題ごとに表記しています。A レベルに関しては、引用・実用する為の習得は不要ですが、用語として認知すべきものとしています。また、習得のレベルの表記がないもの（ ）は、対象のレベルでは習得の不要を意味します。

- 本書は、ミドルレベルのレベル 3 とレベル 4 の I T 検証技術者を対象とします。
- 本書は、随時更新される場合があります。定期的に、バージョンを確認することをお勧めします。