

テスト技術者資格制度
モデルベースドテスト担当者(**CT-MBT**)
シラバス

V1.1.J01

International Software Testing Qualifications Board



著作権について

Copyright Notice © International Software Testing Qualifications Board (以降ではISTQB®と参照)

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

Copyright © 2015 the authors: Stephan Christmann (chair), Anne Kramer, Bruno Legeard, Armin Metzger, Natasa Micuda, Thomas Mueller, Stephan Schulz

ここに、本書の著者グループは、著作権を International Software Testing Qualifications Board (ISTQB®) に移転する。本書の著者グループ（現在の所有権保持者）とISTQB®（将来の所有権保持者）は、以下の使用条件に合意している。

著作者やISTQB®が本シラバスの出典および著作権の保有者であることを明記する限りにおいて、個人またはトレーニング会社が本シラバスをトレーニングコースの基礎に利用してもよい。また、ISTQB®が承認する各国委員会にトレーニング教材の公式な認定のために提出した後は、それらのトレーニングコースの広告にて、本シラバスについて言及してもよい。

著作者やISTQB®が本シラバスの出典および著作権の保有者であることを明記する限りにおいて、個人または個人のグループが本シラバスを記事、書籍、その他の派生著作物に使用してもよい。

ISTQB®の書面による承認を得ることなく、このシラバスを他の用途に使用することは禁じられている。

ISTQB®が承認する各国委員会は本シラバスを翻訳し、シラバスのライセンス（またはその翻訳）を他の団体に付与してもよい。

Translation Copyright © 2020, Japan Software Testing Qualifications Board (JSTQB®), all rights reserved. 日本語翻訳版の著作権は JSTQB®が有するものです。本書の全部、または一部を無断で複製し利用することは、著作権法の例外を除き、禁じられています。

改訂履歴

ISTQB®

バージョン	日付	備考
V1.0	2015年10月23日	CT-MBT 1.0 - 一般リリース版
V1.1	2024年2月23日	CT-MBT 1.1 - 一般リリース版

JSTQB®

バージョン	日付	備考
V1.1.J01	2025年9月1日	Version 1.1 の日本語翻訳版

目次

著作権について	2
改訂履歴	3
目次	4
謝辞	7
0 イントロダクション	9
0.1 本シラバスの目的	9
0.2 ソフトウェアテスト技術者資格制度 モデルベースドテスト担当者	9
0.3 テスト担当者のキャリアパス	9
0.4 ビジネス成果	9
0.5 試験対象の学習の目的と知識レベル	10
0.6 モデルベースドテスト担当者の認定資格試験	10
0.7 認定審査	11
0.8 標準の取り扱い	11
0.9 最新情報の維持	11
0.10 詳細レベル	11
0.11 本シラバスの構成	11
1 モデルベースドテスト入門 - 90 分	13
1.1 MBT の目的とモチベーション	13
1.1.1 MBT に対するモチベーション	13
1.1.2 MBT への誤解を招く期待と落とし穴	14
1.2 テストプロセスにおける MBT 活動と作業成果物	15
1.2.1 MBT 特有の活動	15
1.2.2 MBT に不可欠な作業成果物（入力と出力）	16
1.3 ソフトウェア開発ライフサイクルへの MBT の統合	17
1.3.1 シーケンシャルソフトウェア開発ライフサイクル・イテレーティブソフトウェア開発ライフサイクルにおける MBT	17
1.3.2 要件のエンジニアリングのサポート	18
2 MBT モデリング - 250 分 (K3)	19
2.1 MBT のモデリング	19
2.1.1 MBT のモデリング活動	20
2.1.2 MBT モデルの対象と焦点	20
2.1.3 テスト目的に応じた MBT モデル	21
2.2 MBT モデルのための言語	22
2.2.1 MBT のモデリング言語の主要な分類	22
2.2.2 システムやプロジェクトの目的に応じた言語の分類	23

2.3	MBT モデリング活動のよい実践例	23
2.3.1	MBT モデルの品質特性	23
2.3.2	MBT モデル設計における典型的な間違いや落とし穴	24
2.3.3	要件とプロセス関連情報の MBT モデルへのリンク	24
2.3.4	MBT のモデリングガイドライン	25
2.3.5	既存のシステム設計や要件モデルの再利用	25
2.3.6	モデリング活動のためのツールサポート	26
2.3.7	反復的なモデルの開発、レビュー、検証	26
3	テストケース生成のための選択基準 - 205 分	28
3.1	MBT のテスト選択基準の分類	28
3.1.1	テストの選択基準	28
3.1.2	テストケース選択の実際	30
3.1.3	テスト選択基準の例	30
3.1.4	テスト選択基準の例	30
3.2	テスト選択基準の適用	31
3.2.1	テスト生成における自動化の度合い	31
3.2.2	特定のテスト選択基準の長所と短所	31
3.2.3	MBT のテスト選択のよい実践例	31
4	MBT テストの実装と実行 - 120 分	33
4.1	MBT テストの実装と実行の具体的な方法	33
4.1.1	MBT コンテキストにおけるハイレベルテストケースとローレベルテストケース	33
4.1.2	さまざまな種類のテスト実行	34
4.1.3	変更による MBT の作業成果物への影響	34
4.2	MBT におけるテストの適合の活動	35
5	MBT アプローチの評価と導入 - 60 分	37
5.1	MBT の導入を評価する	37
5.1.1	MBT 導入の ROI 要因	37
5.1.2	組織の目的と MBT アプローチの特徴との関係	39
5.1.3	メトリクスと重要業績評価指標 (KPI)	39
5.2	MBT アプローチ導入の管理とモニタリング	40
5.2.1	MBT を導入する際のよい実践例	40
5.2.2	MBT のコスト要因	41
5.2.3	MBT ツールの統合	42
6	略語について	43
7	参考文献	44
	標準	44

ISTQB ドキュメント.....	44
本シラバスで参照している文献.....	44
その他の文献.....	44
8 付録 A - 簡易的なモデリング言語.....	45
8.1 ワークフローのための簡易的なグラフィカルモデリング言語.....	45
8.2 状態遷移図のための簡易的なグラフィカルモデリング言語.....	47
9 付録 B - 学習の目的と知識の認知レベル.....	48
10 付録 C ビジネス成果と学習の目的のトレーサビリティマトリクス.....	50

謝辞

この文書は、International Software Testing Qualifications Board Foundation Level Working Group のチームによって作成された。

ISTQB® Foundation Level Certified Model-Based Tester (CTFL-MBT) チームは、レビューチームとすべてのナショナルボードからの提案と意見に感謝したい。

CTFL-MBT シラバスが完成した時点で、モデルベースドテスト担当者ワーキンググループのメンバーは以下の通りであった。Stephan Christmann (MBT Working Group Chair), Anne Kramer, Bruno Legeard (MBT Author Group Co-Chair), Armin Metzger (MBT Author Group Co-Chair), Natasa Micuda (MBT Exam Question Group Chair), Thomas Mueller (ISTQB CTFL Working Group Chair), Stephan Schulz (MBT Review Group Chair)

著者 : Stephan Christmann, Lars Frantzen, Anne Kramer, Bruno Legeard, Armin Metzger, Thomas Mueller, Ina Schieferdecker, Stephan Weissleder.

試験問題グループ : Bob Binder, Renzo Cerquozzi, Debra Friedenber, Willibald Krenn, Karl Meinke, Natasa Micuda, Michael Mlynarski, Ana Paiva

試験問題レビューグループ : Eddie Jaffuel, Ingvar Nordstrom, Adam Roman, Lucjan Stapp

レビューグループ : Stephen Bird, Thomas Borchsenius, Mieke Gevers, Paul Jorgensen, Beata Karpinska, Vipul Kocher, Gary Mogyorodi, Ingvar Nordström, Hans Schaefer, Romain Schmechta, Stephan Schulz, Szilard Szell, Tsuyoshi Yumoto

また、Foundation Level Certified Model-Based Tester Syllabus のレビュー、コメント、投票に参加した、National Boards およびモデルベースドテストの専門家コミュニティの以下の方々にも感謝したい。Patricia Alves, Clive Bates, Graham Bath, Rex Black, Armin Born, Bertrand Cornanguer, Carol Cornelius, Winfried Dulz, Elizabeta Fourneret, Debra Friedenber, Kobi Halperin, Kimmo Hakala, Matthias Hamburg, Kari Kakkonen, Jurian van de Laar, Alon Linetzki, Judy McKay, Ramit Manohar, Rik Marselis, Natalia Meergus, Ninna Morin, Klaus Olsen, Tal Pe'er, Michael Pilaeten, Meile Posthuma, Ian Ross, Mark Utting, Ester Zabar

本ドキュメントは、2015年10月23日に開催されたISTQB®の総会で正式に承認され、発行された。

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

日本語翻訳ワーキンググループメンバー：

井芹 久美子 (ASTER)
井芹 洋輝 (トヨタ自動車)
大段 智広 (DMM.com)
蛸島 昭之 (サイバネット MBSE)
須原 秀敏 (ベリサーブ)
福田 里奈 (メルカリ)
山上 直宏 (デンソー)

0 イントロダクション

0.1 本シラバスの目的

本シラバスは、国際ソフトウェアテスト技術者資格モデルベースドテスト担当者のベースとなる。ISTQB®は、本シラバスを以下の趣旨で提供する。

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

ISTQB®では、事前に書面による申請があった場合に限り、第三者が本シラバスを先に定めた以外の目的での使用を許諾することがある。

0.2 ソフトウェアテスト技術者資格制度 モデルベースドテスト担当者

モデルベースドテスト担当者資格は、ソフトウェアテストに関与するあらゆる人々を対象にする。資格の対象者には、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、テストマネージャー、ユーザー受け入れテスト担当者、ソフトウェア開発者などが含まれる。この資格は、プロジェクトマネージャー、品質マネージャー、ソフトウェアアーキテクト、ソフトウェア開発マネージャー、ビジネスアナリスト、IT部門長、経営コンサルタントなど、ソフトウェアテストの基本的な理解を求めるすべての人にとっても適切である。認定資格の保持者は、ソフトウェアテスト資格での上位レベルに進むことが可能である。

0.3 テスト担当者のキャリアパス

ISTQB®スキームは、キャリアのあらゆる段階にあるテストのプロフェッショナルを支援し、知識の幅と深さの両方を提供する。ISTQB®モデルベースドテスト担当者認定資格取得者は、**Core Advanced Levels**（テストアナリスト、テクニカルテストアナリスト、テストマネージャー）その後、**Expert Levels**（テストマネジメントまたはテストプロセス改善担当）へと興味を広げられる。

0.4 ビジネス成果

モデルベースドテスト担当者の認定取得者に求められるビジネス成果を掲載している。

MBT-BO1	標準的な用語と確立された MBT の概念、プロセス、技法を使用して、モデルベースのテストチームで協力する。
MBT-BO2	テストプロセスにモデルベースのテストを適用し、統合する。
MBT-BO3	確立された技術とモデルベースドテストのベストプラクティスを使用して、効果的に MBT モデルを作成および維持する。
MBT-BO4	リスクとテスト対象の機能の価値を考慮して、MBT モデルからテスト作業成果物を選択、作成、および維持する。
MBT-BO5	組織が品質保証プロセスをより建設的かつ効率的に改善するのを支援する。

0.5 試験対象の学習の目的と知識レベル

学習の目的はビジネス成果を支援し、モデルベースドテスト担当者の認定資格試験を作成するために使用する。

全体を通して、本シラバスのすべての内容は、導入と付録を除いて K1 レベルで試験対象となる。つまり、受験志願者は、本シラバスの 5 つの章で説明されているキーワードと概念について認識し、記憶し、想起することになる。本シラバスでは各章の先頭で以下の分類にて、「学習の目的」を示している。

- K1: 記憶
- K2: 理解
- K3: 適用

学習の目的の詳細と例は付録 B に添付する。各章の章見出しの下にキーワードとしてリストアップされているすべての用語の定義は、「学習の目的」には明示的に述べられていないとしても「記憶」しておくべき (K1) レベルとなる。

0.6 モデルベースドテスト担当者の認定資格試験

モデルベースドテスト担当者の認定資格試験は本シラバスに基づく。試験問題に対する解答は本シラバスの複数の節に基づく資料の使用が要件となる場合がある。本シラバスのすべての節は、「イントロダクション」と「付録」を除いて試験対象である。標準、文献は参考資料として含まれているが、それらに関して本シラバス自体の中で要約されている以上の内容は試験対象ではない。

ISTQB®ソフトウェアテスト向けテスト技術者資格制度モデルベースドテスト担当者の構造と規則については [ISTQB_EXAM_S&R] を参照されたい。ISTQB®ソフトウェアテスト向けテスト技術者資格制度モデルベースドテスト担当者を受験するための基準は、候補者がソフトウェアテストに興味を持っていることである。しかし、候補者には以下のことが強く推奨される：

- 少なくともソフトウェア開発またはソフトウェアテストの基本的な経験があること。例えば、システムまたはユーザー受け入れテスト担当者としての 6 ヶ月間の経験、またはソフトウェア開発者としての 6 ヶ月間の経験があること。
- ISTQB 基準に認定されたコースを受講すること (ISTQB 認定委員会のいずれかによって認定されたコース)。

受験基準に関する注意事項：ISTQB®ソフトウェアテスト向けテスト技術者資格制度 Foundation Level [ISTQB_FL_SYL] を取得した後で ISTQB®ソフトウェアテスト向けテスト技術者資格制度モデルベースドテスト担当者を受験すること。

0.7 認定審査

ISTQB®のメンバー委員会にて、教育コースの教材が本シラバスに従っている教育機関を認定する。教育機関はメンバー委員会または認定を行う機関から認定ガイドラインを入手しなければならない。教育コースがシラバスに従っていると認定されると、教育コースの一部として ISTQB®の試験を実施することができる。本シラバスの認定ガイドラインは、プロセスマネジメント・コンプライアンスワーキンググループが発行する一般的な認定ガイドラインに準ずる。

0.8 標準の取り扱い

Foundation シラバスには、参照する標準がある（IEEE や ISO 標準など）。これらの参考文献は、（品質特性に関する ISO 25010 の参考文献のように）枠組みを提供し、読者が望む場合には追加情報のソースを提供するためのものである。標準ドキュメントは試験を目的としたものではない。標準の詳細については、第 6 章を参照のこと。

0.9 最新情報の維持

ソフトウェア業界は急速に変化している。これらの変化に対応し、関係者が関連する最新の情報にアクセスできるように、ISTQB ワーキンググループは、www.istqb.org ウェブサイトにリンクを設け、サポートドキュメントや標準の変更点などを紹介している。これらの情報は、Foundation のシラバスでは試験対象外である。

0.10 詳細レベル

本シラバスの詳細レベルは国際的に一貫した教育と試験を可能にする。このゴールを達成するために本シラバスは以下の内容で構成されている。

- モデルベースドテスト担当者の意図について説明する全般的な指導の目的
- 想起することができなければならない用語（キーワード）のリスト
- 達成すべき認知的な学習の成果について説明している、知識領域ごとの学習の目的
- 認められた情報源の参照を含む主要なコンセプトの説明

本シラバスの内容はソフトウェアテストの全知識領域の説明ではない。詳細レベルは、モデルベースドテスト担当者のトレーニングコースでカバーされることを示している。本シラバスはモデルベースドテストを採用際に、すべてのソフトウェアプロジェクトに適用できるテストの概念と技法に焦点を置いて説明している。

0.11 本シラバスの構成

6 つの章で構成しており、すべて試験対象である。各章の一番上の見出しは、章の学習時間を指定している。章より下のレベルでは、時間は指定されていない。認定トレーニングコースでは、本シラバスは 1135 分（18 時間 55 分）の講義を必要とし、6 つの章で以下のように配分する。

5つの章で構成されており、すべて試験対象である。各章の一番上の見出しは章の時間を指定している。章より下のレベルでは、時間は指定されていない。認定トレーニングコースでは、本シラバスは **725 分 (12 時間 5 分)** の講義を必要とし、**5つの章**で以下のように配分する。

- **第1章：モデルベースドテスト入門 - 90 分**
 - **MBT** の目的、動機、潜在的な落とし穴について学び、それをソフトウェア開発ライフサイクル内でどのように統合するかを理解する。要求エンジニアリングにおける **MBT** の役割についても学ぶ。
 - また、テストプロセス内での具体的な **MBT** 活動と成果物を掘り下げ、それらの重要性と機能に重点を置く。
- **第2章：MBT モデリング - 250 分**
 - ワークフローに基づいたモデリング言語と状態遷移に基づいたモデリング言語を用いて、テスト対象のシンプルな **MBT** モデルを開発する方法を学び、テスト目的がこれらのモデルに与える影響を理解する。
 - **MBT** で使用されるさまざまなモデリング言語のカテゴリについて学び、それらがさまざまなテスト目的とどのように一致するかを理解し、**MBT** モデリング活動における品質特性と古典的な間違いを想起する。
 - **MBT** モデルに要求とプロセス情報を統合する重要性、効果的なモデリングのガイドライン、モデル再利用の適切なシナリオ、**MBT** モデリングをサポートするツールの種類、および反復的な **MBT** モデル開発、レビュー、検証のプロセスについて学ぶ。
- **第3章：テストケース生成のための選択基準 - 205 分**
 - モデルベースドテストのさまざまなテスト選択基準のファミリーを分類し、特定の目的を達成するためのテストケースを生成する方法を学び、**MBT** テスト選択基準と **ISTQB Foundation Level** テスト技術の関係を理解する。
 - テスト成果物生成の自動化の程度、特定のテスト選択基準を **MBT** モデルに適用する方法、およびこれらの基準を選択する際のベストプラクティスについて学ぶ。
- **第4章：MBT テストの実装と実行 - 120 分**
 - **MBT** テストの実装と実行に関する具体的な事項、高レベルと低レベルのテストケースの違い、さまざまな種類のテスト実行、および要求、テスト対象、目的の変更に応じた **MBT** モデルとテスト生成の更新方法について学ぶ。
 - モデルベースドテストの文脈でのテスト実行のための必要なテスト適応の種類について学ぶ。
- **第5章：MBT アプローチの評価と導入 - 60 分**
 - **ROI** 要素を説明し、プロジェクトの目的を **MBT** の特性に関連付け、**MBT** 活動のためのメトリクスと主要な指標を想起することで、**MBT** アプローチの展開を評価する方法を学ぶ。
 - テストと変更管理のベストプラクティスを含む **MBT** アプローチの導入を管理および監視する方法、コスト要素の理解、構成管理、要求管理、テスト管理、およびテスト自動化ツールと **MBT** ツールの統合の重要性について学ぶ。

1 モデルベースドテスト入門 - 90 分

キーワード

MBT モデル、モデルベースドテスト

第 1 章の学習の目的

1.1 MBT の目的とモチベーション

MBT-1.1.1 (K2) MBT に期待される利点を説明する。

MBT-1.1.2 (K2) MBT に対する誤解を招くような期待と落とし穴を説明する。

1.2 テストプロセスにおける MBT 活動と作業成果物

MBT-1.2.1 (K2) テストプロセスに展開されたときの MBT に特有の活動を要約する。

MBT-1.2.2 (K1) MBT に不可欠な作業成果物（入力と出力）を想起する。

1.3 ソフトウェア開発ライフサイクルへの MBT の統合

MBT-1.3.1 (K2) MBT がソフトウェア開発ライフサイクルプロセスにどのように統合されるかを説明する。

MBT-1.3.2 (K2) MBT が要件のエンジニアリングをどのようにサポートするか説明する。

1.1 MBT の目的とモチベーション

モデルベースドテスト (MBT) は、テストのためのモデルを使用する高度なテストアプローチである。同値分割法、境界値分析、デシジョンテーブルテスト、状態遷移テストなどの古典的なテスト技法を拡張し、サポートする。その基本的な考え方は、以下のようにテスト設計とテスト実装の活動の品質と効率を向上させることである。

- プロジェクトのテスト目的に基づいて、一般的にツールを使用して包括的な MBT モデルを設計する。
- テスト設計仕様として MBT モデルを提供する。このモデルには、MBT モデルから直接テストケースを自動生成するのに十分な、高度に形式化されていてかつ詳細な情報が含まれている。

MBT とその作業成果物は、手法、技術環境、ツール、特定のライフサイクルプロセスだけでなく、組織のプロセスと密接に統合されている。

1.1.1 MBT に対するモチベーション

テストの活動には以下の 2 つの主な側面がある。これらは MBT に対する基本的なモチベーションを提供し、MBT がどのようにテストの品質の向上をサポートするかを説明する。

- 効果
 - モデリングは、ステークホルダーとの密接なコミュニケーションを育むプロセスである。

- コミュニケーションを改善することで、特定のドメインにおける要件に対する共通の認識と理解を生み出し、潜在的な誤解を検出できる。
 - グラフィカルな **MBT** モデルの場合、プロジェクトのステークホルダー（例えば、ビジネスアナリスト）がより簡単に関わられるようにする。
 - モデリングは、特定のドメインにおけるテスト担当者の継続的な能力向上をサポートする。
 - **MBT** モデルの抽象度により、テストで欠陥や不正が発見されやすいシステムの部分を特定しやすくなる。
- 実際のシステムができる前に、テストケースの生成と分析が可能になる。
- 効率性
 - 早期のモデリングとモデルの検証・妥当性確認により、開発プロセスの初期段階で欠陥を回避することができる。**MBT** のモデルをプロジェクトのステークホルダー（例えば、ビジネスアナリスト、開発者）と共有して、要件を検証し、要件内の相違を特定できる。このようにして、**MBT** は「早期テスト」の原則をサポートする。
 - 過去のプロジェクトの **MBT** の作業成果物は、テストのさらなる開発のために（再）使用することができる。
 - **MBT** は（例えば、テストウェア生成の）自動化を支援し、テストウェアの作成やメンテナンスを手作業で行った場合に発生する欠陥を低減する。
 - 同一のモデルから異なるテストスイートを生成でき（例えば、テスト選択基準の差異に応じて）、テストベースやテスト目的の変更への効率的な適応を促進できる。
 - **MBT** はさまざまなテスト目的に対して使用でき、テスト期間中、さまざまなテストレベルやテストタイプをカバーできる。
 - **MBT** モデルが単一のメンテナンスポイントを提供するため、**MBT** は要件変更時のメンテナンスコストの削減に役立つ。

ツールのサポートがある場合、モデルベースのテストアプローチは、**MBT** ツールによる自動テスト生成機能を通じて、テストカバレッジを拡張する。この自動テストケース生成はテストサイクルを短縮し、テストのボトルネックを減少させる。

1.1.2 MBT への誤解を招く期待と落とし穴

MBT を使おうとしているテストチームは、**MBT** のテストアプローチの利点と限界について、現実的な期待を持つ必要がある。典型的な誤解を招く期待、勘違い、落とし穴には以下が含まれる。

- **MBT** はすべての問題を解決する
MBT は、従来のテスト技法（[ISTQB_FL_SYL]参照）に取って代わるものではなく、テスト担当者がドメインの理解を深め、より効果的かつ効率的にテストを実施できるようにサポートするものである。従来のテスト技法を無視、および／または、誤用しているテストチームは、**MBT** を導入しても問題を解決することはできない。
- **MBT** は単にツールの問題にすぎない
MBT を成功させるためには適切なツールのサポートが不可欠だが、ツールの入手が最初のステップであってはならない。むしろ、**MBT** を導入するかどうかの決定は、テストの改善に関する測定

可能な目的の定義に基づくべきである。MBT は、テストプロセス全体に影響を与える（1.2 節）。従って、MBT を導入するには、組織内のプロセスやツールの変更をリードするために、マネジメントからの強力なサポートが必要となる。

- モデルは常に正しい
手動でのテスト設計と同様に、テスト担当者が MBT モデルを作成する際に欠陥が入り込む可能性がある。MBT モデルは、レビュー、モデルの静的解析、モデルのシミュレーションなどを適用して、徹底的に検証・妥当性確認をする必要がある。MBT モデルの変更は、変更されたモデル要素に関連する、すべての生成したテストウェアに伝搬するため、各変更は実装前に慎重にレビューする必要がある。
- テストケースの爆発的増加が起こる
MBT を適用することで、テスト設計の手法やカバレッジを改善できる。純粋な組み合わせテストケースの生成の場合、これはテストケースの爆発的増加につながる可能性がある。この課題は、テストケース生成の戦略やアルゴリズムを調整したり、インテリジェントなフィルタの仕組みを適用したりすることで解決できる。

1.2 テストプロセスにおける MBT 活動と作業成果物

1.2.1 MBT 特有の活動

モデルベースドテストを実施する場合、テストプロセスには、従来のテスト分析・設計では通常行われない MBT 特有の活動が含まれる。これには以下が含まれる。

- MBT モデリング活動（MBT モデルの管理に関する活動、モデリングアプローチの開発と統合、モデリングガイドラインの定義、MBT モデルの構造の開発、MBT モデルの特定の図などのモデル要素の開発、ツール関連の活動）。
- MBT アプローチと選択基準に基づいたテストウェアの生成（例えば、テストケースの生成）。

少なくとも、モデルベースドテストは、テストプロセスにおけるテスト分析とテスト設計の活動に影響を与える[ISTQB_FL_SYL]。テスト目的に応じて、MBT は、テストプロセスのすべての部分に関連するより広い範囲で使用することもできる。以下のような MBT 特有の活動を考慮する必要がある。

- テストの計画には、MBT 特有の活動（MBT ツール、ガイドライン、特定のメトリクス、プロジェクト計画のベースラインの一部としての MBT の作業成果物など）の実装が含まれることがある。
- テストの分析と設計には、MBT のモデリング活動、テスト選択基準、高度なテストカバレッジメトリクスの選択と適用が含まれることがある。シフトレフトアプローチにおいては、テスト分析とテスト設計が 1 つの活動に統合される。
- テストの実装と実行には、MBT のテスト生成と MBT のテスト適合が含まれることがある。
- テストのモニタリングとコントロールには、（構造的および／または明示的なモデル情報に基づく）高度なカバレッジメトリクスとモデルベースの影響度分析を含むことがある。
- テスト完了作業には、将来のプロジェクトで再利用できるようにモデルライブラリの構築が含まれることがある。

テストプロセスへの MBT の影響で焦点となるのが、プロセスの自動化と作業成果物の生成である。MBT はテスト設計活動のシフトレフトをサポートする。MBT は、プロジェクト初期でも要件の検証方法とし

て役に立ち、特にグラフィカルなモデルを使用することで、コミュニケーションの改善を促進することがある。モデルベースドテストは、他のテスト技術（[ISTQB_FL_SYL] – 第4章）、例えば探索的テストと補完することができる。

MBT を用いたテストプロセスの活動は、従来のテストプロセスで行われていたものと似ているが、その実施方法は大きく変わる可能性がある。1.1 節で述べたように **MBT** は、以下の特徴を持つ。

- 品質、工数、コミュニケーション、テストプロセスに関わるステークホルダーに影響を与える。
- テスト設計活動をテストプロセスの初期段階に移行させる。

MBT の採用を確かなものにするためには、**MBT** に取り組むチームとともに、手法や関連する変更を受け入れることが重要である。

1.2.2 MBT に不可欠な作業成果物（入力と出力）

MBT モデルを確保する方法には、専用の **MBT** モデルを開発する方法か、システム設計のために開発されたモデルを再利用する方法がある（2.3.5 項参照）。**MBT** モデルは、適用するモデリングのコンセプトに応じて、異なる抽象度と特定のテストに関する情報をサポートする。**MBT** モデルから得られる作業成果物は、これらの異なる抽象化レベルを反映することになる。

情報と抽象度に応じて、**MBT** は異なる入出力の作業成果物とともにテストプロセスに統合する。

入力の作業成果物は以下の通りである。

- テスト戦略
- 要件やその他のテスト対象、テスト条件、口頭での情報、既存の設計やモデルなどのテストベース
- 過去のテスト実行活動におけるインシデントレポートや欠陥レポート、テスト結果記録、テスト実行結果記録
- 手法やプロセスのガイドライン、ツールのドキュメント

出力の作業成果物には、以下のようなさまざまな種類のテストウェアがある。

- **MBT** モデル
- テスト計画の一部（テストすべきフィーチャー、テスト環境など）、テストスケジュール、テストメトリクス
- テストシナリオ、テストスイート、テスト実行スケジュール、テスト設計仕様
- テストケース、テスト手順、テストデータ、テストスクリプト、テスト適合レイヤー（仕様、コード）など
- 生成されたテストとテストベース、特に要件および欠陥レポートとの間の双方向のトレーサビリティマトリクス

1.3 ソフトウェア開発ライフサイクルへの MBT の統合

1.3.1 シーケンシャルソフトウェア開発ライフサイクル・イテレーティブソフトウェア開発ライフサイクルにおける MBT

この節では、ソフトウェア開発ライフサイクルモデルの 2 つの主要な分類、すなわち、V 字モデルのようなシーケンシャルなもの、アジャイルソフトウェア開発のようなイテレーティブ・インクリメンタルなものについて考える。いずれの場合も、MBT アプローチは、主にプロジェクトのテスト目的に適合させる必要があり、テスト目的に対応した MBT のメリットを得ることができる。

組織は、ソフトウェア開発ライフサイクルをさまざまな方法で具体化する。それに応じて、MBT を用いたテストプロセスも適応させる必要がある。例えば、MBT は、プロジェクトのテスト目的に応じて、あるプロジェクトでは受け入れテストに焦点を当て、別のプロジェクトでは自動化されたシステムテストに焦点を当てる場合がある。

シーケンシャルなソフトウェア開発ライフサイクルとイテレーティブなソフトウェア開発ライフサイクルに共通するものとして、以下のものがある。

- テストレベル
 - MBT は、機能要件や期待する振る舞いの複雑度を抽象化することができるため、主に上位のテストレベル（システム、システム統合、受け入れ）で使用される。
 - MBT は、コンポーネントとコンポーネント統合テストではあまり使われていないが、コードアノテーション技術をベースにしたアプローチもある。
- テストタイプ
 - MBT は主に、システムおよび/またはその環境で期待する振る舞いを表す MBT モデルを用いた機能テストに使用する。
 - 改良された MBT モデルや専用の MBT モデルは、非機能テスト（セキュリティテスト、負荷/ストレステスト、信頼性テスト）に使用することができる。

MBT モデルはテスト担当者が所有することができる。しかし、アジャイルソフトウェア開発では、チーム内で共有の責任となることがある。開発者は、テスト自動化を目的としたモデルに興味を持つ場合がある。特に、MBT モデルから生成された自動化スクリプトを、早期かつ頻繁なフィードバックを提供するために継続的インテグレーションに使用する場合は、なおさらである。

シーケンシャルなソフトウェア開発ライフサイクルにおける MBT の具体的な活動は以下の通りである。

- プロジェクトのできるだけ早い段階で MBT モデリングを開始する。
 - ステークホルダー間のコミュニケーションを活性化するため。
 - 不明確な要件、不完全な要件、矛盾した要件を早期に発見するため。
- 以下を含むようにテスト計画の活動と役割を適応させる。
 - 新しいテストウェア要素（例えば、MBT モデル、テスト選択基準）
 - MBT 活動の進捗レポート

アジャイルソフトウェア開発における MBT の主な適応は以下の通りである。

- MBT モデルはイテレーティブかつインクリメンタルに開発され、テストはイテレーションの内容に応じて生成する。

- ユーザーストーリーは、テストベースの一部である。対象となる各ユーザーストーリーはモデル内で関連付けられ、ユーザーストーリーとテストの間の双方向のトレーサビリティを自動的に管理する。
- **MBT** を、受け入れテスト駆動開発（ATDD）を実装するために使用できる。 [ISTQB_FL_SYL] の 2.1.3 節を参照
- アジャイルのよい実践例であるチーム全体アプローチに関しては、モデルベースドテストを使用するテスト担当者は、開発者やビジネス代表者とともにアジャイルチームの一員である。 [ISTQB_FL_SYL] の 1.5.2 節を参照。

1.3.2 要件のエンジニアリングのサポート

MBT は、以下のような要件のエンジニアリングに役立つ。

- ビジネスプロセスモデルや状態遷移図などのグラフィカルな **MBT** モデルを提供することで、開発するソフトウェアに期待する振る舞いについて、ビジネス、開発、テストの間のコミュニケーションを促進する。これらのモデルは、すべてのステークホルダーが、期待する振る舞いの詳細について議論し検証するのに役立つ。
- **MBT** モデルの全部または一部をビジネス担当者や開発者と共有し、期待するソフトウェアの振る舞いについて共通の理解を得ることで、要件および／またはユーザーストーリーの品質を明確にし、向上させることができる。**MBT** モデルから生成されたテストは、ビジネス担当者や開発者がレビューできる可能性のあるユーザーシナリオを構成する。
- 要件を変更する可能性がある段階でも、早期の妥当性確認を支援する。

要件のエンジニアリングと **MBT** は連携している。要件と関連するリスクは、**MBT** のモデリングとテスト生成活動のための入力となる。**MBT** 活動は要件とテストの間の双方向のトレーサビリティの関係の作成と維持を自動化可能にする。

2 MBT モデリング - 250 分 (K3)

キーワード

MBT モデル、テストモデル

2 章の学習の目的

2.1 MBT のモデリング

- MBT-2.1.1 (K3) ワークフローベースのモデリング言語を使用して、テスト対象と事前に定義されたテスト目的のための単純な MBT モデルを開発する (8.1 節参照。「簡易的」とはモデリング要素が 15 個以下のものを意味する)。
- MBT-2.1.2 (K3) 状態遷移ベースのモデリング言語を使用して、テスト対象と事前に定義されたテスト目的のための単純な MBT モデルを開発する (8.2 節参照。「簡易的」とは 15 個以下のモデリング要素を意味する)。
- MBT-2.1.3 (K2) システムや環境、テストを記述した MBT モデルを例示する。
- MBT-2.1.4 (K2) MBT モデルがテスト目的にどのように依存するかの例を示す。

2.2 MBT モデルのための言語

- MBT-2.2.1 (K1) MBT によく使われるモデリング言語の分類の例を想起する。
- MBT-2.2.2 (K2) 異なるシステムやプロジェクトの目的に対するテスト目的に適したモデリング言語の分類の代表的なものを想起する。

2.3 MBT モデリング活動のよい実践例

- MBT-2.3.1 (K1) MBT モデルの品質特性を想起する。
- MBT-2.3.2 (K2) MBT のモデリング活動における典型的な間違いと落とし穴を説明する。
- MBT-2.3.3 (K2) 要件やプロセス関連情報を MBT モデルにリンクさせることの利点を説明する。
- MBT-2.3.4 (K2) MBT モデリングのガイドラインの必要性を説明する。
- MBT-2.3.5 (K2) (要件フェーズや開発フェーズにおける) 既存モデルの再利用が適切な場合とそうでない場合の例を示す。
- MBT-2.3.6 (K1) 特定の MBT モデリング活動をサポートするツールタイプを想起する。
- MBT-2.3.7 (K2) 反復的な MBT モデルの開発、レビュー、検証について要約する。

2.1 MBT のモデリング

テストを行うという状況では、MBT モデルは、何をテストすべきかを表現し、関係者間でコミュニケーションを行い、テスト設計に関連するすべての情報をまとめるのに適している。また、MBT モデルは、テストマネジメントの活動でも活用できる。

MBT モデルは、テストケースを導き出す (あるいは特定する) ことを目的として開発する。MBT モデルは、後のテストケースの生成に必要な情報を持っていないなければならない。例えば、プロジェクトのテストの目的に応じて合理的なテストケースを選択したり、テストオラクルの生成を実現したり、要件と生成さ

れたテストケースの間の双方向のトレーサビリティ確保をサポートしたりするための情報を持っていないなければならない。

MBT モデルの設計は、MBT において不可欠かつ大部分を構成する活動である。MBT モデルの品質は、MBT ベースのテストプロセスの成果物の品質に強い影響を与える。MBT モデルは多くの場合、ツールによって解析されるものであり、厳密な構文に従わなければならない。また、MBT モデルは、モデリングガイドラインで定義されているような他の品質特性をサポートしなければならない場合もある。

2.1.1 MBT のモデリング活動

すべてのモデルは、特定のモデリング言語で表現する。モデリング言語は、モデル要素を構成する作業成果物と、その言語でモデルを構築するために従うべきルールを定義する。モデリング言語の中には、構造、振る舞い、ドメイン固有の作業成果物を表現するものがある。

MBT モデリングでは、次の重要な質問に答えなければならない：

- テスト対象のどのような品質特性をモデル化するか？
機能性に焦点を当てるのが最も一般的である。ただし、性能やセキュリティなどの非機能の側面も MBT モデルの一部となる場合がある。モデルの管理しやすさ、MBT ベースのテストプロセスの効率を維持するために、テストやテストの目的に関係するシステムやその環境のみをモデル化することを推奨する。
- どのモデリング言語が適しているのか？
すべてのモデルは、特定のモデリング言語で表現する。モデリング言語は、モデルを構成する作業成果物と、その言語でモデルを構築するために従うべきルールを定義している。テスト対象物やその環境の構造、振る舞い、その他の側面（例えば、データ、ワークフロー、通信プロトコル）を表現するために、さまざまなモデリング言語がある。
- 適切な抽象度はどのようなものか？
抽象化は、複雑さを克服し、MBT モデルをテストの目的に集中させるのに有効である。しかし、モデルが抽象的であればあるほど、テスト実行のためのテストの適合がより困難になる。さらに、抽象度によって、その特定の抽象度でテスト設計を議論できる対象者やステークホルダーが決まることがしばしばある。

これらの質問に対する答えと、選択した MBT ツールセットは、MBT 活動に影響を与える。

注：2つの簡略化したグラフィカルモデリング言語を付録 A に記載している。トレーニングや試験の準備では、この2つの言語の使用をカバーする必要がある。

2.1.2 MBT モデルの対象と焦点

テスト目的に依存して、MBT モデルは異なる側面に着目する：

- 構造モデルとは、静的な構造を記述する。構造モデルの例としては、データオブジェクトの関係をテストするためのクラス図やデータモデリングのためのクラシフィケーションツリーなどがある。

- 振る舞いモデルは、動的な相互作用を記述する。振る舞いモデルの例としては、アクティビティやワークフローを記述したアクティビティ図やビジネスプロセスモデル、システムの入出力を記述した状態遷移図などがある。

MBT モデルがどう書かれるかは、MBT モデルの目的に強く依存する。このシラバスでは、3 種類のモデルの区分を検討している：

- システムモデル
システムモデルは、意図された通りにシステムを記述している。このモデルから生成されたテストケースは、システムがこのモデルに準拠しているかどうかを確認する。システムモデルの例としては、オブジェクト指向システムのクラス図や、リアクティブシステムの状態や状態遷移を記述した状態遷移図がある。
- 環境モデル／使用モデル
環境モデルとは、システムの環境を記述するものである。環境モデルの例としては、システムの予想される使用状況を記述したマルコフ連鎖モデルなどがある。
- テストモデル
テストモデルとは、（1 つまたは複数の）テストケースのモデルである。これには通常、テスト対象物に対して期待する動作とその評価が含まれる。テストモデルの例としては、（抽象的な）テストケースの記述や、テスト手順のグラフィカルな表現などがある。

MBT モデルは、通常、これらの区分のいくつかを組み合わせる。例えば、MBT モデルは、データ要素を含むテスト対象と、与えられたコンテキストにおけるテスト対象の使用方法を表現する場合がある。

2.1.3 テスト目的に応じた MBT モデル

MBT モデルを開発する際には、テストの目的を考慮することが重要である。これは、モデルの対象と焦点を当てる箇所を決定するためである。テストの目的とその目的に適した MBT モデルの例を以下の表に示す。

テストの目的	モデル例
ビジネスワークフローが正しく実装されているかを検証する	ワークフローを記述したビジネスプロセスモデル
特定の状態にあるときに、システムが要件に対して正しい応答を提供することを検証する	UML (Unified Modeling Language) のステートマシン
インターフェースの可用性を確認する	インターフェースの構造の説明
テスト対象がユーザーの期待する使用方法に適しているという確信を得る	ユーザーの行動を記述した利用モデル

2.2 MBT モデルのための言語

2.2.1 MBT のモデリング言語の主要な分類

モデルは、モデリング言語を使うことによって表現する。モデリング言語は次の要素で定義する：

- コンセプト（抽象構文とも呼ばれ、テキストで記述することが多いが、メタモデルで指定することもある）
- シンタックス（具象構文とも呼ばれ、しばしば文法規則によって定義する）
- セマンティクス（多くの場合、静的および動的なセマンティックルールによって定義する）

ダイアグラムは、UML のクラス図、シーケンス図、状態遷移図のような、グラフィカルモデリング言語でモデルを表現したものである。

MBT で使用可能なモデリング言語は数多くある。目的に合ったよい言語を選択するために、テスト担当者は MBT のモデリング言語の分類の主な特徴を知る必要がある：

- モデリングのコンセプト
モデリング言語は、サポートするコンセプトのセットが異なる。テストの目的に応じて、モデルは、構造の側面（例えば、ソフトウェアのアーキテクチャー、コンポーネント、インターフェース）、データの側面（例えば、ソフトウェアのパスオブジェクトのフォーマットやセマンティクス）、振る舞いの側面（例えば、ソフトウェアのアクティブオブジェクトのシナリオ、インタラクション、実行）を表現できる。2.1.2 項も参照のこと。
- 形式化
モデリング言語の形式化の度合いは、限定的な形式化から完全な形式化までさまざまである。後者は最も厳密であり、モデルの本格的な分析を可能にする。しかし前者の方がより実用的であることが多い。最低限、言語は形式化された構文（例えば構造化テキストや構造化テーブル）を持つ必要がある。また、形式的に定義されたセマンティクスを持つこともある。
- 表示形式
モデリング言語では、テキスト形式からグラフィカルな形式まで、さまざまな表示形式を使用し、組み合わせる場合がある。グラフィカルなフォーマットは、よりユーザーフレンドリーで読みやすく、テキストフォーマットは、よりツールフレンドリーで、記述とメンテナンスの両方において効率的であることが多い。

モデリング言語の分類には、以下のようなものがある：

- 構造モデルのための言語
ソフトウェアの構造要素（例えば、インターフェース、コンポーネント、階層）を指定するための言語。例えば、UML のコンポーネント図などがある。
- データモデルのための言語
データの型や値の指定をサポートする言語である。例えば、UML のクラス図や値の指定がある。
- 振る舞いモデルのための言語
UML のアクティビティ図や相互作用図、状態遷移図、ビジネスプロセスモデリング記法

(BPMN) などがある。

- 統合した言語
一般的に、モデリング言語は1つの特定の観点に限定するのではなく、複数の観点によるコンセプトを提供するものである。例えば、UMLでは、ソフトウェアのさまざまな側面を表すために、さまざまなダイアグラムを組み合わせ使用できる。

2.2.2 システムやプロジェクトの目的に応じた言語の分類

MBTのモデリング言語の選択は、ソフトウェアを開発しテストする必要があるプロジェクトの当初の目的と密接に関係している。また、ソフトウェアを構築するシステムの特性にも関係する。これにより、一連のモデリング言語の基準が生まれる。システム固有の属性との関連性に基づいて、これらの基準を評価し、あるものは受け入れ、あるものは取り除く必要がある場合がある。

以下のようなモデリング言語の使用例がある：

- コントロール/コマンドテスト対象の期待する動作を表すための状態遷移図
- 情報システムのエンドツーエンドテストのためのワークフローを表現するアクティビティ図またはBPMNモデル
- システムテストにおいてビジネスルールを表現するためのデシジョンテーブルと原因結果グラフ

また、モデリング言語とプロジェクトの目的の関係について、次のような例がある：

- ソフトウェア要件をコードやテストケースに紐付けることを必須とする、セーフティクリティカルあるいはセキュリティクリティカルなソフトウェアの認証に関する非機能要件
- 文書化のためのモデルへの注釈付けを行う場合での、監査や認証を目的とした、プロセスの文書化に関する非機能要件

2.3 MBT モデリング活動のよい実践例

2.3.1 MBT モデルの品質特性

モデルの品質は生成した出力物に対して直接的に影響する。モデルの品質特性を次に示す：

- シンタックスの品質（正しさ）
MBTモデルを、その形式的表現に関するルール（モデリング言語、モデリングガイドライン）と一致させることで、テストケースやテストデータの生成において、構文の誤りによる問題が発生することなく作業成果物を生成できる。
- セマンティックの品質（妥当性）
モデルの内容が、表現すべき内容に関して正しいこと。生成した作業成果物が「使用可能」であること（つまり、テストケースがテストケースの誤りで失敗することなく、テスト対象上で実行可能である）。
- 実用的な品質（適合性）
MBTモデルは、与えられたテスト目的および与えられたテストジェネレーターに適合しており、生成した作業成果物が期待を満たしていること。

ツールは、モデルのシンタックスと、最低限の一部のセマンティクスを確認する場合がある。レビューは、セマンティクスおよび実用性の品質を確認する。モデルシミュレーターは、モデルの品質を動的に確認することで、これらの静的な技術を補完する。

MBT モデルを段階的に開発し、テストを繰り返し生成（および実行）して、モデル、生成した作業成果物、テスト対象を、早期かつ定期的に確認することを推奨する。

2.3.2 MBT モデル設計における典型的な間違いや落とし穴

MBT の初心者は、モデルの設計において、以下に示すような間違いを犯しがちである：

- **MBT** モデルの詳細度が高すぎたり、低すぎたりする（与えられたテスト目的に対して、抽象度を誤っている）。
- すべてをカバーする1つのモデルを書こうとする。

MBT モデルは、テストの目的達成に関連する側面に焦点を当てるべきである。1つのモデルですべての側面をカバーするのではなく、特定の側面に重点を置いた2つのモデル（例えば、システムの実装を検証するための状態遷移図と、ワークフローを検証するためのアクティビティ図）を描くのがよい。

2.3.3 要件とプロセス関連情報の **MBT** モデルへのリンク

要件、**MBT** モデル、生成されたテストの間のトレーサビリティの確立は、モデルベースドテストにおけるよい実践例である。モデル要素と要件を結びつけることで、以下のようなメリットが得られる：

- **MBT** モデルのレビューが容易になる。
- **MBT** モデルから生成されたテストを、要件に自動的にリンクできる。
- 選択した要件に基づいてテストを生成できる。そして選択した要件の優先順位に基づいてどのテストを最初に実行するかを決定できるようになる。
- **MBT** で生成されたテストの要件カバレッジを測定できる。
- すべてのステークホルダー（例えばテスト担当者やテストマネージャー）が、要件変更の影響を分析して、必要なリグレッションテストの範囲を決定できる。
- テストケースジェネレーターが、トレーサビリティ文書（トレーサビリティマトリクス）を自動的に生成できる場合がある。

要件をモデル要素にリンクさせる実現方法はいくつかある。例えば、特定の図形で示した印や文字で示したキーワードによって、モデル内で要件を表現できる。

テスト担当者は、特定の意味を持つ新しいモデル要素として、あるいは既存のモデル要素の属性として、他の追加情報を追加できる。**MBT** モデルでのその追加情報の例に次のようなものがある：

- テスト手順（テスト実行のための一連の動作）の詳細内容
- テストスクリプトのセグメント（関数呼び出しやキーワード）
- リスク／ハザード（機能性や非機能の品質属性に関連するものや、プロジェクトマネジメントに関連するもの）
- （機能やテストの）優先順位
- テスト期間（予想実行時間）
- プロダクトの構成をテストするために必要なテストの機器やルール

追加情報は、次の例に示すように、モデルベースドテストをテストプロセス全体に統合するのに役立ち、さまざまな方法でテストマネジメントをサポートする：

- リスクや優先度の情報を、**MBT** モデルにリンクし、そしてそれらを生成したテストケースにリンクさせることで、リスクや優先度を利用可能にする（例えばテスト実行の優先順位付けに利用できる）。
- 他のプロジェクトの制約や目的を **MBT** モデルに反映することで、テスト計画への適応に役立つ。

インターフェース（例えばグラフィカルユーザーインターフェース）が仕様書でまだ安定していない場合、**MBT** モデルは定義済みの機能要件を包含する場合がある。その場合、機能性についてのキーワード駆動スクリプトを生成できることがある。そこでインターフェースが指定されたら、自動テストスクリプトの準備が整い、テスト適合レイヤーのみを実装する。

2.3.4 MBT のモデリングガイドライン

モデリングガイドラインとは、**MBT** モデルの設計、書き方、読み方の方法を文書化したものである。このガイドラインは次の役割を持つ：

- 関係するすべてのステークホルダーによる、**MBT** モデルの理解とレビューを支援する。
- プロセス、ドメイン、組織、または特定のツールに由来するシンタックス要件への適合をサポートする。
- 選択した表記法の範囲を制限／拡張する（例えば、**UML** のサブセットの定義や、モデル要素の異なる意味付けの定義）。
- さまざまなモデル作成者によって作られた **MBT** モデルの類似したシンタックスとセマンティクスを醸成する。
- よい実践例を広める。
- **MBT** モデルの保守性と再利用性を支える。

モデリングガイドラインのない **MBT** プロジェクトは、モデリングガイドラインのある **MBT** プロジェクトと比較して、テスト担当者がモデリングの典型的な間違いを犯しやすくなるため、新たなリスクが増大する。

2.3.5 既存のシステム設計や要件モデルの再利用

「古典的な」テストベースド文書からの **MBT** モデルの開発だけでなく、既存のシステム設計や要件モデル（例：ビジネスプロセスモデルやアクティビティ図）を **MBT** モデルの一部として直接再利用するアプローチは、可能ではあるものの、制限がある。再利用性の程度は、与えられたモデルとテストの目的によって異なる。例えば、システム設計の状態遷移図は、コマンド制御システムのテストに役立つ。しかしクラス図は、ワークフローベースのシステムテストには不十分である。

既存のシステム設計モデルを再利用すると、シングルソース問題が発生する。この文脈におけるシングルソース問題とは、元のシステム設計モデルの誤りが **MBT** モデルに伝搬することである。テストは、システムの開発や実装から必要な程度まで独立しているわけではない。そのため、システム設計用と **MBT** 用にそれぞれ分けたモデルを作成することを勧める。これにより、責任の分離が実現され、独立性が高まる。

システム設計モデルを何も変更せずに再利用した場合、システム設計モデルから派生したテストは、（妥当性確認ではない）単なる検証テストとなる。システムの実装がシステム設計モデルで記述された仕様に対応しているかどうかを確認するものになる。

妥当性確認の目的のために、テスト担当者は既存のモデル（例えば、要件元）から取り組み、それをテストの観点で強化する場合がある。MBT モデルに変換すると、最初のシステム設計モデルから成長する傾向にある。テスト担当者は、通常とは異なるシナリオやエラー状況をテストするために、モデルの要素（例えば、予期しない遷移）を追加する。最終的には、MBT モデルは、初期のシステム設計ではなく、テスト担当者の考え方を反映したものになる。

既存モデルの作成者がテスト担当者ではない場合を考えてみよう。その場合、以前の開発フェーズからのモデルは、MBT 関連の要件やガイドラインを必ずしも尊重するものでなくなる結果となる。再利用されたモデルは、必ずしも（ツール依存の）推奨する実践例に従っているとは限らず、テストケースの爆発的増加につながる可能性がある。

2.3.6 モデリング活動のためのツールサポート

モデルエディタは、モデルを記述するために使用する。これは、特定のモデリングツールの他に、フローチャートエディタ（グラフィカルモデルの場合）、テキストエディタ/スクリプトエディタ（テキストモデルの場合）になる場合がある。理論的には鉛筆と紙で MBT モデルを書くことも可能である。文書化の目的のためには単純な描画ツールで十分かもしれない。しかし MBT モデルでの単純な描画以上の処理をサポートしない。

MBT モデルエディタは、事前に定義されたモデル要素と、場合によってはいくつかのシンタックスおよびセマンティクスの確認機能を提供する場合がある。モデルを複数のパスで実行し、MBT モデルの妥当性確認を行うツールは、モデルシミュレーターと呼ばれる。

MBT モデルからテストケースやテストデータを自動的に得るためには、テストケース（やテストデータ）ジェネレーターが必要になる。

2.3.7 反復的なモデルの開発、レビュー、検証

複数のダイアグラムにまたがる、より複雑な振る舞いのモデル化は、MBT モデルのレビューだけでは十分でない状況を容易に引き起こす。テスト担当者は、MBT モデルから生成されたテストが期待通りのものか確認する必要がある。この状況をコントロールする方法は、イテレーティブ開発、モデルのレビュー、生成されたテストの作業成果物の頻繁なレビューである。そのレビューは、まずテスト担当者が行い、さらにソフトウェア開発ライフサイクルに関わる同僚や他のステークホルダーも参加する。

これらの最良な実践例を適用することで以下が実現する：

- テスト担当者は、他のステークホルダーとともに、テストすべき側面に関する自分の観点の妥当性を確認できる。
- テスト担当者は、MBT モデルの開発初期（テスト実行前）に、エラーの発見・修正や、モデルに欠けている部分の発見を実施できるようになる。
- テスト担当者は、不完全な要件や矛盾した要件を特定して共有できる。
- テストマネージャーは、プロジェクトに関連するリスクを管理できる。
- チームは、テストプロセスでの MBT モデリング活動の完了に必要な時間全体を短縮できる。

反復的な MBT モデルの開発は、反復的なテスト生成にも関連する。つまり、MBT モデルにいくつかの側面が追加されるたびに、テスト生成も更新される。本質的に、テスト生成は MBT モデルのシミュレーションを実現し、テストすべき振る舞いの側面についてのフィードバックを提供する。

ビジュアルな ATDD のアプローチ [ISTQB_AcT_SYL]においては、モデルのレビューと妥当性確認はバックログリファインメントの一部である。その場合、MBT モデルはユーザーストーリーのセットの受け入れ基準の生成に使える。

3 テストケース生成のための選択基準 - 205 分

キーワード

カバレッジアイテム、モデルカバレッジ、テストケースの爆発的増加、テスト選択基準

第 3 章の学習の目的

3.1 MBT テスト選択基準の分類

- MBT-3.1.1 (K2) モデルからのテスト生成に使用するテスト選択基準のさまざまな種類を分類する。
- MBT-3.1.2 (K3) 与えられたコンテキストで与えられたテスト目的を達成するために、MBT モデルからテストケースを生成する。
- MBT-3.1.3 (K2) モデルカバレッジ、データ関連、パターンおよびシナリオベース、プロジェクトベースのテスト選択基準の例を示す。
- MBT-3.1.4 (K2) MBT テスト選択基準が ISTQB Foundation Level のテスト設計技法とどのように関連しているかを認識する。

3.2 テスト選択基準の適用

- MBT-3.2.1 (K1) テストの作業成果物の生成の自動化の程度を想起する。
- MBT-3.2.2 (K3) 与えられたテスト選択基準を与えられた MBT モデルに適用する。
- MBT-3.2.3 (K2) MBT テスト選択基準のよい実践例を説明する。

3.1 MBT のテスト選択基準の分類

同じ MBT モデルから、さまざまなテストスイートを生成する可能性がある。テスト選択基準を使用することで、テスト担当者は、テスト目的に最も適した、意味のあるテストケースのサブセットを選択することができる。

MBT による効率化を実現するためには、テストケースの作成が重要な役割を果たす。

MBT のためのテスト選択基準は、多くの文献で取り上げられている ([Utt07]、[Zan11]参照)。このシラバスでは、テスト選択基準の 6 つの分類を紹介する。それらのうちのいくつかは、特定のアイテムのカバレッジに焦点を当て、他のものは、プロジェクト管理の側面、特定のシナリオのテスト、またはランダムな側面をサポートする。

3.1.1 テストの選択基準

カバレッジベースのテスト選択基準は、MBT モデルからのカバレッジアイテムとテスト生成を関連付ける。カバレッジベースのテスト選択基準は、テストの実行時の対応するカバレッジ基準を達成することを狙いとする。カバレッジアイテムの例は以下の通りである。

- MBT モデルにリンクした要件

この基準では、**MBT** モデルの要素が選択された要件にリンクしていることを要求している。完全な要件カバレッジは、選択された要件のセットを完全にカバーするテストケースのセットに対応している（各要件は、少なくとも 1 つのテストケースでカバーする）。

- **MBT** モデルの要素

モデルカバレッジは、**MBT** モデルの内部構造に基づいている。テスト担当者やテストマネージャーなどの関係者は、カバレッジアイテムを定義し、カバレッジアイテムで求められる量をカバーするテストケースのセットを選択する。選択可能なカバレッジアイテムは、以下のようなモデル要素である。

- 状態遷移図における状態、遷移、判定（[ISTQB_FL_SYL]の「状態遷移テスト」参照）
- ビジネスプロセスモデルにおけるアクティビティとゲートウェイ
- デシジョンテーブルの条件とアクション
- テキストモデルにおけるステートメントと条件

- データ関連のテスト選択基準

これらの基準は、以下のようなテスト技法（[ISQB_FL_SYL]参照）に関するものである。

- 同値分割法
- 境界値分析

また、ペアワイズや、より一般的には n ワイズ組み合わせのテストケース生成などのヒューリスティックも含まれている（詳細は[ISTQB_ATA_SYL]を参照）。

その他のテスト選択基準は以下の通りである。

- ランダム

この選択基準は、モデルの（多かれ少なかれランダムな）ウォークスルーに相当する。ランダムなテストケースの選択では、異なる選択肢はすべて同じ確率である。確率的テストケース選択では、異なる選択肢が異なる確率を持つことを考慮する。

- シナリオベースとパターンベース

テストケースの選択は、あらかじめ定義されたシナリオやパターンに基づいて行われる。シナリオは、明示的に定義されたユースケースや、明示的に定義されたユーザーストーリー（[ISTQB_FL_SYL]の 4.5.2 参照）である。パターンは、部分的に定義されたシナリオであり、複数のテストを生成するために **MBT** モデルに適用することができる。

- プロジェクト主導

プロジェクト主導のテストケースの選択は、テストマネジメントをサポートするため、またはプロジェクトの特定のテスト目的を達成するためにモデルに追加された、プロジェクト関連の追加情報に基づいて行われる。プロジェクト関連の情報とは、リスク、優先順位、必要なテスト機器など、特定のプロジェクトで重要となるあらゆる側面を含む。特定のテスト機器を必要とするすべてのテストケースを選択することは、プロジェクト主導の選択基準を適用することに相当する。

MBT ツールは、上述のテスト選択基準のうち、少なくとも 1 つをサポートしているが、通常はすべてではない。

3.1.2 テストケース選択の実際

実際には、テスト目的に最も適合するテストケースのサブセットを得るために、いくつかのテスト選択基準を組み合わせることを検討すべきである。例えば、要件カバレッジとモデルカバレッジや、シナリオ／パターンとデータ関連のテスト選択基準などがその例である。

目標とする目的とは別に、テスト選択基準の適用は以下にも依存する。

- MBT ツールが提供するメカニズム
- モデルの設計
- 基準に対するテスト担当者の経験

MBT モデルは形式的に正しいが、ツール固有の生成アルゴリズムによりテストケースが爆発的に増える可能性がある。

3.1.3 テスト選択基準の例

いくつかのモデリング言語に関連したテスト選択基準の例がある ([Utt07]-第 4 章参照)。

- アクティビティ図やビジネスプロセスモデルについて
 - アクティビティのカバレッジ (100% = 図中の各アクティビティのカバレッジ)
 - ブランチ／ゲートウェイのカバレッジ (100% = 図中の各ブランチポイントのカバレッジ)
 - パスカバレッジ (100% = 図中のすべてのビジネスシナリオをカバー)、ループあり／なし
- 状態遷移図について
 - 状態および遷移のカバレッジ (100% = 各状態または遷移のカバレッジ)
 - 遷移ペアのカバレッジ (100% = 図中の連続した遷移ペアをそれぞれカバーする)
 - パスカバレッジ (100% = 図中のすべてのパスをカバーし、ループがないことが望ましい)
- デシジョンテーブルについて
 - すべての条件 / 判定
 - すべてのアクション
 - すべてのルール
- モデルの構造部で定義されたデータドメインについて
 - 同値分割カバレッジ (例: 1つのクラスに 1つの代表値を取る)
 - 境界値カバレッジ (例: 数値データの範囲の境界を考慮する)
- テキストモデルについて
 - ステートメントカバレッジ (100% = 各実行可能なステートメントがカバーされている)
 - ブランチカバレッジ (100% = 各ブランチがカバーされている)

3.1.4 テスト選択基準の例

MBT は、状態遷移テスト、同値分割法、境界値分析、デシジョンテーブルテスト、ユースケーステストなど、ISTQB Foundation Level の標準的なテスト設計技法をサポートしている。MBT モデル自体が、それらのテスト設計技法のカバレッジ項目 (例えば、MBT モデルに記述された同値パーティションや境界値) を含んでいる場合がある。あるいは、MBT モデルは、それらの項目を含む他のグラフィックやテキストの表現 (例えば、デシジョンテーブルや追加の図) と組み合わせてもよい。

3.2 テスト選択基準の適用

3.2.1 テスト生成における自動化の度合い

MBT というと、テストケースの自動生成というイメージがあるが、テストケースの生成プロセスは必ずしもツールベースではない。テストケース生成には、以下のような手法がある。

- 手動テスト生成 - MBT モデルから手動でテストケースを導き出し、パスをたどり、対応するテストケースを「手動」で記述する。しかし、この MBT アプローチは、成熟度が低く、1.1 節で述べた自動テストケース生成の効率性や有効性に関する利点を提供しない。
- テストの自動生成 - テストケースなどの出力した作業成果物は、MBT ツールによって自動的に生成され、後処理をすることなくそのまま使用することができる。
- 半自動テスト生成 - ツールを使用しながらも、ツールによって自動化されたステップの間や最後に、特定のテストケースを選択するなどの手動のステップを必要とする中間的なソリューションも存在する。

3.2.2 特定のテスト選択基準の長所と短所

各テストの選択基準には長所と短所があり、テストの目的に応じて選択する。以下は、さまざまなテスト選択基準の長所（プラス）と短所（マイナス）の例である。

- 要件カバレッジ
 - (+) 規制上の要件により必須となることが多い
 - (-) 正確な定義が必要である（例えば、すべての振る舞いをカバーすることで達成とする場合と、1つでもカバーすれば達成とする場合がある）。
- モデルカバレッジ
 - (+) モデルのどこがカバーできないのかを理解し、モデルから見たテストカバレッジの完全性を把握するのに役立つ。
 - (-) 構造的なテストの選択基準によっては、テスト生成時にテストケースの爆発的増加が発生する可能性がある（例：アクティビティ図のすべてのパスカバレッジなど）。
- データ関連のテスト選択基準
 - (+) ドメインの同値分割をカバーすることはしばしば必須である。
 - (-) 大量の組み合わせになる可能性がある。
- ランダム
 - (+) 想定外のテストケースの選択に役立つ。
 - (-) ランダムなアルゴリズムは、ビジネス上の意味のない長いテストを生成することがある。
- シナリオベース/パターンベース
 - (+) ユースケースの選択をサポートする（例：リグレッションテスト用）。
 - (-) シナリオやパターンを定義し、維持するために余分な工数が必要となる。
- プロジェクト主導
 - (+) テストマネジメントに役立つ。
 - (-) 特定の情報をモデルにリンクさせるために余分な工数を必要とする。

3.2.3 MBT のテスト選択のよい実践例

しばしば、1つの基準だけでは、テスト目的を達成するために必要なすべてのテストの側面をカバーするのに十分ではない。そのような場合、テスト担当者は複数の基準を組み合わせなければならない。

基準の組み合わせには2つのアプローチがある。

- 基準の合成 - 生成されたテストスイートには、適用されたすべての基準を満たすテストケースのみが含まれる（共通集合）。
- 基準の加算 - 生成されたテストスイートには、少なくとも1つの基準を満たすすべてのテストケースが含まれる（和集合）。

選択基準は、MBT のモデリングのコンセプトに影響を与える可能性がある。さらに、テスト選択の結果を再現できなければならない。すなわち、同じモデルに対して同じ選択基準を適用した場合、同一のテストケースが得られなければならない。従って、テストの選択活動を計画し、選択とその理由の両方を文書化することが不可欠である。

モデルのカバレッジ基準によっては、要求するカバレッジを満たすパスのセットが複数存在する。MBT モデルに対する小さな変更が全く異なるテスト選択結果となることがあり、この事実は考え方を大きく変えることを必要とする。すなわち、モデルがマスターであり、テストケースはそこから生成する作業成果物である。

テスト選択基準は、MBT においてテストケースの爆発的増加を抑制する方法でもある。単一の基準または複数の基準の組み合わせを使用することで、テスト担当者はテスト生成を精密に調整することで、テストの目的を達成し、テストケースの爆発的増加を抑えることができる。

4 MBT テストの実装と実行 - 120 分

キーワード

オフライン MBT、オンライン MBT、テスト適合レイヤー

第 4 章の学習の目的

4.1 MBT テストの実装と実行の具体的な方法

- MBT-4.1.1 (K2) MBT のコンテキストにおけるハイレベルテストケースとローレベルテストケースの違いを説明する。
- MBT-4.1.2 (K2) MBT のコンテキストにおけるさまざまな種類のテスト実行について説明する。
- MBT-4.1.3 (K3) 要件、テスト対象、テスト目的の変更による MBT モデルの更新とテスト生成を行う。

4.2 MBT におけるテストの適合の活動

- MBT-4.2.1 (K2) MBT におけるテスト実行のために、どのようなテストの適合が必要であることを説明する。

4.1 MBT テストの実装と実行の具体的な方法

テストスイートが生成されたら、次はテストケースを実行する。テストの実行は手動で行う場合と、テストを自動的に実行してテスト結果を記録する機能を備えたテスト実行環境を使用して自動化する場合がある。いずれの場合も MBT モデルとテスト実行の作業成果物の間には関連性がなくてはならない。

4.1.1 MBT コンテキストにおけるハイレベルテストケースとローレベルテストケース

MBT のテスト生成では、ハイレベルテストケースやローレベルテストケースを生成することができる。

- ハイレベルテストケースとは、入力データや期待する結果について、具体的な（実装レベルの）値を持たないテストケースのことである。また、テストステップを詳細に指定しない。
- ローレベルテストケースとは、入力データや期待する結果に具体的な（実装レベルの）値を設定し、テストステップを非常に詳細に記述したテストケースのことである。

MBT モデルは、生成したさまざまな出力した作業成果物やさまざまなステークホルダーに対応するために、異なる抽象度のものを含むことができる。例えば以下ようになる：

- ハイレベルテストケースは、ビジネスアナリストのレビューの対象となることがある。
- ローレベルテストケースは、テスト担当者が直接実行することができる。

MBT で生成されたハイレベルテストケースは、テスト条件、入力データ、期待結果に関する情報を具体的な値を伴わずに提供する。テスト情報は抽象的なレベルで表現する。例えば以下ようになる：

- 詳細で完全に定義されたテストアクションではなく、ハイレベルのテストのアクションの連続でテスト手順を定義する。
- パーティションの具体的な代表値ではなく、同値パーティションを与える。

ハイレベルテストケースから、（手動または自動で）実行可能なローレベルテストケースに移行するには、ハイレベルテストケースを以下の要素で補完する必要がある。

- 完全に定義されたテストのアクション
- 具体的で完全な入力データの値
- 期待する結果の具体的な値

この補完情報は、**MBT** モデルの中に定義しても（例えば、アクション/検証やデータなどのモデルの作業成果物のドキュメントとして）、**MBT** モデルの外に定義しても（例えば、データテーブルを使って抽象的なデータ値と具体的なデータ値をマッピングするなど）どちらでも構わない。

4.1.2 さまざまな種類のテスト実行

MBT で生成されたテストケースは、手動または自動で実行することができる。

手動テストの実行では、**MBT** ツールで生成されたテストケースをテスト担当者が実行する。これらのテストケースは、手動でのテスト実行に使用できる形式で生成されなければならない。また、テストマネジメントツールへテストケースをエクスポートすることも有益である。テスト実行と欠陥マネジメントは、**ISTQB** の基本的なテストプロセスに従って行われる。

自動テストの実行では、テストケースを実行可能な形で生成する必要がある。1つの手段としては、**MBT** で生成されたテストケースを直接、自動テストスクリプトに変換する。他の手段としては、テスト対象のラッパーとしてのテスト適合レイヤーを使う（キーワード駆動テストのアプローチに似ている）。テスト適合レイヤーの利用は、テストケース設計と詳細な実装を区別するという利点がある。詳細な実装は、**SUT** のバージョンやバリエーション、構成に依存することがあるからである。

生成された自動テストスクリプトは、テスト実行ツールによって実行する。**MBT** ツールは生成されたテストケースを、テスト実行ツールのテスト自動化スクリプト言語に変換してもよい。また、**MBT** ツールは、生成された自動テストスクリプトをテストマネジメントツールに格納してもよい。

MBT のコンテキストにおける自動テスト実行には、2つの方法がある。

- オフライン **MBT** - 自動化されたテストスクリプトが最初に生成され（期待結果を含む）、その後実行する。
- オンライン **MBT** ("on-the-fly"とも呼ばれる) - テストの生成と実行が同時に実現する。そのため、各テストステップは、テスト中に前のテストステップを実行した後に生成する。実行結果が、モデルのどのパスを通るかに影響を与える可能性がある。

4.1.3 変更による **MBT** の作業成果物への影響

ソフトウェアプロジェクトでは変更は避けられない。以下のような変更が発生する可能性があり、それらを想定しておく必要がある。

- 影響を与える可能性のある要件、テスト対象またはその環境の変更。
 - **MBT** モデル
 - アクション
 - 条件/期待結果
 - データ

- MBT テストの選択基準
 - 適合レイヤーが存在する場合はその仕様（存在する場合）
- 機能要件ではなく、テスト対象のインターフェースの変更（例えば、機能的な動作に影響を与えないグラフィカルユーザーインターフェースの小さな変更）で、影響を与えるもの。
 - 適合レイヤーの仕様のみ（存在する場合）
- 影響を与える可能性のあるテスト目的またはテスト条件の変更。
 - MBT モデル
 - MBT テストの選択基準
 - 適合レイヤーの仕様が存在する場合はその仕様（存在する場合）

MBT の作業成果物の変更管理は、影響分析、変更オプションの検討、作業成果物への変更の適用、およびレビュー活動を含むプロセスに基づいて行う必要がある。

4.2 MBT におけるテストの適合の活動

手動テストを実行する場合、テストの適合は MBT モデルで作られた抽象化と、テスト対象システムの具体的なインターフェースやテストデータとの間のギャップを埋めるために、生成されたテストの文書化に関連がある。例えば具体的なデータ値は特定の境界値の代表として提供することがある。この適合プロセスにより完全かつ十分に文書化された手動テストスクリプトが提供され、手動テストに直接使用できるようになる。

自動テストを実行する場合、テストの適合とは MBT モデルから生成された作業成果物をテスト実行フレームワークに統合するプロセスである。このプロセスはキーワード駆動テストやデータ駆動テストの実践例をサポートする。

キーワード駆動テストではキーワードが MBT モデルで定義され、生成されたテストケースで使用する。完全に自動化されたテストスクリプトを得るためには、次のような手順と活動が必要となる：

- 1) テストケースを、テスト実行ツールの言語で書かれたスクリプトとしてエクスポートする。これは手動で行うことも、エクスポート機能（MBT ツールが提供している場合もある）を使って自動的に行うこともできる。
- 2) キーワードをテスト実行ツールの言語で実装する。これはテスト自動化を担当するテスト担当者か、テスト対象の開発者のどちらかが行うことができる。

データ駆動テストでは、MBT モデルには抽象的な入力データと期待結果（例：同値分割法に基づく）が記述されている。生成されたテストケースやスクリプトは、これらの抽象的な入力データと期待結果を参照する。

完全に自動化されたテストスクリプトを得るためには、次のような手順と活動が必要となる。

- 1) 自動テストスクリプトに必要な具体的な入力データと期待結果を与える。このデータは表またはスプレッドシートに格納することができる。
- 2) MBT モデルで形式化されたテストデータを、テスト実行ツールやテストハーネスの具体的なテストデータにリンクさせる。

各テストスクリプトは特定の事前条件を想定している。自動テストスクリプトの実行を連鎖的に行うためには、各テストスクリプトの実行前に事前条件が正しく設定されていることを確認する必要がある。これには次のような方法がある。

- いつでも可能とは限らないが、テストスクリプトの中で事後条件を用意し、次のテストの事前条件として使用する。
- 各テストスクリプトの最初に事前条件を設定する。

例えばモデル要素の開発と同時にテストの適合の仕様を作成するなど、MBT モデリング活動中にテストの適合の準備を行う必要がある。

5 MBT アプローチの評価と導入 - 60 分

キーワード
なし

第 5 章の学習の目的

5.1 MBT の導入を評価する

- MBT-5.1.1 (K2) MBT 導入時の ROI (投資収益率) に影響する要因を説明する。
- MBT-5.1.2 (K2) プロジェクトの目的が、MBT アプローチの特徴とどのように関連しているかを説明する。
- MBT-5.1.3 (K1) MBT 活動の進捗状況と結果を測定するために選択されたメトリクスと重要業績評価指標 (KPI) を想起する。

5.2 MBT アプローチ導入の管理とモニタリング

- MBT-5.2.1 (K1) MBT を導入する際のテストマネジメント、変更管理、共同作業のためのよい実践例を想起する。
- MBT-5.2.2 (K1) MBT のコストに影響する要因を想起する。
- MBT-5.2.3 (K1) MBT ツールと構成管理、要件マネジメント、テストマネジメントおよびテスト自動化ツールを統合する必要性を想起する。

5.1 MBT の導入を評価する

MBT アプローチを組織に導入する際には、多くの場合、プロダクト採用時の典型的なプロセスをたどる。

1. 認識 - テストプロセスにおける改善目標を設定し、その目標のすべてまたは一部を解決する可能性のある技術として MBT を識別する。このような改善の機会を特定することは、テストチームの働き方を変える動機付けの重要な要因となる。
2. 興味 - MBT についてさらに学ぶ。
3. 評価 - MBT の主要な原則と既存の MBT アプローチを、与えられたプロジェクトのコンテキストへの適用性の点で分析する。
4. 試用 - 重要業績評価指標 (KPI) を定義し、パイロットプロジェクトを立ち上げて、改善点を測定する。
5. 採用 - 組織のスキルの向上と行動の変化を導き、強化する。

従って、ROI (投資収益率) の見積りや、テストプロジェクトのパフォーマンスに対する MBT の影響を (KPI に基づいて) 評価する機能は、採用サイクルの一部になる。

5.1.1 MBT 導入の ROI 要因

MBT は、導入時と運用時の両方でコストが発生する。MBT の導入を成功させるためには、コストに見合うメリットが得られるよう以下の点に注意しなければならない：

- 長期的なテストプロセス全体を考慮した収支バランス
- テスト設計の品質と開発プロセス全体へのプラスの影響

テストプロセスにおけるコスト、省力化、改善とテストの品質を考慮に入れるべきである。

コスト (5.2.2 項参照) :

- MBT の導入
- テストプロセスで発生する MBT の再利用コスト

省力化:

- 早期の要件の妥当性確認により、後の開発フェーズでのコストのかかるバグ修正を回避することができる：
MBT の特徴であるトップダウンアプローチにより、要件がまだ確定していない、あるいは大まかにしか定義されていないプロジェクトの段階で MBT の設計を行うことができる。このことで、その時点での要件の成熟度に応じた要件の妥当性確認が可能になる。
- テスト設計の作業成果物を再利用し、重複を回避することで、テスト設計の工数を削減する：
MBT モデルでは、すべての可能なパスが 1 つの作業成果物に統合されており、手動テストや自動テストスクリプトとして実装されたテストシーケンスの集合と比較して、重複がない。
- MBT モデルからのテストの作業成果物の自動生成により、テスト実装の工数を削減する：
MBT モデルを開発すると、テストケースを自動的に生成するため、メンテナンス活動の工数が軽減される。
- 早期に欠陥を発見し、市場投入までの時間を短縮する：
これは、MBT のテスト設計が早期に開始できることと、MBT が提供する高度な効率性によって促進する。
- テストマネジメントの支援：
さまざまな MBT のテスト生成戦略とフィルタリング機構を使用することで、テストマネージャーは、テストのゴールに最も適合する「理想的な」テストケースのセットを選択することができる。
- 最適化されたアルゴリズムにより、MBT を使用して定義されたテストケースの数を (MBT を使用しないテストケース設計技法と比較して) 削減できる可能性があり、それにより実行しなければならないテストケースの数を減らすことができる。一様で体系的な生成戦略に基づいて、テストケースを選択し、体系的に数を減らすことができる。

テストの品質におけるメリット:

- テスト設計方法の改善とテスト設計の一貫性
- テスト設計の優先順位付けと品質保証を含む、MBT のアウトプットを使用したテストマネジメント支援
- テスト設計の内容と構成の改善によるトレーサビリティの向上

5.1.2 組織の目的と MBT アプローチの特徴との関係

MBT は、テストの品質、工数の削減、コミュニケーションの点においてテストプロセスの改善ができる。組織は、必要な改善を優先しなければならない。改善の範囲や、組織に与える質的、量的な影響は、MBT の手法の特性に大きく依存する。従って、MBT アプローチは、優先順位付けされた改善の範囲に基づいて定義する必要がある。次の表は、組織の目的の例と、それを達成するために MBT をどのように利用できるかを示している。

組織の目標	MBT 改善フォーカス	達成度
テストの質の向上	<ul style="list-style-type: none"> テストカバレッジを高めるためのテスト設計方法 トレーサビリティ（カバレッジメトリクスと影響分析の能力向上を実現） ヒューマンエラーを避けるためのプロセス自動化 	<ul style="list-style-type: none"> 開発と MBT の活動を別々のモデルで行う（テスト担当者のマインドセットを活用し、独立性を尊重する） テスト計画に基づいて明確に定義された抽象度 ヒューマンエラーを減らすためのテストの作業成果物の生成とテストの実行を含む高度なプロセス自動化 特定のテスト目的に適応可能な、明確に定義されたテスト選択基準
工数の削減	<ul style="list-style-type: none"> プロセス自動化 トレーサビリティ（プロセス自動化を支援 	<ul style="list-style-type: none"> MBT モデルの共有（MBT モデルの再利用） 高度なプロセス自動化とテストの作業成果物の自動生成 最も効率的なテストケースのセットを生成するための、明確に定義されたテスト選択基準
コミュニケーションの向上	<ul style="list-style-type: none"> ステークホルダーの思考の抽象度を反映した、適切で理解しやすく使いやすい MBT アプローチ 	<ul style="list-style-type: none"> 関係するすべてのステークホルダーに適した抽象度（例：ビジネスアナリストにはハイレベルでビジネス指向、テスト担当者には詳細でテスト指向）

組織の目的の組み合わせにより、MBT アプローチの要件が矛盾することがある。これは、異なるテスト目的のために、異なる MBT モデルを開発することで解決できる（2.1.3 項参照）。

5.1.3 メトリクスと重要業績評価指標（KPI）

組織への MBT の導入は、MBT 活動の進捗状況や成果を測るための明確な目標、メトリクス、KPI に基づいて行う必要がある。

監視すべきメトリクスや KPI としては、以下のようなものが考えられる：

- MBT モデルで管理され、トレースされた要件の数、および生成されたテストケースによる要件のカバレッジ（割合）
- MBT モデルのサイズと複雑さ
- 生成されたテストケース/スクリプトの数、生成されたテストケース/スクリプトの 1 人日あたりの数
- MBT のモデリング時に見つかった要件中の欠陥の数

- プロジェクト間での **MBT** モデル要素の再利用性レベル
- プロジェクトのステークホルダー（ビジネス／開発／テスト）による **MBT** モデルの使用レベル
- 従来のテスト設計のアプローチと比較して、生産性の面で効率化が図られた割合（テストが安くなった）
- 従来のテスト設計のアプローチと比較して、欠陥検出の面で効率化が図られた割合（テストがよくなった）

メトリクスや **KPI** を定義して監視することは、**MBT** アプローチを導入する際のプロジェクト管理の最適な実践例の一部である。

5.2 MBT アプローチ導入の管理とモニタリング

5.2.1 MBT を導入する際のよい実践例

MBT ベースのテストプロセスは、その作業成果物やツールとともに、既存の開発プロセス、テストプロセス、ツールチェーンと密接に統合すべきである。これは、アプリケーションライフサイクル管理のツールチェーンに統合する際にも重要であり、例えば、**MBT** を要件のエンジニアリングのプロセスやツールと統合する場合などが挙げられる。

シームレスな統合は、**MBT** の導入を成功させるための重要な要素である。ここでのよい実践例は以下の通り。

- 以下を含む、すべての **MBT** の作業成果物の構成管理：
 - テストベース
 - **MBT** モデルとテスト選択基準
 - テストケースとテストスクリプト
 - 適合レイヤーの仕様とコード

開発プロセスでは、リリース対象やデプロイ対象の作業成果物をバージョン管理する。これは、開発やデプロイメントのプロセスを機能させるために必須である。このようなプロセスで **MBT** の作業成果物を統合し、関連付けるためには、**MBT** の作業成果物の構成管理も必須である。

- **MBT** のテスト生成プロセスと継続的インテグレーションの統合
ビルドを自身でテストできるようにすることは、継続的インテグレーションの重要な価値の 1 つである。コードをビルドすると、継続的インテグレーションサーバーはテストツールを呼び出し、新しい内容を確認する。ユニットテストツールだけでなく、**MBT** ツールもここに統合すべきであり、特に **MBT** が継続的なリグレッションテストに使用される場合はそうである。
- 要件のエンジニアリングやバックログ管理の実践例との統合
要件やバックログ項目（計画されたプロダクトの機能）は、（完成（done）の定義 [ISTQB_FL_SYL] に従って）完了する前にテストされなければならない。例えば、アジャイルソフトウェア開発では、バックログ管理プロセスにテストプロセスを反映させる必要がある。バックログ項目のテストに **MBT** が使用されている場合、対応する **MBT** の作業成果物はバックログ管理ツールで追跡可能でなければならない。すなわち、「どのテスト目的でどのバージョンのモデルからどのテストケースがこのバックログ項目のテストに使用されたか」ということである。

5.2.2MBT のコスト要因

以下の表は、MBT のイニシャルコストとランニングコストを、ISTQB のテストプロセスのテスト活動に関連付けたものである。

(組織とプロジェクトのための) イニシャルコスト :

組織のイニシャルコスト	プロジェクトのイニシャルコスト
<ul style="list-style-type: none"> ● MBT 導入のための既存リソースや知識の確認 ● MBT アプローチとツールの評 ● MBT の手法とプロセスの定義と導入 ● 要件マネジメント、テストマネジメント、継続的インテグレーションとの統合 ● MBT レポートの自動化と統合 ● MBT の作業成果物のアーカイブ手段の確立 ● 一般的な MBT のモデリングとプロセスのガイドラインの作成 ● MBT のコーチングとトレーニング ● MBT ツールのライセンス 	<ul style="list-style-type: none"> ● プロジェクト固有の MBT のモデリングおよびプロセスガイドラインの作成 ● 初期の MBT モデルの作成 ● 既存の資産の変換 (例: テキストのテストケースから MBT モデルへの変換) ● MBT モデルの移植

再利用コスト :

テスト活動とタスク	再利用コスト
<ul style="list-style-type: none"> ● 一般 	<ul style="list-style-type: none"> ● ツールのライセンス (ライセンスモデルによる) および保守費用 ● 新しいチームメンバーのコーチングとトレーニング
<ul style="list-style-type: none"> ● テスト計画 	<ul style="list-style-type: none"> ● MBT のテスト容易性に関するテストベースの分析 ● MBT モデルの開発/拡充/導出の計画
<ul style="list-style-type: none"> ● テスト分析とテスト設計 	<ul style="list-style-type: none"> ● MBT のモデリング ● MBT モデルのリファクタリング ● モデルの検証と妥当性確認
<ul style="list-style-type: none"> ● テスト実装とテスト実行 	<ul style="list-style-type: none"> ● 適切なテスト選択基準の選択 ● 実行可能なテストケースの生成 ● テスト適合レイヤーの開発 (テスト自動実行の場合) ● テストケースの実行 (手動または自動)
<ul style="list-style-type: none"> ● テストのモニタリングとコントロール 	<ul style="list-style-type: none"> ● MBT モデルの品質の継続的なチェック ● 欠陥のトレーサビリティの確保 ● テスト完了基準の文書化 ● MBT の評価と他のテストの評価を共通のレポートへ統合
<ul style="list-style-type: none"> ● テスト完了 	<ul style="list-style-type: none"> ● MBT の作業成果物のアーカイブ ● MBT に関して得られた知識の文書化 ● MBT の作業成果物とプロセスの保守フェーズへの移行

5.2.3 MBT ツールの統合

MBT ツール評価の重要な側面は、MBT ツールと構成管理ツール、要件マネジメントツール、テストマネジメントツール、テスト自動化ツールとの統合に関連している。

MBT を組み込んだテストツールチェーンの典型的な例を下図に示す。

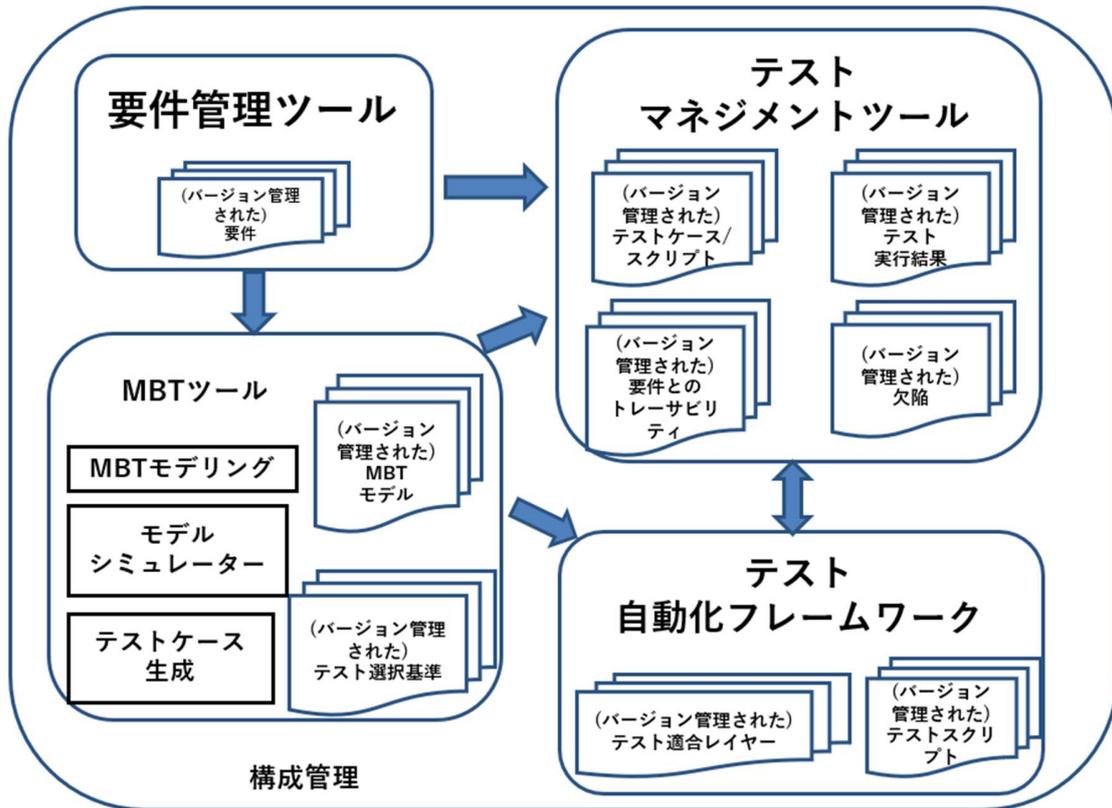


図 1- MBT を組み込んだ典型的なツールチェーン

この典型的なツールチェーンは、MBT ベースの基本的なテストプロセスの主な活動をサポートしており、その活動は以下を含む：

- モデリングツール、モデルシミュレーター、要件とのトレーサビリティを用いた MBT モデルの反復的な開発、レビュー、妥当性確認
- テストケース生成ツールがサポートする MBT モデルにテスト選択基準を適用することによるテスト生成
- テストマネジメントツールやテスト自動化フレームワーク（自動テスト実行の場合）にエクスポート可能な、生成されたテストケースやテストスクリプト、およびトレーサビリティリンク
- MBT モデルや適合レイヤーなどの MBT のテストウェアの構成管理

6 略語について

略称	意味
ATDD	Acceptance Test-Driven Development (受け入れテスト駆動開発)
BPMN	Business Process Modeling Notation (ビジネスプロセスモデリング記法)
CTFL	Certified Tester Foundation Level (ISTQB テスト技術資格制度 Foundation Level)
ISTQB	International Software Testing Qualifications Board (国際ソフトウェアテスト資格認定委員会)
KPI	Key Performance Indicator (重要業績評価指標)
MBT	Model-Based Testing (モデルベースドテスト)
ROI	Return On Investment (投資収益率)
UML	Unified Modeling Language (統一モデリング言語)

7 参考文献

標準

- [ISO25000] ISO/IEC 25000:2005, Software Engineering - Software Product Quality 4 - Requirements and Evaluation (SQuaRE)
- [ETSI_MBT] ETSI ES 202 951, Methods for Testing and Specification (MTS) - Model-Based Testing (MBT) - Requirements for Modeling Notations, Version 1.1.1 (2011-07)
- [ISO29119-8.2] ISO/IEC CD TR 29119-8.2 - Software and systems engineering - Software testing - Part 8: Model-based testing

ISTQB ドキュメント

- [ISTQB_FL_SYL] ISTQB Foundation Level Syllabus, CTFL V4.0
JSTQB 訳注) 日本では「ISTQB テスト技術者資格制度 Foundation Level シラバス 日本語版 Version 2023V4.0.J02」として発行されている
- [ISTQB_ATA_SYL] ISTQB Advanced Level Syllabus Test Analyst, Version 2012
JSTQB 訳注) 日本では「テスト技術者資格制度 Advanced Level シラバス日本語版 テストアナリスト Version 2012.J01」として発行されている
- [ISTQB_GLOSSARY] Standard Glossary of Terms used in Software Testing, <https://glossary.istqb.org/>
- [ISTQB_Act_SYL] ISTQB Certified Tester Acceptance Testing (CT-AcT), V1.0.

本シラバスで参照している文献

- [Utt07] Mark Utting and Bruno Legeard, "Practical Model-Based Testing - A Tools approach," Morgan&Kauffmann, 2007.
- [Zan11] Justyna Zander (Editor), Ina Schieferdecker (Editor) and Pieter J. Mosterman (Editor), "ModelBased Testing for Embedded Systems," CRC Press, 2011.
- [Kra16] Anne Kramer and Bruno Legeard, "Model-Based Testing Essentials: Guide to the ISTQB Certified Model-Based Tester Foundation Level", John Wiley & Sons Inc, 2016
- [Jor17] Paul C. Jorgensen, "The Craft of Model-Based Testing", Auerbach Publishers Inc., 2017

その他の文献

- [Sch12] Ina Schieferdecker: Model-Based Testing. IEEE Software 29(1): 14-18, 2012.
- [Utt12] Mark Utting, Alexander Pretschner, Bruno Legeard, "A Taxonomy of Model-Based Testing Approaches," Softw. Test.Test.Verif.Reliab.22 (5), 297–312, 2012.

8 付録 A - 簡易的なモデリング言語

K3 レベルの学習の目的を達成するために、ここでは 2 つの簡易的なグラフィカルモデリング言語について説明する。

- 1 つ目は UML アクティビティ図のサブセットである。
- 2 つ目は UML 状態遷移図のサブセットである。

次の 2 つの節では、MBT モデリングを実践するために使用できる、これらの UML のサブセットについて例を挙げて定義する。

8.1 ワークフローのための簡易的なグラフィカルモデリング言語

このモデリング言語は、ワークフローを表す MBT モデルまたはアクティビティ図を構築するために使用することができる。ここでは、中間的な判断によって制御する一連のアクションのフローをモデル化することが主眼となる。この言語で作成された MBT モデルは、テストの選択基準に基づいてテストケースを生成するために使用することができる（第 3 章参照）。

このグラフィカルモデリング言語は、UML アクティビティ図のサブセットであり、以下の要素で構成する。

- 黒丸はワークフローの開始（初期状態）を表し、黒丸を丸で囲んだものは終了（最終状態）を表す。
- ▭ 角丸長方形はアクションを表す。
- ◇ ひし形は判断とマージを表し、テキストによるラベルを付与してもよい。
- 矢印はフローを表し、テキストや論理式（算術およびブール演算子を含む）などの表現を付与してもよい。
- ┆ バーは並行処理の開始（フォーク）または終了（ジョイン）を表す。処理のシーケンスは、すべての並行処理がジョインに到達した後に継続する。
- 破線は要件の識別子のアクションへの関連を表す。
- ▭ 矩形はサブダイアグラムを表す。

このようなモデリング言語の記述は、非公式で簡易的なものであるが、このシラバスの文脈で MBT モデルを作成するには十分な情報を含んでいる。

図 2 は、このモデリング言語で開発された MBT モデルの抽象的な例で、常にアクション 1 から始まる振る舞いを示している。「xx EQUAL yy」が真の場合、プロセスは終了する。それ以外の場合は、アクション 2 が実行する。アクション 3 とアクション 2 は判断 B の結果が「No」になるまで繰り返し実行する。その場合、アクション 4 とサブダイアグラム 1 の振る舞いが実行する。判断 C の結果が「Exp 1」となった場合、振る舞いは停止する。そうでなければ、アクション 5 とアクション 6 が並行して実行され、振る舞いが停止する。図 2 は、アクション 2 が要件「Req 1」に、アクション 5 が要件「Req 2」に関連していることを示している。

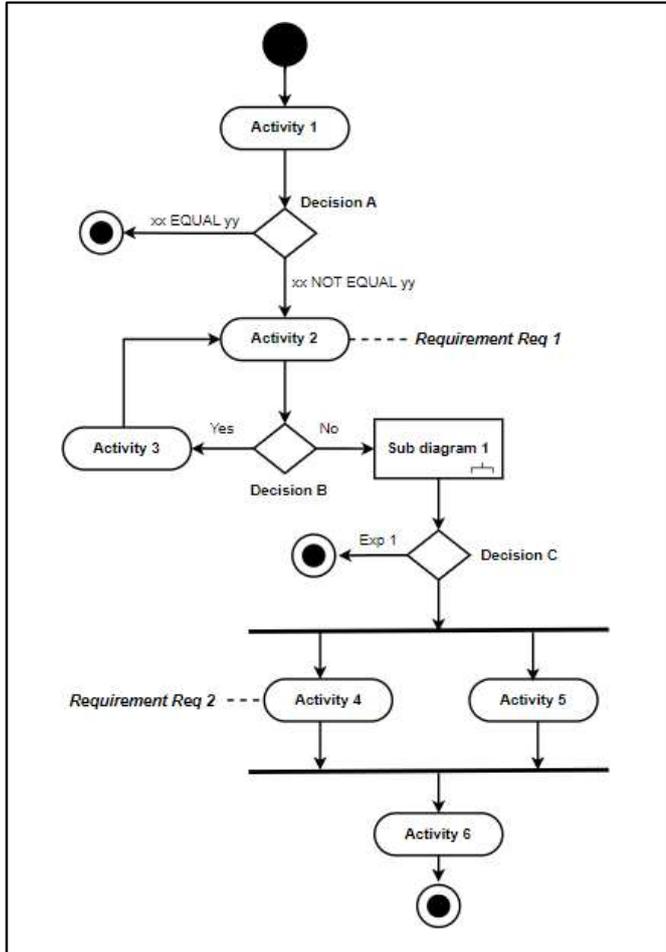


図 2- アクティビティ図の例

9 付録 B - 学習の目的と知識の認知レベル

本シラバスに適用される具体的な学習目標は各章の冒頭に示されている。学習の目的に従ってシラバスのそれぞれのトピックを試験する。

学習の目的は、以下にリストされているように、知識の認知レベルに対応する動詞で記載する。

レベル 1：記憶レベル (K1)

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。

動作動詞：想起する (recall)、認識する (recognize)

例：

- 「テストピラミッドの概念を想起する。」
- 「典型的なテスト目的を識別する。」

レベル 2：理解レベル (K2)

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、テスト概念に関して要約、比較、分類、例の提示を行うことができる。

動作動詞：分類する (classify)、比較する (compare)、区別する (differentiate)、区別する (distinguish)、説明する (explain)、例を挙げる (give examples)、解釈する (interpret)、要約する (summarize)

例：

- テストツールをその意図する用途および具体的なテスト活動に基づいて分類する。
- テストレベルの違いを比較する。(類似点、相違点、または両方を探す)
- テストのデバッグを区別する。(概念の違いを探す)
- プロジェクトリスクとプロダクトリスクを区別する。(2つ以上の概念を区別することができる)
- テストを取り巻くコンテキストがテストプロセスに与える影響を説明する。
- テストが必要な理由について例を挙げる。
- 失敗の傾向を分析し、それに基づいて血管の根本原因を推測する。
- レビュープロセスの活動を要約する。

レベル 3：適用レベル (K3)

精通したタスクに直面した際にプロシジャーに沿って実施、もしくはプロシジャーを正しく選択することができ、それを特定の事例に適用することができる。

動作動詞：適用する (apply) 、実装する (implement) 、準備する (prepare) 、使用する (use)

例：

- テストケースを導出するために境界値分析を使用する。(プロシジャー、技法、プロセス、アルゴリズムなどを参照すべきである)
- 技術的および管理的要件をサポートするためのメトリクス収集方法を実装する。
- モバイルアプリのインストール性テストを準備する。
- トレーサビリティを活用して、テストの目的、テスト戦略、およびテスト計画との整合性と完全性を確保しながら、テストの進捗を監視する。(候補者が技術や手順を使用できるようにすることを目的とした学習の目的で使用される可能性がある。「適用する」に似ている)

参考資料

(学習の目的の認知レベル用)

Anderson, L. W. and Krathwohl, D. R. (eds) (2001) A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives, Allyn & Bacon.

10 付録 C ビジネス成果と学習の目的のトレーサビリティマトリクス

この節では、テスト技術者資格制度モデルベースドテスト担当者シラバス V1.1 のビジネス成果と学習の目的の間のトレーサビリティを列挙する。

ビジネス成果: CT-MBT		BO1	BO2	BO3	BO4	BO5
MBT-BO1	標準的な用語と確立された MBT の概念、プロセス、技法を使用して、モデルベースのテストチームで協力する。	36				
MBT-BO2	テストプロセスにモデルベースのテストを適用し、統合する。		17			
MBT-BO3	確立された技術とモデルベースドテストのベストプラクティスを使用して、効果的に MBT モデルを作成および維持する。			16		
MBT-BO4	リスクとテスト対象の機能の価値を考慮して、MBT モデルからテスト作業成果物を選択、作成、および維持する。				10	
MBT-BO5	組織が品質保証プロセスをより建設的かつ効率的に改善するのを支援する。					8

ビジネス成果: CT-MBT			BO1	BO2	BO3	BO4	BO5
章/節/項	学習の目的	K-レベル					
1	モデルベースドテスト入門						
1.1	MBT の目的とモチベーション						
MBT-1.1.1	MBT に期待される利点を説明する。	K2	X				X
MBT-1.1.2	MBT に対する誤解を招くような期待と落とし穴を説明する。	K2	X	X			X
1.2	テストプロセスにおける MBT 活動と作業成果物						
MBT-1.2.1	テストプロセスに展開されたときの MBT に特有の活動を要約する。	K2	X	X			
MBT-1.2.2	MBT に不可欠な作業成果物（入力と出力）を想起する。	K1	X	X			
1.3	ソフトウェア開発ライフサイクルへの MBT の統合						
MBT-1.3.1	MBT がソフトウェア開発ライフサイクルプロセスにどのように統合されるかを説明する。	K2	X	X			
MBT-1.3.2	MBT が要件のエンジニアリングをどのようにサポートするか説明する。	K2	X	X	X		
2	MBT モデリング						
2.1	MBT のモデリング						
MBT-2.1.1	ワークフローベースのモデリング言語を使用して、テスト対象と事前に定義されたテスト目的のための単純な MBT モデルを開発する。	K3	X		X		
MBT-2.1.2	状態遷移ベースのモデリング言語を使用して、テスト対象と事前に定義されたテスト目的のための単純な MBT モデルを開発する。	K3	X		X		
MBT-2.1.3	システムや環境、テストを記述した MBT モデルを例示する。	K2	X		X		
MBT-2.1.4	MBT モデルがテスト目的にどのように依存するかを例示する。	K2	X	X	X		
2.2	MBT モデルのための言語						
MBT-2.2.1	MBT によく使われるモデリング言語の分類の例を想起する。	K1	X		X		
MBT-2.2.2	テストの目的とモデリング言語の分類の適切な組み合わせを例示する。	K2	X		X		
2.3	MBT モデリング活動のよい実践例						
MBT-2.3.1	MBT モデルの品質特性を想起する。	K1	X		X		
MBT-2.3.2	MBT のモデリング活動における典型的な間違いと落とし穴を説明する。	K2	X		X		
ビジネス成果: CT-MBT			BO1	BO2	BO3	BO4	BO5

MBT-2.3.4	MBT モデリングのガイドラインの必要性を説明する。	K2	X		X		
MBT-2.3.5	(要件フェーズや開発フェーズにおける)既存モデルの再利用が適切な場合とそうでない場合の例を示す。	K2	X	X	X		
MBT-2.3.6	特定の MBT モデリング活動をサポートするツールタイプを想起する。	K1	X		X		
MBT-2.3.7	反復的な MBT モデルの開発、レビュー、検証について要約する。	K2	X	X	X		
3	テストケース生成のための選択基準						
3.1	MBT テスト選択基準の分類						
MBT-3.1.1	モデルからのテスト生成に使用するテスト選択基準のさまざまな種類を分類する。	K2	X			X	
MBT-3.1.2	与えられたコンテキストで与えられたテスト目的を達成するために、MBT モデルからテストケースを生成する。	K3	X			X	
MBT-3.1.3	モデルカバレッジ、データ関連、パターンおよびシナリオベース、プロジェクトベースのテスト選択基準の例を示す。	K2	X			X	
MBT-3.1.4	MBT テスト選択基準が ISTQB Foundation Level のテスト設計技法とどのように関連しているかを認識する。	K2	X			X	
3.2	テスト選択基準の適用						
MBT-3.2.1	テストの作業成果物の生成の自動化の程度を想起する。	K1	X	X		X	
MBT-3.2.2	与えられたテスト選択基準を与えられた MBT モデルに適用する。	K3	X			X	
MBT-3.2.3	MBT テスト選択基準のよい実践例を説明する。	K2	X			X	
4	MBT テストの実装と実行						
4.1	MBT テストの実装と実行の具体的な方法						
MBT-4.1.1	MBT のコンテキストにおけるハイレベルテストケースとローレベルテストケースの違いを説明する。	K2	X	X		X	
MBT-4.1.2	MBT のコンテキストにおけるさまざまな種類のテスト実行について説明する。	K2	X	X			
MBT-4.1.3	要件、テスト対象、テスト目的の変更による MBT モデルの更新とテスト生成を行う。	K3	X		X	X	
ビジネス成果: CT-MBT			BO1	BO2	BO3	BO4	BO5
4.2	MBT におけるテストの適合の活動						

MBT-4.2.1	MBT におけるテスト実行のために、どのようなテストの適合が必要であるかを説明する。	K2	X		X	X	
5	MBT アプローチの評価と導入						
5.1	MBT の導入を評価する						
MBT-5.1.1	MBT 導入時の ROI (投資収益率) に影響する要因を説明する。	K2	X	X			X
MBT-5.1.2	プロジェクトの目的が、MBT アプローチの特徴とどのように関連しているかを説明する。	K2	X	X			X
MBT-5.1.3	MBT 活動の進捗状況と結果を測定するために選択されたメトリクスと重要業績評価指標 (KPI) を想起する。	K1	X	X			X
5.2	MBT アプローチ導入の管理とモニタリング						
MBT-5.2.1	MBT を導入する際のテストマネジメント、変更管理、共同作業のためのよい実践例を想起する。	K1	X	X			X
MBT-5.2.2	MBT のコストに影響する要因を想起する。	K1	X	X			X
MBT-5.2.3.	MBT ツールと構成管理、要件マネジメント、テストマネジメントおよびテスト自動化ツールを統合する必要性を想起する。	K1	X	X			X