

ISTQB® Testeur Certifié Niveau Fondation Model-Based Testing

Syllabus

Version 2015

International Software Testing Qualifications Board



Copyright

Ce document peut être copié dans son intégralité, ou partiellement, si la source est mentionnée.

Copyright Notice © International Software Testing Qualifications Board (ci-après appelé ISTQB®)
ISTQB est une marque enregistrée de l'International Software Testing Qualifications Board.

Copyright © Comité Français des Tests Logiciels – CFTL – pour la version française.

Groupe de travail ISTQB® – Model-Based Testing: Stephan Christmann (Responsable), Anne Kramer, Bruno Legeard, Armin Metzger, Natasa Micuda, Thomas Mueller, Stephan Schulz; 2014-2015.

Historique des modifications

Version	Date	Remarques
Beta 1.0	8 Mai 2015	Version Beta.
Version 2015 - Final	23 Octobre 2015	Version Finale.
Version Français en	8 février 2016	Version en Français

Table des matières

Historique des modifications.....	3
Table des matières.....	4
Remerciements.....	6
0 Introduction à ce Syllabus.....	7
0.1 Objectif.....	7
0.2 Présentation générale.....	7
0.3 <i>Objectifs de connaissance /niveaux de connaissance</i>	7
1 Introduction au Model-Based Testing – 90 minutes.....	8
1.1 Objectifs et motivations du MBT.....	8
1.1.1 Motivations principales du MBT.....	8
1.1.2 Attentes erronées et pièges du MBT.....	9
1.2 Activités et artefacts du MBT dans le processus de test.....	10
1.2.1 Activités spécifiques du MBT.....	10
1.2.2 Principaux artefacts du MBT (en entrée et en sortie).....	11
1.3 Intégration du MBT dans le cycle de développement logiciel.....	11
1.3.1 Le MBT dans les cycles de vie séquentiels et agiles.....	11
1.3.2 MBT et Ingénierie des exigences.....	12
2 Modélisation MBT – 250 minutes.....	14
2.1 Modélisation MBT.....	14
2.1.1 Activités de modélisation MBT.....	15
2.1.2 Sujet et focus du modèle MBT.....	15
2.1.3 La modélisation MBT dépend des objectifs de test.....	16
2.2 Langages de modélisation pour le MBT.....	17
2.2.1 Catégories principales de langages de modélisation pour le MBT.....	17
2.2.2 Catégories de langages de modélisation adaptées pour différents types de systèmes et d'objectifs de test.....	18
2.3 Bonnes pratiques de la modélisation MBT.....	18
2.3.1 Caractéristiques de qualité des modèles MBT.....	18
2.3.2 Erreurs types et pièges à considérer lors de la modélisation MBT.....	19
2.3.3 Relier les informations d'exigences et sur le processus de test avec le modèle MBT.....	19
2.3.4 Guide de modélisation MBT.....	20
2.3.5 Réutilisation pour le MBT de modèle de conception ou d'exigences existants.....	20
2.3.6 Outillage des activités de modélisation MBT.....	21
2.3.7 Modélisation, revue et validation itératives.....	21
3 Critères de sélection pour la génération des tests – 205 minutes.....	22
3.1 Classification des critères de sélection de tests utilisés en Model-Based Testing.....	22
3.1.1 Familles de critères de sélection de tests.....	23
3.1.2 La sélection des cas de tests en pratique.....	24

3.1.3 Exemples de critères de sélection de tests	24
3.1.4 Relation avec les techniques de conception de tests du niveau fondation	25
3.2 Application des critères de sélection de tests	25
3.2.1 Niveau d'automatisation de la génération de tests	25
3.2.2 Analyse de la mise en œuvre des critères de sélection de tests	25
3.2.3 Bonne pratique de la mise en œuvre de critères de sélection de tests en MBT	
26	
4 Implémentation et exécution des tests en MBT – 120 minutes.	27
4.1 Aspects spécifiques au MBT de l'implémentation et de l'exécution des tests	27
4.1.1 Notion de tests abstraits et concrets en MBT	27
4.1.2 Différents types d'exécution des tests	28
4.1.3 L'impact de changement sur les artefacts MBT	29
4.2 Activités MBT d'adaptation des tests pour l'exécution	29
5 Évaluation et déploiement d'une approche MBT – 60 minutes.....	31
5.1 Évaluer un déploiement MBT	31
5.1.1 Les facteurs de ROI de l'introduction du Model-Based Testing	32
5.1.2 Objectifs d'amélioration du processus de test et mise en œuvre de l'approche MBT	
33	
5.1.3 Métriques et principaux indicateurs de performance	33
5.2 Piloter le déploiement d'une approche MBT	34
5.2.1 Bonnes pratiques du déploiement MBT	34
5.2.2 Facteurs de coût du MBT	35
5.2.3 Intégration de l'outillage MBT	36
6 Définition des termes spécifiques au MBT	37
7 Abréviations et marques déposées	39
8 Références	40
Standards	40
Documents ISTQB – Versions françaises.....	40
Documents référencés dans le syllabus	40
Autres références	40
9. Annexe A – Langages de modélisation retenus dans le syllabus Model-Based Testing	
41	
9.1 Un langage simple de modélisation de workflow	41
9.2 Un langage simple de modélisation pour diagramme d'états-transitions	42

Remerciements

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.

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 Fournernet, 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.

Ce document a été formellement approuvé dans sa version finale 2015 lors de l'assemblée générale de l'ISTQB® du 23 Octobre 2015 à Shanghai (Chine).

0 Introduction à ce Syllabus

0.1 Objectif

Ce syllabus forme la base de l'extension Model-Based Testing du niveau fondation de la qualification Internationale en test de logiciels. L'International Software Testing Qualifications Board (ISTQB) et le Comité Français des Tests Logiciels (ci-après appelé CFTL) fournissent ce syllabus aux comités nationaux d'examens pour accréditer les fournisseurs de formations et pour en dériver des questions d'examen dans leur langue nationale. Les fournisseurs de formations produiront leurs cours et détermineront les méthodes de formation appropriées pour l'accréditation, et le syllabus aidera les candidats à se préparer aux examens.

0.2 Présentation générale

La certification Model-Based Testing de l'ISTQB® est un module de spécialisation de la certification Testeur Certifié au niveau fondation.

La certification ISTQB® Testeur Certifié au Niveau Fondation est un prérequis obligatoire pour le passage de la certification Model-Based Testing.

La qualification Model-Based Testing vise toutes les personnes impliquées dans les tests de logiciels. Ceci inclut les personnes ayant des rôles de testeurs, analystes de test, ingénieurs de test, consultants en test, gestionnaires de test, testeurs en phase d'acceptation utilisateur et développeurs de logiciels. Cette qualification est aussi appropriée pour toute personne souhaitant une compréhension de base de l'apport du Model-Based Testing pour optimiser les processus de test de logiciels, tels que les gestionnaires de projets, responsables qualité, responsables de développements logiciels, analystes métier et consultants en management.

Les possesseurs du Certificat Model-Based Testing ont étendu les connaissances acquises au niveau fondation pour mettre en œuvre une approche de test dirigée par les modèles pour améliorer l'efficacité des tests.

Les objectifs métier de la certification Model-Based Testing sont présentés dans le document [ISTQB_MBT_OVIEW].

0.3 Objectifs de connaissance /niveaux de connaissance

Comme pour le syllabus ISTQB® au niveau fondation, les niveaux de connaissance sont fournis pour chaque section de ce syllabus et classés de la façon suivante :

- K1 : se souvenir ;
- K2 : comprendre ;
- K3 : utiliser.

Toutes les définitions des termes listés sous la rubrique "Termes", après les titres de chapitre, correspondent à un niveau de connaissance K1. La définition de chaque terme peut être trouvée dans le glossaire de l'ISTQB® des termes du domaine des tests logiciels, et traduit en français par le CFTL.

1 Introduction au Model-Based Testing – 90 minutes.

Termes

Modèle MBT, Model-Based Testing (MBT)

Objectifs de connaissance pour le chapitre 1

1.1 Objectifs et motivations du MBT

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

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

1.2 Activités et artefacts du MBT dans le processus de test

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

FM-1.2.2 (K1) Rappeler les principaux artefacts du MBT (en entrée et en sortie)

1.3 Intégration du MBT dans le cycle de développement du logiciel

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

FM-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 s'appuyant sur la modélisation. Cette approche étend les techniques classiques de conception de tests telles que le partitionnement par classes d'équivalence, l'analyse des valeurs aux limites, le test par tables de décision, et le test à partir de diagrammes d'états-transitions ou de cas d'utilisation. L'idée essentielle de l'approche MBT est d'améliorer la qualité et l'efficacité des pratiques d'analyse, de conception et d'implémentation des tests par :

- la mise en œuvre outillée de modélisation pour le test permettant de satisfaire les objectifs de test du projet,
- la mise en œuvre de la modélisation MBT comme spécification de conception des cas de test. Les modèles MBT incluent des informations suffisamment précises pour permettre la génération automatique des cas de test directement à partir du modèle.

Le Model-Based Testing, ses activités et ses artefacts sont totalement intégrés aux processus, méthodes, environnements techniques et outils du cycle de vie du logiciel.

1.1.1 Motivations principales du MBT

Il y a deux aspects principaux dans les activités du test logiciel qui expliquent les motivations principales du MBT et qui expliquent comment le MBT contribue à l'amélioration des pratiques de test :

- L'amélioration de l'efficacité
 - La modélisation MBT renforce la communication entre les parties prenantes du projet.
 - L'amélioration de la communication permet une compréhension partagée des exigences et facilite la détection de mécompréhension potentielle dans ces exigences.

- Dans le cas de modèles MBT graphiques, les participants au projet (tels que les Analystes Métier) peuvent être impliqués plus facilement.
 - La modélisation MBT facilite une augmentation continue de la compétence des testeurs sur le domaine Métier.
 - Le niveau d'abstraction des modèles MBT rend plus facile l'identification des parties du système où se poseront le plus probablement des problèmes.
 - La génération et l'analyse des cas de test est possible avant que le système soit réellement implémenté.
- L'amélioration de l'efficacité
 - Une modélisation MBT et une vérification/validation du modèle en amont permettent d'éviter des défauts dans les premières étapes du cycle de vie du logiciel. Les modèles MBT peuvent être partagés avec les parties prenantes du projet (tels que les Analystes Métier et les développeurs) pour vérifier les exigences et identifier les manques à l'intérieur de ces exigences. Le MBT permet ainsi de mettre en œuvre le principe appelé "tester tôt" dans le syllabus ISTQB® niveau fondation.
 - Les artefacts MBT issus de précédents projets peuvent être réutilisés et adaptés pour de nouveaux projets.
 - Le MBT facilite l'automatisation (par exemple de la génération des tests et de la traçabilité entre exigences et tests) et permet la diminution des défauts qui peuvent être introduits dans les tests lorsque ceux-ci sont créés et maintenus manuellement.
 - Différentes suites de tests peuvent être générées à partir d'un même modèle MBT (par exemple en utilisant différents critères de sélection de tests), permettant ainsi une adaptation efficace des tests à des changements dans les exigences ou les objectifs de test.
 - Le MBT peut être utilisé pour différents objectifs de test et pour couvrir différents niveaux et types de test.
 - Le MBT permet de réduire les coûts de maintenance lorsque les exigences évoluent car le modèle MBT crée un point de maintenance unique.

1.1.2 Attentes erronées et pièges du MBT

Les équipes de test qui veulent mettre en œuvre le MBT doivent avoir des attentes réalistes sur les bénéfices et limitations de cette approche de test. Les attentes erronées, incompréhension et pièges classiques incluent :

- Penser que le MBT va résoudre tous les problèmes
Le MBT ne remplace pas une maîtrise des techniques de conception des tests (telles que celles présentées dans le syllabus ISTQB® niveau fondation par exemple), mais le MBT facilite et automatise leur mise en œuvre, permettant aux testeurs de réaliser des tests plus efficaces et plus efficaces. Une équipe de test qui méconnaît ces techniques de conception de test ne résoudra pas ce problème avec le MBT.
- Croire que le MBT est uniquement une question d'outillage
Des outils adaptés sont essentiels pour la réussite d'un projet MBT, mais l'acquisition d'outil n'est pas la première étape du déploiement de l'approche. Il faut d'abord définir des critères mesurables et objectifs sur l'amélioration attendue des pratiques du test. Le MBT modifie l'ensemble du processus de test (voir la section 1.2 de ce syllabus). L'introduction du MBT nécessite donc un soutien déterminé du management pour piloter les changements de processus et d'outillage dans l'organisation.
- Croire que les modèles MBT sont forcément corrects

Comme dans la conception manuelle de tests, le testeur peut introduire des erreurs lors du développement du modèle MBT. Les modèles MBT doivent donc être vérifiés et validés, en particulier par des revues, des analyses statiques outillées ou des simulations du modèle. Les modifications du modèle MBT se propagent dans les artefacts générés (tels que les cas et scripts de test, la matrice de traçabilité entre les exigences et les tests). Ces modifications doivent donc être validées avant leur mise en œuvre opérationnelle.

- 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 artefacts du MBT dans le processus de test

1.2.1 Activités spécifiques du MBT

Lorsque le MBT est déployé sur un projet, le processus de test est adapté avec les activités spécifiques du MBT, telles que :

- les activités de modélisation MBT (définition des règles de modélisation, développement et maintenance des modèles, administration des modèles, mise en place des outils),
- la génération d'artefacts (par exemple des cas et scripts de test, de la matrice de traçabilité entre les exigences et les tests) à partir des modèles MBT et des critères de sélection de tests.

Au minimum, le MBT modifie les phases d'analyse, de conception et d'implémentation des tests. En fonction des objectifs de test du projet, le MBT peut influencer la totalité des phases du processus de test.

Cela concerne en particulier :

- La planification et le suivi des tests qui doivent prendre en compte les activités spécifiques du MBT (avec des métriques spécifiques – voir chapitre 5 de ce syllabus).
- L'analyse et la conception des tests qui s'appuient sur la modélisation MBT, la mise en œuvre de critères de sélection de tests et de métriques spécifiques de couverture des tests.
- L'implémentation et l'exécution qui s'appuient sur la génération automatique des tests avec les outils MBT et l'adaptation des tests.
- L'évaluation des critères d'arrêt des tests et le reporting qui utilisent des métriques de couverture spécifiques (par exemple fondées sur la couverture du modèle MBT) et une analyse d'impact supportée par les modèles.
- La phase de clôture des tests peut inclure la mise en place de bibliothèques de modèles MBT pour une réutilisation ultérieure.

Le MBT contribue à automatiser le processus de test et la production des artefacts de test (tels que les cas et scripts de test et la matrice de traçabilité entre les exigences et les tests). Le MBT favorise un déplacement du test vers les phases en amont du cycle de vie du logiciel comparé aux techniques de conception classique de tests. Cela contribue à une vérification/validation en amont des exigences et améliore la communication, en particulier dans le cas de modèles graphiques. Les tests MBT viennent souvent renforcer des tests existants créés manuellement. Cela permet à l'équipe de test de vérifier la consistance entre les deux types de test.

Le MBT ne modifie par l'organisation des phases de test, mais la façon dont elles sont réalisées. Comme vu précédemment en section 1.1, le MBT :

- a un impact sur la qualité, les efforts, la communication et les parties prenantes du processus de test,
- permet aux activités de conception des tests d'intervenir plus tôt dans le projet.

Pour faciliter l'adoption du MBT, il est important d'obtenir une acceptation de l'approche et des changements que cela implique dans le processus de test par l'équipe en charge de sa mise en œuvre.

1.2.2 Principaux artefacts du MBT (en entrée et en sortie)

Les modèles MBT peuvent provenir soit de modélisations spécifiques au test soit d'une réutilisation de modèles développés lors de la conception du logiciel (voir la section 2.3.5).

La modélisation MBT peut être réalisée à différents niveaux d'abstraction et intégrer différentes informations pour le test. Les artefacts générés à partir des modèles MBT reflètent les choix de niveau d'abstraction.

En fonction des informations de test prises en compte et du niveau d'abstraction, le MBT s'intègre au processus de test au travers de différents artefacts d'entrée et de sortie.

Artefacts en entrée des activités du MBT :

- Stratégie de test.
- Bases de tests, en particulier les exigences, cibles de test, conditions de test, informations orales et modélisations ou éléments de conception existants.
- Rapports d'anomalies et d'incidents, registres de test, rapports d'exécution des tests issus de tests précédents.
- Guides méthodologiques décrivant le processus de test, documentations des outils.

Artefacts de sortie produits par les activités du MBT :

- Modèles MBT.
- Des parties du plan de test (configuration de l'environnement de test), ordonnancement des tests, métriques de test.
- Scénarios de test, suites de test, ordonnancement de l'exécution des tests, spécifications de conception des tests.
- Cas de test, spécifications des procédures de test, données de test, scripts de test, couche d'adaptation des tests (spécifications et code).
- Matrice bidirectionnelle de traçabilité entre les tests générés et les bases de test, en particulier les exigences, et les rapports d'anomalies.

1.3 Intégration du MBT dans le cycle de développement logiciel

1.3.1 Le MBT dans les cycles de vie séquentiels et agiles

Deux types principaux de cycles de vie du logiciel sont considérés dans cette section : séquentiel (tel que le modèle en V) et itératif-incrémental (tel que le développement agile). Dans les deux cas, le MBT doit en premier lieu être adapté aux objectifs de test du projet, de telle façon que les bénéfices du MBT soient obtenus en adéquation avec ces objectifs.

Les organisations implémentent le cycle de développement de façon très variée. Le processus de test utilisant le MBT doit s'adapter en conséquence. Par exemple, le MBT peut être utilisé pour le test d'acceptation sur un projet, et pour du test automatisé au niveau système pour un autre projet, en fonction des objectifs de test.

Les caractéristiques suivantes de mise en œuvre du MBT s'appliquent aussi bien aux cycles de développement séquentiel qu'en mode agile :

- Niveaux de test :
 - Le Model-Based Testing est principalement utilisé sur les niveaux de test les plus élevés (intégration, systèmes, acceptation), du fait de sa propension à abstraire la complexité des exigences fonctionnelles et des comportements attendus ;
 - Le Model-Based Testing est moins utilisé au niveau composant (test unitaire), mais il existe des approches du MBT fondées sur l'annotation du code.
- Types de test :
 - Le Model-Based Testing est en premier lieu utilisé pour le test fonctionnel avec des modèles MBT représentant le comportement attendu du système sous test et/ou de son environnement.
 - Des modèles MBT enrichis ou dédiés peuvent être utilisés pour le test non-fonctionnel (par exemple pour le test de sécurité, de performance ou de sûreté).

Les testeurs sont généralement en charge des modèles MBT, qu'ils peuvent partager avec des partie-prenantes du projet. Par exemple, les modèles MBT (ou des portions de ceux-ci) peuvent être partagés avec des partie-prenantes du Métier pour valider et vérifier les exigences. Les développeurs peuvent être intéressés par les modèles MBT en cas d'automatisation des tests, particulièrement lorsque les scripts de test automatisés sont utilisés en intégration continue.

Les activités spécifiques du MBT dans les cycles de développement de type séquentiel incluent :

- L'intégration du Model-Based Testing comme une composante de la stratégie de test du projet, tenant compte de l'évaluation du risque, du contexte et des objectifs de test du projet, et intégrant la traçabilité des exigences avec les éléments des modèles MBT.
- La réalisation de la modélisation MBT le plus tôt possible dans le cycle du projet :
 - pour favoriser la communication entre les partie-prenantes ;
 - pour permettre la détection, en amont du développement, des exigences mal formulées, incomplètes ou inconsistantes.
- L'adaptation du planning des activités et rôles du test doivent prendre en compte :
 - les nouveaux artefacts apportés par le MBT (par exemple les modèles et les critères de sélection de tests);
 - le suivi d'avancement des activités du MBT.

Les adaptations principales du MBT en contexte de développement agile (voir le syllabus ISTQB® Testeur agile [ISTQB_AT_FL_SYL]) sont les suivantes :

- Les modèles MBT sont développés de façon itérative et incrémentale en lien avec le contenu des itérations du cycle agile.
- Les 'User Story' font partie des bases de test. Chaque 'User Story' à couvrir est liée avec le modèle MBT pour automatiser la gestion de la traçabilité bidirectionnelle entre les 'User Story' et les tests.
- Le MBT peut être utilisé pour implémenter le développement guidé par les tests d'acceptation (ATDD – Acceptance Test-Driven Development).
- 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.

1.3.2 MBT et Ingénierie des exigences

Le Model-Based Testing facilite l'ingénierie des exigences sur les aspects suivants :

- Le MBT facilite la communication entre les équipes Métier, de développement et de test autour des comportements attendus du logiciel en fournissant des modèles graphiques MBT tels que des

modèles de processus métier utilisés pour la génération de tests, ou des machines à états. Ces modèles MBT aident les partie-prenantes à discuter et valider dans les détails les comportements attendus.

- Le MBT aide à clarifier et améliorer les exigences et/ou 'User Story' par le partage (en totalité ou en partie) des modélisations MBT avec les représentants du Métier et/ou les développeurs pour faciliter une compréhension partagée des comportements attendus de l'objet de test. Les tests générés à partir des modèles MBT comprennent des scénarios utilisateur qui peuvent être revus par les représentants Métier et/ou les développeurs.
- Le MBT facilite une validation des exigences en amont, y compris lorsque des changements sur ces exigences sont encore possibles.

L'ingénierie des exigences et le MBT sont synergiques. Les exigences et les risques associés sont des entrées pour les activités de modélisation et génération de tests en MBT, et cela permet d'automatiser la création et la maintenance de la matrice de traçabilité bidirectionnelle entre les exigences et les tests.

2 Modélisation MBT – 250 minutes.

Termes

Modèle de test

Objectifs de connaissance pour le chapitre 2

2.1 Modélisation MBT

- FM-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 9.1 – 'simple' signifie moins de 15 éléments de modèle)
- FM-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 9.2 – 'simple' signifie moins de 15 éléments de modèle)
- FM-2.1.3 (K2) Classifier un modèle MBT par rapport au sujet et au focus du modèle
- FM-2.1.4 (K2) Donner des exemples sur la relation entre un modèle MBT et les objectifs de test pour lesquels il est réalisé

2.2 Langages de modélisation pour le MBT

- FM-2.2.1 (K1) Donner des exemples de langage de modélisation utilisé couramment pour le Model-Based Testing
- FM-2.2.2 (K1) Citer des représentants de familles de langages de modélisation adaptés pour différents types de systèmes et objectifs de test du projet

2.3 Bonnes pratiques de la modélisation MBT

- FM-2.3.1 (K1) Donner les caractéristiques de qualité d'un modèle MBT
- FM-2.3.2 (K2) Décrire les erreurs classiques et les pièges des activités de modélisation MBT
- FM-2.3.3 (K2) Expliquer les avantages de lier les exigences et des informations relatives au processus de test au sein du modèle MBT
- FM-2.3.4 (K2) Expliquer la nécessité d'un guide pour la modélisation MBT
- FM-2.3.5 (K2) Fournir des exemples où la réutilisation de modèles (issus des phases d'analyse des exigences, ou du développement) est/n'est pas appropriée
- FM-2.3.6 (K1) Rappeler les types d'outils adaptés aux différentes activités de la modélisation MBT
- FM-2.3.7 (K2) Présenter la démarche itérative de développement de modèle MBT, de revue et de validation.

2.1 Modélisation MBT

La modélisation Model-Based Testing permet de formaliser ce qui doit être testé sur un projet, de faciliter la communication entre les parties-prenantes du projet, pour rassembler toutes les informations pour la conception. Les modèles MBT peuvent aussi faciliter la gestion des activités de test.

Un modèle MBT est développé dans l'objectif de produire (ou identifier) les cas de test. Le modèle MBT doit donc inclure l'ensemble des informations nécessaires à la génération des cas de test, pour permettre la couverture des objectifs de test, la définition pour chaque cas de test généré des pas de test et des résultats attendus et pour produire la matrice de traçabilité bidirectionnelle entre les exigences et les tests.

Concevoir un modèle MBT est une activité essentielle du Model-Based Testing. La qualité du modèle MBT a un impact déterminant sur la qualité de la production avec une approche MBT. Les modèles MBT sont en général interprétés par des outils (par exemple pour la génération des tests) et doivent respecter une syntaxe précise. 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

Tout modèle est exprimé avec un langage de modélisation spécifique. Le langage de modélisation définit en particulier les éléments de modélisation pouvant être utilisés et les règles de modélisation à respecter avec ce langage.

Il faut considérer les questions suivantes pour le choix d'un langage de modélisation MBT :

- Quelles caractéristiques de qualité de l'objet de test seront modélisées pour le test ?
Il s'agit généralement des fonctionnalités du système (permettant ainsi de produire des tests fonctionnels), mais les facteurs de performance et d'autres aspects non-fonctionnels tels que la sécurité peuvent aussi faire partie du modèle MBT. Pour conserver le modèle aussi simple que possible et le processus MBT efficace, il est recommandé de ne modéliser que les aspects du système ou de son environnement qui sont utiles pour la satisfaction des objectifs de test du projet.
- Quels langages de modélisation sont adaptés ?
Chaque modèle étant exprimé dans un langage spécifique, intégrant des éléments de modèle à utiliser et les règles de modélisation, le choix doit être adapté à la nature du système testé, aux objectifs de test et aux différentes contraintes du projet. Il existe une grande variété de langages de modélisation, pouvant exprimer par exemple la structure, les comportements, les données, les flux métier, les protocoles ou d'autres aspects d'un système ou de son environnement.
- Quel est le niveau d'abstraction approprié pour la modélisation ?
Une certaine abstraction du modèle facilite la maîtrise de la complexité et la focalisation sur les objectifs de test. Mais, plus le modèle sera abstrait et plus l'adaptation des tests produits avec ce modèle pour l'exécution demandera un effort. De plus, le niveau d'abstraction du modèle détermine les possibilités de partage du modèle avec d'autres parties prenantes.

Les réponses à ces questions et le choix de l'outillage MBT influenceront la mise en œuvre des activités de modélisation MBT.

Note : Deux langages (simplifiés) de modélisation sont fournis en section 9 pour les besoins de la formation associée à ce syllabus. La compréhension et l'utilisation de ces deux langages fait partie de l'examen de passage de la certification.

2.1.2 Sujet et focus du modèle MBT

Trois types de sujets de la modélisation sont considérés dans ce syllabus :

- Modèle du système
Un modèle du système décrit le système attendu. Les cas de test générés par ce type de modèle vérifient ainsi si le système sous test est conforme au modèle. Le diagramme de classes d'un système orienté objet ou un diagramme d'états-transitions d'un système réactif sont des exemples de modèles du système.

- **Modèle de l'environnement / Modèle d'usage**
 Un modèle d'environnement décrit l'environnement du système. Un modèle par chaîne de Markov décrivant l'usage du système sous test est un exemple de modèle d'environnement.
- **Modèle de test**
 Un modèle de test est un modèle d'un (ou de plusieurs) cas de test. Cela peut inclure le comportement attendu de l'objet de test et son évaluation. La description abstraite de cas de test ou la représentation graphique d'une procédure de test sont des exemples de modèles de test.

Un modèle MBT combine généralement plusieurs ou tous ces types de sujet. Par exemple, le modèle peut représenter l'objet de test, y compris des éléments de données de test, ainsi que l'usage de l'objet de test dans un contexte donné.

Le focus d'un modèle MBT peut être structurel, comportemental ou une combinaison des deux :

- Les modèles structurels décrivent la structure statique. Les diagrammes de classes (pour spécifier des interfaces) ou les arbres de classification (pour modéliser des données) sont des exemples de modèles structurels.
- Les modèles comportementaux décrivent des interactions dynamiques. Les diagrammes d'activités, les modèles de processus décrivant les activités et les flots métier, ainsi que les diagrammes états-transitions décrivant les entrées et sorties d'un système sont des exemples de modèles comportementaux

Un modèle MBT combine généralement des aspects structurels (par exemple pour la description des interfaces de l'objet de test) et des aspects comportementaux (par exemple pour les comportements attendus de l'objet de test).

2.1.3 La modélisation MBT dépend des objectifs de test

Lors du développement d'un modèle MBT, il est important de considérer les objectifs de test car ils déterminent le sujet et le focus du modèle. Des exemples d'objectifs de test associés à des caractéristiques de modèle MBT sont listés dans la table ci-dessous :

Objectif de test	Exemple de modélisation MBT	Sujet du modèle	Focus du modèle
Vérifier que le flot métier est implémenté correctement	Un modèle de processus métier décrivant le flot	Système	Comportemental
Vérifier que le système fournit des réponses correctes à des stimulations dans des états spécifiques	Une machine à états UML	Système	Comportemental
Vérifier la disponibilité d'une interface	La description de la structure de l'interface	Système	Structurel
S'assurer que l'objet de test est adapté aux usages effectifs	Un modèle d'usage décrivant le comportement de l'utilisateur	Environnement	Comportemental
Vérifier les configurations d'un système	Un modèle de données utilisant un arbre de classification	Données	Structurel

2.2 Langages de modélisation pour le MBT

2.2.1 Catégories principales de langages de modélisation pour le MBT

Les langages de modélisation sont définis de plusieurs façons :

- par les concepts que ces langages manipulent (aussi appelé la syntaxe abstraite des langages, pouvant être spécifiée textuellement ou à l'aide d'un méta-modèle);
- par leur syntaxe (aussi appelée la syntaxe concrète et souvent définie par des règles de grammaire);
- par leur sémantique (souvent définie par des règles statiques et sémantiques).

Les diagrammes sont des représentations graphiques de modélisation, tels que les diagrammes de classes, de séquences ou les machines à états UML.

Un grand nombre de langages de modélisation peut être utilisé pour le MBT. Pour sélectionner un langage bien adapté à son projet, un testeur doit connaître les principales caractéristiques qui différencient les langages de modélisations pour le MBT :

- Concepts de modélisation

Les langages de modélisation se différencient par l'ensemble des concepts qu'ils supportent. En fonction des objectifs de test du projet, les modèles MBT peuvent représenter les aspects structurels (décrivant par exemple l'architecture, les composants, les interfaces du logiciel), les aspects données (par exemple le format et la sémantique des données) ou les aspects comportementaux (par exemple les scénarios, interactions, exécutions – voir section 2.1.2).

- Formalisme

Le degré de formalisme d'un langage de modélisation va d'une formalisation limitée (en général plus facile à pratiquer) à une formalisation complète (plus rigide mais permettant une analyse formelle exhaustive). Au minimum, le langage de modélisation MBT doit posséder une syntaxe formalisée (par exemple incluant du texte structuré et des tables). Le langage peut aussi posséder une sémantique définie formellement.

- Formats de présentation

Les langages de modélisation peuvent utiliser différents formats allant du tout textuel au tout graphique, avec souvent une combinaison des deux. Les formats graphiques sont souvent plus conviviaux et faciles à lire, tandis que les modèles textuels sont souvent traités plus efficacement pour les outils ce qui facilite leur écriture et leur maintenance.

Les catégories de langages de modélisations suivantes sont à considérer :

- Les 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 d'un langage de modélisation MBT dépend des objectifs de test du projet et de la nature et caractéristiques du système testé. Ces éléments déterminent des critères de choix de langage qui guident le choix d'un outillage de Model-Based Testing supportant un langage de modélisation donné.

Voici quelques exemples d'adéquations entre un type de langage de modélisation et des caractéristiques du projet de test :

- Les diagrammes d'états-transitions pour représenter le comportement attendu d'un système de type contrôle-commande ;
- Les diagrammes d'activités ou modèles de processus en BPMN pour représenter des workflows pour le test de bout en bout d'un système d'information ;
- Les tables de décision ou les graphes causes-effets pour représenter les règles métier pour des tests au niveau système ;
- Les modèles de Features pour représenter les variantes du système dans le contexte d'une ligne de produits logiciels.

Les exigences non-fonctionnelles, par exemple dans le cas de systèmes critiques (pour la sûreté ou la sécurité) lorsque les exigences doivent être tracées avec le code et les tests, peuvent aussi être à prendre en compte dans le choix d'un langage de modélisation MBT. Les exigences non-fonctionnelles relatives à l'audit et la certification des processus sont aussi à considérer lorsqu'elles sont présentes.

2.3 Bonnes pratiques de la modélisation MBT

2.3.1 Caractéristiques de qualité des modèles MBT

La qualité des modèles MBT se reflète de façon directe dans les artefacts générés. Les caractéristiques de qualité à considérer pour les modèles MBT sont les suivantes :

- Qualité syntaxique (correction)
Le modèle MBT est correct vis-à-vis des règles syntaxiques (au niveau du langage de modélisation et des règles de modélisation) de telle façon que les cas et données de test générés sont eux-mêmes corrects et syntaxiquement bien formés.
- Qualité sémantique (validité)
Le contenu du modèle MBT est correct vis-à-vis de ce qu'il doit décrire. Les artefacts générés sont exploitables (les scripts de test peuvent être exécutés par un automate de test, et les cas de test manuels peuvent être exécutés par les testeurs sans produire d'échec dû à des tests incorrects).
- Qualité pragmatique (pertinence)

Le modèle MBT est adapté aux objectifs de test du projet et à l'outil de génération de tests utilisé de telle façon que les résultats de la génération satisfont les attentes.

Les outils MBT peuvent vérifier la syntaxe, et au moins partiellement la sémantique du modèle. Les revues de modèle peuvent permettre de vérifier la qualité sémantique et pragmatique du modèle. Les simulateurs de modèles complètent les techniques statiques de vérification en réalisant des vérifications dynamiques de la qualité du modèle.

Il est recommandé de développer le modèle MBT de façon incrémentale et de générer les tests (et de les exécuter) régulièrement, permettant de vérifier le modèle et les tests obtenus de façon régulière lors du projet de test MBT.

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

Les débutants en MBT doivent être attentifs aux erreurs et pièges courants dans la modélisation MBT, tels que :

- Être trop détaillé (ou parfois pas assez) au niveau du modèle MBT au regard des objectifs de test du projet (erreur dans le niveau d'abstraction mis en œuvre).
- Réaliser un modèle MBT qui essaie de couvrir une multitude d'objectifs différents.

Les modèles MBT doivent être focalisés sur les aspects utiles et nécessaires à la couverture des objectifs de test. Il est préférable d'utiliser deux modèles MBT pour couvrir certains aspects (par exemple un diagramme d'états pour vérifier l'implémentation du système et un diagramme d'activités pour valider un workflow), plutôt qu'un seul qui chercherait à tout couvrir d'un coup.

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

Mettre en place la traçabilité entre les exigences, les modèles MBT et les tests générés est une bonne pratique du Model-Based Testing. Relier les éléments du modèle MBT avec les exigences :

- Facilite la revue du modèle MBT (par rapport à la couverture des exigences).
- Permet de produire automatiquement la matrice de traçabilité entre les tests générés et les exigences.
- Permet de réaliser une sélection des tests à partir de la couverture des exigences, et de décider la priorisation d'exécution des tests en fonction des priorités associées aux exigences.
- Permet de mesurer et monitorer la couverture des exigences par les tests générés.
- Permet aux parties-prenantes (par exemple les testeurs et les manageurs de test) d'analyser l'impact de changements et de déterminer le périmètre du test de régression.

D'un point de vue technique, il y a plusieurs façons de réaliser la liaison entre les exigences et les éléments de modèles. Par exemple, un symbole graphique ou un mot clé spécifique peut être utilisé.

Le testeur peut souhaiter ajouter d'autres informations que les exigences au sein du modèle MBT, tels que :

- Le contenu détaillé des spécifications des procédures de test (séquences des actions pour l'exécution des tests).
- Les parties de scripts de test (appels de fonctions ou mots d'action).
- Les risques (en lien avec les caractéristiques fonctionnelles et/ou non-fonctionnelles ou avec la gestion du projet).
- Les priorités (des fonctionnalités et/ou des tests).
- Les durées d'exécution des tests.

- Les équipements nécessaires à la réalisation des tests et les paramètres pour tester différentes configurations du système.

Ces informations additionnelles facilitent l'intégration du Model-Based Testing dans le processus de test et contribuent à la gestion des tests sur plusieurs aspects :

- Les risques et priorités peuvent être intégrés au modèle MBT, permettant ainsi d'être tracés dans les tests générés, pour par exemple être utilisés pour prioriser l'exécution des tests.
- D'autres contraintes et objectifs peuvent être intégrés au modèle MBT et utilisés pour les activités de planification et gestion des activités de test.
- Lorsque les interfaces (par exemple les interfaces graphiques) ne sont pas encore stabilisées dans les spécifications, le modèle MBT peut se focaliser sur les exigences fonctionnelles. Ainsi, les scripts à base de mots-clés (aussi appelés mots d'action) peuvent être générés plus tôt dans le processus, et les phases d'automatisation démarrer plus tôt. Lorsque les interfaces sont spécifiées, les scripts automatisés sont prêts et seulement la couche d'adaptation reste à implémenter (voir chapitre 4).

2.3.4 Guide de modélisation MBT

Le guide de modélisation MBT dans une organisation documente la façon de concevoir, développer et lire les modèles MBT. Cela permet :

- De faciliter une compréhension commune des modèles MBT par toutes les parties-prenantes.
- De faciliter la conformité du modèle avec ses exigences syntaxiques en phase avec le processus de test, le domaine d'application, les spécificités de l'organisation et de l'outillage.
- De limiter ou d'étendre le langage de modélisation utilisé (par exemple en définissant un sous-ensemble d'UML ou une signification spécifique de certains éléments de modélisation)
- De forcer une syntaxe et une sémantique similaire pour les modèles MBT quel que soit leur auteur.
- D'enseigner les bonnes pratiques de modélisation.
- De faciliter la maintenance et la réutilisabilité des modèles 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 pour le MBT de modèle de conception ou d'exigences existants

Les modèles MBT peuvent être développés en partant de la feuille blanche à partir des bases de test (telles que les exigences) ou en réutilisant des modélisations existantes, de conception ou d'exigences (par exemple des modèles de processus métier, ou des diagrammes d'activités). Mais, la réutilisation peut poser certaines difficultés. Par exemple, un modèle de conception de type diagramme d'états-transitions peut servir pour tester un système de contrôle/commande, mais un diagramme de classes d'implémentation n'est pas adéquat pour le test du workflow métier.

Réutiliser les modèles de conception existant peut aussi créer un problème de propagation d'erreurs (aussi appelé problème de source de connaissance unique). Ainsi, les erreurs de conception se propagent dans le modèle MBT (donc dans les tests). Dans ce cas, les tests ne sont pas aussi indépendants du développement du système et de l'implémentation qu'ils pourraient l'être. Il est plutôt recommandé de développer des modèles MBT indépendant des modèles d'implémentation. Cela favorise la séparation des responsabilités et facilite l'indépendance du test.

Lorsqu'ils sont utilisés sans modification, les tests issus d'un modèle de conception vérifient si l'implémentation correspond aux modèles de conception, plutôt qu'être des tests de validation par rapport aux besoins.

Pour la validation, le testeur peut partir de modélisations d'exigences et l'enrichir avec les aspects pertinents pour le test. Transformer ainsi en modèle MBT, les modélisations de conception sont enrichies d'éléments de modèle pour tester des scénarios spécifiques et de cas d'erreur. In fine, le modèle reflète plus les aspects de test que la conception initiale.

Le fait que les modèles existants aient été réalisés par des auteurs non testeurs peut aussi faire que ces modélisations ne respectent pas les exigences, contraintes et bonnes pratiques d'une approche MBT. Ceci peut ensuite poser des problèmes lors de leur réutilisation pour le MBT, tels que l'explosion du nombre de cas de test par exemple

2.3.6 Outillage des activités de modélisation MBT

Les éditeurs de modèles sont utilisés pour leur écriture. Il peut s'agir d'outils de modélisation spécifiques ou de diagrammes (pour les modèles graphiques), ou un éditeur de texte et/ou de scripts pour les modèles textuels. En principe, il est possible de réaliser des modèles MBT sans outillage spécifique, mais l'utilisation d'outils simples de dessin empêche le traitement ultérieur de modèles MBT.

Un éditeur de modèle MBT peut fournir des éléments de modélisation prédéfinis, et des vérifications syntaxiques et sémantiques. Les outils qui permettent d'exécuter différents chemins sur les modèles MBT sont appelés simulateurs de modèles.

Les générateurs de tests permettent d'obtenir automatiquement les cas de test, scripts de test et données de test à partir du modèle MBT.

2.3.7 Modélisation, revue et validation itératives

La modélisation de fonctionnalités complexes au travers d'un ensemble de diagrammes peut faire qu'une simple revue du modèle soit insuffisante. Le testeur doit s'assurer que les tests issus du modèle sont conformes aux attentes. Le développement itératif des modèles, leur revue, ainsi que la revue fréquente des tests générés, d'abord par le testeur mais aussi par les pairs et parties-prenantes, permettent de gérer cette complexité.

Ainsi, appliquer ces bonnes pratiques permet :

- aux testeurs de valider avec les autres parties-prenantes leur point de vue sur les aspects à tester ;
- aux testeurs de corriger les erreurs et omissions le plus tôt possibles dans les modèles MBT (avant même l'exécution des tests) ;
- aux testeurs d'identifier et d'informer sur les exigences incomplètes ou inconsistantes ;
- aux managers de test de mieux gérer les risques du projet ;
- à l'équipe de réduire le temps global pour finaliser les activités de modélisation MBT au sein du processus de test.

Le développement itératif des modèles MBT est associé à la génération itérative des tests. Cela signifie que chaque fois qu'un ajout est réalisé dans le modèle MBT, la génération de tests est aussi mise à jour. La génération de tests constitue une simulation du modèle MBT, fournissant un feedback sur les comportements modélisés du système sous test.

3 Critères de sélection pour la génération des tests – 205 minutes.

Termes

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

- FM-3.1.1 (K2) Classifier les différentes familles de critères de sélection de tests utilisées pour la génération de tests à partir de modèles
- FM-3.1.2 (K3) Générer des cas de test à partir d'un modèle MBT pour satisfaire des objectifs de test dans un contexte donné
- FM-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.
- FM-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

A partir d'un même modèle MBT, différentes suites de tests peuvent être générées. L'utilisation de critères de sélection de tests facilite, pour le testeur, la sélection de cas de test adaptés aux objectifs de test du projet.

Les outils de génération automatique de tests à partir de modèles automatisent la mise en œuvre les critères de sélection de tests et jouent un rôle important pour l'efficacité (i.e. la productivité) du MBT.

Les critères de sélection de tests pour le MBT ont été largement étudiés dans la littérature sur le sujet (voir les références [Utt07], [Wei09], [Zan11] en section 8).

Ce syllabus introduit 6 familles de critères de sélection de tests. Certaines familles concernent la couverture d'items spécifiques et d'autres familles introduisent des critères variés liés à la gestion de projet, à des scénarios spécifiques ou à des aspects aléatoires.

3.1.1 Familles de critères de sélection de tests

Les critères de sélection de tests fondés sur la couverture visent à guider la génération de tests par la couverture d'éléments du modèle MBT. Ces éléments de couverture peuvent correspondre :

- Aux exigences liées au modèle MBT
La couverture des exigences implique la couverture des éléments de modèles reliés aux exigences à couvrir. La couverture complète des exigences correspond à l'ensemble des cas de test qui couvrent la totalité des exigences sélectionnées (chaque exigence est couverte par au moins un test).
- Aux éléments du modèle MBT
La couverture du modèle s'appuie sur la structure interne du modèle MBT. Le testeur, le manager de test et/ou les autres parties-prenantes définissent les éléments de couverture pour sélectionner les tests couvrant les éléments de modélisation considérés. Ces éléments dépendent du langage de modélisation, par exemple :
 - Les états, les transitions et les décisions dans un diagramme d'états-transitions (voir le test états-transitions dans le syllabus ISTQB® niveau fondation [ISTQB_FL_SYL]).
 - Les activités et les branchements dans les modèles de processus métier.
 - Les conditions et les actions dans les tables de décision.
 - Les instructions et les conditions dans les modèles textuels.
- A la couverture des données et types de données
Ces critères sont liés aux techniques de conception de test (voir le syllabus ISTQB® niveau fondation [ISQB_FL_SYL]) tels que:
 - Le partitionnement par classes d'équivalence.
 - L'analyse des valeurs aux limites.Cela inclut aussi des heuristiques de test combinatoire tel que le pair-wise (voir le syllabus [ISTQB_ATA_SYL]).

Les autres familles de critères de sélection de tests sont :

- Les critères fondés sur l'aléatoire
Ces critères de sélection de tests s'appuient sur un parcours (plus ou moins aléatoire) du modèle MBT. Avec la sélection fondée sur l'aléatoire, les différentes alternatives ont la même probabilité d'être couvertes. La sélection stochastique prend en compte les différentes alternatives avec potentiellement des probabilités différentes.
- Les critères fondés sur les scénarios et sur les patterns
La sélection des cas de test est basée sur des scénarios et/ou des patterns prédéfinis. Un scénario peut être un cas d'utilisation du système (voir le test des cas d'utilisation dans le syllabus ISTQB® niveau fondation [ISTQB_FL_SYL]) ou une 'user story' définie dans un contexte agile (voir le syllabus agile au niveau fondation [ISTQB_AT_FL_SYL]). Un pattern peut correspondre à un scénario partiellement défini, qui, appliqué sur le modèle MBT, permet de produire plusieurs ou une multitude de cas de test.
- 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

En pratique, le testeur doit généralement mettre en œuvre une combinaison de critères de sélection de tests pour obtenir les tests requis pour satisfaire les objectifs de test définis. Un exemple de combinaison est la couverture des exigences et d'éléments de modèles, ou la mise en œuvre de critères de type scénarios/patterns avec une sélection fondée sur les données.

En plus des objectifs de test, l'application des critères de sélection de tests dépend des éléments suivants :

- Les mécanismes fournis par l'outil MBT utilisé.
- La conception du modèle MBT.
- L'expérience du testeur avec les critères de sélection mis en œuvre.

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

3.1.3 Exemples de critères de sélection de tests

Voici quelques exemples de critères de sélection de tests en relation avec le langage de modélisation utilisé (voir aussi [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 chacune 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
 - Couverture de chaque action
 - Couverture de chaque règle
- Sur les domaines de données définis dans la partie structurelle du modèle MBT :
 - Couverture des partitions d'équivalence (par exemple avec 1 représentant par classe)
 - Couverture des valeurs aux limites (par exemple en prenant les valeurs aux limites des intervalles de données numériques)
 - Mise en œuvre de pair-wise sur les domaines de données définis (en permettant ainsi d'assurer que chaque combinaison de paires de valeurs est produite)
- Sur les modèles textuels :
 - Couverture des instructions (100% = chaque instruction exécutable est couverte)
 - Couverture des décisions (100% = chaque décision est couverte)

- Couverture des conditions dans les décisions (100% = chaque résultat de conditions et chaque résultats de décision sont couvertes dans la suite de test).
- Couverture des décisions multiples (chaque combinaison de conditions dans chaque décision est couverte – ce critère peut conduire à un grand nombre de tests générés).

3.1.4 Relation avec les techniques de conception de test du niveau fondation

Le Model-Based Testing implémente les techniques standards de conception de test telles que définies dans le niveau fondation de la certification ISTQB® : test de transition d'états, partition d'équivalence, analyse des valeurs limites, test par tables de décision et test de cas d'utilisation.

De plus, les modèles MBT peuvent combiner différentes représentations intégrant ces éléments (par exemple les tables de décision ou autres diagrammes additionnels).

3.2 Application des critères de sélection de tests

3.2.1 Niveau d'automatisation de 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 la production des tests peut aussi être réalisée sans outils. Les différents niveaux d'automatisation de la génération de tests MBT sont les suivants :

- Génération manuelle des tests – les tests sont générés manuellement à partir du modèle MBT, en suivant par exemple les chemins dans le modèle et en écrivant les tests correspondant. Cette approche manuelle dénote une faible maturité et ne permet pas d'obtenir tous les bénéfices du MBT en termes d'efficacité et d'efficience (voir section 1.1).
- Génération automatique des tests – les artefacts de sortie du processus MBT sont générés automatiquement avec l'outil MBT et peuvent être utilisés sans traitement postérieur.
- Génération semi-automatique des tests – il s'agit d'une solution intermédiaire, qui peut aussi s'appliquer lorsqu'un outil est utilisé, pour laquelle des étapes manuelles sont nécessaires en cours ou à la fin du processus MBT, par exemple pour sélectionner des tests spécifiques.

3.2.2 Analyse de la mise en œuvre des critères de sélection de tests

Chaque type de critère de sélection de tests possède des avantages et inconvénients. Le choix d'un type de critère dépend des objectifs de test. L'analyse ci-dessous dénote les avantages (+) et les inconvénients potentiels (-) de chaque type de critère :

- Couverture des exigences
 - (+) Souvent obligatoire du fait des exigences sur le processus de test
 - (-) 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
 - (+) Permet de déterminer ce qui est exactement couvert dans le modèle et facilite la mesure de la complétude de la couverture dans les termes de la modélisation
 - (-) Certains critères de sélection structurels peuvent conduire à une explosion du nombre de cas de test générés (par exemple la couverture de chemins dans un diagramme d'activités en présence de boucles).
- Critères de sélection orientés sur la couverture de données
 - (+) Souvent requis pour couvrir les partitions d'équivalence des domaines de données
 - (-) Peut conduire à une forte combinatoire

- Critères fondés sur l'aléatoire
 - (+) Utile pour produire des tests non prévus
 - (-) Les algorithmes de génération de tests fondés sur l'aléatoire peuvent générer de longues séquences de test sans réel sens d'un point de vue métier
- Critères fondés sur les scénarios et les patterns
 - (+) Ces critères permettent de sélectionner des cas d'utilisation (par exemple pour le test de régression)
 - (-) Ils nécessitent un effort supplémentaire pour définir et maintenir les scénarios
- Critères fondés sur des caractéristiques du projet
 - (+) Ces critères facilitent la gestion du projet de test
 - (-) Ils nécessitent la gestion d'informations spécifiques dans le modèle

3.2.3 Bonne pratique de la mise en œuvre de critères de sélection de tests en 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.

Deux approches différentes peuvent être utilisées pour combiner les critères de sélection de tests :

- La composition de critères – la suite de test générée contient seulement les tests satisfaisant l'ensemble des critères appliqués (intersection).
- L'addition de critères – la suite de test générée contient tous les tests 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 le 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.

Termes

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

- FM-4.1.1 (K2) Expliquer la différence entre les tests abstraits et concrets dans le contexte MBT
- FM-4.1.2 (K2) Expliquer la différence entre différents types d'exécution des tests dans le contexte MBT
- FM-4.1.3 (K3) Réaliser une mise à jour du modèle MBT et de la génération de tests prenant en compte des changements dans les exigences, l'objet de test ou les objectifs de test

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

- FM-4.2.1 (K2) Expliquer quel type d'adaptation peut être nécessaire à l'exécution des tests dans le contexte MBT

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

Lorsque la suite de tests a été générée, l'étape suivante consiste à exécuter les cas de test. Cette exécution peut être réalisée manuellement ou peut être automatisée en utilisant un environnement d'exécution des tests qui permet leur exécution automatique et le recueil des résultats de l'exécution. Quel que soit le mode d'exécution, il doit y avoir un lien entre le modèle et l'exécution des tests.

4.1.1 Notion de tests abstraits et concrets en MBT

La génération de tests en MBT peut produire des cas de test abstrait (aussi appelé cas de test de haut niveau) ou des cas de test concrets (aussi appelé cas de test de bas niveau) :

- Les cas de test abstraits sont des cas de test ne définissant pas les valeurs concrètes (de niveau implémentation) des données d'entrée des tests et des résultats attendus. Ils peuvent aussi ne pas définir les étapes de test dans tout leur détail.
- Les cas de test concrets sont des cas de test définissant des valeurs concrètes (de niveau implémentation) des données d'entrée des tests et des résultats attendus, ainsi qu'une définition très détaillée des étapes de test.

Les modèles MBT peuvent être réalisés à différents niveaux d'abstraction, permettant la production de différents types d'artefacts en sortie, et aussi pouvant concerner différents acteurs du processus de test. Par exemple :

- Les cas de test abstraits peuvent être utilisés lors de revue par les analystes métier.
- Les cas de test concrets peuvent être exécutés directement par les testeurs.

Les cas de test abstraits générés en MBT fournissent des informations concernant les conditions de test, les données d'entrée du test et les résultats attendus sans définir totalement les données concrètes pour l'exécution. Les informations sont ainsi définies à un niveau plus abstrait, par exemple :

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

Passer des cas de test abstraits à des cas de test concrets prêts pour l'exécution (que ce soit de façon manuelle ou automatisée) nécessite que les tests abstraits soient complets aux trois niveaux suivants :

- des actions de test complètement définies ;
- des données d'entrée du test qui soient concrètes et complètes ;
- des valeurs concrètes pour les résultats attendus.

Ce complément d'information peut être défini au sein du modèle MBT (par exemple au travers de la documentation du modèle définissant les actions / vérification et les données concrètes) ou être réalisé en dehors du modèle MBT (par exemple en utilisant des tables de données liant les données abstraites avec des valeurs de données concrètes).

4.1.2 Différents types d'exécution des tests

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

Pour l'exécution manuelle des tests, les testeurs exécutent les tests qui ont été générés par l'outil MBT. Ces cas de test doivent alors être générés dans un format adapté à l'exécution manuelle des tests. Il est aussi utile d'exporter ces tests dans l'outil de management des tests. L'exécution des tests et la gestion des défauts sont alors réalisées de façon synchronisée dans le cadre du processus de test ISTQB.

Pour l'exécution automatique, les cas de test doivent être générés dans un format exécutable. On considère trois approches principales pour gérer l'exécution automatique de tests à partir de la génération de cas de test abstraits (voir [Utt07] – chapitre 8) :

- L'adaptation des tests – avec cette approche, un code dit de *couche d'adaptation des tests* est développé pour faire le pont entre le niveau d'abstraction des tests générés et leur concrétisation pour l'exécution. Cette couche d'adaptation s'intercale au-dessus du système sous test de façon similaire à l'approche du test automatisé à partir de mots-clés.
- La transformation des tests – avec cette approche, les tests MBT générés sont directement convertis en scripts de test exécutables pour l'automatisation (la couche d'adaptation n'est pas nécessaire dans ce cas).
- L'approche mixte – il s'agit de combiner les deux approches précédentes de telle façon que les cas de test abstraits sont transformés en cas de test concrets (scripts de test), pour lesquels l'exécution s'appuie sur la couche d'adaptation développée au-dessus du système sous test.

Une fois générés, les scripts de test automatisés sont exécutés par un outil d'automatisation. Les outils MBT peuvent convertir les cas de test en scripts de test dans un langage de script adapté à l'outil d'automatisation. L'outil MBT peut aussi publier les scripts de test automatisés ainsi générés dans l'outil de management des tests.

Il y a deux façons d'exécuter automatiquement des tests dans le contexte MBT :

- Le MBT hors-ligne – les scripts automatisés sont d'abord générés (en incluant les résultats attendus) puis ensuite exécutés.

- MBT en-ligne (aussi appelé 'à-la-volée') – la génération et l'exécution des tests sont réalisées simultanément. Ainsi, chaque pas de test est généré suite à l'exécution de pas de test précédents. Les résultats d'exécution peuvent ainsi influencer le chemin pris dans le modèle.

4.1.3 L'impact de changements sur les artefacts MBT

Les changements sont inévitables dans un projet logiciel. Les types suivants de changement sont à considérer :

- Des changements dans les exigences, sur l'objet de test ou son environnement peuvent conduire à modifier les éléments suivants :
 - Le modèle MBT
 - Les actions
 - Les conditions et résultats attendus
 - Les données de test
 - Les critères de sélection de tests
 - La spécification (et le code s'il existe) de la couche d'adaptation des tests
- Des changements dans les interfaces de l'objet de test, mais pas dans les exigences fonctionnelles (par exemple des évolutions mineures sur l'IHM sans impact sur les comportements du système) peuvent conduire à modifier l'élément suivant :
 - La spécification (et le code s'il existe) de la couche d'adaptation des tests
- Des changements dans les objectifs de test ou dans les conditions de test peuvent conduire à modifier les éléments suivants :
 - Le modèle MBT
 - Les critères de sélection de tests
 - La spécification (et le code s'il existe) de la couche d'adaptation des tests

La gestion des évolutions sur les artefacts MBT doit s'appuyer sur un processus défini qui inclut l'analyse d'impact des évolutions, l'exploration des options de prise en compte des changements, l'application de ces changements sur les artefacts et les activités de revue.

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

Dans le cas d'une exécution manuelle des tests générés, l'adaptation pour l'exécution consiste à faire le lien entre actions et données abstraites et les interfaces concrètes du système et données de test concrètes. Cela peut être réalisé sous la forme de documentation dans le modèle. 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. Cette adaptation fournit des procédures de test exécutables suffisamment documentées pour être mises en œuvre directement en test manuel.

Dans le cas de l'automatisation de l'exécution des tests, l'adaptation est un processus qui vise à intégrer les tests générés avec l'outillage d'exécution automatique des tests sur la base de la spécification de la couche d'adaptation. Ce processus met en œuvre les bonnes pratiques du test automatisé piloté par mots-clés et/ou du test automatisé piloté par les données.

Concernant le test piloté par les mots-clés, ceux-ci sont définis dans le modèle MBT et mis en œuvre dans les tests générés. Pour obtenir des scripts de test totalement exécutables, les activités suivantes doivent être mises en œuvre :

- 1) Exportation des cas de test dans le langage de l'outil d'automatisation. Cela peut être réalisé manuellement ou automatiquement en utilisant un module d'export des cas de test vers les scripts automatisés (ce module pouvant être intégré à l'outil MBT).
- 2) Implémentation des mots-clés, en s'appuyant sur la spécification de la couche d'adaptation, dans le langage d'exécution automatique des tests. Cette implémentation correspond au code de la

couche d'adaptation. Les testeurs en charge de l'automatisation, ou les développeurs, peuvent réaliser l'implémentation de la couche d'adaptation.

Concernant le test automatisé piloté par les données, le modèle MBT décrit les données d'entrée et les résultats attendus de façon abstraite (par exemple au travers de partitions d'équivalence). Les cas ou scripts de test générés utilisent ces données abstraites en paramètres d'entrée et en résultats attendus des actions de test. Pour obtenir des scripts de test totalement exécutables dans une approche pilotée par les données, les activités suivantes doivent être mise en œuvre :

- 1) Fournir les données de test et résultats attendus concrets nécessaires pour spécifier la couche d'adaptation des tests. Ces données peuvent être gérées dans une table ou un tableur.
- 2) Lier les données abstraites formalisées dans le modèle MBT avec les données concrètes pour l'exécution dans l'environnement d'automatisation des tests (outil d'automatisation et harnais de test).

Chaque script de test suppose des conditions initiales spécifiques. Pour permettre d'enchaîner automatiquement l'exécution automatique des scripts de test, il est nécessaire d'assurer que ces conditions initiales sont positionnées correctement avant l'exécution de chaque script :

- Soit en établissant les conditions du script suivant au niveau de chaque script de test de telle façon que les tests puissent s'enchaîner (ce qui n'est pas toujours possible).
- Soit en établissant les préconditions au début de chaque script de test de façon indépendante des autres scripts.

L'adaptation des tests doit être préparée dès la phase de modélisation MBT, par exemple en élaborant une spécification de la couche d'adaptation en parallèle de la création des éléments du modèle MBT.

5 Évaluation et déploiement d'une approche MBT – 60 minutes.

Termes

aucun

Objectifs de connaissance pour le chapitre 5

5.1 Évaluer un déploiement MBT

- FM-5.1.1 (K2) Décrire les facteurs de ROI de l'introduction du MBT dans le processus de test
- FM-5.1.2 (K2) Expliquer comment les objectifs d'amélioration du processus de test influencent la mise en œuvre de l'approche MBT
- FM-5.1.3 (K1) Rappeler les métriques et indicateurs clés de performance pour mesurer la progression et les résultats des activités du MBT

5.2 Piloter le déploiement d'une approche MBT

- FM-5.2.1 (K1) Rappeler les bonnes pratiques de la gestion des tests, de la gestion du changement et de travail collaborative lors d'un déploiement MBT
- FM-5.2.2 (K1) Rappeler les facteurs de coût du MBT
- FM-5.2.3 (K2) Donner des exemples d'intégration de l'outillage MBT avec les outils de gestion de configuration, de gestion des exigences, de management des tests et d'automatisation.

5.1 Évaluer un déploiement MBT

Le déploiement d'une approche MBT dans une organisation suit généralement le cycle classique d'adoption d'un produit ou d'une technologie en cinq étapes :

1. La prise de conscience – cette étape consiste à établir des objectifs d'amélioration du processus de test et à identifier le Model-Based Testing comme une technologie pertinente pour satisfaire tout ou partie de ces objectifs. L'identification des améliorations possibles est un facteur de motivation pour faire évoluer les pratiques de l'équipe de test.
2. La prise de connaissance – cette étape consiste à mieux comprendre le MBT.
3. L'évaluation – cette étape consiste en l'analyse des caractéristiques principales des approches MBT et de leur mise en œuvre dans le contexte spécifique de l'organisation et/ou du projet.
4. L'expérimentation – cette étape consiste à définir des indicateurs clés de performance (KPI) et à réaliser un projet pilote pour mesurer les améliorations obtenues.
5. L'adoption – cette étape consiste à réaliser le pilotage et la gestion du changement induit par l'introduction du MBT avec l'amélioration des compétences de l'équipe et l'évolution des pratiques dans l'organisation en lien avec la mise en œuvre du MBT.

Ainsi, l'estimation du retour sur investissement (ROI) et la capacité à évaluer l'impact du MBT sur les performances des projets de test (sur la base de KPI) font partie du cycle d'adoption.

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

Le MBT induit des coûts spécifiques dans la phase d'introduction et de mise en œuvre qui sont contrebalancés par les bénéfices suivants pour obtenir le ROI :

- Résultats nets financiers positifs dus aux gains sur le processus de test au fur et à mesure des projets réalisés.
- Amélioration de la qualité des tests et impact 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.

Les coûts d'introduction et d'opération sont présentés en section 5.2.2.

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 :

- L'amélioration des méthodes de conception des tests et de la cohérence de cette conception.
- L'amélioration sur suivi de la couverture des tests et ainsi de la qualité de la conception des tests.
- L'apport du MBT pour les activités du processus de test telles que la priorisation des tests et l'assurance qualité de la conception des tests.
- L'amélioration de la traçabilité par une amélioration du contenu et de l'organisation des tests.

5.1.2 Objectifs d'amélioration du processus de test et mise en œuvre 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 en fonction de leur priorité. Le tableau suivant propose des exemples d'objectifs d'amélioration et indique comment l'approche MBT peut permettre de les atteindre.

Objectifs d'amélioration	Apports du MBT	Mise en œuvre
Améliorer la qualité du test	<ul style="list-style-type: none"> Méthode de conception de test améliorant la couverture des tests Traçabilité (permettant la mise en place de métriques de couverture et de meilleures capacités d'impact) Plus grande automatisation du processus (évitant des erreurs) 	<ul style="list-style-type: none"> Séparation des modèles de développement et de la modélisation MBT (favorisant le focus sur l'amélioration des tests et leur indépendance) Niveau d'abstraction clairement défini en phase avec le plan de test Haut degré du processus d'automatisation comprenant la génération des artefacts de test et de leur exécution diminuant le risque d'erreurs
Réduire les efforts	<ul style="list-style-type: none"> Automatisation du processus de test Traçabilité (mise en œuvre de façon automatisée) 	<ul style="list-style-type: none"> Partage et réutilisation des modèles MBT Haut niveau d'automatisation du processus de test par génération automatique des artefacts de test (cas et scripts de test, données de test, matrice de traçabilité, ...) Critères de sélection de tests bien définis pour générer les suites de tests permettant une exécution efficace et économe en moyens.
Améliorer la communication	<ul style="list-style-type: none"> Utilisation d'une approche MBT adéquate, compréhensible et utilisable pour adapter le niveau d'abstraction en fonction des parties-prenantes du projet 	<ul style="list-style-type: none"> Niveau d'abstraction adapté aux parties-prenantes (par exemple de haut niveau et orienté métier pour les analystes métier et orienté actions de test pour les testeurs)

Une combinaison des objectifs d'amélioration peut parfois conduire à des exigences contradictoires pour la mise en œuvre du MBT. Cela peut être traité en utilisant des modélisations différentes en fonction des objectifs (voir la section 2.1.3 de ce syllabus).

5.1.3 Métriques et principaux indicateurs de performance

L'introduction du MBT dans une organisation devrait être basée sur des objectifs clairs, des métriques et des indicateurs de performance, permettant ainsi de mesurer les progrès réalisés et les résultats atteints par la mise en œuvre du Model-Based Testing.

Les exemples de métriques et d'indicateurs de performance suivants sont à considérer :

- Le nombre d'exigences prises en charge 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 et de scripts de test générés, et le nombre de cas et de scripts de test générés par personne.jour.
- Le nombre de défauts détectés dans les exigences durant les activités de modélisation MBT.

- Le niveau de réutilisation des éléments de modèle MBT d'un projet à l'autre.
- Le niveau d'utilisation du modèle MBT par les parties-prenantes du projet (au niveau du métier, du développement et des tests).
- Le pourcentage d'amélioration de la productivité de la conception des tests par rapport à l'approche sans MBT (des tests moins coûteux).
- Le pourcentage d'augmentation de la détection d'anomalies par l'amélioration, due au MBT, de la conception des tests (des tests plus productifs).

Définir et mesurer des métriques et des indicateurs de performances fait partie des bonnes pratiques dans le déploiement d'une approche MBT.

5.2 Piloter le déploiement d'une approche MBT

5.2.1 Bonnes pratiques du déploiement MBT

L'introduction du MBT dans le processus de test avec ses artefacts propres et l'outillage associé doit être parfaitement intégrée avec le processus de développement de façon générale. C'est aussi le cas au niveau de l'intégration de l'outillage MBT dans la chaîne outillée du développement, par exemple dans l'intégration avec le processus d'ingénierie des exigences et son outillage.

Une intégration de qualité est un facteur de succès de l'introduction du MBT dans une organisation. Cela implique :

- La mise en gestion de configuration de l'ensemble des artefacts utilisés ou produits par le MBT :
 - les bases de test ;
 - les modèles MBT et les critères de sélection de tests ;
 - les cas et scripts de test générés ;
 - la spécification et le code de la couche d'adaptation.

Dans le processus de développement, les artefacts, tels que les livraisons logicielles et les éléments du déploiement, sont placés en gestion de configuration. Ceci permet de réaliser un processus de développement et de déploiement opérationnel. Dans ce contexte, les artefacts MBT doivent aussi être placés en gestion de configuration et intégrés dans le processus de développement existant.

- Intégration de la génération de test MBT au sein de 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 l'ingénierie des exigences et la gestion des besoins
Les exigences et les besoins exprimés (par exemple les 'user story' dans le contexte agile – voir [ISTQB_AT_FL_SYL]) doivent être validés lors du développement. Par exemple, dans un processus agile de développement, le processus de test doit être synchronisé avec le processus de gestion de la liste des besoins. Si l'approche MBT est utilisée pour tester les besoins, les artefacts MBT doivent pouvoir être tracés dans le système de gestion de la liste de besoins et permettre de savoir "quel cas de test issu de quelle version de modélisation MBT avec quel objectif de test a été mis en œuvre".

5.2.2 Facteurs de coût du MBT

Le tableau suivant présente les coûts initiaux et opérationnels d'un déploiement MBT au travers des différentes étapes du processus de test.

Coûts initiaux (au niveau global de l'organisation et au niveau du projet) :

Coûts initiaux au niveau de l'organisation	Coûts initiaux au niveau du projet
<ul style="list-style-type: none"> • Vérifier les ressources et compétences existantes pour l'introduction du MBT • Évaluer différentes approches et outils MBT • Mettre en place le processus MBT • Intégrer le MBT avec la gestion des exigences, la gestion des tests et l'intégration continue • Automatiser et intégrer les rapports de test MBT • Mettre en place la gestion et l'archivage des artefacts MBT • Définir un guide général du processus et de la modélisation MBT • Réaliser la formation et le coaching MBT • Prendre en compte le coût des licences de l'outillage MBT 	<ul style="list-style-type: none"> • Créer (si besoin) un guide du processus et de la modélisation MBT spécifique au projet • Développer la modélisation MBT initiale • Reprendre les artefacts existant utiles pour le MBT (par exemple, reprendre des tests existant pour les transformer en modèle MBT) • Migrer les modèles MBT réutilisés

Coûts opérationnels récurrents de mise en œuvre du MBT :

Activité de test considérée	Coûts opérationnels récurrents
<ul style="list-style-type: none"> • Général 	<ul style="list-style-type: none"> • Coût de licence des outils (en fonction du mode de licence) et/ou coût de maintenance des outils • Formation et coaching des nouveaux membres de l'équipe
<ul style="list-style-type: none"> • Planification et suivi 	<ul style="list-style-type: none"> • Analyse des bases de test en fonction de la mise en œuvre du MBT • Planification du développement et de l'évolution des modèles MBT • Vérification continue de la qualité des modèles MBT
<ul style="list-style-type: none"> • Analyse et conception 	<ul style="list-style-type: none"> • Modélisation MBT • Re-factorisation des modèles MBT • Validation et vérification des modèles MBT
<ul style="list-style-type: none"> • Implémentation et exécution 	<ul style="list-style-type: none"> • Application des critères de sélection de tests adaptés • Génération de tests exécutables • Développement de la couche d'adaptation des tests (dans le cas de l'exécution automatisée des tests) • Exécution des tests (manuelle ou automatique)
<ul style="list-style-type: none"> • Évaluation des critères d'arrêt du test et suivi 	<ul style="list-style-type: none"> • Mise en place de la traçabilité des défauts • Documentation des critères d'arrêt des tests • Combinaison de l'évaluation MBT avec d'autres évaluation (si pertinent) dans un rapport commun
<ul style="list-style-type: none"> • Clôture 	<ul style="list-style-type: none"> • Archivage des artefacts MBT • Documentation des connaissances acquises avec le MBT • Mise en maintenance des artefacts MBT

5.2.3 Intégration de l'outillage MBT

L'introduction de l'outillage MBT au sein des équipes de test suit les mêmes principes que pour n'importe quel outil de test (voir à ce sujet le syllabus ISTQB® niveau fondation [ISTQB_FL_SYL], section 6.3). Mais comme le MBT est une approche récente, l'évaluation de l'outillage est une étape importante qui ne doit pas être sous-estimée. Un aspect important de cette évaluation concerne l'intégration dans la chaîne outillée, incluant la gestion de configuration, la gestion des exigences, la gestion des tests et les outils d'automatisation.

La figure suivante représente un exemple de chaîne outillée intégrant le MBT.

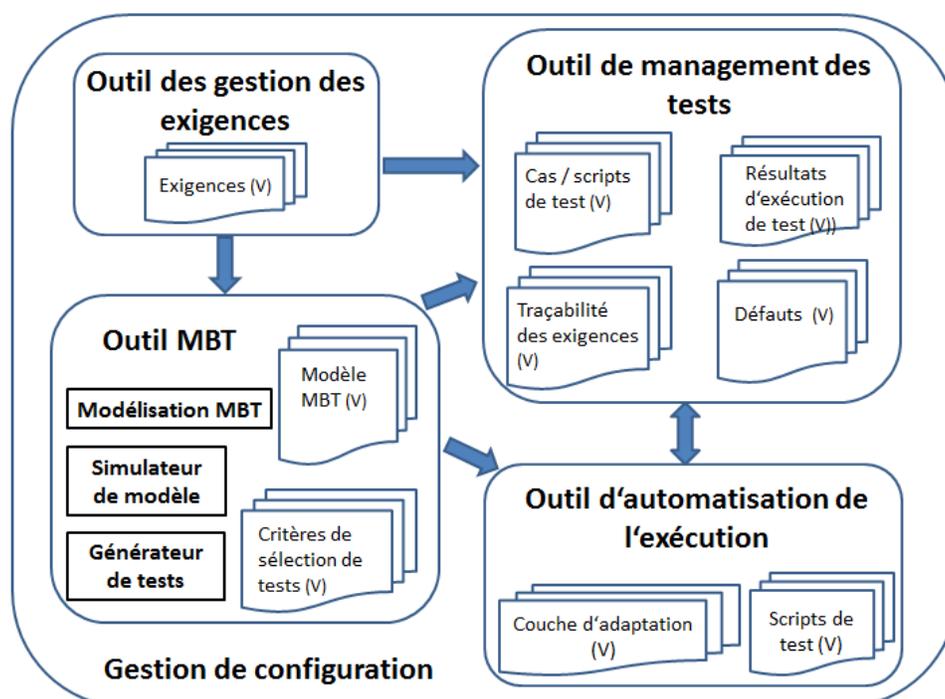


Figure 1 – Exemple de chaîne outillée intégrant le MBT – le symbole (V) indique que l'artefact considéré est mis en gestion de configuration

Cette chaîne outillée supporte les activités principales d'un processus de test intégrant le MBT :

- Développement itératif des modèles MBT, revue et validation avec l'outil de modélisation, le simulateur de modèle et la traçabilité avec les exigences.
- Génération automatique des tests par application des critères de sélection de tests sur le modèle MBT mise en œuvre avec le générateur de tests.
- Génération des cas et des scripts de test ainsi que des liens de traçabilité entre les tests et les exigences, avec export au sein de l'outil de management des tests et de l'outil d'automatisation (dans le cas d'une exécution automatique des tests).
- Mise en gestion de configuration des artefacts MBT tels que le modèle MBT, les critères de sélection de tests et la couche d'adaptation (spécification et code).

6 Définition des termes spécifiques au MBT

Le tableau suivant présente la traduction des termes du syllabus Model-Based Testing issu du glossaire ISTQB® dans sa version 3.01 et ultérieure. Pour chaque terme, les versions en anglais et en français sont données.

Term - English	Terme - Français	Definition - English	Définition - Français
Model-Based Testing	Model-Based Testing	Testing based on or involving models.	Approche du test mettant en œuvre la modélisation
MBT	MBT	Acronym for Model-Based Testing.	Acronyme pour Model-Based Testing
MBT model	modèle MBT	Any model used in Model-Based Testing.	Tout modèle utilisé pour le Model-Based Testing
coverage item	Élément de couverture	An entity or property used as a basis for test coverage, e.g., equivalence partitions or code statements.	Une entité ou propriété utilisée comme élément de couverture, par exemple les partitions d'équivalence ou des instructions du code.
model coverage	couverture de modèle	The degree, expressed as a percentage, to which model elements are planned to be or have been exercised by a test suite.	Le degré, exprimé en pourcentage, indiquant dans quelle mesure les éléments du modèle sont exercés dans une suite de tests
offline MBT	MBT hors-ligne	Model-Based Testing approach whereby test cases are generated into a repository for future execution.	Une approche du MBT dans laquelle les cas de test sont générés dans un référentiel en vue d'une exécution ultérieure.
online MBT Synonym: on-the-fly MBT	MBT en-ligne Synonyme : MBT à la volée	Model-Based Testing approach whereby test cases are generated and executed simultaneously.	Une approche du MBT dans laquelle les cas de test sont générés et exécutés de façon simultanée.
test adaption layer	couche d'adaptation des tests	The layer in a test automation architecture which provides the necessary code to adapt test scripts on an abstract level to the various components, configuration or interfaces of the SUT.	Dans une architecture d'automatisation des tests, la couche qui fournit le code nécessaire pour adapter les scripts, générés à un certain niveau d'abstraction, aux composants, configuration et interfaces du système sous test.
test case explosion	explosion du nombre de cas de test	The disproportionate growth of the number of test cases with growing size of the test basis, when using a certain test design technique. Test case explosion may also happen when applying the test design technique systematically for the first time.	La croissance disproportionnée, par rapport à la taille des bases du test, du nombre de cas de test obtenus avec une technique de conception de test. L'explosion du nombre de cas de test peut aussi survenir lorsqu'on applique une technique de conception de test de façon systématique pour la première fois.

test model	modèle de test	A model describing testware that is used for testing a component or a system under test.	Un modèle décrivant des artefacts de test et utilisé pour tester un composant ou un système sous test.
test selection criteria	critère de sélection des tests	The criteria used to guide the generation of test cases or to select test cases in order to limit the size of a test.	Un critère utilisé pour guider la génération de cas de test ou pour sélectionner des cas de test dans l'objectif d'en diminuer le nombre.

7 Abréviations et marques déposées

Abréviation	Signification
BPMN	Business Process Modeling Notation
IHM	Interface Homme Machine
ISTQB	International Software Testing Qualifications Board
KPI	Key Performance Indicator – Indicateur clé de performance
MBT	Model-Based Testing
ROI	Return On Investment – Retour sur Investissement
UML	Unified Modeling Language

Marque déposée	Propriétaire
ISTQB®	International Software Testing Qualification Board
UML®	Object Management Group, Inc.

8 Références

Standards

- [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)

Documents ISTQB – Versions françaises

- [ISTQB_AT_FL_SYL] ISTQB Testeur agile – Niveau fondation - Syllabus, Version 2014
[ISTQB_ATA_SYL] ISTQB Niveau avancé – Analyste de test, Version 2012
[ISTQB_FL_SYL] ISTQB Testeur certifié – Niveau fondation, Version 2011
[ISTQB_GLOSSARY] Standard Glossary of Terms used in Software Testing, Version 3.0, 2015
[ISTQB_MBT_OVIEW] ISTQB Testeur certifié – Model-Based Testing – Présentation générale, Version 2015

Ces documents sont disponibles en français sur le site du CFTL.

Documents référencés dans le syllabus

- [Utt07] Mark Utting and Bruno Legeard, "Practical Model-Based Testing – A tools approach," Morgan&Kauffmann, 2007.
[Wei09] Stephan Weißleder. "Test Models and Coverage Criteria for Automatic Model-Based Test Generation with UML State Machines," PhD Thesis, Humboldt-Universität zu Berlin, 12/2009.
[Zan11a] Justyna Zander, Ina Schieferdecker and Pieter J. Mosterman, "A Taxonomy of Model-Based Testing for Embedded Systems from Multiple Industry Domains," In Model-based testing for embedded systems. CRC Press.

Autres références

- [Bak08] Paul Baker, Zhen Ru Dai, Jens Grabowski, Øystein Haugen, Ina Schieferdecker and Clay Williams, "Model-Driven Testing," Springer, 2008.
[Jac07] Jonathan Jacky, Margus Veanes, Colin Campbell and Wolfram Schulte, "Model-Based Software Testing and Analysis with C#," Cambridge University Press, 2007.
[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.
[Zan11b] Justyna Zander (Editor), Ina Schieferdecker (Editor) and Pieter J. Mosterman (Editor), "Model-Based Testing for Embedded Systems," CRC Press, 2011.

9. Annexe A – Langages de modélisation retenus dans le syllabus Model-Based Testing

Pour les besoins des objectifs d'apprentissage de niveau K3, deux langages simples de modélisation sont proposés dans le syllabus :

- Le premier est un sous-ensemble des diagrammes d'activités UML.
- Le second est un sous-ensemble des machines à états UML.

Ces deux langages, sous-ensembles de diagrammes UML, sont chacun présentés dans les sections suivantes au travers d'un exemple.

9.1 Un langage simple de modélisation 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.

Il s'agit d'un sous-ensemble des diagrammes d'activités UML. Il est composé des éléments de modélisation suivants :

- Un cercle noir représente le nœud initial du flot métier et un cercle noir encerclé représente un nœud de fin d'activité.
- Les rectangles à bord rond représentent les actions.
- Les losanges représentent des nœuds de contrôle (décision ou fusion) auxquels peuvent être associés des labels (texte).
- Les flèches représentent les flots, auxquels peuvent être associées des expressions logiques (incluant les opérateurs arithmétiques et booléens).
- Les barres horizontales représentent le point de départ (bifurcation) et de fin (union) de séquences d'activités en parallèle. La séquence continue lorsque toutes les activités en parallèle arrivent à la barre d'union.
- Le lien vers les identifiants d'exigence avec des lignes en pointillé.
- Les sous-diagrammes sont définis par des rectangles.

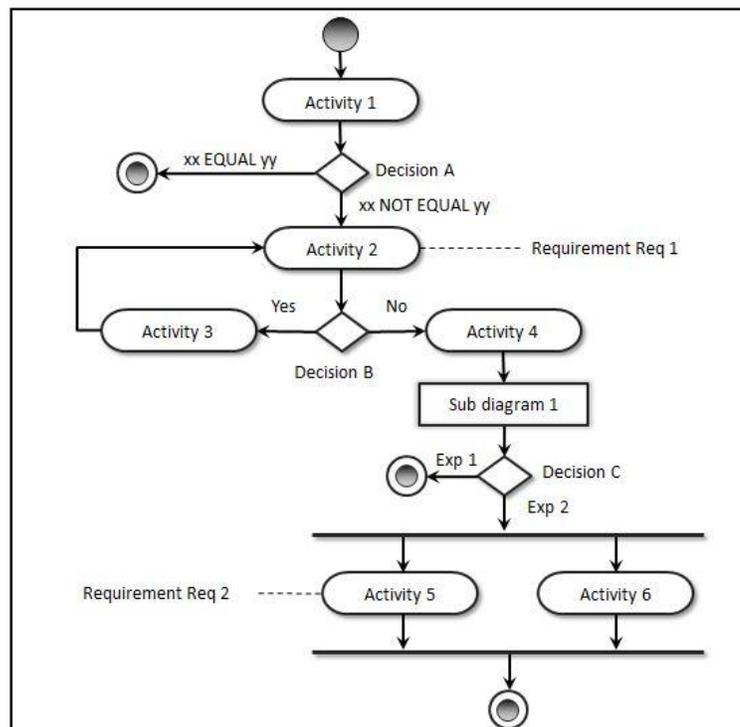


Figure 2 – Exemple de diagramme d'activités

Cette description du langage de modélisation reste informelle et simple, mais contient suffisamment d'information pour utiliser le langage 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. Il montre un flot d'activités qui démarre toujours par *Activity 1*.

Le processus s'arrête si "*xx EQUAL yy*". Sinon, l'activité *Activity 2* est réalisée.

Les activités *Activity 3* et *Activity 2* sont répétées tant que la décision *Decision B* est "Yes".

Si la décision *Decision B* est "No", alors l'activité *Activity 4* est réalisée et le sous-diagramme *Sub diagram 1* est appelé.

Le flot d'activités s'arrête si le résultat de la décision *Decision C* est égal à l'expression *Exp 1*.

Sinon, *Activity 5* et *Activity 6*, réalisées en parallèle, sont réalisées puis le flot d'activités s'arrête.

Le modèle indique que l'activité *Activity 2* est reliée à l'exigence *Req 1* et que l'activité *Activity 5* est reliée à l'exigence *Req 2*.

9.2 Un langage simple de modélisation pour diagramme d'états-transitions

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

- Un cercle noir représente l'état initial et un cercle noir encerclé représente un état final.
- Les rectangles à bord rond représentent les états.
- Les losanges représentent des nœuds de contrôle (décision ou fusion) auxquels peuvent être associés des labels (texte).
- Les flèches représentent les transitions annotées par un texte "événement [condition] / action" où chaque partie est optionnelle.
- Le lien vers les identifiants d'exigence avec des lignes en pointillé.
- Les sous-diagrammes sont définis par des rectangles.

La figure 3 représente un exemple de modèle MBT développé avec ce langage de modélisation. Il montre le comportement d'un distributeur de café ou de thé.

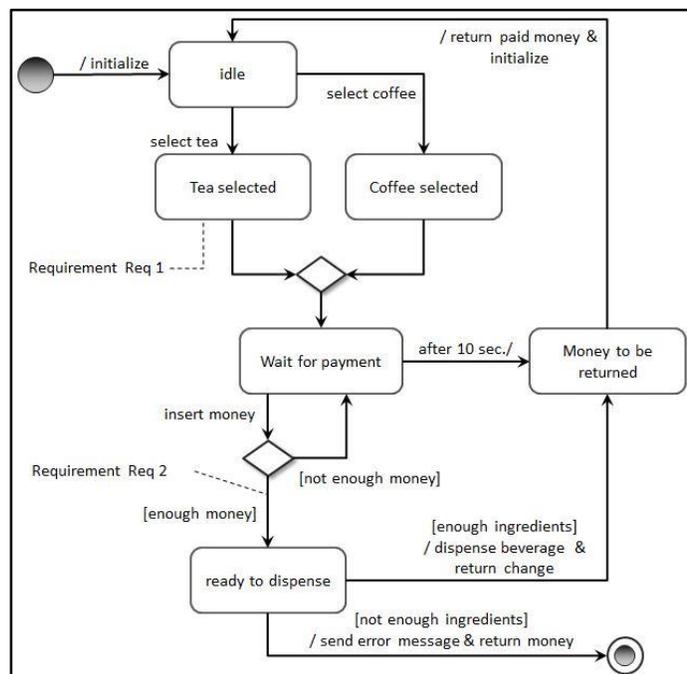


Figure 3 – Exemple de diagramme d'états-transitions

Une fois initialisé, le distributeur se trouve en attente d'entrée utilisateur (état '*idle*'). Les entrées utilisateur peuvent être la sélection du thé ('*select tea*') ou la sélection du café ('*select coffee*').

En fonction de la sélection, la préparation correspondante est réalisée en positionnement une demande de paiement en fonction de la boisson sélectionnée, et mise en attente du paiement par l'utilisateur.

Le modèle possède une boucle qui permet au distributeur de fournir la boisson demandée (état '*ready to dispense*') seulement si le paiement réalisé par l'utilisateur est suffisant.

Si l'utilisateur n'insère pas suffisamment d'argent pour le paiement de la boisson demandée, le distributeur attend 10 secondes avant de restituer la monnaie déjà introduite et se remettre en attente (état '*idle*').

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

Dans le cas où il manque des ingrédients (eau, thé ou café), le distributeur de boisson rend la monnaie introduite et se met en arrêt.

On peut aussi voir que l'exigence *Req1* est liée à l'état '*Tea selected*', indiquant qu'un test qui atteint cet état couvre cette exigence.

L'exigence *Req2* est liée à la transition '*ready to dispense*' et porte sur le comportement du distributeur lorsque suffisamment d'argent a été introduit en fonction de la boisson sélectionnée.