

ISTQB® Testeur certifié
Model-Based Testing (CT-MBT)
Syllabus

Version 1.1

International Software Testing Qualifications Board



Avis de Copyright

Avis de Copyright © International Software Testing Qualifications Board (ci-après dénommé ISTQB®)

ISTQB® est une marque déposée de l'International Software Testing Qualifications Board.
Copyright © 2015 Groupe de travail Model-Based Testing : Stephan Christmann (président), Bruno Legeard, Armin Metzger, Natasa Micuda, Thomas Mueller, Stephan Schulz ; 2014-2015.

Copyright © 2024, les auteurs pour la mise à jour 2024 : Bruno Legeard (Responsable du groupe de travail), Abbas Ahmad, Anne Kramer.

Les auteurs transfèrent par la présente le droit d'auteur à l'ISTQB®. Les auteurs (en tant que détenteurs actuels des droits d'auteur) et l'ISTQB® (en tant que futur détenteur des droits d'auteur) ont accepté les conditions d'utilisation suivantes :

- Des extraits de ce document peuvent être copiés, à des fins non commerciales, à condition que la source soit mentionnée. Tout organisme de formation accrédité peut utiliser ce syllabus comme base d'un cours de formation si les auteurs et l'ISTQB® sont reconnus comme la source et les détenteurs des droits d'auteur du syllabus et à condition que toute publicité d'un tel cours de formation ne puisse mentionner le syllabus qu'après avoir reçu l'accréditation officielle du matériel de formation de la part d'un Membre reconnu par l'ISTQB®.
- Tout individu ou groupe d'individus peut utiliser ce syllabus comme base pour des articles et des livres, à condition que les auteurs et l'ISTQB® soient reconnus comme la source et les détenteurs des droits d'auteur du syllabus.
- Toute autre utilisation de ce syllabus est interdite sans l'accord préalable et écrit de l'ISTQB®.
- Tout Membre reconnu par l'ISTQB® peut traduire ce syllabus à condition de reproduire l'Avis de copyright susmentionné dans la version traduite du syllabus.

La traduction française est la propriété du CFTL – Comité Français des Tests Logiciels. Elle a été réalisée par un groupe d'experts en tests logiciels : Eric Riou du Cosquer, Olivier Denoo et Bruno Legeard.

Historique des modifications

Version	Date	Remarques
v1.0	23 octobre 2015	CT-MBT 1.0 – Version majeure générale
v1.1	23 février 2024	CT-MBT 1.1 – Révision 2024
v1.1 FR	16 avril 2024	CT-MBT 1.1 – Version en français

Table des matières

Avis de Copyright	2
Historique des modifications	3
Table des matières.....	4
Remerciement.....	6
0. Introduction	8
0.1 Objectif de ce document	8
0.2 Testeur certifié Model-Based Testing.....	8
0.3 Parcours de carrière pour les testeurs	8
0.4 Objectifs métier.....	9
0.5 Objectifs d'apprentissage examinables et niveaux de connaissance	9
0.6 L'examen de certification du testeur certifié en tests basés sur des modèles	9
0.7 Accréditation	10
0.8 Prise en compte des normes	10
0.9 Rester à jour	10
0.10 Niveau de détail	10
0.11 Organisation du syllabus	11
1 Introduction au Model-Based Testing - 90 minutes (K2)	13
1.1 Objectifs et motivations du MBT.....	13
1.1.1 Motivations principales du MBT	13
1.1.2 Attentes erronées et pièges du MBT.....	14
1.2 Activités et produits d'activités de MBT dans le processus de test	15
1.2.1 Activités spécifiques aux MBT.....	15
1.2.2 Produits d'activités essentiels du MBT (intrants et extrants).....	16
1.3 Intégrer l'approche MBT dans le cycles de vie du développement logiciel	17
1.3.1 MBT dans les cycles de vie du développement logiciel séquentiel et itératif	17
1.3.2 Soutien à l'ingénierie des exigences	18
2 Modélisation MBT - 250 minutes (K3).....	19
2.1 Modélisation MBT	19
2.1.1 Activités de modélisation MBT	20

2.1.2	Focus des modèles MBT	20
2.1.3	Les modèles MBT dépendent des objectifs du test	21
2.2	Langages de modélisation pour le MBT	21
2.2.1	Catégories principales de langages de modélisation pour le MBT	21
2.2.2	Catégories de langages de modélisation adaptées pour différents types de systèmes et d'objectifs de test.....	22
2.3	Bonnes pratiques de la modélisation MBT	23
2.3.1	Caractéristiques de qualité des modèles MBT.....	23
2.3.2	Erreurs types et pièges à considérer lors de la modélisation MBT.....	23
2.3.3	Relier les informations d'exigences et sur le processus de test avec le modèle MBT	24
2.3.4	Guide de modélisation MBT	25
2.3.5	Réutilisation des modèles de conception ou d'exigences des systèmes existants	25
2.3.6	Outils d'aide aux activités de modélisation	26
2.3.7	Modélisation, revue et validation itératives	26
3.	Critères de sélection pour la génération des cas de test – 205 minutes (K3)	27
3.1	Classification des critères de sélection de tests utilisés en Model-Based Testing.....	27
3.1.1	Critères de sélection des tests	27
3.1.2	La sélection des cas de test en pratique.....	29
3.1.3	Exemples de critères de sélection des tests	29
3.1.4	Exemples de critères de sélection des tests	29
3.2	Application des critères de sélection des tests	30
3.2.1	Degré d'automatisation dans la génération de tests	30
3.2.2	Avantages et inconvénients des critères spécifiques de sélection des tests.....	30
3.2.3	Bonnes pratiques de sélection des tests MBT	31
4.	Implémentation et exécution des tests en MBT - 120 minutes (K3)	32
4.1	Aspects spécifiques au MBT de l'implémentation et de l'exécution des tests	32
4.1.1	Cas de test de haut niveau et de bas niveau dans le contexte du MBT	32
4.1.2	Différents types d'exécution des tests.....	33
4.1.3	L'impact de changements sur les produits d'activités du MBT	34
4.2	Activités d'adaptation des tests pour l'exécution en MBT.....	34
5.	Évaluation et déploiement d'une approche MBT - 60 minutes (K2)	36
5.1	Évaluer le déploiement d'une approche MBT.....	37
5.1.1	Les facteurs de ROI de l'introduction du Model-Based Testing.....	37

5.1.2 Les objectifs organisationnels et leur relation avec les caractéristiques de l'approche MBT ...	38
5.1.3 Métriques et indicateurs clés de performance	39
5.2 Gérer et contrôler le déploiement d'une approche MBT	40
5.2.1 Bonnes pratiques pour le déploiement du MBT	40
5.2.2 Facteurs de coût du MBT	40
5.2.3 Intégration de l'outil MBT	42
6 Abréviations	44
7 Références	45
Standards.....	Erreur ! Signet non défini.
Documents ISTQB®	45
Ouvrages	45
Articles	45
8 Annexe A - Langages de modélisation simple	46
8.1 Un langage de modélisation graphique simple de workflow	46
8.2 Un langage de modélisation graphique simple pour les diagrammes de transition d'état	48
9 Annexe B - Objectifs d'apprentissage/Niveau de connaissances cognitives	50
Niveau 1 : Se souvenir (K1)	50
Niveau 2 : Comprendre (K2)	50
Niveau 3 : Appliquer (K3)	51
10 Annexe C - Matrice de traçabilité des résultats opérationnels avec les objectifs de formation	52
11 Annexe D - Notes de mise à jour	59
12 Index.....	Erreur ! Signet non défini.

Remerciement

Ce document a été officiellement livré par l'assemblée générale de l'ISTQB® le 23 février 2024.

Ce document a été produit par une équipe internationale du groupe de travail Model-Based Testing (MBT) au niveau fondation de l'ISTQB® (International Software Testing Qualifications Board).

Le groupe de travail Model-Based Testing remercie l'ensemble des personnes ayant revu le document et les membres des bureaux nationaux de l'ISTQB® pour leurs suggestions et propositions d'amélioration.

Le groupe de travail Model-Based Testing pour la réalisation de ce syllabus était composé des personnes suivantes : Stephan Christmann (Responsable du groupe de travail MBT), Anne Kramer, Bruno Legeard (Co-responsable du contenu), Armin Metzger (Co-responsable du contenu), Natasa Micuda (Responsable du sous-groupe en charge des questions d'examen), Thomas Mueller (Responsable du

groupe de travail ISTQB® Niveau Fondation) et Stephan Schulz (Responsable du sous-groupe en charge des revues).

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

Les auteurs pour la mise à jour version 1.1 : Bruno Legeard (Responsable du groupe de travail), Abbas Ahmad, Anne Kramer.

Questions d'examen : Bob Binder, Renzo Cerquozzi, Debra Friedenberg, Willibald Krenn, Karl Meinke, Natasa Micuda, Michael Mlynarski, Ana Paiva.

Revue des questions d'examen : Eddie Jaffuel, Ingvar Nordstrom, Adam Roman, Lucjan Stapp

Revue du syllabus : 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.

Le groupe de travail MBT remercie les personnes suivantes, membres des bureaux nationaux de l'ISTQB® ou experts de la communauté MBT, pour leurs commentaires du syllabus lors des phases de revue : Patricia Alves, Clive Bates, Graham Bath, Rex Black, Armin Born, Bertrand Cornanguer, Carol Cornelius, Winfried Dulz, Elizabeta Fournere, Debra Friedenberg, 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 and Ester Zabar.

Version 1.1 : Graham Bath, Filipe Carlos, Kaiheng Guo, Matthias Hamburg, Meile Posthuma, Klaus Skaft

0. Introduction

0.1 Objectif de ce document

Ce syllabus forme la base du module spécialisé en Model-Based Testing de la qualification internationale en tests de logiciels. L'International Software Testing Qualifications Board (ISTQB®) et le Comité Français des Tests Logiciels (ci-après appelé CFTL) fournissent ce syllabus comme suit :

1. aux Membres de l'ISTQB®, afin de traduire dans leur langue locale et d'accréditer les fournisseurs de formation. Les Membres peuvent adapter ce syllabus à leurs besoins linguistiques particuliers et ajouter des références pour l'adapter à leurs publications locales.
2. aux organismes de certification, afin de produire les questions d'examen dans leur langue locale en fonction des objectifs d'apprentissage pour ce syllabus.
3. aux organismes de formation, afin de produire le matériel de formation et déterminer les méthodes pédagogiques appropriées.
4. aux candidats à la certification, pour se préparer à l'examen de certification (dans le cadre d'un cours de formation ou indépendamment).
5. à la communauté internationale de l'ingénierie des logiciels et des systèmes, pour faire progresser les pratiques professionnelles en tests de logiciels et de systèmes, et comme base pour des livres articles.

L'ISTQB® et le CFTL peuvent permettre à d'autres entités d'utiliser ce syllabus à d'autres fins, à condition qu'elles demandent et obtiennent une autorisation écrite préalable.

0.2 Testeur certifié Model-Based Testing

La qualification "Testeur certifié Model Based Testing" s'adresse à toute personne impliquée dans les tests de logiciels. Il s'agit notamment de testeurs, d'analystes de test, d'ingénieurs de test, de consultants en test, de test managers, de testeurs d'acceptation des utilisateurs et de développeurs de logiciels. Cette qualification Testeur certifié Model-Based Testing convient également à toute personne souhaitant acquérir une compréhension de base des tests de logiciels, comme les managers de projets, les responsables qualité, les architectes logiciels, les responsables du développement de logiciels, les analystes métier, les directeurs des technologies de l'information et les consultants en management. Les titulaires du certificat de Testeur Certifié Model-Based Testing pourront accéder à des qualifications de test logiciel de niveau supérieur.

0.3 Parcours de carrière pour les testeurs

Le programme de l'ISTQB® apporte un soutien aux professionnels du test à tous les stades de leur carrière, en leur proposant des connaissances à la fois étendues et approfondies. Les personnes qui obtiennent la certification ISTQB® Testeur certifié Model-Based Testing peuvent également être

intéressées par les niveaux principaux avancés (Analyste Test, Analyste Technique Test, et Test Manager) et par la suite le niveau expert (Test Management ou Improving the Test Process).

0.4 Objectifs métier

Cette section énumère les objectifs métiers attendus d'un candidat ayant obtenu la certification ISTQB® Testeur certifié Model-Based Testing.

Un testeur certifié en Model-Based Testing peut :

MBT-BO1	Collaborer au sein d'une équipe de tests en utilisant la terminologie standard et les concepts, processus et techniques MBT établis.
MBT-BO2	Appliquer et intégrer les tests basés sur des modèles dans un processus de test.
MBT-BO3	Créer et maintenir efficacement les modèles MBT en utilisant les techniques établies et les meilleures pratiques des tests basés sur les modèles.
MBT-BO4	Sélectionner, créer et maintenir les produits d'activités de test à partir des modèles MBT en tenant compte du risque et de la valeur des fonctionnalités testées.
MBT-BO5	Soutenir l'organisation dans l'amélioration de son processus d'assurance qualité afin qu'il soit plus constructif et plus efficient.

0.5 Objectifs d'apprentissage examinables et niveaux de connaissance

Les objectifs d'apprentissage soutiennent les objectifs métiers et sont utilisés pour créer les examens de Testeur certifié en Model-Based Testing.

En général, tous les contenus de ce syllabus sont examinables au niveau K1, à l'exception de l'Introduction et des Annexes. Cela signifie que le candidat peut être amené à reconnaître, mémoriser ou rappeler un mot-clé ou un concept mentionné dans l'un des cinq chapitres. Les niveaux des objectifs d'apprentissage spécifiques sont indiqués au début de chaque chapitre et classés comme suit :

- K1 : Se souvenir
- K2 : Comprendre
- K3 : Appliquer

Des détails supplémentaires et des exemples d'objectifs d'apprentissage sont donnés dans l'annexe B.

0.6 L'examen de certification du Testeur certifié Model-Based Testing

L'examen du certificat Testeur certifié Model-Based Testing sera basé sur ce syllabus. Les réponses aux questions de l'examen peuvent nécessiter l'utilisation d'exigences basées sur plus d'une section de ce syllabus. Toutes les sections du syllabus sont examinables, à l'exception de l'Introduction et des Annexes. Les standards et les livres sont inclus comme références, mais leur contenu n'est pas examinable, au-delà de ce qui est résumé dans le syllabus lui-même à partir de ces standards et livres.

Se référer à [ISTQB_EXAM_S&R] pour la structure et les règles de l'examen de l'ISTQB® Testeur certifié Model-Based Testing.

Le critère d'entrée pour passer la certification CT-MBT de l'ISTQB® est que les candidats aient un intérêt pour les tests de logiciels. Cependant, il est fortement recommandé aux candidats :

- D'avoir au moins une expérience minimale soit dans le développement de logiciels, soit dans les tests de logiciels, par exemple six mois d'expérience en tant que testeur de système ou d'acceptation des utilisateurs ou en tant que développeur de logiciels.

- De suivre un cours accrédité selon les standards de l'ISTQB® (par l'un des membres reconnus par l'ISTQB®).

Note sur les exigences d'entrée : La certification ISTQB® Testeur certifié de niveau Fondation [ISTQB_FL_SYL] doit être obtenue avant de passer l'examen de certification ISTQB® Testeur certifié Model-Based Testing.

0.7 Accréditation

Un Membre de l'ISTQB® peut accréditer les organismes de formation dont le matériel de cours suit ce syllabus. Les organismes de formation doivent obtenir les directives d'accréditation auprès du Membre ISTQB® ou de l'organisme qui effectue l'accréditation. Un cours accrédité est reconnu comme étant conforme à ce syllabus et peut comporter un examen de l'ISTQB®.

Les directives d'accréditation pour ce syllabus suivent les directives générales d'accréditation publiées par le groupe de travail sur la gestion des processus et la conformité.

0.8 Prise en compte des normes

Certaines normes sont référencées dans le syllabus de base (par exemple, IEEE, ISO, etc.). L'objectif de ces références est de fournir un cadre (comme dans les références à la norme ISO 25010 concernant les caractéristiques de qualité) ou de fournir une source d'informations supplémentaires si le lecteur le souhaite. Veuillez noter que le syllabus utilise les normes comme référence. Ces documents ne sont pas destinés à l'examen. Reportez-vous à la section 6- Références pour plus d'informations sur les normes.

0.9 Rester à jour

L'industrie du logiciel évolue rapidement. Pour faire face à ces changements et permettre aux parties prenantes d'accéder à des informations pertinentes et actuelles, les groupes de travail de l'ISTQB® ont créé des liens sur le site web www.istqb.org, qui renvoient à des documents complémentaires et à des modifications apportées aux normes. Ces informations ne sont pas examinables dans le cadre de ce syllabus.

0.10 Niveau de détail

Le niveau de détail de ce syllabus permet des cours et des examens cohérents au niveau international. Afin d'atteindre cet objectif, le syllabus se compose de :

- Des objectifs pédagogiques généraux décrivant l'objet du test basé sur des modèles du testeur certifié.
- Une liste de termes que les étudiants doivent être capables de se rappeler
- Des objectifs d'apprentissage pour chaque domaine de connaissance, décrivant le résultat d'apprentissage cognitif à atteindre
- Une description des concepts clés, y compris des références à des sources telles que la littérature acceptée ou les standards.

Le contenu du syllabus n'est pas une description de l'ensemble du domaine de connaissance des tests sur des modèles de logiciels ; il reflète le niveau de détail à couvrir dans les cours de formation Testeur certifié Model-Based Testing. Il se concentre sur les concepts et techniques de test qui peuvent s'appliquer à tous les projets logiciels lors de l'utilisation d'une approche de tests basés sur des modèles.

0.11 Organisation du syllabus

Il y a cinq chapitres dont le contenu peut faire l'objet d'un examen. L'en-tête de chaque chapitre précise la durée du chapitre ; le calendrier n'est pas indiqué au-dessous du niveau du chapitre. Pour les cours de formation agréés, le syllabus exige un minimum de 725 minutes (12 heures et 05 minutes) d'enseignement, réparties sur les cinq chapitres comme suit :

- **Chapitre 1 : Introduction aux tests basés sur des modèles** (90 minutes)
 - Le testeur apprend les objectifs, les motivations et les pièges potentiels du MBT, ainsi que leur intégration dans le cycle de vie du développement logiciel, y compris leur rôle dans l'ingénierie des exigences.
 - Le testeur se penchera également sur les activités et les produits d'activités spécifiques du MBT au sein du processus de test, en soulignant leur importance et leur fonction.
- **Chapitre 2 : Modélisation MBT** (250 minutes)
 - Le testeur apprend à développer des modèles MBT simples pour les objets de test en utilisant des langages de modélisation basés sur les flux et les transitions d'état, tout en comprenant l'influence des objectifs du test sur ces modèles.
 - Le testeur apprend les différentes catégories de langage de modélisation utilisées en MBT, comment elles s'alignent sur les différents objectifs du test, et rappelle les caractéristiques de qualité et les erreurs classiques dans les activités de modélisation en MBT.
 - Le testeur apprend l'importance de l'intégration des exigences et des informations sur les processus dans les modèles MBT, les lignes directrices pour une modélisation efficace, les scénarios appropriés pour la réutilisation des modèles, les types d'outils soutenant la modélisation MBT, et le processus de développement de modèle MBT itératifs, la revue et la validation.
- **Chapitre 3 : Critères de sélection des tests pour la génération de cas de test** (205 minutes)
 - Le testeur apprend à classer les différentes familles de critères de sélection des tests basés sur des modèles, à générer des cas de test pour atteindre des objectifs spécifiques, et à comprendre la relation entre les critères de sélection des tests MBT et les techniques de test du niveau Fondation de l'ISTQB®.

- Le testeur apprend les degrés d'automatisation de la génération des produits tests, comment appliquer des critères de sélection des tests spécifiques à un modèle MBT, et les bonnes pratiques dans la sélection de ces critères.
- **Chapitre 4 : Implémentation et exécution des tests en MBT (120 minutes)**
 - Le testeur apprend les spécificités de l'implémentation et de l'exécution des tests MBT, y compris la distinction entre les cas de test de haut niveau et de bas niveau, les différents types d'exécution des tests, et comment mettre à jour les modèles MBT et la génération des tests en réactivité avec les changements d'exigences, d'objets de test ou d'objectifs.
 - Le testeur apprend à connaître les types d'adaptabilité nécessaires à l'exécution du test dans le contexte des tests basés sur des modèles.
- **Chapitre 5 : Évaluer et déploiement d'une approche MBT (60 minutes)**
 - Le testeur apprend à évaluer le déploiement d'une approche MBT en décrivant les facteurs de retour sur investissement, en établissant un lien entre les objectifs du projet et les caractéristiques MBT, et en rappelant les métriques et les indicateurs clés de performance pour les activités MBT.
 - Le testeur apprend à gérer et à suivre le déploiement d'une approche MBT, y compris les bonnes pratiques pour la gestion des tests et des changements, la compréhension des facteurs de coût, et l'importance de l'intégration des outils MBT avec la gestion de la configuration, des exigences et des tests, ainsi qu'avec les outils d'automatisation des tests.

1 Introduction au Model-Based Testing - 90 minutes (K2)

Mots-clés : modèle MBT, Model-Based Testing (MBT)

Objectifs d'apprentissage pour le chapitre 1

1.1 Objectifs et motivations du MBT

MBT-1.1.1 (K2) Décrire les bénéfices attendus du MBT.

MBT-1.1.2 (K2) Décrire les attentes erronées et les pièges du MBT.

1.2 Activités et produits d'activités du MBT dans le processus de test

MBT-1.2.1 (K2) Synthétiser les activités spécifiques du MBT déployées dans un processus de test.

MBT-1.2.2 (K1) Rappeler les produits d'activités essentiels en MBT (intrants et extrants).

1.3 Intégrer l'approche MBT dans le cycle de vie du développement logiciel

MBT-1.3.1 (K2) Expliquer comment le MBT s'intègre dans les différents types de cycles de développement logiciel.

MBT-1.3.2 (K2) Expliquer comment le MBT facilite la démarche d'ingénierie des exigences.

1.1 Objectifs et motivations du MBT

Le Model-Based Testing (MBT) est une approche de test avancée qui utilise des modèles pour tester. Cette approche étend et soutient les techniques de test classiques telles que la partition d'équivalence, l'analyse des valeurs limites, les tests par tableaux de décisions et les tests de transition d'états. L'idée de base est d'améliorer la qualité et l'efficacité des activités de conception des tests et d'implémentation des tests en :

- Concevant un modèle MBT complet, généralement à l'aide d'outils, basé sur les objectifs de test du projet.
- Fournir un modèle MBT en tant que spécification de conception des tests. Ce modèle comprend un degré élevé d'informations formelles et détaillées qui sont suffisantes pour générer automatiquement les cas de test directement à partir du modèle MBT.

Le MBT et leurs produits d'activités sont étroitement intégrés aux processus de l'organisation ainsi qu'aux méthodes, aux environnements techniques, aux outils et à tout processus spécifique du cycle de vie.

1.1.1 Motivations principales du MBT

Deux aspects principaux des activités de test expliquent la motivation du Model-Based Testing et décrivent comment le MBT contribue à l'amélioration de la qualité des tests :

- L'amélioration de l'efficacité
 - La modélisation est un processus qui favorise une communication étroite avec les parties prenantes.

- Une meilleure communication permet de créer une perception et une compréhension communes des exigences dans un domaine donné et de détecter les malentendus potentiels.
- Dans le cas des modèles graphiques MBT, les parties prenantes du projet (par exemple, les Analyste Métier) peuvent être impliquées plus facilement.
- La modélisation soutient l'amélioration continue des compétences des testeurs dans un domaine donné.
- Le niveau d'abstraction du modèle MBT permet d'identifier plus facilement les parties d'un système où les tests auront plus de chances d'identifier des défauts ou des anomalies.
- Il est possible de générer et d'analyser des cas de test avant qu'un système réel n'existe.
- L'amélioration de l'efficacité
 - La modélisation précoce ainsi que la vérification et la validation du modèle permettent d'éviter les défauts dès les premières étapes du processus de développement. Les modèles MBT peuvent être partagés avec les parties prenantes du projet (par exemple, les Analystes Métier, les développeurs) afin de vérifier les exigences et d'identifier les lacunes au sein de ces exigences. De cette manière, le MBT soutient le principe des "tests précoces".
 - Les produits d'activités du MBT issus de projets antérieurs peuvent être (ré)utilisés pour le développement ultérieur de tests.
 - Le MBT soutient l'automatisation (par exemple, pour la génération des produits d'activités du test) et contribue à réduire les défauts qui peuvent être introduits lorsque ces produits d'activités sont créés et maintenus manuellement.
 - Différentes suites de tests peuvent être générées à partir du même modèle (par exemple, avec différents critères de sélection des tests), ce qui favorise une adaptabilité efficace aux changements de la base de test ou des objectifs du test.
 - Le MBT peut être utilisé pour différents objectifs de test et pour couvrir différents niveaux de test et types de test pendant les tests.
 - Le MBT peut contribuer à réduire les coûts de maintenance lorsque les exigences changent, car le modèle MBT constitue un point de maintenance unique.
 - Lorsqu'elle est soutenue par des outils, l'approche des tests basés sur des modèles permet d'étendre la couverture des tests grâce aux capacités de génération automatique de tests fournies par les outils MBT. Cette génération automatique de cas de test raccourcit également le cycle de test, réduisant ainsi le goulet d'étranglement des tests.

1.1.2 Attentes erronées et pièges du MBT

Les équipes de test qui vont utiliser le MBT doivent avoir des attentes réalistes quant aux avantages et aux limites de cette approche de test. Les attentes trompeuses, les malentendus et les pièges typiques sont les suivants :

- Penser que le MBT va résoudre tous les problèmes
Le MBT ne remplace pas les techniques de test traditionnelles (voir [ISTQB_FL_SYL]), mais les soutient d'une manière qui permet aux testeurs d'améliorer leur compréhension du domaine et d'effectuer des tests de manière plus efficace et efficiente. Une équipe de test qui ignore et/ou utilise mal ces techniques de test ne résoudra pas ce problème en introduisant le MBT.

- Croire que le MBT est uniquement une question d'outillage
Un soutien approprié des outils est essentiel au succès du MBT, mais l'acquisition de l'outil ne doit pas être la première étape. Au lieu de cela, la décision d'introduire le MBT devrait être basée sur la définition d'objectifs mesurables concernant l'amélioration des tests. Le MBT a un impact sur l'ensemble du processus de test (section 1.2). Par conséquent, l'introduction du MBT nécessite un soutien fort du management pour conduire les changements de processus et d'outils dans l'organisation.
- Croire que les modèles MBT sont forcément corrects
Comme dans la conception manuelle des tests, le testeur peut introduire des défauts sur les tests lors de la création du modèle MBT. Les modèles MBT nécessitent une vérification et une validation approfondies en appliquant des revues, une analyse statique du modèle, une simulation du modèle, etc. Toute modification du modèle MBT se propage à tous les produits d'activités générés liés à l'élément de modèle modifié, de sorte que chaque modification doit être soigneusement vérifiée avant d'être implémentée.
- Ignorer les problèmes d'explosion du nombre de cas de test
La mise en œuvre du MBT améliore les méthodes de conception des tests et leur couverture. Dans le cas de génération de tests purement combinatoire, cela peut amener à une explosion du nombre de cas de test. Ce problème peut être évité par la mise en œuvre de stratégies et d'algorithmes de génération de tests adaptés et en appliquant des mécanismes de sélection intelligents lors de la génération.

1.2 Activités et produits d'activités de MBT dans le processus de test

1.2.1 Activités spécifiques aux MBT

Lors du déploiement d'une approche Model-Based Testing, le processus de test est adapté avec les activités spécifiques du MBT, telles que :

- Les activités de modélisation MBT (activités liées à l'administration du modèle MBT, au développement et à l'intégration de l'approche de modélisation, à la définition des directives de modélisation, au développement de la structure du modèle MBT, au développement des éléments du modèle, par exemple les diagrammes spécifiques du modèle MBT, activités liées à l'outillage,).
- La génération de testware basé sur l'approche MBT et les critères de sélection, par exemple, la génération de cas de tests.

Au minimum, le Model-Based Testing modifie les phases d'analyse et de conception des tests. [ISTQB_FL_SYL]. En fonction des objectifs du test, le MBT peut également être utilisé avec un périmètre plus large concernant toutes les parties du processus de test. Les activités suivantes, spécifiques aux MBT, doivent être prises en compte :

- La planification des tests peut inclure l'implémentation des activités spécifiques aux MBT (outillage MBT, approche de la conception des tests, métriques spécifiques, produits d'activités MBT en tant que partie de la planification du projet, etc.)
- L'analyse et la conception des tests peuvent inclure des activités de modélisation MBT, le choix et l'application de critères de sélection des tests et de métriques avancées de couverture des

tests. Dans une approche shift-left, l'analyse de test et la conception des tests fusionnent en une seule activité.

- L'implémentation et l'exécution des tests peuvent inclure la génération et l'adaptation des tests à partir du modèle MBT.
- Le suivi et le contrôle des tests peuvent inclure l'utilisation de métriques de couverture avancées (basées sur des informations structurelles et/ou des modèles explicites) et l'analyse d'impact basée sur les modèles.
- Les activités de clôture des tests peuvent inclure l'établissement de bibliothèques de modèles en vue de leur réutilisation dans des projets futurs.

L'automatisation des tests et la génération de produits d'activités constituent un point central de l'impact du MBT sur le processus de test. Le MBT facilite le shift-left des activités de conception des tests. Cela peut servir de méthode de vérification précoce des exigences et favoriser une meilleure communication, notamment grâce à l'utilisation de modèles graphiques. Les tests basés sur des modèles peuvent compléter d'autres techniques de test (cf. [ISTQB_FL_SYL] - Chapitre 4), comme les tests exploratoires.

Bien que les activités du processus de test utilisant le MBT soient similaires à celles réalisées dans le processus de test traditionnel, la réalisation de ces activités peut changer de manière significative. Comme décrit dans la section 1.1, le MBT :

- A un impact sur la qualité, l'effort, la communication et les parties prenantes impliquées dans le processus de test ;
- Déplace les activités de conception des tests vers des étapes plus précoces du processus de test.

Pour garantir l'adoption du MBT, il est important de s'assurer de l'acceptation de la méthode et des changements connexes au sein des équipes qui travailleront avec le MBT.

1.2.2 Produits d'activités essentiels du MBT (intrants et extrants)

Les modèles MBT peuvent être obtenus soit en développant des modèles MBT spécifiques, soit en réutilisant des modèles développés pour la conception des systèmes (voir le point 2.3.5). Les modèles MBT soutiennent différents niveaux d'abstraction et des informations de test spécifiques en fonction du concept de modélisation appliqué. Les produits d'activités obtenus à partir des modèles MBT reflètent ces différents niveaux d'abstraction.

Selon le niveau d'information et d'abstraction, le MBT est intégré dans le processus de test avec différents produits d'activités d'entrée et de sortie.

Produits d'activités en entrée de l'approche MBT :

- Stratégie de test.
- Bases de test, y compris les exigences et autres cibles de test, les conditions de test, les informations orales et la conception ou les modèles existants.
- Rapports d'incidents et de défauts, les logs des tests et les journaux d'exécution des tests des activités d'exécution des tests précédentes.
- Guides méthodologiques relatives aux méthodes et aux processus, et les documents relatifs aux outils.

Les produits d'activités de sortie pourraient inclure différents types de testware, tels que :

- Des modèles MBT.
- Parties du plan de test (Feature à tester, environnement de test, ...), calendrier des tests, métriques de test.
- Scénarios de test, suites de tests, ordonnancement d'exécution des tests, spécifications de conception des tests.
- Cas de test, procédures de test, données de test, scripts de test, couche d'adaptation des tests (spécifications et code).
- Matrice de traçabilité bidirectionnelle entre les tests générés et les bases de test, en particulier les exigences et les rapports de défaut.

1.3 Intégrer l'approche MBT dans les cycles de vie du développement logiciel

1.3.1 MBT dans les cycles de vie du développement logiciel séquentiel et itératif

Deux grandes catégories de modèles de cycle de vie du développement logiciel sont considérées dans cette section : séquentiel, comme le modèle en V, et itératif-incrémental, comme le développement logiciel Agile. Dans les deux cas, l'approche de MBT doit avant tout être adaptée aux objectifs de test du projet de telle façon que les bénéfices du MBT soient obtenus en adéquation avec ces objectifs.

Les organisations implémentent les cycles de vie du développement logiciel de nombreuses manières différentes. Le processus de test utilisant le MBT doit être adapté en conséquence. Par exemple, le MBT peut se concentrer sur les tests d'acceptation dans un projet, et sur les tests automatisés au niveau système dans un autre, en fonction des objectifs de test du projet.

Les éléments suivants sont communs aux cycles de vie du développement logiciel séquentiel et itératif :

Niveaux de test :

- Le MBT est principalement utilisé sur les niveaux de test les plus élevés (système, intégration du système et acceptation), en raison de sa capacité à abstraire la complexité des exigences fonctionnelles et du comportement attendu.
- Le MBT est moins couramment utilisée dans les tests aux niveaux composants et d'intégration des composants, mais certaines approches sont basées sur des techniques d'annotation du code.

Types de tests :

- Le MBT est principalement utilisé pour les tests fonctionnels avec des modèles MBT représentant le comportement attendu du système et/ou de son environnement.
- Des modèles MBT enrichis ou dédiés peuvent être utilisés pour les tests non fonctionnels (sécurité, charge/stress, fiabilité).

Les testeurs sont généralement en charge des modèles MBT. Mais, dans le cadre du développement logiciel Agile, il peut s'agir d'une responsabilité partagée au sein de l'équipe. Les développeurs peuvent être intéressés par les modèles à des fins d'automatisation des tests, en particulier lorsque les scripts

automatisés générés à partir du modèle MBT seront utilisés pour l'intégration continue afin de fournir un retour d'information précoce et fréquent.

Les activités spécifiques pour le MBT dans un cycle de vie du développement logiciel séquentiel comprennent :

- Commencer la modélisation MBT le plus tôt possible dans le projet :
 - Pour stimuler la communication entre les parties prenantes.
 - Pour permettre une détection précoce des exigences peu claires, incomplètes ou incohérentes.
- Adapter les activités et les rôles de planification des tests pour inclure :
 - De nouveaux éléments de testware (par exemple, les modèle MBT, les critères de sélection des tests).
 - Des rapports sur l'état d'avancement des activités du MBT.

Les adaptations principales du MBT dans un contexte de développement en mode Agile sont les suivantes :

- Les modèles MBT sont développés de manière itérative et incrémentale et les tests sont générés en fonction du contenu de l'itération.
- Les User Stories font partie de la base de test. Chaque User Story couverte est liée au sein du modèle pour gérer automatiquement la traçabilité bidirectionnelle entre les User Stories et les tests.
- Le MBT peut être utilisé pour piloter le développement de logiciels dans le cadre d'un développement piloté par les tests d'acceptation (ATDD) - voir [ISTQB_FL_SYL] section 2.1.3.
- Dans le contexte des pratiques d'équipe intégrée de l'agilité, les testeurs utilisant le MBT font partie intégrante de l'équipe agile avec les développeurs et les représentants du Métier - voir [ISTQB_FL_SYL] section 1.5.2.

1.3.2 Soutien à l'ingénierie des exigences

Le MBT facilite l'ingénierie des exigences de la manière suivante :

- Le MBT facilite la communication entre le Métier, le développement et les tests autour du comportement attendu du logiciel à développer en fournissant des modèles MBT graphiques tels que des modèles de processus métier et/ou des diagrammes de transition d'état. Ces modèles aident toutes les parties prenantes à discuter et à valider les détails du comportement attendu.
- Le MBT aide à clarifier et améliorer la qualité des exigences et/ou des User Stories en partageant des modèles MBT complets ou partiels avec les représentants du Métier et/ou les développeurs afin de produire une compréhension commune du comportement attendu du logiciel. Les tests générés à partir des modèles MBT comprennent des scénarios d'utilisation possibles qui peuvent être révisés par les représentants des métiers et/ou les développeurs.
- Le MBT facilite une validation précoce des exigences, même à des stades où elles sont encore susceptibles d'être modifiées.

L'ingénierie des exigences et le MBT fonctionnent ensemble. Les exigences et les risques qui y sont associés constituent les données d'entrée des activités de modélisation et de génération de tests en MBT, et les activités de MBT peuvent automatiser la création et la maintenance de liens de traçabilité bidirectionnels entre les exigences et les tests.

2 Modélisation MBT - 250 minutes (K3)

Mots-clés : modèle MBT, modèle de test

Objectifs d'apprentissage pour le chapitre 2 :

2.1 Modélisation MBT

MBT-2.1.1 (K3) Développer un modèle MBT simple pour un objet de test et des objectifs de test prédéfinis en utilisant un langage de modélisation de workflow (voir section 8.1 - "simple" signifie moins de 15 éléments de modélisation).

MBT-2.1.2 (K3) Développer un modèle MBT simple pour un objet de test et des objectifs de test prédéfinis en utilisant un langage de modélisation états-transitions (voir section 8.2 - "simple" signifie moins de 15 éléments de modélisation).

MBT-2.1.3 (K2) Fournir des exemples de modèle MBT décrivant le système, l'environnement ou les tests.

MBT-2.1.4 (K2) Donner des exemples de la manière dont un modèle MBT dépend des objectifs du test.

2.2 Langages de modélisation pour le MBT

MBT-2.2.1 (K1) Donner des exemples de catégories de langage de modélisation couramment utilisées pour le MBT.

MBT-2.2.2 (K2) Donner des exemples d'adéquation entre les objectifs des tests et les catégories de langage de modélisation pour différents systèmes et objectifs de test du projet.

2.3 Bonnes pratiques de la modélisation MBT

MBT-2.3.1 (K1) Rappeler les caractéristiques de qualité des modèles MBT.

MBT-2.3.2 (K2) Décrire les erreurs et les pièges classiques lors des activités de modélisation MBT.

MBT-2.3.3 (K2) Expliquer les avantages de lier les exigences et des informations relatives au processus de test au sein du modèle MBT.

MBT-2.3.4 (K2) Expliquer la nécessité d'un guide de bonnes pratiques de modélisation MBT.

MBT-2.3.5 (K2) Donner des exemples où la réutilisation de modèles existants (de la phase des exigences ou de la phase de développement) est, ou n'est pas, appropriée.

MBT-2.3.6 (K1) Rappeler les types d'outils adaptés aux différentes activités de la modélisation MBT.

MBT-2.3.7 (K2) Résumer le développement, la revue et la validation itératifs des modèles MBT.

2.1 Modélisation MBT

Dans le contexte des tests, les modèles MBT sont excellents pour exprimer ce qui doit être testé, pour communiquer entre les parties prenantes et pour résumer toutes les informations pertinentes pour la conception des tests. Ces modèles sont également utilisables pour les activités de gestion des tests.

Un modèle MBT est développé dans le but de produire (ou d'identifier) des cas de test. Le modèle MBT doit inclure les informations nécessaires à la génération ultérieure de cas de test, par exemple pour permettre la sélection de cas de test adaptés aux objectifs de test du projet, pour permettre la génération de l'oracle de test et pour réaliser la traçabilité bidirectionnelle entre les exigences et les cas de test générés.

La conception du modèle MBT est une activité essentielle et exigeante en MBT. La qualité du modèle MBT a un impact important sur la qualité du résultat du processus de test basé sur les modèles MBT. Les modèles MBT sont souvent interprétés par des outils et doivent suivre une syntaxe stricte. Les modèles

MBT peuvent aussi respecter différents critères de qualité tels que ceux définis dans le guide de modélisation MBT du projet.

2.1.1 Activités de modélisation MBT

Chaque modèle est exprimé dans un langage de modélisation spécifique. Le langage de modélisation définit les produits d'activités qui comprennent les éléments du modèle et les règles à suivre pour construire des modèles dans ce langage. Les langages de modélisation permettent d'exprimer la structure, le comportement ou les produits d'activités spécifiques à un domaine, entre autres.

Il convient de se poser les questions importantes suivantes concernant la modélisation MBT :

- Quelles caractéristiques de qualité de l'objet du test seront modélisées ?
Le plus souvent, l'accent est mis sur les fonctionnalités, mais les facteurs de performance et d'autres aspects non fonctionnels tels que la sécurité peuvent également faire partie du modèle MBT. Pour que les modèles restent gérables et que le processus de test basé sur le MBT soit efficace, il est recommandé de ne modéliser que les aspects du système ou de son environnement qui sont pertinents pour le test et pour les objectifs de test.
- Quels sont les langages de modélisation appropriés ?
Chaque modèle est exprimé dans un langage de modélisation spécifique. Le langage de modélisation définit les produits d'activités qui composent le modèle et les règles à suivre pour construire des modèles dans ce langage. Il existe une grande variété de langages de modélisation pour représenter la structure, le comportement ou d'autres aspects (par exemple, les données, les flux de travail, les protocoles de communication) de l'objet du test ou de son environnement.
- Quel est le niveau d'abstraction approprié ?
L'abstraction est utile pour maîtriser la complexité et pour centrer le modèle MBT sur les objectifs du test. Cependant, plus le modèle est abstrait, plus l'adaptation des tests produits avec ce modèle pour l'exécution demandera un effort. En outre, le niveau d'abstraction détermine souvent l'audience/les parties prenantes qui sont en mesure de discuter de la conception des tests au niveau d'abstraction donné.

Les réponses à ces questions et l'ensemble d'outils de MBT choisi influenceront les activités de MBT.

Remarque : deux langages de modélisation graphique simplifiés sont présentés à l'annexe A. La formation et la préparation à l'examen doivent couvrir l'utilisation de ces deux langages.

2.1.2 Focus des modèles MBT

En fonction de l'objectif du test, les modèles MBT se concentrent sur différents aspects.

- Les modèles structurels décrivent la structure statique. Il s'agit par exemple de diagrammes de classes pour tester les relations entre les objets de données ou d'arbres de classification pour la modélisation des données.
- Les modèles comportementaux décrivent les interactions dynamiques. Des exemples de modèles comportementaux sont les diagrammes d'activité ou les modèles de processus métier décrivant les activités et les flux de travail, et les diagrammes de transition d'état décrivant les entrées et les sorties d'un système.

La manière dont les modèles MBT sont rédigés dépend fortement de l'objectif du modèle MBT. Trois types de modèles sont considérés dans ce syllabus :

- **Modèle du système**
Un modèle du système décrit le système tel qu'il est censé se comporter. Les cas de test générés à partir de ce type de modèle vérifient si le système est conforme à ce modèle. Les diagrammes de classes pour les systèmes orientés objet ou les diagrammes de transition d'état décrivant les états et les transitions d'état des systèmes réactifs sont des exemples de modèles de système.
- **Modèle d'environnement / Modèle d'utilisation**
Un modèle d'environnement décrit l'environnement du système. Parmi les exemples de modèles d'environnement, on peut citer les modèles de chaîne de Markov décrivant l'utilisation prévue du système.
- **Modèle de test**
Un modèle de test est un modèle de (un ou plusieurs) cas de test. Il comprend généralement le comportement attendu de l'objet de test et son évaluation. Les descriptions (abstraites) des cas de test ou la représentation graphique d'une procédure de test sont des exemples de modèles de test.

Un modèle MBT combine généralement ces trois aspects.

2.1.3 Les modèles MBT dépendent des objectifs du test

Lors de l'élaboration d'un modèle MBT, il est important de tenir compte de l'objectif du test, car il détermine l'orientation du modèle. Le tableau ci-dessous présente des exemples de bonnes adéquations entre les objectifs du test et le modèle MBT adapté à cet objectif.

Objectif du test	Exemple de modèle
Vérifier que le flux métier est implémenté correctement	Un modèle de processus métier décrivant le flux
Vérifier que le système fournit des réponses correctes aux demandes lorsqu'il se trouve dans un état spécifique	Une machine à états UML
Vérifier la disponibilité d'une interface	La description de la structure de l'interface
S'assurer que l'objet testé conviendra à l'utilisation prévue par les utilisateurs	Un modèle d'utilisation décrivant le comportement de l'utilisateur

2.2 Langages de modélisation pour le MBT

2.2.1 Catégories principales de langages de modélisation pour le MBT

Les modèles sont réalisés par l'utilisation de langages de modélisation. Les langages de modélisation sont définis de la manière suivante :

- Par leurs concepts (également connus sous le nom de syntaxe abstraite et souvent décrits textuellement, mais parfois spécifiés dans des méta-modèles).
- Par leur syntaxe (également appelée syntaxe concrète et souvent définie par des règles de grammaire).
- Par leur sémantique (souvent définie par des règles sémantiques statiques et dynamiques).

Les diagrammes sont des représentations de modèles dans un langage de modélisation graphique, comme les diagrammes de classes, les diagrammes de séquence ou les diagrammes de machines à états dans UML. De nombreux langages de modélisation peuvent être utilisés pour le MBT.

Les catégories de langages de modélisation sont les suivantes :

- **Langages de modélisation structurelle**
Ces langages permettent la modélisation des éléments structurels du logiciel tels que les interfaces, les composants et les hiérarchies. Il s'agit par exemple en UML des diagrammes de composants.
- **Langages pour la modélisation de données**
Ces langages permettent la modélisation des types et valeurs des données. Il s'agit par exemple en UML des diagrammes de classes et des spécifications de valeurs.
- **Langages pour la modélisation comportementale**
Ces langages permettent la modélisation des événements, des actions – réactions et interactions du logiciel. Il s'agit par exemple en UML des diagrammes d'activités et des machines à états, et de la notation BPMN pour la modélisation des processus métier.
- **Langages intégrés de modélisation**
Souvent, un langage de modélisation n'est pas limité à un aspect mais peut supporter différents concepts et types de modélisation. Par exemple, UML propose différents diagrammes qui peuvent être combinés pour modéliser différents aspects d'un logiciel.

2.2.2 Catégories de langages de modélisation adaptées pour différents types de systèmes et d'objectifs de test

Le choix du langage de modélisation pour le MBT est étroitement lié aux objectifs initiaux du projet pour lequel le logiciel est développé et doit être testé. Ce choix dépend également des propriétés du système pour lequel le logiciel est conçu. Il en résulte un ensemble de critères relatifs au langage de modélisation. Il peut être nécessaire d'évaluer ces critères, d'en accepter certains et d'en éliminer d'autres, en fonction de leur pertinence par rapport aux attributs propres au système.

Voici trois exemples :

- Des diagrammes de transition d'état pour représenter le comportement attendu d'un objet de test de contrôle/commande.
- Des diagrammes d'activité ou des modèles BPMN pour représenter le flux de travail pour le test de bout en bout d'un système d'information.
- Des tables de décision pour représenter les règles de gestion pour les tests de systèmes.

Voici d'autres exemples d'aspects à considérer dans la relation entre les langages de modélisation et les objectifs du projet :

- Les exigences non fonctionnelles relatives à la certification de logiciels critiques en matière de sûreté ou de sécurité, lorsque la certification du logiciel exige la mise en relation des exigences logicielles avec le code et les cas de test.
- Les exigences non fonctionnelles concernant la documentation des processus à des fins d'audit ou de certification, lorsque les modèles sont dotés d'annotations pour la documentation.

2.3 Bonnes pratiques de la modélisation MBT

2.3.1 Caractéristiques de qualité des modèles MBT

La qualité du modèle influe directement sur le résultat généré. Les caractéristiques de qualité des modèles sont les suivantes :

- **Qualité syntaxique (exactitude)**
Le modèle MBT est cohérent avec les règles relatives à sa description formelle (langage de modélisation, lignes directrices de modélisation) de sorte que la génération de cas de test/données de test puisse produire des produits d'activités sans rencontrer de problèmes dus à une syntaxe incorrecte.
- **Qualité sémantique (validité)**
Le contenu du modèle est correct par rapport à ce qu'il doit décrire ; les produits d'activités dérivés sont "utilisables" (c'est-à-dire que les cas de test peuvent être exécutés sur l'objet de test sans produire d'échecs dus à des cas de test incorrects).
- **Qualité pragmatique (adéquation)**
Le modèle MBT est adapté à l'objectif de test donné et au générateur de test donné, de sorte que les produits dérivés répondent aux attentes.

Les outils peuvent vérifier la syntaxe et, au moins en partie, la sémantique d'un modèle. Les revues vérifient la qualité sémantique et pragmatique. Les simulateurs de modèles complètent ces techniques statiques en fournissant des contrôles dynamiques de la qualité du modèle.

Il est recommandé de développer le modèle MBT de manière incrémentale et de générer (et d'exécuter) les tests dérivés de manière répétée, ce qui permet de vérifier le modèle, les produits d'activités dérivés et l'objet du test à un stade précoce et sur une base régulière.

2.3.2 Erreurs types et pièges à considérer lors de la modélisation MBT

Les nouveaux venus dans le domaine du MBT ont tendance à commettre un certain nombre d'erreurs courantes dans la conception des modèles, notamment :

- Mettre trop ou trop peu de détails dans le modèle MBT (mauvais niveau d'abstraction pour des objectifs de test donnés).
- Essayer de construire un modèle qui couvre tout.

Les modèles MBT doivent se concentrer sur les aspects pertinents pour atteindre les objectifs du test. Il est préférable d'écrire deux modèles qui mettent l'accent sur des aspects particuliers (par exemple, un diagramme de transition d'état pour vérifier l'implémentation du système et un diagramme d'activité pour valider le flux de travail) plutôt qu'un seul modèle couvrant tous les aspects.

2.3.3 Relier les informations d'exigences et sur le processus de test avec le modèle MBT

L'établissement d'une traçabilité entre les exigences, les modèles MBT et les tests générés est une bonne pratique dans les tests basés sur des modèles. En reliant les éléments du modèle aux exigences, les avantages suivants peuvent être obtenus :

- La revue du modèle MBT est facilitée.
- Les tests générés à partir du modèle MBT peuvent être automatiquement liés aux exigences.
- Il devient possible de générer des tests sur la base d'une sélection d'exigences et de décider quels tests exécuter en premier en fonction de la priorité des exigences sélectionnées.
- Cela permet de mesurer la couverture des exigences par les tests générés par le modèle MBT.
- Cela permet à toutes les parties prenantes (par exemple, les testeurs et les gestionnaires de tests) d'analyser l'impact des changements d'exigences et de déterminer l'étendue nécessaire des tests de régression.
- Les générateurs de cas de test peuvent générer automatiquement une documentation de traçabilité (matrice de traçabilité).

Il existe plusieurs choix de mise en œuvre pour lier les exigences aux éléments du modèle. Par exemple, un symbole graphique spécifique ou un mot-clé textuel peut représenter l'exigence dans le modèle.

Le testeur peut ajouter d'autres informations supplémentaires sous la forme de nouveaux éléments de modèle ayant une signification spécifique, ou sous la forme d'attributs à des éléments de modèle existants.

Voici quelques exemples d'informations supplémentaires dans les modèles MBT :

- Contenu détaillé de la procédure de test (séquence d'actions pour l'exécution du test).
- Segments de script de test (appels de fonction ou mots-clés).
- Risques / dangers (liés aux attributs de qualité fonctionnels et non fonctionnels ou à la gestion de projet).
- Priorités (de la fonctionnalité ou des tests).
- Durée (temps d'exécution estimé).
- Équipement de test requis ou règles pour tester les configurations du produit.

Les informations supplémentaires aident à intégrer les tests basés sur des modèles dans l'ensemble du processus de test et soutiennent la gestion des tests de plusieurs manières :

- Les informations sur les risques ou les priorités peuvent être liées au modèle MBT et aux cas de test générés, qui peuvent être utilisés (par exemple, pour établir des priorités dans l'exécution des tests).
- Les autres contraintes et objectifs du projet peuvent être reflétés dans le modèle MBT et contribuer ainsi à adapter la planification des tests.

Lorsque les interfaces (par exemple, l'interface utilisateur graphique) ne sont pas encore stabilisées dans la spécification, le modèle MBT peut englober les exigences fonctionnelles qui sont définies. Si c'est le cas, les scripts fonctionnels pilotés par mots-clés peuvent être générés. Une fois les interfaces spécifiées,

les scripts de test automatisés sont prêts et seule la couche d'adaptation des tests reste à mettre en œuvre.

2.3.4 Guide de modélisation MBT

Un guide de modélisation MBT définit des instructions documentées sur la manière de concevoir, d'écrire et de lire les modèles de MBT. Ces instructions documentées permettent de :

- Favoriser la compréhension et l'examen des modèles MBT par tous les acteurs concernés.
- Favoriser la conformité aux exigences syntaxiques découlant du processus, du domaine, de l'organisation ou de l'outil spécifique.
- Limiter/étendre le champ d'application de la notation choisie (par exemple, définir un sous-ensemble d'UML ou des significations différentes d'éléments).
- Favoriser une syntaxe et une sémantique similaires pour les modèles MBT de différents auteurs.
- Enseigner les bonnes pratiques.
- Favoriser la maintenabilité et la réutilisation des modèles de MBT.

Ne pas définir de guide de modélisation sur un projet MBT augmente les risques par rapports aux projets définissant et utilisant un tel guide, dans la mesure où les testeurs feront plus facilement des erreurs courantes.

2.3.5 Réutilisation des modèles de conception ou d'exigences des systèmes existants

Outre le développement de modèles MBT à partir de documents de base de test "classiques", il est possible de réutiliser des modèles de conception ou d'exigences de systèmes existants (par exemple, des modèles de processus métier ou des diagrammes d'activité) directement dans le cadre d'un modèle MBT, mais cette approche a ses limites. Le degré de réutilisation dépend du modèle donné et des objectifs du test. Par exemple, un diagramme de machine à états issu de la conception du système peut servir pour tester un système de commande et de contrôle, mais un diagramme de classes d'implémentation n'est pas adéquat pour le test du workflow métier.

La réutilisation de modèles de conception peut aussi créer un problème de propagation d'erreurs (aussi appelé problème de source de connaissance unique). Toute erreur dans le modèle de conception du système d'origine se propage au modèle MBT. Les tests ne sont pas aussi indépendants du développement et de la mise en œuvre du système qu'ils devraient l'être. Il est donc recommandé de rédiger des modèles indépendants par des auteurs différents : l'un pour la conception du système et l'autre pour le MBT. Cela permet de séparer les responsabilités et de promouvoir l'indépendance.

S'ils sont réutilisés sans aucune modification, les tests dérivés des modèles de conception du système sont uniquement des tests de vérification (par opposition à la validation). Ils vérifient si la mise en œuvre du système correspond à la spécification décrite par le modèle de conception du système.

À des fins de validation, le testeur peut partir d'un modèle existant (par exemple, issu de l'élicitation des exigences) et l'enrichir d'aspects de test. Une fois transformé en modèle MBT, le modèle initial de conception du système a tendance à évoluer. Le testeur ajoutera des éléments de modélisation (par exemple, des transitions imprévues) pour tester des scénarios inhabituels ou des situations d'erreur. Finalement, le modèle MBT qui en résulte reflète les aspects liés aux tests à réaliser plutôt que la conception initiale du système.

Prenons le cas où l'auteur du modèle existant n'est pas un testeur. Dans ce cas, les modèles issus des phases de développement précédentes ne respectent pas nécessairement les exigences et les lignes directrices relatives à la modélisation MBT. Un modèle réutilisé ne suit pas nécessairement les pratiques recommandées (dépendant de l'outil) et peut conduire par exemple à une explosion des cas de test.

2.3.6 Outils d'aide aux activités de modélisation

Les éditeurs de modèles sont utilisés pour écrire des modèles. Il peut s'agir d'un outil de modélisation spécifique ou de tout autre éditeur d'organigramme (pour les modèles graphiques), ou d'un éditeur de texte / de script (pour les modèles textuels). En principe, il est possible de rédiger des modèles MBT avec un crayon et du papier. Si de simples outils de dessin peuvent suffire à des fins de documentation, ils ne permettent toutefois pas un traitement ultérieur du modèle MBT (par exemple pour la génération automatique de tests).

Un éditeur de modèles MBT peut fournir des éléments de modèle prédéfinis et éventuellement quelques vérifications syntaxiques et sémantiques. Les outils qui exécutent différents parcours dans le modèle, validant ainsi le modèle MBT, sont appelés simulateurs de modèle.

Pour obtenir automatiquement des cas de test et/ou des données de test à partir du modèle MBT, un générateur de cas de test (et/ou de données de test) est nécessaire.

2.3.7 Modélisation, revue et validation itératives

La modélisation d'un comportement plus complexe à travers plusieurs diagrammes peut facilement conduire à des situations où une simple revue du modèle MBT seul peut ne plus suffire. Un testeur doit être sûr que les tests générés à partir du modèle MBT répondront aux attentes. Le développement itératif, les revues de modèles, ainsi que les revues fréquentes des produits de test générés par le testeur, mais aussi par ses pairs et d'autres parties prenantes impliquées dans le cycle de vie du développement logiciel, sont des moyens de gérer cette situation.

L'application de ces bonnes pratiques permet :

- Aux testeurs de valider leur point de vue sur les aspects à tester avec les autres parties prenantes.
- Aux testeurs d'identifier et de corriger les erreurs ainsi que de détecter les parties manquantes du modèle dès le début du développement du modèle MBT (avant l'exécution du test).
- Aux testeurs d'identifier et de communiquer sur les exigences incomplètes ou incohérentes.
- aux responsables des tests de gérer les risques associés au projet.
- A l'équipe de réduire le temps global nécessaire à la réalisation de l'activité de modélisation MBT dans le processus de test.

Le développement itératif de modèles MBT est également lié à la génération itérative de tests. Cela signifie que chaque fois que certains aspects sont ajoutés au modèle MBT, la génération de tests est également mise à jour. Par essence, la génération de tests constitue une simulation du modèle MBT, qui fournit un retour d'information sur les aspects comportementaux à tester.

Dans une approche ATDD visuelle [ISTQB_AcT_SYL], la revue et la validation du modèle font partie des activités de raffinement du backlog. Le modèle MBT peut alors être utilisé pour formuler des critères d'acceptation pour un ensemble de User Stories.

3. Critères de sélection pour la génération des cas de test – 205 minutes (K3)

Mots-clés : couverture de modèle, élément de couverture, critère de sélection des tests, explosion du nombre de cas de test

Objectifs de connaissance pour le chapitre 3

3.1 Classification des critères de sélection de tests utilisés en Model-Based Testing

- MBT-3.1.1 (K2) Classer les différentes familles de critères de sélection des tests utilisés pour la génération de tests à partir de modèles.
- MBT-3.1.2 (K3) Générer des cas de test à partir d'un modèle MBT pour atteindre des objectifs de test dans un contexte donné.
- MBT-3.1.3 (K2) Fournir des exemples de critères de sélection de tests de type couverture de modèle, orienté donnée, à base de scénarios et de patterns, et spécifique au projet.
- MBT-3.1.4 (K2) Montrer comment les critères de sélection de tests du Model-Based Testing sont en rapport avec les techniques de conception de test ISTQB® de niveau Fondation.

3.2 Application des critères de sélection de tests

- FM-3.2.1 (K1) Rappeler les niveaux d'automatisation de la génération de tests.
- FM-3.2.2 (K3) Appliquer des critères de sélection de tests sur un modèle MBT donné.
- FM-3.2.3 (K2) Décrire les bonnes pratiques de la mise en œuvre des critères de sélection de tests en MBT.

3.1 Classification des critères de sélection de tests utilisés en Model-Based Testing

À partir du même modèle MBT, plusieurs suites de tests peuvent être générées. L'utilisation de critères de sélection des tests aide le testeur à sélectionner un sous-ensemble significatif de cas de test qui correspond le mieux aux objectifs de test visés.

Les générateurs de cas de test jouent un rôle essentiel dans l'amélioration de l'efficacité du MBT.

Les critères de sélection des tests pour le MBT ont été largement étudiés dans la littérature (voir [Utt07], [Zan11]). Ce syllabus présente six familles de critères de sélection des tests. Certains d'entre eux se concentrent sur la couverture d'éléments spécifiques ; d'autres visent les aspects de gestion de projet, les tests de scénarios spécifiques ou les aspects aléatoires.

3.1.1 Critères de sélection des tests

Les critères de sélection des tests basés sur la couverture établissent un lien entre la génération des tests et les éléments de couverture du modèle MBT. Les critères de sélection des tests basés sur la couverture visent à atteindre les critères de couverture correspondants lors de l'exécution du test. Les éléments de couverture peuvent être :

- Les exigences liées au modèle MBT
Ce critère requiert que les éléments du modèle MBT soient liés à des exigences identifiées. La

couverture complète des exigences correspond à un ensemble de cas de test qui couvre entièrement un ensemble sélectionné d'exigences (chaque exigence est couverte par au moins un cas de test).

- Des éléments du modèle MBT
La couverture du modèle est basée sur la structure interne du modèle MBT. Le testeur, le responsable des tests ou tout autre rôle impliqué définissent les éléments de couverture pour sélectionner les tests couvrant les éléments de modélisation considérés. Les éléments de couverture possibles peuvent être des éléments de modèle tels que :
 - États, transitions et décisions dans les diagrammes de transition d'état (voir "test de transition d'état" dans [ISTQB_FL_SYL]).
 - Activités et branchements dans les modèles de processus métier.
 - Conditions et actions dans les tables de décision.
 - Déclarations et conditions dans les modèles textuels.
- Critères de sélection des tests liés aux données
Ces critères concernent les techniques de test (voir [ISTQB_FL_SYL]) telles que :
 - Partitions d'équivalence.
 - Analyse des valeurs limites.
 - Ces critères comprennent également des heuristiques telles que la génération de cas de test combinatoire pair-wise ou, plus généralement, la génération de cas de test combinatoire n-wise (pour plus d'informations, voir [ISTQB_ATA_SYL]).

Les autres critères de sélection des tests sont les suivants :

- Les critères fondés sur l'aléatoire
Ce critère de sélection est comparable à un parcours (plus ou moins aléatoire) du modèle. Dans la sélection aléatoire des cas de test, les différentes alternatives ont toutes la même probabilité. La sélection stochastique de cas de test tient compte du fait que les différentes alternatives peuvent avoir des probabilités différentes.
- Les critères fondés sur les scénarios et sur les patterns
La sélection des cas de test est basée sur des scénarios ou des modèles prédéfinis. Un scénario peut être un cas d'utilisation explicitement défini ou une User Story explicitement définie (voir [ISTQB_FL_SYL] section 4.5.2). Un pattern de génération de tests est un scénario partiellement défini, qui peut être appliqué au modèle MBT pour produire plusieurs tests.
- Les critères fondés sur des caractéristiques du projet
Les critères de sélection de tests fondés sur des caractéristiques du projet s'appuient sur des informations additionnelles ajoutées au modèle MBT, soit pour prendre en compte la gestion du projet, soit pour couvrir des objectifs spécifiques de test. Ces informations peuvent être relatives aux risques, aux priorités, aux équipements nécessaires pour les tests, ou tout autre aspect spécifique du projet. Par exemple, la sélection des tests qui utilisent un équipement spécifique correspond à la mise en œuvre de ce type de critères de sélection de tests fondés sur des caractéristiques du projet.

Les outils MBT mettent en œuvre une ou plusieurs de ces familles de critères de couverture de test (mais généralement pas toutes).

3.1.2 La sélection des cas de test en pratique

Dans la pratique, le testeur devrait envisager de combiner plusieurs critères de sélection des tests afin d'obtenir un sous-ensemble de cas de test qui corresponde le mieux aux objectifs de test visés. Des exemples de ces combinaisons sont la couverture des exigences et la couverture du modèle, ou encore les critères de sélection des tests liés aux scénarios et aux données.

Outre les objectifs visés, l'application des critères de sélection des tests dépend également des éléments suivants :

- Les mécanismes fournis par l'outil MBT utilisé.
- La conception du modèle MBT.
- L'expérience du testeur en matière de critères.

Il peut arriver que le modèle MBT soit formellement correct, mais qu'une explosion du nombre de cas de test générés se produise en raison des spécificités des algorithmes de génération de tests mis en œuvre.

3.1.3 Exemples de critères de sélection des tests

Voici quelques exemples de critères de sélection des tests en relation avec certains langages de modélisation (voir [Utt07] - chapitre 4) :

- Sur les diagrammes d'activités et les modèles de processus métier :
 - Couverture des activités (100% = couverture de chacune des activités du diagramme)
 - Couverture des décisions et branchements (100% = couverture de chacune des décisions ou branchements du diagramme)
 - Couverture des chemins (100% = couverture de chacun des scénarios métier du diagramme), avec ou sans boucles
- Sur les diagrammes d'états :
 - Couverture des états et des transitions (100% = couverture de chacun des états ou transitions)
 - Couverture des paires de transitions (100% = couverture de chaque paire consécutive de transitions du diagramme)
 - Couverture des chemins (100% = couverture de l'ensemble des chemins, souvent sans boucles, sur le diagramme)
- Sur les tables de décision :
 - Couverture de chaque condition / décision
 - Couverture de chaque action
- Sur les domaines de données définis dans la partie structurale du modèle MBT :
 - Couverture des partitions d'équivalence (par exemple avec 1 représentant par classe)
 - Couverture des valeurs limites (par exemple en prenant les valeurs limites des intervalles de données numériques)
- Sur les modèles textuels :
 - Couverture des instructions (100% = chaque instruction exécutable est couverte)
 - Couverture des décisions (100% = chaque décision est couverte)

3.1.4 Exemples de critères de sélection des tests

Le MBT prend en charge les techniques de test standard du niveau Fondation de l'ISTQB®, telles que les tests de transition d'état, les partitions d'équivalence, l'analyse des valeurs limites et les tests de table de décision. Le modèle MBT lui-même peut contenir des éléments de couverture de ces techniques de test (par exemple, des partitions d'équivalence ou des valeurs limites décrites dans le modèle MBT). Le

modèle MBT peut également être combiné à d'autres représentations graphiques ou textuelles contenant ces éléments (par exemple, des tables de décision ou des diagrammes supplémentaires).

3.2 Application des critères de sélection des tests

3.2.1 Degré d'automatisation dans la génération de tests

En pratique, le Model-Based Testing met généralement en œuvre la génération automatique de tests à partir du modèle MBT et des critères de sélection de tests. Mais le processus de génération de cas de test n'est pas nécessairement basé sur un outil. Les méthodes suivantes peuvent être utilisées pour la génération de cas de test :

- Génération manuelle de tests - Dérivation manuelle de cas de test à partir du modèle MBT, en suivant les chemins et en écrivant les cas de test correspondants "manuellement". Cependant, cette approche MBT a un faible niveau de maturité et n'offre pas les avantages en termes d'efficacité et d'efficacité de la génération automatisée de cas de test mentionnés dans la section 1.1.
- Génération automatisée de tests - Les produits de sortie tels que les cas de test sont générés automatiquement par un outil de MBT et peuvent être utilisés tels quels sans autre post-traitement.
- Génération semi-automatique de tests - Il existe également des solutions intermédiaires dans lesquelles un outil est utilisé, mais des étapes manuelles sont nécessaires en cours ou à la fin du processus MBT, par exemple pour sélectionner des cas de test spécifiques.

3.2.2 Avantages et inconvénients des critères spécifiques de sélection des tests

Chaque critère de sélection des tests présente des avantages et des inconvénients. Le choix dépend des objectifs de test. Voici des exemples d'avantages (+) et d'inconvénients (-) pour divers critères de sélection des tests :

- Couverture des exigences
 - (+) Souvent obligatoire en raison d'exigences réglementaires sur le processus de développement.
 - (-) Une définition précise de la notion de couverture d'exigences est généralement requise (par exemple, si une exigence est reliée à plusieurs comportements modélisés, la couverture d'exigences doit définir si un comportement ou tous les comportements doivent être couverts).
- Couverture du modèle
 - (+) Utile pour comprendre où exactement un modèle n'a pas pu être couvert et pour avoir une idée de l'exhaustivité de la couverture des tests vis-à-vis des éléments du modèle.
 - (-) Certains critères de sélection des tests structurels peuvent entraîner une explosion des cas de test lors de la génération des tests (par exemple : couverture de tous les chemins dans un diagramme d'activité).
- Critères de sélection des tests liés aux données
 - (+) Souvent obligatoires pour couvrir les partitions d'équivalence des domaines.

- (-) Peuvent être fortement combinatoires.
- Critères fondés sur l'aléatoire
 - (+) Utile pour sélectionner des cas de test inattendus.
 - (-) Les algorithmes aléatoires peuvent générer de longs tests qui n'ont aucune signification métier.
- Critères fondés sur les scénarios et les patterns
 - (+) Facilite la sélection des cas d'utilisation (par exemple, pour les tests de régression).
 - (-) Nécessite un effort supplémentaire pour définir et maintenir les scénarios.
- Critères fondés sur des caractéristiques du projet
 - (+) Utile pour la gestion des tests.
 - (-) Nécessite un effort supplémentaire pour relier des informations spécifiques au modèle.

3.2.3 Bonnes pratiques de sélection des tests MBT

Souvent, un critère n'est pas suffisant à lui tout seul pour couvrir l'ensemble des aspects nécessaires à la satisfaction des objectifs de test définis. Dans ce cas, le testeur peut combiner plusieurs critères de sélection.

Il existe deux approches pour combiner les critères :

- Composition de critères - La suite de tests générée ne contient que les cas de test qui remplissent tous les critères qui ont été appliqués (intersection).
- Ajout de critères - La suite de tests générée contient tous les cas de test satisfaisant chaque critère (union).

Les critères de sélection peuvent influencer la modélisation MBT. De plus, il doit être possible de reproduire une sélection, c'est-à-dire d'obtenir à chaque fois les mêmes tests avec le même modèle et les mêmes critères de sélection de tests. Il est ainsi nécessaire de définir les activités de sélection de tests et de documenter les choix qui procèdent à cette sélection.

Pour certains critères de couverture de test, plusieurs ensembles de chemins peuvent satisfaire au critère choisi. De plus, des petites modifications dans le modèle MBT peuvent modifier en profondeur les résultats de l'application des critères, ce qui impose au testeur une nouvelle façon de penser la conception des tests. Le modèle est l'élément maître à partir duquel sont dérivés les artefacts de sortie du processus MBT.

Les critères de sélection de tests sont aussi un moyen de maîtriser l'explosion du nombre de cas de test générés. En utilisant un critère déterminé ou une combinaison de critères, le testeur peut piloter de façon précise la génération de tests pour satisfaire les objectifs de test et éviter l'explosion du nombre de tests générés.

4. Implémentation et exécution des tests en MBT - 120 minutes (K3)

Mots-clés : MBT hors-ligne, MBT en-ligne, couche d'adaptation des tests

Objectifs de connaissance pour le chapitre 4 :

4.1 Aspects spécifiques au MBT de l'implémentation et de l'exécution des tests

MBT-4.1.1 (K2) Expliquer la différence entre les cas de test de haut niveau et de bas niveau dans le contexte du MBT.

MBT-4.1.2 (K2) Expliquer les différents types d'exécution des tests dans le contexte de le MBT.

MBT-4.1.3 (K3) Effectuer des mises à jour d'un modèle MBT et de la génération de tests suite à des changements d'exigences, d'objet de test ou d'objectifs de test.

4.2 Activités d'adaptation des tests pour l'exécution en MBT

MBT-4.2.1 (K2) Expliquer quel type d'adaptation des tests peut être nécessaire pour l'exécution des tests en MBT.

4.1 Aspects de l'implémentation et de l'exécution des tests spécifiques au MBT

Une fois la suite de tests générée, l'étape suivante consiste à exécuter les cas de test. L'exécution des tests peut être manuelle ou automatisée en utilisant un environnement d'exécution des tests qui permet d'exécuter automatiquement les tests et d'en enregistrer les résultats. Dans les deux cas, il doit y avoir un lien entre le modèle MBT et les produits de l'exécution des tests.

4.1.1 Cas de test de haut niveau et de bas niveau dans le contexte du MBT

La génération de tests MBT peut produire des cas de test de haut niveau ou des cas de test de bas niveau :

- Les cas de test de haut niveau sont des cas de test sans valeurs concrètes (au niveau de l'implémentation) pour les données d'entrée et les résultats attendus. Ils ne spécifient pas non plus les étapes du test dans les moindres détails.
- Les cas de test de bas niveau sont des cas de test avec des valeurs concrètes (au niveau de l'implémentation) pour les données d'entrée et les résultats attendus, et avec des descriptions très détaillées des étapes de test.

Les modèles MBT peuvent contenir différents niveaux d'abstraction en fonction des différents produits d'activités générés et des différentes parties prenantes. Par exemple :

- Les cas de test de haut niveau peuvent être destinés à être examinés par les Analystes Métier
- Les cas de test de bas niveau peuvent être directement exécutés par les testeurs.

Les cas de test de haut niveau générés par MBT fournissent des informations sur les conditions de test, les données d'entrée et les résultats attendus sans valeurs concrètes. Les informations relatives aux tests sont exprimées à un niveau abstrait, par exemple :

- Des procédures de test avec la séquence des actions de haut niveau peuvent être définies plutôt que tous les détails des actions de test.
- Des partitions d'équivalence de données peuvent être définies plutôt que des instances concrètes de ces partitions.

Pour passer des cas de test de haut niveau aux cas de test de bas niveau prêts à être exécutés (manuellement ou automatiquement), il faut compléter les cas de test de haut niveau par les éléments suivants :

- Des actions de test complètement définies.
- Des valeurs concrètes et complètes pour les données d'entrée.
- Des valeurs concrètes pour les résultats attendus.

Ces informations complémentaires peuvent être définies dans le modèle MBT (par exemple, en tant que documentation du modèle MBT tels que les actions/vérifications et les données) ou en dehors du modèle MBT (par exemple, en utilisant une table de données pour mettre en correspondance des valeurs de données abstraites et concrètes).

4.1.2 Différents types d'exécution des tests

Les cas de test générés par MBT peuvent être exécutés manuellement ou automatiquement.

Pour l'exécution manuelle des tests, les testeurs exécutent les cas de test générés par l'outil MBT. Ces cas de test doivent être générés dans un format utilisable pour l'exécution manuelle des tests. Il peut également être utile de pouvoir exporter les cas de test dans un outil de gestion des tests.

Pour l'exécution automatisée des tests, les cas de test doivent être générés sous une forme exécutable. Dans certains cas, il est possible de convertir les cas de test générés par MBT directement en scripts de test automatisés. Dans d'autres cas, une couche d'adaptation des tests est fournie, qui est essentiellement une enveloppe autour de l'objet de test (similaire à l'approche des tests par mots-clés – Keyword-Driven Testing). L'utilisation d'une couche d'adaptation des tests présente l'avantage de séparer la conception des cas de test des détails d'implémentation qui peuvent dépendre de la version, de la variante ou de la configuration exacte du système testé.

Une fois générés, les scripts de test automatisés sont exécutés par un outil d'exécution des tests. Les outils MBT peuvent convertir les cas de test générés dans le langage de script d'automatisation des tests de l'outil d'exécution des tests. L'outil MBT peut également publier les scripts de test automatisés générés dans un outil de gestion des tests.

Il existe deux méthodes d'exécution automatisée des tests dans le contexte du MBT :

- MBT hors-ligne - Les scripts de test automatisés sont d'abord générés (y compris les résultats attendus) et exécutés ensuite.
- MBT en-ligne (également appelée "à la volée") - La génération et l'exécution des tests sont réalisées simultanément. Par conséquent, chaque étape de test est générée après l'exécution de l'étape de test précédente pendant le test. Le résultat de l'exécution peut influencer le chemin emprunté dans le modèle.

4.1.3 L'impact de changements sur les produits d'activités du MBT

Les changements sont inévitables dans un projet logiciel. Les changements suivants peuvent se produire et doivent être anticipés :

- Changement dans les exigences, dans l'objet du test ou dans son environnement qui peut avoir un impact sur :
 - Le modèle MBT.
 - Les actions de test.
 - Les conditions/résultat attendu.
 - Les données de test.
 - Les critères de sélection du test MBT.
 - La couche d'adaptation (si elle existe).
- Changement dans les interfaces de l'objet du test, mais pas dans les exigences fonctionnelles (par exemple, un petit changement dans l'interface graphique de l'utilisateur sans impact sur le comportement fonctionnel), qui a un impact sur :
 - La couche d'adaptation uniquement (si elle existe).
- Modification des objectifs de test ou des conditions du test pouvant avoir un impact :
 - Le modèle MBT.
 - Les critères de sélection des tests MBT.
 - La couche d'adaptation (si elle existe).

La gestion des modifications apportées aux produits d'activités en MBT doit reposer sur un processus comprenant l'analyse d'impact, l'exploration des options de modification, l'application des modifications aux produits d'activités et les activités de revue.

4.2 Activités d'adaptation des tests pour l'exécution en MBT

Dans le cas de l'exécution manuelle des tests, l'adaptation des tests concerne la documentation des tests générés pour combler l'écart entre les abstractions faites dans le modèle MBT et les interfaces concrètes et les données de test du système testé. Par exemple, les valeurs concrètes des données de test peuvent être fournies par des valeurs aux limites des partitions d'équivalence définies dans le modèle. Ce processus d'adaptation permet d'obtenir des scripts de test manuels complets et suffisamment documentés pour pouvoir être directement utilisés pour des tests manuels.

Dans le cas de l'exécution automatisée des tests, l'adaptation des tests est le processus d'intégration des produits d'activités générés à partir d'un modèle MBT dans le cadre d'exécution des tests. Ce processus soutient les pratiques de tests basés sur des mots-clés et/ou de tests basés sur des données (Keyword-Driven Testing / Data-Driven Testing).

En ce qui concerne les tests pilotés par mots-clés, les mots-clés sont définis dans le modèle MBT et utilisés dans les cas de test générés. Afin d'obtenir des scripts de test entièrement automatisés, les étapes et activités suivantes sont nécessaires :

1. Exporter les cas de test sous forme de scripts dans le langage du framework d'automatisation des tests. Cette opération peut être effectuée manuellement ou automatiquement à l'aide d'un exportateur (qui peut être fourni par l'outil MBT).
2. Implémenter les mots-clés dans le langage du framework d'automatisation des tests. Cette opération peut être effectuée par le testeur (automaticien de tests) chargé de l'automatisation des tests ou par le développeur de l'objet de test.

En ce qui concerne les tests pilotés par les données (Data-Driven Testing), le modèle MBT décrit les données d'entrée abstraites et les résultats attendus (par exemple, sur la base d'un partitionnement par classes équivalence). Les cas de test ou les scripts générésinstancient ces données d'entrée abstraites et aux résultats attendus par des valeurs concrètes.

Pour obtenir des scripts de test entièrement automatisés, les étapes et activités suivantes sont nécessaires :

1. Fournir les données d'entrée concrètes et les résultats attendus requis par les scripts de test automatisés. Ces données peuvent être stockées dans un tableau ou une feuille de calcul.
2. Lier les données de test formalisées dans le modèle MBT aux données de test concrètes dans l'outil d'exécution des tests ou dans le harnais de test.

Chaque script de test suppose des conditions préalables initiales spécifiques. Afin de pouvoir enchaîner l'exécution des scripts de test automatisés, il est nécessaire de s'assurer que les conditions préalables sont correctement définies avant l'exécution de chaque script de test. Cela peut se faire de la manière suivante :

- En fournissant des post-conditions à l'intérieur des scripts de test (ce qui n'est pas toujours possible) qui seront utilisées comme conditions préalables pour le script suivant.
- En définissant la condition préalable au début de chaque script de test.

L'adaptation des tests doit être préparée pendant les activités de modélisation en MBT, par exemple en développant une spécification de l'adaptation des tests en même temps que les éléments du modèle.

5. Évaluation et déploiement d'une approche MBT - 60 minutes (K2)

Mots-clés : Aucun

Objectifs d'apprentissage pour le chapitre 5 :

5.1 Évaluer le déploiement d'une approche MBT

MBT-5.1.1 (K2) Décrire les facteurs de retour sur investissement pour l'introduction du MBT.

MBT-5.1.2 (K2) Expliquer comment les objectifs du projet sont liés aux caractéristiques de l'approche MBT.

MBT-5.1.3 (K1) Rappeler les métriques sélectionnées et les indicateurs clés de performance pour mesurer les progrès et les résultats des activités de MBT.

5.2 Gérer et contrôler le déploiement d'une approche MBT

MBT-5.2.1 (K1) Rappeler les bonnes pratiques en matière de gestion des tests, de gestion du changement et de travail collaboratif lors du déploiement du MBT.

MBT-5.2.2 (K1) Rappeler les facteurs de coût du MBT.

MBT-5.2.3 (K1) Rappeler la nécessité d'intégrer l'outil MBT avec les outils de gestion de configuration, de gestion des exigences, de gestion des tests et d'automatisation des tests.

5.1 Évaluer le déploiement d'une approche MBT

Le déploiement d'une approche MBT dans une organisation suit souvent le processus classique d'adoption d'un produit :

1. Prise de conscience - Établir des objectifs d'amélioration dans le processus de test et identifier le MBT comme une technologie susceptible de répondre à tout ou partie de ces objectifs.
L'identification des opportunités d'amélioration est un facteur important pour motiver les équipes de test à changer leur façon de travailler.
2. Intérêt - En savoir plus sur le MBT.
3. Évaluation - Analyser les grands principes du MBT et les approches MBT existantes quant à leur applicabilité dans le contexte d'un projet donné.
4. Essai - Définir des indicateurs clés de performance (KPI – Key Performance Indicators) et mettre en place un projet pilote pour mesurer les améliorations obtenues.
5. Adoption - Conduire et renforcer le changement en améliorant les compétences et en modifiant les comportements au sein de l'organisation.

Par conséquent, l'estimation du retour sur investissement (ROI – Return of Investment) et la capacité à évaluer l'impact du MBT sur la performance du projet de test (sur la base des KPI) font partie du cycle d'adoption.

5.1.1 Les facteurs de ROI de l'introduction du Model-Based Testing

L'approche MBT génère des coûts à la fois pendant la phase d'introduction et pendant la phase d'exploitation. Pour une introduction réussie du MBT, les coûts doivent être contrebalancés par des bénéfices sur les aspects suivants :

- L'équilibre financier net en tenant compte du processus de test complet sur une échelle à long terme.
- La qualité de la conception des tests et un impact positif sur l'ensemble du processus de développement

Ainsi, pour calculer le ROI, l'ensemble des coûts, des économies et des résultats des améliorations sur le processus de test doit être pris en compte.

Coûts (voir section 5.2.2) :

- Introduction du MBT
- Coûts récurrents du MBT dans le processus de test

Gains attendus du MBT :

- Une validation en amont des exigences lors de la modélisation MBT permet d'éviter des anomalies coûteuses dans les phases ultérieures du développement :
Le MBT permet aux activités de modélisation d'intervenir tôt dans les phases projet, lorsque les exigences ne sont pas encore finalisées, apportant du feedback et une forte amélioration de la maturité des exigences.
- La réduction de l'effort de conception des tests par la réutilisation des artefacts MBT et la suppression des redondances :

Dans un modèle MBT, les différents chemins et scénarios métier sont définis sans aucune redondance, alors que les suites de tests définies manuellement portent, par construction, de nombreuses redondances coûteuses lors de la maintenance des tests manuels ou des scripts automatisés.

- La réduction de l'effort d'implémentation des tests par l'automatisation de la génération à partir des modèles MBT :
Lorsque le modèle MBT est développé, les tests sont générés automatiquement, réduisant l'effort d'implémentation et de maintenance des tests et de la traçabilité entre les exigences et les tests.
- La détection des défauts en amont du processus accélère le projet :
Ceci est induit par la capacité du MBT à démarrer la conception des tests via la modélisation, très tôt dans le processus de test, évitant ainsi les goulets d'étranglement du test.
- L'apport pour la gestion du projet de test :
En utilisant les différentes stratégies de génération et les critères de sélection de tests, l'équipe de test peut sélectionner l'ensemble des cas de test les plus adaptés à l'objectif défini.
- Une réduction potentielle du nombre de tests (par rapport à la conception de test sans MBT) due aux algorithmes de génération de tests, optimisant le nombre de pas de test et le nombre de tests pour un objectif de test donné. Cela induit une réduction de l'effort d'exécution des tests et peut être mis en œuvre en appliquant de façon systématique des stratégies de génération optimisées.

Bénéfices pour l'amélioration de la qualité des tests :

- Amélioration des méthodes de conception des tests, de la couverture des tests et de la cohérence de la conception des tests.
- Soutien à la gestion des tests en utilisant les résultats du MBT, y compris la priorisation et l'assurance qualité de la conception des tests
- Amélioration de la traçabilité grâce à l'amélioration du contenu et de l'organisation de la conception des tests

5.1.2 Les objectifs organisationnels et leur relation avec les caractéristiques de l'approche MBT

Le Model-Based Testing peut améliorer la qualité du processus de test, réduire les efforts et faciliter la communication. Le périmètre des améliorations et leur impact qualitatif et quantitatif sur l'organisation dépendent fortement des caractéristiques de l'approche MBT mise en œuvre. Ainsi, la mise en œuvre de l'approche MBT doit être réalisée en visant ce périmètre des améliorations recherchées. Le tableau suivant propose des exemples d'objectifs d'amélioration et indique comment l'approche MBT peut permettre de les atteindre.

Objectifs organisationnels	Axe d'amélioration du MBT	Obtenu par
Amélioration de la qualité des tests.	<ul style="list-style-type: none"> • Méthodes de conception des tests pour une meilleure couverture des tests. • Traçabilité (permettant des mesures de couverture et une meilleure capacité d'analyse d'impact). 	<p>Modèles distincts pour les activités de développement et de MBT (permettant l'état d'esprit du testeur et encourageant l'indépendance).</p> <p>Niveau d'abstraction bien défini basé sur le plan de test.</p> <p>Haut degré d'automatisation des processus, y compris la génération de produits des activités</p>

	<ul style="list-style-type: none"> Automatisation des processus pour éviter les erreurs humaines. 	<p>de test et l'exécution des tests afin de réduire les erreurs humaines.</p> <p>Critères de sélection des tests bien définis et adaptables à des objectifs de test spécifiques.</p>
Réduction de l'effort	<ul style="list-style-type: none"> Automatisation des processus. Traçabilité (à l'appui de l'automatisation des processus). 	<p>Modèle MBT partagés (réutilisation des modèles MBT).</p> <p>Haut degré d'automatisation des processus et produits de test générés automatiquement.</p> <p>Critères de sélection des tests bien définis pour générer l'ensemble le plus efficace de cas de test à exécuter.</p>
Amélioration de la communication	<ul style="list-style-type: none"> Approche MBT adéquate, compréhensible et utilisable, reflétant le niveau d'abstraction de la pensée des acteurs concernés. 	<p>Niveau d'abstraction approprié pour tous les acteurs concernés (par exemple, haut niveau et orienté métier pour les analystes métier ou détaillé et orienté test pour les testeurs).</p>

Une combinaison d'objectifs organisationnels peut conduire à des exigences contradictoires pour la mise en œuvre du MBT. Il est possible de résoudre ce problème en développant différents modèles MBT pour différents objectifs de test (voir section 2.1.3).

5.1.3 Métriques et indicateurs clés de performance

L'introduction du MBT dans une organisation doit se fonder sur des objectifs, des mesures et des indicateurs clés de performance clairs qui peuvent être utilisés pour mesurer les progrès et les résultats des activités de MBT.

Les mesures et les indicateurs clés de performance à surveiller peuvent être les suivants :

- Le nombre d'exigences gérées et tracées dans le modèle MBT, et la couverture des exigences (en pourcentage) par les cas de test générés.
- La taille et la complexité du modèle MBT.
- Le nombre de cas de test / scripts générés et le nombre de cas de test / scripts générés par jour-personne.
- Le nombre de défauts trouvés dans les exigences au cours des activités de modélisation en MBT.
- Le niveau de réutilisation des éléments du modèle MBT d'un projet à l'autre.
- Le niveau d'utilisation des modèles MBT par les parties prenantes du projet (entreprise / développement / tests).
- Le pourcentage de gain d'efficacité par rapport aux approches précédentes pour la conception des tests en termes de productivité (tests moins chers).

- Le pourcentage de gain d'efficacité par rapport aux approches précédentes pour la conception des tests en termes de détection des défauts (meilleurs tests).

La définition et le suivi de mesures et d'indicateurs clés de performance font partie des meilleures pratiques de gestion de projet lors du déploiement d'une approche MBT.

5.2 Gérer et contrôler le déploiement d'une approche MBT

5.2.1 Bonnes pratiques pour le déploiement du MBT

Le processus de test basé sur le MBT, ainsi que ses produits et outils de travail, doivent être étroitement intégrés dans le processus de développement, le processus de test et la chaîne d'outils existants. Ceci est également important pour l'intégration dans la chaîne d'outils de gestion du cycle de vie de l'application, par exemple l'intégration du MBT dans les processus et outils d'ingénierie des exigences.

Une intégration de qualité est un facteur clé du succès de l'introduction du MBT. Les bonnes pratiques en la matière sont les suivantes :

- Gestion de la configuration de tous les produits d'activités MBT, y compris :
 - La base de test.
 - Les modèles MBT et les critères de sélection des tests.
 - Les cas de test et scripts de test générés.
 - La couche d'adaptation.

Dans un processus de développement, les produits d'activités tels que les produits de mise en production ou de déploiement sont versionnés. Ceci est obligatoire pour établir un processus de développement et de déploiement fonctionnel. Pour intégrer et relier les produits d'activités MBT dans un tel processus, la gestion de la configuration des produits d'activités MBT est également obligatoire.

- Intégration du processus de génération de tests MBT à l'intégration continue
- L'intégration continue permet de lier la construction automatique de l'application et son test automatisé. Dès que l'application est construite, le serveur d'intégration continue lance les outils de test automatisé pour vérifier la nouvelle version. Comme les tests unitaires, les tests automatisés MBT ont vocation à faire partie de l'intégration continue en particulier pour le test de régression.
- Intégration avec les pratiques d'ingénierie des exigences et de gestion du backlog
- Les exigences et les éléments du backlog (fonctionnalités planifiées du produit) doivent être testés avant d'être réalisés (selon la définition de "réalisé" [ISTQB_FL_SYL]). Par exemple, dans le cadre d'un développement logiciel en mode Agile, le processus de test doit se refléter dans le processus de gestion du backlog. Si le MBT est utilisé pour tester les éléments du backlog, les produits d'activités MBT correspondants doivent être traçables dans l'outil de gestion du backlog, c'est-à-dire "quels cas de test, de quel modèle versionné, avec quel objectif de test, ont été utilisés pour tester cet élément".

5.2.2 Facteurs de coût du MBT

Les tableaux suivants mettent en relation les coûts initiaux et les coûts de fonctionnement du MBT avec les activités de test du processus de test de l'ISTQB.

Coûts initiaux (pour l'organisation et pour le projet) :

Coûts initiaux pour l'organisation	Coûts initiaux du projet
<p>Vérifier les ressources et les connaissances existantes pour l'introduction du MBT.</p> <p>Évaluer les approches et les outils en matière de MBT.</p> <p>Définir et mettre en œuvre les méthodes et le processus de MBT.</p> <p>Intégration avec la gestion des exigences, la gestion des tests, l'intégration continue.</p> <p>Automatisation et intégration des rapports MBT.</p> <p>Mise en place de moyens d'archivage des produits du travail MBT.</p> <p>Création de lignes directrices générales en matière de modélisation et de processus MBT.</p> <p>Accompagnement et formation en MBT.</p> <p>Octroi de licences pour les outils de MBT.</p>	<p>Création d'une modélisation MBT spécifique au projet et de lignes directrices pour le processus.</p> <p>Création du modèle initial de MBT.</p> <p>Transformer les actifs (par exemple, des cas de test textuels aux modèles de MBT).</p> <p>Migration des modèles de MBT.</p>

Coûts récurrents :

Activité et tâches de test	Coûts récurrents
Général	<p>Licences des outils (en fonction du modèle de licence) et coûts de maintenance.</p> <p>Coaching et formation des nouveaux membres de l'équipe.</p>
Planification des tests	<p>Analyse de la base de test en ce qui concerne la mise en œuvre du MBT.</p> <p>Planification du développement, de l'enrichissement et de la dérivation des modèles MBT.</p>
Analyse et conception des tests	<p>Modélisation MBT.</p> <p>Refonte des modèles MBT.</p> <p>Validation et vérification des modèles.</p>
Mise en œuvre et exécution des tests	<p>Choisir les critères de sélection des tests appropriés.</p> <p>Générer des cas de test exécutables.</p>

	Développement de la couche d'adaptation des tests (en cas d'exécution automatisée des tests). Exécution des cas de test (manuellement ou automatiquement).
Pilotage et suivi des tests	Contrôler en permanence la qualité des modèles MBT. Assurer la traçabilité des défauts. Documenter les critères de complétion des tests. Combiner l'évaluation du MBT et d'autres évaluations de tests dans un rapport commun.
Clôture des tests	Archiver les produits du travail en MBT. Documenter les connaissances acquises en matière de MBT. Transition des produits et processus de travail MBT vers la maintenance.

5.2.3 Intégration de l'outil MBT

Un aspect important de l'évaluation des outils de MBT est lié à l'intégration de l'outil de MBT avec les outils de gestion de la configuration, de gestion des exigences, de gestion des tests et d'automatisation des tests.

Une chaîne d'outils de test typique intégrant le MBT est décrite dans le diagramme ci-dessous.

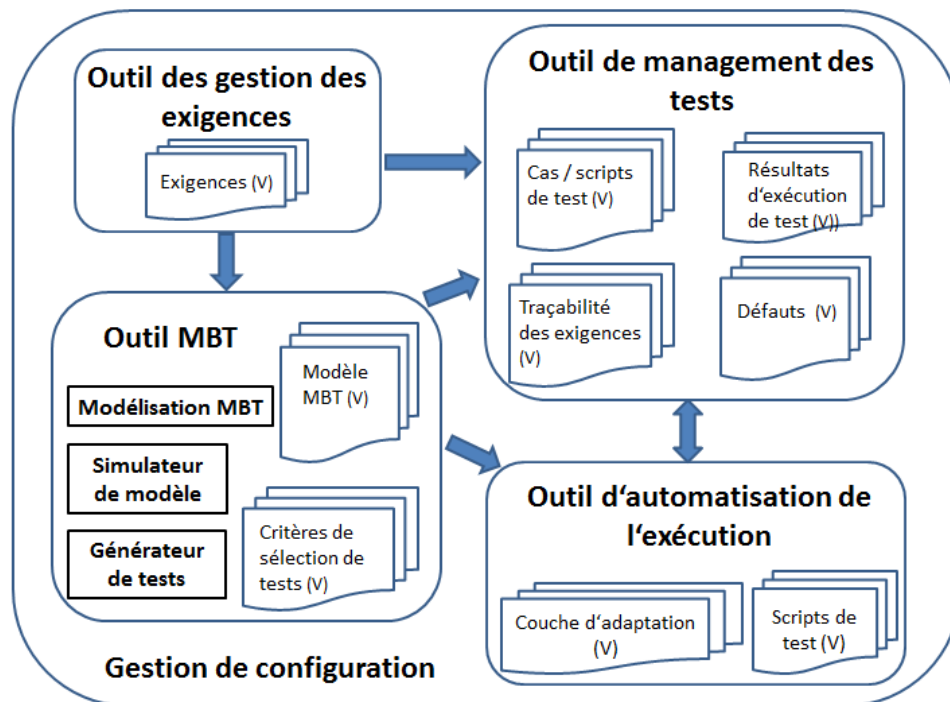


Figure 1 – Exemple de chaîne outillée intégrant le MBT – le symbole (V) indique que le produit d'activités considéré est mis en gestion de configuration

Cette chaîne d'outils typique prend en charge les principales activités du processus de test basé sur le MBT, y compris :

- Le développement, la révision et la validation itérative du modèle MBT à l'aide de l'outil de modélisation, du simulateur de modèle et de la traçabilité avec les exigences.
- La génération de tests en appliquant des critères de sélection de tests au modèle MBT soutenu par le générateur de cas de test.
- La génération de cas de test et de scripts de test ainsi que de liens de traçabilité pouvant être exportés vers l'outil de gestion des tests et le cadre d'automatisation des tests (en cas d'automatisation de l'exécution des tests).
- La gestion de la configuration du logiciel de test avec MBT tel que le modèle MBT et la couche d'adaptation.

6 Abréviations

Abréviation	Signification	Traduction
ATDD	Acceptance Test-Driven Development	Développement piloté par les tests d'acceptation
BPMN	Business Process Modeling Notation	Notation pour la modélisation des processus métiers
CTFL	Certified Tester Foundation Level	Testeur certifié de niveau Fondation
ISTQB	International Software Testing Qualifications Board	International Software Testing Qualifications Board
KPI	Key Performance Indicator	Indicateur clé de performance
MBT	Model-Based Testing	Tests basés sur des modèles
ROI	Return On Investment	Retour sur investissement
UML	Unified Modeling Language	Langage de modélisation unifié

7 Références

Normes

[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

Documents ISTQB®

[ISTQB_FL_SYL] Syllabus du niveau Fondation de l'ISTQB, CTFL V4.0

[ISTQB_ATA_SYL] Syllabus niveau avancé de l'ISTQB Analyste de Test, CTAL-TA V3.1.2

[ISTQB_GLOSSARY] Glossaire standard des termes utilisés dans les tests de logiciels, <https://glossary.istqb.org/>

[ISTQB_AcT_SYL] ISTQB Testeur certifié Test d'acceptation (CT-AcT), V1.0.

Ouvrages

[Utt07] Mark Utting et Bruno Legeard, "Practical Model-Based Testing – A tools approach," Morgan&Kauffmann, 2007

[Zan11] Justyna Zander (Editeur), Ina Schieferdecker (Editeur) and Pieter J. Mosterman (Editor), "Model-Based Testing for Embedded Systems," CRC Press, 2011.

[Kra16] Anne Kramer et 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

Articles

[Sch12] Ina Schieferdecker: Model-Based Testing. IEEE Software 29(1): 14-18, 2012.

[Utt12] Mark Utting, Alexander Pretschner and Bruno Legeard, "A Taxonomy of Model-Based Testing Approaches," Softw. Test. Verif. Reliab. 22 (5), 297–312, 2012.

8 Annexe A - Langages de modélisation simple

Pour les objectifs d'apprentissage du niveau K3, deux langages de modélisation graphique simples sont abordés ici :

- Le premier est une sous-partie des diagrammes d'activité UML.
- Le second est une sous-partie des machines à états UML.

Dans les deux sections suivantes, ces sous-parties d'UML, qui peuvent être utilisées pour pratiquer la modélisation MBT, sont définies à l'aide d'exemples.

8.1 Un langage de modélisation graphique simple de workflow

Ce langage de modélisation peut être utilisé pour développer des modèles MBT représentant des workflows ou des diagrammes d'activités. Il permet de modéliser un flot d'activités contrôlé par des décisions intermédiaires. Les modèles MBT développés avec ce langage permettent de générer des tests par application de critères de sélection tels que ceux vus au chapitre 3.

Ce langage de modélisation graphique est une sous-partie du diagramme d'activité UML. Il est composé des éléments suivants :

- Un cercle noir représentant le début (état initial) du flux de travail et un cercle noir encadré représentant la fin (état final).
- Des rectangles arrondis représentant les actions.
- Des losanges représentant les nœuds de décision et de fusion avec des étiquettes possibles (texte).
- Des flèches représentant les flux, éventuellement accompagnées d'expressions telles que du texte ou des expressions logiques (y compris des opérateurs arithmétiques et booléens).
- Des barres représentant le début (scission) ou la fin (jonction) d'activités parallèles concurrentes - la séquence se poursuit après que toutes les activités parallèles sont arrivées à la jonction.
- Des identificateurs d'exigences reliant les activités à l'aide de lignes pointillées.
- Les sous-diagrammes sont représentés par des rectangles.

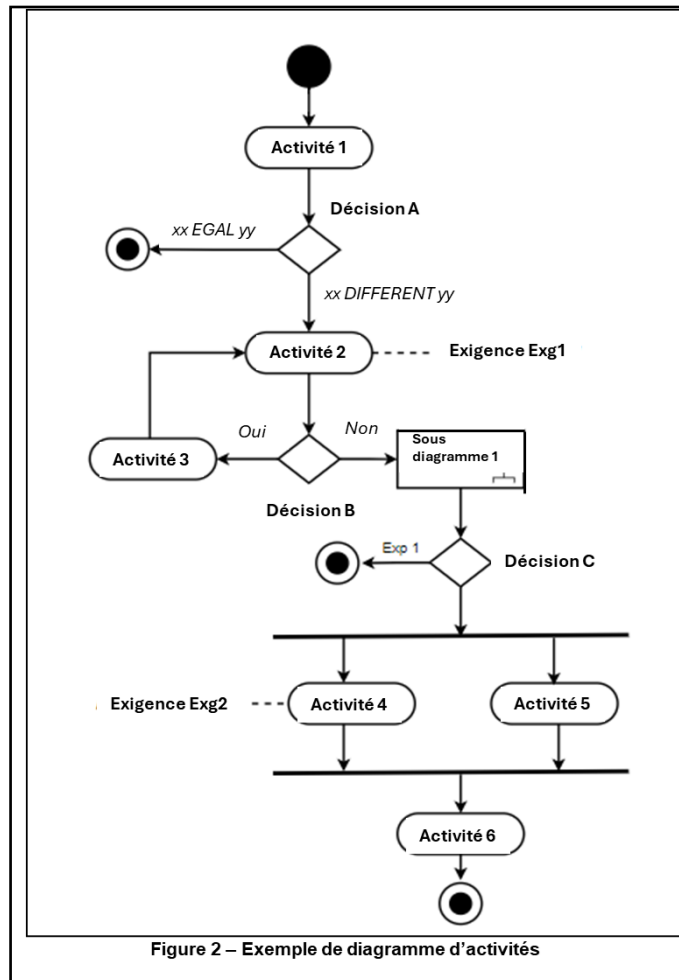


Figure 2 – Exemple de diagramme d'activités

Figure 2 – Exemple de diagramme d'activités

Une telle description d'un langage de modélisation est plutôt informelle et simple, mais elle contient suffisamment d'informations pour être utilisée afin de créer des modèles MBT dans le contexte de ce syllabus.

La figure 2 représente un exemple abstrait de modèle MBT développé avec ce langage de modélisation. Elle montre un comportement qui commence toujours par l'activité 1.

Le processus se termine si "xx EGAL yy". Dans le cas contraire, l'activité 2 est exécutée.

L'activité 3 et l'activité 2 sont exécutées à plusieurs reprises jusqu'à ce que le résultat de la décision B soit "Non".

Dans ce cas, l'activité 4 et le comportement du sous-schéma 1 sont exécutés.

Le comportement s'arrête si le résultat de la décision C est égal à Exp 1.

Dans le cas contraire, l'activité 5 et l'activité 6 sont exécutées en parallèle, puis le comportement s'arrête. La notation montre que l'activité 2 est liée à l'exigence Exg 1 et que l'activité 5 est liée à l'exigence Exg 2.

8.2 Un langage de modélisation graphique simple pour les diagrammes de transition d'état

Ce langage de modélisation graphique est une sous-partie des machines à états UML. Il est composé des éléments suivants :

- Un cercle noir représentant le début (état initial) du flux de travail et un cercle noir encerclé représentant la fin (état final).
- Des rectangles arrondis représentant les états.
- Des flèches représentant les transitions avec l'annotation textuelle "événement [condition] / action" (le déclencheur se référant à tout événement, la garde contenant des opérateurs arithmétiques et booléens qui forment des conditions, l'effet appelant toute action).
- Les identificateurs d'exigences sont reliés à tout type d'élément par des lignes pointillées.
- Les sous diagrammes sont indiqués par des rectangles.

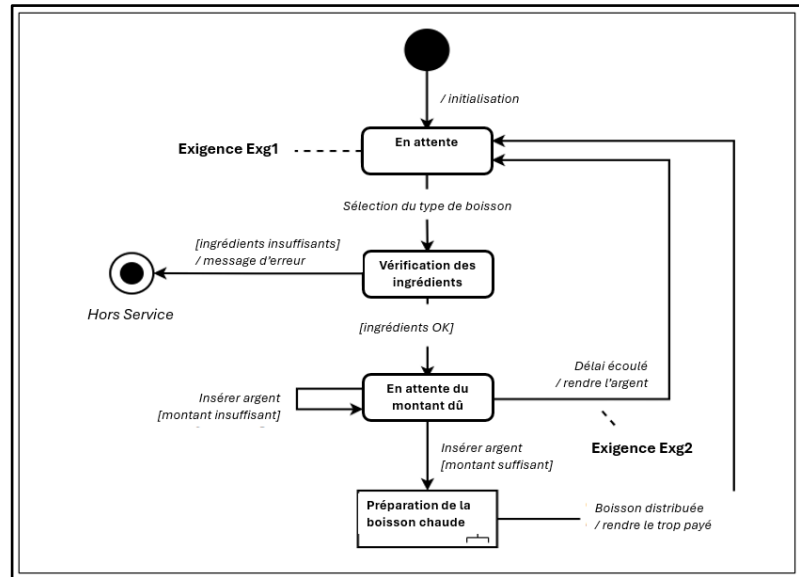


Figure 3 – Exemple de diagramme de machine à états

Une telle description d'un langage de modélisation est plutôt informelle et simple, mais elle contient suffisamment d'informations pour être utilisée afin de créer des modèles MBT dans le contexte de ce syllabus.

La figure 3 représente un exemple de modèle MBT développé avec ce langage de modélisation. Elle montre le comportement d'un distributeur de boissons distribuant du thé et du café.

Si l'automate est activé, la quantité déposée est initialisée et il attend les entrées de l'utilisateur. Ces entrées de l'utilisateur peuvent être "sélectionner le thé" ou "sélectionner le café", indiquant la boisson souhaitée.

En réaction à la sélection, les préparations correspondantes sont effectuées (par exemple, en fixant des prix différents pour le thé et le café). Ensuite, l'utilisateur doit payer la boisson.

Le modèle contient une boucle qui garantit que l'automate n'entre dans l'état "prêt à distribuer" que lorsqu'une quantité suffisante d'argent a été introduite.

Si l'utilisateur cesse d'insérer des pièces avant d'avoir inséré le montant nécessaire, l'automate attend 10 secondes avant de rendre l'argent inséré et de recommencer.

Si suffisamment d'argent a été introduit, la boisson est distribuée.

Seule exception, l'automate s'arrête et rend l'argent inséré s'il n'y a pas assez d'ingrédients (eau, thé, café).

On peut également voir que l'exigence Exg 1 est liée à l'état "En attente", ce qui indique qu'un test qui atteint cet état couvre également cette exigence. L'exigence Exg 2 est liée à la transition "Rendre l'argent si temps écoulé" pour indiquer que le remboursement du paiement en cas de non consommation est un aspect important.

9 Annexe B - Objectifs d'apprentissage/Niveau de connaissances cognitives

Les objectifs d'apprentissage spécifiques s'appliquant à ce programme sont indiqués au début de chaque chapitre. Chaque sujet du programme sera examiné en fonction de l'objectif d'apprentissage qui lui est associé.

Les objectifs d'apprentissage commencent par un verbe d'action correspondant à son niveau cognitif de connaissance, comme indiqué ci-dessous.

Niveau 1 : Se souvenir (K1)

Le candidat se souviendra, reconnaîtra et rappellera un terme ou un concept.

Verbes d'action : Rappeler, reconnaître.

Exemples
Rappeler les concepts de la pyramide des tests.
Reconnaître les objectifs typiques des tests.

Niveau 2 : Comprendre (K2)

Le candidat peut sélectionner les raisons ou les explications des affirmations liées au sujet, et peut résumer, comparer, classer et donner des exemples pour le concept de test.

Verbes d'action : Classer, comparer, différencier, distinguer, expliquer, donner des exemples, interpréter, résumer

Exemples	Notes
Classer les outils de test en fonction de leur objectif et des activités de test qu'ils soutiennent.	
Comparer les différents niveaux de test.	Il peut être utilisé pour rechercher des similitudes, des différences ou les deux.
Différencier le test du débogage.	Recherche des différences entre les concepts.
Distinguer les risques liés au projet de ceux liés au produit.	Permet de classer séparément deux concepts (ou plus).

Exemples	Notes
Expliquer l'impact du contexte sur le processus de test.	
Donnez des exemples de raisons pour lesquelles les tests sont nécessaires.	
Déduire la cause première des défauts à partir d'un profil de défaillances donné.	
Résumer les activités du processus d'examen du produit du travail.	

Niveau 3 : Appliquer (K3)

Le candidat peut exécuter une procédure lorsqu'il est confronté à une tâche familière, ou sélectionner la procédure correcte et l'appliquer à un contexte donné.

Verbes d'action : Appliquer, mettre en œuvre, préparer, utiliser

Exemples	Notes
Appliquer l'analyse de la valeur limite pour dériver des cas de test à partir d'exigences données.	Doit faire référence à une procédure / une technique / un processus, etc.
Mettre en œuvre des méthodes de collecte de données pour répondre aux exigences techniques et de gestion.	
Préparer des tests de facilité d'installation pour les applications mobiles.	
Utiliser la traçabilité pour contrôler l'avancement des tests afin de s'assurer de leur complétion et de leur cohérence avec les objectifs, la stratégie et le plan de test.	Peut être utilisé dans un LO qui veut que le candidat soit capable d'utiliser une technique ou une procédure. Semblable à "appliquer".

Référence

(Pour les niveaux cognitifs des objectifs d'apprentissage)

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 Annexe C - Matrice de traçabilité des résultats opérationnels avec les objectifs de formation

Cette section liste la traçabilité entre les Résultats métier et les Objectifs d'Apprentissage du Syllabus de Tests Basés sur le Modèle du Testeur Certifié de l'ISTQB® v1.1.

Objectifs métier : CT-MBT		BO1	BO2	BO3	BO4	BO5
MBT-BO1	Collaborer au sein d'une équipe de tests en utilisant la terminologie standard et les concepts, processus et techniques MBT établis.	36				
MBT-BO2	Appliquer et intégrer les tests basés sur des modèles dans un processus de test.		17			
MBT-BO3	Créer et maintenir efficacement les modèles MBT en utilisant les techniques établies et les meilleures pratiques des tests basés sur les modèles.			16		
MBT-BO4	Sélectionner, créer et maintenir les produits d'activités de test à partir des modèles MBT en tenant compte du risque et de la valeur des fonctionnalités testées.				10	
MBT-BO5	Soutenir l'organisation dans l'amélioration de son processus d'assurance qualité afin qu'il soit plus constructif et plus efficient.					8

Objectifs métier : CT-MBT			BO1	BO2	BO3	BO4	BO5
	Objectif d'apprentissage	K-Level					
1	Introduction aux tests basés sur des modèles						
1.1	Objectifs et motivations du MBT						
MBT-1.1.1	Décrire les bénéfices attendus du MBT	K2	X				X
MBT-1.1.2	Décrire les attentes erronées et les pièges du MBT	K2	X	X			X
1.2	Activités et produits d'activités du MBT dans le processus de test						
MBT-1.2.1	Synthétiser les activités spécifiques du MBT déployées dans un processus de test	K2	X	X			
MBT-1.2.2	Rappeler les produits d'activités essentiels en MBT (intrants et extrants)	K1	X	X			
1.3	Intégrer l'approche MBT dans le cycle de développement des logiciels						
MBT-1.3.1	Expliquer comment le MBT s'intègre dans les différents types de cycles de développement logiciels	K2	X	X			
MBT-1.3.2	Expliquer comment le MBT facilite la démarche d'ingénierie des exigences	K2	X	X	X		
2	Modélisation MBT						

Objectifs métier : CT-MBT			BO1	BO2	BO3	BO4	BO5
2.1	Modélisation MBT						
MBT-2.1.1	Développer un modèle MBT simple pour un objet de test et des objectifs de test prédéfinis en utilisant un langage de modélisation de workflow	K3	X		X		
MBT-2.1.2	Développer un modèle MBT simple pour un objet de test et des objectifs de test prédéfinis en utilisant un langage de modélisation états-transitions	K3	X		X		
MBT-2.1.3	Fournir des exemples de modèle MBT décrivant le système, l'environnement ou les tests	K2	X		X		
MBT-2.1.4	Donner des exemples de la manière dont un modèle MBT dépend des objectifs du test	K2	X	X	X		
2.2	Langages de modélisation pour le MBT						
MBT-2.2.1	Donner des exemples de catégories de langage de modélisation couramment utilisées pour le MBT	K1	X		X		
MBT-2.2.2	Donner des exemples d'adéquation entre les objectifs des tests et les catégories de langage de modélisation pour différents systèmes et objectifs de test du projet	K2	X		X		
2.3	Bonnes pratiques de la modélisation MBT						
MBT-2.3.1	Rappeler les caractéristiques de qualité des modèles MBT	K1	X		X		

Objectifs métier : CT-MBT			BO1	BO2	BO3	BO4	BO5
MBT-2.3.2	Décrire les erreurs et les pièges classiques lors des activités de modélisation MBT	K2	X		X		
MBT-2.3.3	Expliquer les avantages de lier les exigences et des informations relatives au processus de test au sein du modèle MBT	K2	X		X		
MBT-2.3.4	Expliquer la nécessité d'un guide de bonnes pratiques de modélisation MBT	K2	X		X		
MBT-2.3.5	Donner des exemples où la réutilisation de modèles existants (de la phase des exigences ou de la phase de développement) est ou n'est pas appropriée	K2	X	X	X		
MBT-2.3.6	Rappeler les types d'outils adaptés aux différentes activités de la modélisation MBT	K1	X		X		
MBT-2.3.7	Résumer le développement, la revue et la validation itératifs des modèles MBT	K2	X	X	X		
3	Critères de sélection pour la génération des cas de test						
3.1	Classification des critères de sélection de tests utilisés en Model-Based Testing						
MBT-3.1.1	Classer les différentes familles de critères de sélection des tests utilisés pour la génération de tests à partir de modèles	K2	X			X	

Objectifs métier : CT-MBT			BO1	BO2	BO3	BO4	BO5
MBT-3.1.2	Générer des cas de test à partir d'un modèle MBT pour atteindre des objectifs de test dans un contexte donné	K3	X			X	
MBT-3.1.3	Fournir des exemples de critères de sélection de tests de type couverture de modèle, orienté donnée, à base de scénarios et de patterns, et spécifique au projet	K2	X			X	
MBT-3.1.4	Montrer comment les critères de sélection de tests du Model-Based Testing sont en rapport avec les techniques de conception de test ISTQB® de niveau Fondation	K2	X			X	
3.2	Application des critères de sélection de tests						
MBT-3.2.1	Rappeler les niveaux d'automatisation de la génération de tests	K1	X	X		X	
MBT-3.2.2	Appliquer des critères de sélection de tests sur un modèle MBT donné	K3	X			X	
MBT-3.2.3	Décrire les bonnes pratiques de la mise en œuvre des critères de sélection de tests en MBT	K2	X			X	
4	Implémentation et exécution des tests en MBT						
4.1	Aspects spécifiques au MBT de l'implémentation et de l'exécution des tests						
MBT-4.1.1	Expliquer la différence entre les cas de test de haut niveau et de bas niveau dans le contexte du MBT	K2	X	X		X	

Objectifs métier : CT-MBT			BO1	BO2	BO3	BO4	BO5
MBT-4.1.2	Expliquer les différents types d'exécution des tests dans le contexte de le MBT	K2	X	X			
MBT-4.1.3	Effectuer des mises à jour d'un modèle MBT et de la génération de tests suite à des changements d'exigences, d'objet de test ou d'objectifs de test	K3	X		X	X	
4.2	Activités d'adaptation des tests pour l'exécution en MBT						
MBT-4.2.1	Expliquer quel type d'adaptation des tests peut être nécessaire pour l'exécution des tests en MBT	K2	X		X	X	
5	Évaluation et déploiement d'une approche MBT						
5.1	Évaluer le déploiement d'une approche MBT						
MBT-5.1.1	Décrire les facteurs de retour sur investissement pour l'introduction du MBT	K2	X	X			X
MBT-5.1.2	Expliquer comment les objectifs du projet sont liés aux caractéristiques de l'approche MBT	K2	X	X			X
MBT-5.1.3	Rappeler les métriques sélectionnées et les indicateurs clés de performance pour mesurer les progrès et les résultats des activités de MBT	K1	X	X			X
5.2	Gérer et contrôler le déploiement d'une approche MBT						

Objectifs métier : CT-MBT			BO1	BO2	BO3	BO4	BO5
MBT-5.2.1	Rappeler les bonnes pratiques en matière de gestion des tests, de gestion du changement et de travail collaboratif lors du déploiement du MBT	K1	X	X			X
MBT-5.2.2	Rappeler les facteurs de coût du MBT	K1	X	X			X
MBT-5.2.3	Rappeler la nécessité d'intégrer l'outil MBT avec les outils de gestion de configuration, de gestion des exigences, de gestion des tests et d'automatisation des tests	K1	X	X			X

11 Annexe D - Notes de mise à jour

ISTQB® Testeur certifié en Model-Based Testing Syllabus v1.1 est une mise à jour mineure de la version 1.0 (2015) afin de s'aligner sur le Syllabus ISTQB Testeur Certifié de niveau Fondation en version v4.0.

Le nouveau syllabus CT-MBT v1.1 est officiellement lancé le 23 février 2024 et disponible sur le site de l'ISTQB® pour la version en anglais et sur le site du CFTL pour la version en français.

Ce qui a été publié

1. CT-MBT Syllabus Version v1.1
2. CT-MBT Sample Exam Updated for Syllabus Version v1.1 (Document de questions et de réponses)

Liste des modifications du syllabus v1.1 vis-à-vis de la version v1.0

Alignement sur la version actuelle du glossaire et sur le syllabus niveau Fondation v4.0 :

- Spécification de la procédure de test => procédure de test.
- Processus de test fondamental => processus de test.
- Couverture des décisions => couverture des branches.
- 1.2.1 et 5.2.2 : activités alignées sur la section 1.4.1 "Activités et tâches de test" du syllabus de niveau Fondation en version v4.0.

Changements dans les objectifs d'apprentissage :

- FM-2.1.3 : renommé, taxonomie supprimée du texte.
- FM-2.2.2 : Le niveau K devient K2, l'accent est mis sur des exemples d'adéquation avec les objectifs du test.
- FM-5.2.3 : Niveau K réduit à K1, pas assez de temps pour entrer dans les détails requis pour K2.

Des éléments de contenu ont été supprimés car sortant du niveau fondation de ce syllabus :

- 2.1.2 & 2.1.3 : suppression de la notion de taxonomie MBT.
- 2.2.1 : suppression de la classification des langages de modélisation.
- 2.2.2 : modèles de caractéristiques, modèles temporels, graphiques de cause à effet supprimés de la liste d'exemples.
- 3.1.3 : tests par paires, couverture des conditions de décision (multiples) supprimés de la liste d'exemples.
- 4.1.2 : simplification de la description des approches d'adaptation.
- 4.1.3 : suppression de la référence explicite à la spécification de la couche d'adaptation des tests (seule la couche d'adaptation des tests est mentionnée).
- 5.2.3 : suppression de la référence au syllabus niveau Fondation.

Changement dans l'annexe 8 sur les langages de modélisation :

- 8.2 : L'élément de décision a été supprimé du diagramme d'état afin de simplifier le langage de modélisation des diagrammes de transition d'état.
- 8.1 et 8.2 : Mise à jour des figures.