

# **Testeur Certifié**

## **Syllabus Niveau Avancé**

### **Test Manager**

Version 2012

---

International Software Testing Qualifications Board

---



Copyright Notice

Ce document ne peut être copié intégralement ou partiellement que si la source est mentionnée.



---

Copyright © International Software Testing Qualifications Board (ci-après nommé ISTQB®).

Sous-groupe de travail Test Manager niveau avancé : Rex Black (Responsable), Judy McKay (Responsable adjointe), Graham Bath, Debra Friedenberg, Bernard Homès, Kenji Onishi, Mike Smith, Geoff Thompson, Tsuyoshi Yumoto ; 2010-2012.

Traduction française : Eric Riou du Cosquer, Bernard Homès.



## Historique des modifications

Version	Date	Remarques
ISEB v1.1	04SEP01	ISEB Practitioner Syllabus
ISTQB 1.2 <sup>E</sup>	SEP03	ISTQB Advanced Level Syllabus from EOQ-SG
V2007	12OCT07	Certified Tester Advanced Level syllabus version 2007
V2007FR	31DEC08	Testeur Certifié Niveau Avancé, version Française
D100626	26JUN10	Incorporation des changements acceptés en 2009, division des chapitres pour les différents modules
D101227	27DEC10	Acceptation des changements de format et des corrections n'ayant pas d'impact sur la signification des phrases.
D2011	31OCT11	Division du syllabus, reformulation des LOs et modifications de texte pour correspondre aux LOs. Ajout des objectifs métier.
Alpha 2012	09Fèv12	Incorporation de tous les commentaires reçus des comités nationaux sur la version d'Octobre.
Beta 2012	26Mar12	Incorporation des commentaires reçus à temps des comités nationaux sur la version Alpha.
Beta 2012	07APR12	Version Beta soumise à l'AG
Beta 2012	08JUN12	Version éditée distribuée aux comités nationaux
Beta 2012	27JUN12	Incorporation des commentaires des groupes de travail EWG et Glossaire
RC 2012	15AUG12	Livraison de la version candidate – incorporation finale des commentaires des comités nationaux
GA 2012	19OCT12	Corrections finales et mise en forme pour la livraison à l'assemblée générale
FR 2012	31DEC12	Traduction française par le CFTL



## Table des matières

Historique des modifications .....	3
Table des matières .....	4
Remerciements .....	6
0. Introduction à ce Syllabus .....	7
0.1 Objectif de ce Document.....	7
0.2 Vue d'Ensemble .....	7
0.3 Objectifs d'Apprentissage Examinables.....	7
1. Processus de Test – 420 mins. ....	8
1.1 Introduction .....	9
1.2 Planification, Suivi et Contrôle des Tests.....	9
1.2.1 Planification des Tests.....	9
1.2.2 Pilotage et Contrôle des Tests.....	10
1.3 Analyse des Tests.....	11
1.4 Conception des Tests .....	13
1.5 Implémentation des Tests.....	13
1.6 Exécution des Tests.....	14
1.7 Évaluation des Critères de Sortie et Reporting.....	15
1.8 Activités de Clôture des Tests.....	15
2. Gestion des Tests – 750 mins. ....	17
2.1 Introduction .....	19
2.2 Gestion des Tests en Pratique.....	19
2.2.1 Comprendre les Parties-prenantes du Test.....	19
2.2.2 Autres Activités du Cycle de Vie du Développement Logiciel et Livrables .....	20
2.2.3 Alignement des Activités de Test et autres Activités du Cycle de Vie.....	21
2.2.4 Gérer les Tests Non Fonctionnels .....	23
2.2.5 Gérer les Tests Basés sur l'Expérience .....	24
2.3 Test Basé sur les Risques et Autres Approches pour la Priorisation des Tests et l'Allocation de l'Effort.....	25
2.3.1 Test Basé sur les Risques .....	25
2.3.2 Test Basé sur les Risques Techniques .....	29
2.3.3 Autres Techniques de Sélection des Tests .....	33
2.3.4 Priorisation des Tests et Allocation de l'Effort dans le Processus de Test .....	34
2.4 Documentation des Tests et autres Livrables.....	34
2.4.1 Politique de Test .....	35
2.4.2 Stratégie de Test .....	35
2.4.3 Plan de Test Maître .....	37
2.4.4 Plan de Test de Niveau .....	38
2.4.5 Gestion des Risques Projet .....	39
2.4.6 Autres Livrables du Test .....	39
2.5 Estimation des Tests.....	40
2.6 Définition et Utilisation de Métriques de Test.....	42
2.7 Valeur Financière du Test .....	47
2.8 Tests Distribués, Externalisés et Internalisés .....	48
2.9 Gérer l'Application de Standards Industriels.....	48
3. Revues – 180 mins.....	51
3.1 Introduction .....	52
3.2 Revues de Gestion et Audits.....	53
3.3 Gérer les Revues .....	54
3.4 Métriques pour les Revues .....	56



3.5 Gérer des Revues Formelles .....	56
4. Gestion des Anomalies – 150 mins .....	58
4.1 Introduction .....	59
4.2 Cycle de Vie des Anomalies et Cycle de développement logiciel .....	59
4.2.1 Cycle d'une Anomalie et États .....	60
4.2.2 Gérer des Rapports d'Anomalie Invalides ou Dupliqués .....	60
4.2.3 Gestion Fonctionnelle Transverse des Défauts .....	61
4.3 Données d'un Rapport d'Anomalie .....	61
4.4 Évaluer l'Efficacité d'un Processus avec l'Information des Rapports d'Anomalie .....	63
5. Améliorer le Processus de Test – 135 mins .....	64
5.1 Introduction .....	65
5.2 Processus d'Amélioration des Tests .....	65
5.2.1 Introduction à l'amélioration de processus .....	65
5.2.2 Types de processus d'amélioration .....	66
5.3 Améliorer le Processus de Test .....	66
5.4 Améliorer le Processus de Test avec TMMi .....	67
5.5 Améliorer le Processus de Test avec TPI Next .....	68
5.6 Améliorer le Processus de Test avec CTP .....	68
5.7 Améliorer le Processus de Test avec STEP .....	69
6. Outils de Test et Automatisation – 135 min .....	70
6.1 Introduction .....	71
6.2 Sélection d'un Outil .....	71
6.2.1 Outils Open-Source .....	71
6.2.2 Outils Maison .....	72
6.2.3 Retour sur Investissement (ROI) .....	72
6.2.4 Processus de Sélection .....	74
6.3 Cycle de vie d'un Outil .....	75
6.4 Métriques sur les Outils .....	76
7. Compétences – Composition de l'Équipe – 210 mins .....	77
7.1 Introduction .....	78
7.2 Compétences Individuelles .....	78
7.3 Dynamique de l'Équipe de Test .....	80
7.4 Introduire le Test dans une Organisation .....	81
7.5 Motivation .....	83
7.6 Communication .....	84
8. Références .....	85
8.1 Standards .....	85
8.2 Documents ISTQB .....	85
8.3 Marques .....	86
8.4 Ouvrages .....	86
8.5 Autres Références .....	88
9. Index .....	89



## Remerciements

Ce document a été produit par une équipe issue du groupe de travail Niveau Avancé de l'International Software Testing Qualifications Board – Test Manager Avancé: Rex Black (Responsable), Judy McKay (Responsable adjointe), Graham Bath, Debra Friedenberg, Bernard Homès, Paul Jorgensen, Kenji Onishi, Mike Smith, Geoff Thompson, Erik van Veenendaal, Tsuyoshi Yumoto.

L'équipe remercie l'équipe de revue ainsi que tous les Comités Nationaux pour leurs suggestions et entrées.

Au moment de la finalisation du Syllabus Niveau Avancé le groupe de travail Niveau Avancé était constitué des membres suivants (par ordre alphabétique):

Graham Bath, Rex Black, Maria Clara Choucair, Debra Friedenberg, Bernard Homès (Responsable adjoint), Paul Jorgensen, Judy McKay, Jamie Mitchell, Thomas Mueller, Klaus Olsen, Kenji Onishi, Meile Posthuma, Eric Riou du Cosquer, Jan Sabak, Hans Schaefer, Mike Smith (Responsable), Geoff Thompson, Erik van Veenendaal, Tsuyoshi Yumoto.

Les personnes suivantes ont participé aux relectures, revues et choix pour ce document (par ordre alphabétique):

Chris van Bael, Graham Bath, Kimmo Hakala, Rob Hendriks, Bernard Homès, Marcel Kwakernaak, Rik Marselis, Don Mills, Gary Mogyorodi, Thomas Mueller, Ingvar Nordstrom, Katja Piroué, Miele Posthuma, Nathalie Rooseboom de Vries, Geoff Thompson, Jamil Wahbeh, Hans Weiberg.

Ce document a été officiellement approuvé pour publication par l'Assemblée Générale de l'ISTQB® le 19 octobre 2012.



## 0. Introduction à ce Syllabus

### 0.1 Objectif de ce Document

Ce syllabus forme la base de la Qualification Internationale en Test de Logiciels au Niveau Avancé pour le Test Manager. L'ISTQB® fournit ce syllabus comme suit :

1. Aux Comités Nationaux, pour traduction dans leur langue et pour l'accréditation des fournisseurs de formation. Les Comités Nationaux peuvent adapter le syllabus aux particularités de leur langage et modifier les références pour les adapter à leurs publications locales.
2. Aux Comités d'Examens, pour en dériver des questions d'examen adaptées aux objectifs d'apprentissage de chaque module dans la langue locale.
3. Aux fournisseurs de formation, pour concevoir les cours et déterminer les méthodes de formation appropriées.
4. Aux candidats à la certification, pour préparer l'examen (dans le cadre d'une formation ou indépendamment).
5. A la communauté internationale du logiciel et de l'ingénierie des systèmes, pour faire progresser la profession de testeur de logiciels et de systèmes, et comme base pour des livres et articles.

L'ISTQB® peut autoriser d'autres entités à utiliser ce syllabus pour d'autres objectifs à condition qu'elles demandent et obtiennent une autorisation écrite préalable.

### 0.2 Vue d'Ensemble

Le Niveau Avancé est composé de trois syllabi séparés:

- Test Manager
- Analyste de Test
- Analyste Technique de Test

Le document Niveau Avancé Vue Générale [ISTQB\_AL\_OVIEW] contient les informations suivantes:

- Bénéfices Métier pour chaque syllabus
- Résumé pour chaque syllabus
- Relations entre les syllabi
- Description des niveaux cognitifs (Niveaux K)
- Annexes

### 0.3 Objectifs d'Apprentissage Examinables

Les Objectifs d'Apprentissage supportent les Bénéfices Métier et sont utilisés pour créer l'examen pour obtenir la certification de Test Manager Avancé. Toutes les parties de ce syllabus sont examinables au niveau K1, c.à.d. le candidat devra reconnaître, se souvenir et mémoriser un terme ou un concept. Les Objectifs d'Apprentissage aux niveaux K2, K3 et K4 sont indiqués au début de chaque chapitre.

## 1. Processus de Test – 420 mins.

### Mots clé

critère de sortie, cas de test, clôture des tests, condition de test, contrôle des tests, conception de tests, exécution des tests, implémentation des tests, registre de test, planification des tests, procédure de test, script de test, rapport de synthèse de tests

### Objectifs d'Apprentissage pour Processus de Test

#### 1.2 Gestion, Pilotage et Contrôle des Tests

TM-1.2.1 (K4) Analyser les besoins de test pour un système afin de planifier les activités de test et les livrables permettant d'atteindre les objectifs de test

#### 1.3 Analyse des Tests

GT-1.3.1 (K3) Utiliser la traçabilité pour regarder la complétude et la cohérence des conditions de test définies par rapport aux objectifs de test, à la stratégie de test, et au plan de test

TM-1.3.2 (K2) Expliquer les facteurs pouvant affecter le niveau de détail possible pour les conditions de test et les avantages et désavantages à spécifier les conditions de test à un niveau détaillé.

#### 1.4 Conception des Tests

TM-1.4.1 (K3) Utiliser la traçabilité pour vérifier la complétude et la cohérence des cas de test conçus par rapport aux conditions de test définies.

#### 1.5 Implémentation des Tests

TM-1.5.1 (K3) Utiliser les risques, la priorisation, l'environnement de test, les données associées et les contraintes pour développer un planning d'exécution des tests qui soit complet et consistant par rapport aux objectifs de test, à la stratégie de test et au plan de test.

#### 1.6 Exécution des Tests

TM-1.6.1 (K3) Utiliser la traçabilité pour piloter l'avancement du test par rapport à la complétude et la cohérence par rapport aux objectifs de test, à la stratégie de test et au plan de test.

#### 1.7 Évaluation des Critères de Sortie et Reporting

TM-1.7.1 (K2) Expliquer l'importance d'un recueil d'information précis et effectué au bon moment dans le processus de test afin de permettre de réaliser de façon précise les rapports et les évaluations par rapport aux critères de sortie.

#### 1.8 Activités de Clôture des Tests

TM-1.8.1 (K2) Résumer les quatre groupes d'activités de clôture des tests.

TM-1.8.2 (K3) Effectuer un bilan de projet pour évaluer les processus et identifier des axes d'amélioration.

## 1.1 Introduction

Dans le syllabus ISTQB® Niveau Fondation, le processus fondamental de test comprend les activités suivantes :

- Planification et contrôle
- Analyse et conception
- Implémentation et exécution
- Évaluation des critères de sortie et information
- Activités de clôture des tests

Le syllabus Niveau Fondation affirme que malgré un ordre séquentiel, les activités du processus peuvent se recouvrir ou avoir lieu de façon concurrente. Il est en général nécessaire d'adapter ces principales activités dans le contexte d'un système ou projet.

Pour les syllabi Niveau Avancé, certaines de ces activités sont abordées séparément, afin de fournir des précisions complémentaires et une optimisation des processus mieux intégrés avec le cycle de développement logiciel, et pour augmenter l'efficacité du pilotage et contrôle des tests. Les activités sont désormais les suivantes:

- Gestion, pilotage et contrôle
- Analyse
- Conception
- Implémentation
- Exécution
- Évaluation des critères de sortie et information
- Activités de clôture des tests

## 1.2 Planification, Suivi et Contrôle des Tests

Cette partie se concentre sur le processus de gestion, pilotage et contrôle des tests. Comme indiqué dans le syllabus Niveau Fondation, ces activités sont des rôles de Test Manager.

### 1.2.1 Planification des Tests

Pour chaque niveau de test, la planification des tests commence au début du processus de test pour ce niveau et continue tout au long du projet, jusqu'à l'achèvement des activités de clôture pour ce niveau. Cela implique l'identification des activités et ressources nécessaires à la réalisation de la mission et des objectifs définis dans la stratégie de test. La planification des tests inclut aussi l'identification des méthodes pour collecter et suivre les métriques utilisées pour guider le projet, déterminer le respect du plan et mesurer l'atteinte des objectifs. En déterminant des métriques utiles pendant les étapes de planification, des outils peuvent être sélectionnés, des formations peuvent être programmées et des recommandations de documentation peuvent être établies.

La stratégie (ou les stratégies) retenue(s) pour le projet de test aident à identifier les tâches qui devraient arriver pendant les phases de planning. Par exemple, en utilisant la stratégie de test basée sur les risques (voir Chapitre 2), l'analyse des risques est utilisée pour guider le processus de planification des tests au regard des activités de mitigation des risques requises pour identifier les risques produit et pour aider avec le plan d'urgence. Si un nombre de défauts potentiels et sérieux, relatifs à la sécurité sont identifiés, un effort significatif devrait être consacré au développement et à l'exécution de tests de sécurité.



Les informations sur les risques peuvent aussi être utilisées pour déterminer les priorités des diverses activités de test. Par exemple, lorsque la performance du système est un risque élevé, les tests de performances peuvent être mis en place dès que du code intégré est disponible. De même, si une stratégie réactive doit être employée, la planification peut être justifiée d'une création des agréments de test et d'outils pour des techniques de tests dynamiques telles que du test exploratoire.

De plus, l'étape de planification des tests est là où l'approche de test est clairement définie par le Test Manager, incluant les niveaux de test qui seront mis en œuvre, les buts et objectifs de chaque niveau, et les techniques de tests à utiliser à chaque niveau de test. Par exemple, avec le test basé sur les risques pour certains systèmes avioniques, une évaluation des risques prescrira quel niveau de couverture de code est requis et ainsi quelles techniques de test devraient être utilisées.

Des relations complexes peuvent exister entre la base de test (par ex. exigences ou risques spécifiques) les conditions de test et les tests qui les couvriront. Des relations n-n existent souvent entre ces différents livrables. Ces relations doivent être comprises pour permettre une mise en œuvre efficace de la planification des tests, de leur suivi et de leur contrôle. Des décisions relatives aux outils peuvent aussi dépendre de la compréhension des relations entre ces différents livrables.

Des relations peuvent aussi exister entre des livrables produits par l'équipe de développement et l'équipe de test. Par exemple, la matrice de traçabilité peut devoir suivre les relations entre les éléments de spécification de conception détaillée provenant des architectes système, les exigences métier provenant des analystes métier, et les livrables de test définis par l'équipe de test. Si des cas de test bas niveau doivent être conçus et utilisés, une exigence pourra être définie lors de la planification des tests pour préciser que les documents de conception détaillée provenant de l'équipe de développement doivent être approuvés avant que ne puisse commencer la création des cas de test. Dans un cycle de vie Agile, des sessions informelles de transfert d'information peuvent être utilisées pour transférer les informations entre les équipes avant le démarrage du test.

Le plan de test peut aussi lister les caractéristiques spécifiques du logiciel faisant partie de son périmètre (sur la base de l'analyse des risques, si nécessaire), de même qu'identifier précisément les caractéristiques qui sont hors de son périmètre. Selon les niveaux de formalisme et la documentation adaptés au projet, chaque caractéristique à couvrir peut être associée à une spécification de conception de test correspondante.

Il peut aussi y avoir des exigences à cette étape pour que le test manager travaille avec les architectes projet afin de définir la spécification initiale de l'environnement de test, de vérifier la disponibilité des ressources nécessaires, pour s'assurer de l'engagement des personnes prévues pour configurer l'environnement à le faire, pour comprendre les prévisions de coûts et de délais ainsi que le travail requis pour terminer et livrer l'environnement de test.

Enfin, toutes les dépendances externes et les engagements sur le niveau de service (Service Level Agreement) devraient être identifiés et, si nécessaire, un premier contact devrait être pris. Comme exemples de dépendances, on peut citer le recours à des ressources externes, la dépendance envers d'autres projets (dans le contexte d'un programme), à des fournisseurs externes ou sous-traitants en développement, à l'équipe de mise en production et aux administrateurs de bases de données.

## 1.2.2 Pilotage et Contrôle des Tests

Afin de permettre au test manager de fournir un contrôle efficace des tests, un planning des tests et un cadre de mesure doivent être établis pour permettre le suivi des livrables et des ressources par rapport à ce plan. Ce cadre doit inclure les mesures et cibles détaillées qui sont nécessaires pour lier l'état des livrables de test et des activités au plan et aux objectifs stratégiques.



Pour des projets petits ou moins complexes, il peut être assez facile de lier les livrables et les activités de test au plan et aux objectifs stratégiques fixés, mais généralement des objectifs plus détaillés sont nécessaires pour atteindre cela. Ceci peut inclure les mesures et cibles pour atteindre les objectifs de test et la couverture de la base de test.

Il est très important de lier le statut des livrables et des activités de test à la base de test d'une façon compréhensible et utile pour les parties prenantes techniques et métier du projet. Définir des objectifs et mesurer l'avancement par rapport aux conditions de tests et groupes de conditions de test peut être un moyen d'y arriver en reliant d'autres livrables de test à la base de test via les conditions de test. Une traçabilité correctement configurée, incluant la capacité à communiquer sur l'état de traçabilité, rend plus transparentes et compréhensibles les relations complexes entre les livrables du développement, les bases de test, et les livrables de test.

Parfois, les mesures détaillées et les cibles que les parties prenantes souhaitent suivre ne sont pas directement liées aux fonctionnalités du système ou à une spécification, surtout s'il y a peu ou pas de documentation formelle. Par exemple, une partie prenante côté métier peut être plus intéressée par l'établissement d'une couverture par rapport à un cycle opérationnel métier, même si la spécification est définie en termes de fonctionnalités système. L'implication des parties prenantes métier, tôt dans le projet peut aider à définir des mesures et objectifs qui pourront non seulement être utilisés pour aider à fournir un meilleur contrôle pendant le projet, mais peut aussi aider à conduire et orienter les activités de test tout au long du projet. Par exemple, des mesures et objectifs fixés par les parties prenantes peuvent résulter en une structuration des livrables de la conception et de l'implémentation des livrables tests et/ou du planning d'exécution des tests afin de faciliter un contrôle précis de l'avancement du test en rapport avec ces mesures. Ces objectifs aideront aussi à fournir une traçabilité pour un niveau de test spécifique et ont le potentiel d'aider à fournir des informations de traçabilité entre différents niveaux de test.

Le contrôle des tests est une activité récurrente. Cela implique la comparaison de l'avancement réel par rapport à l'avancement prévu et la mise en œuvre d'actions correctives si besoin. Le contrôle des tests oriente le test vers l'accomplissement de la mission, de la stratégies et des objectifs, incluant la réorientation des activités de test selon les besoins. Les réactions appropriées aux informations du contrôle dépendront des informations détaillées de planification des tests.

Le contenu des documents de planification des tests et les activités de contrôle des tests sont couverts dans le Chapitre 2.

### 1.3 Analyse des Tests

Plutôt que d'aborder ensemble l'analyse et la conception des tests, comme décrits dans le syllabus niveau fondation, les syllabi niveau avancé les considèrent comme des activités séparées, tout en reconnaissant qu'elles peuvent être implémentées en parallèle, intégrées, ou itératives pour faciliter la production des livrables de conception des tests.

L'analyse des tests est l'activité qui définit "ce qu'il faut" tester, sous la forme de conditions de test. Les conditions de test peuvent être identifiées par l'analyse de la base de test, des objectifs de test et des risques produit. Elles peuvent être considérées comme les mesures détaillées et les cibles décrivant le succès (par ex. parmi des critères de sortie) et devraient faire l'objet d'une traçabilité amont vers la base de test et des objectifs stratégiques définis, incluant des objectifs de test et les autres critères de succès du projet ou des parties prenantes. Les conditions de test doivent également faire l'objet d'une traçabilité avale vers la conception des tests et des autres livrables du test au fur et à mesure de leur création.

L'analyse des tests, pour un niveau de test donné peut être réalisée dès que les bases pour les tests sont établies pour ce niveau. Des techniques de test formelles et d'autres techniques de test analytique générales (par ex. stratégies analytiques orientées par les risques et stratégies analytiques basées sur les exigences) peuvent être utilisées pour identifier les conditions de test. Les conditions de test peuvent ou non spécifier des valeurs ou des variables selon le niveau de test, l'information disponible lors de la réalisation de l'analyse et le niveau de détail retenu (par ex. le degré de granularité de la documentation).

Il y a un nombre de facteurs à considérer quand l'on décide du niveau de détail avec lequel spécifier les conditions de test, dont :

- Le niveau de test
- Le niveau de détail et de qualité de la base de test
- La complexité du système ou du logiciel
- Les risques projet et produit
- Les relations entre les bases de test, ce qui doit être testé et comment cela doit être testé
- Le cycle de développement logiciel utilisé
- L'outil de Gestion des Tests utilisé
- Le niveau de spécification et de documentation requis pour la conception des tests et des autres livrables de test
- Les compétences et connaissances des Analystes de Test
- Le niveau de maturité du processus de test et de l'organisation dans son ensemble (notez qu'un niveau de maturité élevé peut demander un niveau de détail plus important, ou permettre un niveau de détail moindre)
- La disponibilité d'autres parties prenantes du projet pour répondre à des demandes

Spécifier les conditions de test de façon détaillée aboutira à un nombre plus important de conditions de test. Vous pouvez par exemple avoir une unique condition de test générale, "Tester le paiement", pour une application de commerce électronique. Cependant, dans un document détaillé de condition de test, cela pourrait être scindé en de nombreuses conditions de test, avec une condition par méthode de paiement acceptée, une condition par pays destinataire possible, et ainsi de suite.

Certains des avantages de spécifier les conditions de test de façon détaillée incluent :

- Offrir plus de flexibilité pour relier d'autres livrables (par ex. cas de test) aux bases et objectifs de test, et ainsi offrir un suivi meilleur et plus détaillé pour un Test Manager
- Contribuer à la prévention des défauts, comme discuté dans le syllabus niveau fondation, en se déroulant de façon précoce dans un projet pour les niveaux de test les plus hauts, dès que la base de test est définie et éventuellement avant que l'architecture et la conception détaillée ne soient disponibles
- Lier les livrables de test aux parties prenantes en des termes qu'elles peuvent comprendre (souvent, les cas de test et autres livrables du test n'ont aucune signification pour les parties prenantes métier et des métriques simples comme le nombre de cas de test exécutés n'apportent rien par rapport à la couverture des exigences des parties prenantes.)
- Aider à influencer et orienter non seulement les autres activités de test mais aussi les autres activités du développement
- Permettre d'optimiser la conception, l'implémentation et l'exécution des tests, ainsi que leurs livrables correspondant, par une meilleure couverture de mesures et cibles détaillées
- Permettre la base d'une traçabilité horizontale plus claire au sein d'un niveau de test

Certains inconvénients liés à une spécification détaillée des conditions de test incluent :

- Une consommation potentiellement importante en temps
- La maintenabilité sera plus difficile dans un environnement changeant
- La nécessité de définir et mettre en œuvre le niveau de formalité au sein de l'équipe



Spécifier les conditions de test de façon détaillée peut être particulièrement efficace dans les situations suivantes:

- Des méthodes légères de documentation de conception de tests, comme des check-lists, sont utilisées pour s'adapter aux cycles de vie de développement, aux contraintes de temps ou d'argent ou à d'autres facteurs
- Peu ou pas d'exigences formelles ou autres livrables de développement sont disponibles comme base de test
- Le projet est de grande taille, complexe ou à hauts risques et demande un niveau de suivi et de contrôle qui ne peut être obtenu par la simple liaison entre des cas de test et des livrables de développement

Les conditions de test peuvent être spécifiées avec moins de détails quand la base de test peut être facilement et directement reliée aux livrables de développement. Ceci sera plus probablement le cas pour:

- Les tests au niveau des composants
- Des projets moins complexes où des relations hiérarchiques simples existent entre ce qui doit être testé et comment le tester
- Des tests d'acceptation où des cas d'utilisation peuvent être utilisés pour aider à définir les tests

## 1.4 Conception des Tests

La Conception des tests est l'activité qui définit "comment" quelque chose doit être testé. Cela implique l'identification de cas de tests par une élaboration pas à pas des conditions ou de la base de test en utilisant les techniques identifiées dans la stratégie de test et/ou le plan de test.

Dépendant des approches utilisées pour mesurer, contrôler les tests, et en assurer la traçabilité, les cas de tests peuvent être directement (ou indirectement via les conditions de test) liés à la base de test et aux objectifs définis. Ces objectifs incluent les objectifs stratégiques, les objectifs de test et d'autres critères de succès définis pour le projet ou par les parties prenantes.

La conception des tests pour un niveau de test donné, peut être exécutée quand les conditions de test sont identifiées et que suffisamment d'information est disponible pour permettre la production de cas de test bas ou haut niveau, dépendant de l'approche utilisée pour la conception des tests. Pour les niveaux de test les plus élevés, il est plus probable que la conception des tests soit une activité séparée suivant une analyse des tests antérieure. Pour les niveaux de test inférieurs, l'analyse et la conception des tests seront peut-être menés comme une activité intégrée.

Il est aussi probable que certaines tâches ayant normalement lieu pendant l'implémentation des tests seront intégrées au processus de conception des tests avec une approche itérative pour construire les tests nécessaires à l'exécution ; par ex. la création des données de test. En fait, cette approche peut optimiser la couverture des conditions de test, avec la création des cas de test soit bas niveau, soit haut niveau au cours du processus de test.

## 1.5 Implémentation des Tests

L'Implémentation des tests est l'activité durant laquelle les tests sont organisés et priorisés par les Analystes de Test. Dans des contextes avec une documentation formelle, l'implémentation des tests est l'activité durant laquelle les conceptions de test sont implémentées en cas de test concrets, en procédures de test, et avec des données de test. Certaines organisations suivant le standard IEEE

829 définissent des entrées et les résultats attendus associés dans les spécifications de cas de test, et les étapes de test dans les spécifications de procédure de test. Plus communément, les entrées de test, les résultats attendus et les étapes de test sont documentés ensemble. L'Implémentation des tests comprend aussi la création de données de test enregistrées (par ex. dans des fichiers plats ou des tables de bases de données).

L'Implémentation des tests implique aussi d'ultimes vérifications pour s'assurer que l'équipe de test est prête pour l'exécution des tests. Les recherches peuvent inclure de s'assurer de la livraison de l'environnement de test nécessaire, des données de test et du code (éventuellement l'exécution de quelques tests d'acceptation de l'environnement et du code) et de voir que tous les cas de test ont été écrits, revus et sont prêts à être exécutés. Cela peut aussi comprendre de comparer par rapport aux critères d'entrée explicites et implicites pour le niveau de test en question (voir Section 1.7). L'implémentation des tests peut aussi impliquer le développement d'une description détaillée de l'environnement et des données de test.

Le niveau de détail et la complexité associée du travail effectué durant l'implémentation des tests peuvent être influencés par le niveau de détail des livrables du test (par ex. cas de tests et conditions de test). Dans certains cas, en particulier quand les tests doivent être archivés pour une réutilisation à long terme pour des tests de non-régression, ils peuvent fournir une description détaillée des étapes à suivre nécessaires pour exécuter le test, de façon à assurer une exécution fiable et consistante quel que soit le testeur exécutant le test. Si des normes réglementaires s'appliquent, les tests devraient fournir des preuves du respect des standards applicables (voir Section 2.9).

Pendant l'implémentation des tests, l'ordre dans lequel les tests manuels ou automatisés sont à exécuter devrait être précisé dans un ordonnancement d'exécution des tests. Le Test Manager devrait soigneusement regarder les contraintes, y compris les risques et les priorités, qui pourraient exiger que les tests soient exécutés dans un ordre particulier ou sur un équipement particulier. Les dépendances vis à vis de l'environnement de test ou des données de test doivent être connues et vérifiées.

Il peut y avoir certains désavantages à une implémentation précoce des tests. Avec un cycle de développement Agile, par exemple, le code peut changer radicalement d'itération en itération, rendant obsolète une grande partie du travail d'implémentation. Même sans un cycle de vie aussi avide de changements que l'Agile, tout cycle de développement itératif ou incrémental peut résulter en des changements importants entre les itérations, rendant les scripts de test non fiables, ou sujets à des besoins de maintenance importants. Il en est de même pour les cycles de vie séquentiels mal gérés où les exigences changent fréquemment, même tard dans le projet. Avant de se lancer dans un important effort d'implémentation des tests, il est sage de comprendre le cycle de développement logiciel et le caractère prévisible des caractéristiques du logiciel qui seront disponibles pour le test.

Il peut y avoir des avantages à une implémentation précoce des tests. Par exemple, les cas de test concrets fournissent des exemples du comportement prévu du logiciel, s'ils sont écrits en accord avec la base de test. Des experts métiers trouveront probablement la vérification de cas de test concrets plus facile que la vérification de règles métier abstraites, et pourront ainsi identifier des faiblesses dans les spécifications du logiciel. De tels tests vérifiés pourront fournir des illustrations révélatrices sur le comportement attendu pour les concepteurs logiciels et les développeurs.

## 1.6 Exécution des Tests

L'Exécution des tests commence dès que l'objet de test est livré et que les critères d'entrée pour l'exécution des tests sont satisfaits. Les tests devraient être conçus ou du moins définis avant l'exécution des tests. Des outils devraient être en place, notamment pour la Gestion des Tests, le



suivi des anomalies et (si nécessaire) l'automatisation de l'exécution des tests. Le suivi des résultats de test, y compris le suivi des métriques, devrait être opérationnel et les données suivies devraient être comprises par tous les membres de l'équipe. Des standards pour l'enregistrement des tests et le suivi des anomalies devraient être disponibles et publiés. En s'assurant que tous ces éléments sont en place avant le démarrage de l'exécution des tests, l'exécution des tests peut se dérouler de façon efficace.

Les tests devraient être exécutés selon les cas de tests, cependant le Test Manager devrait envisager de laisser une certaine latitude aux testeurs de couvrir des scénarios de test supplémentaires ou des comportements observés durant les tests. En suivant une stratégie de test au moins partiellement réactive, du temps devrait être réservé pour des sessions de test utilisant les techniques basées sur l'expérience ou sur les défauts. Bien sûr, toute défaillance détectée lors de ce type de test non scripté doit être documentée avec les variations des cas de test qui sont nécessaires pour sa reproduction. Les tests automatisés suivront leurs instructions définies, sans écart.

Le principal rôle du Test Manager pendant l'exécution des tests est de suivre l'avancement par rapport au plan de test et, si nécessaire, d'initier et mettre en œuvre des actions de contrôle pour diriger les tests vers un succès en termes de missions, objectifs, et stratégie. Pour ce faire, le Test Manager peut utiliser la traçabilité ascendante, des résultats de test vers les conditions de test, les bases de test, et enfin les objectifs de test, mais aussi la traçabilité allant des objectifs de test vers les résultats de test. Ce processus est décrit en détail dans la Section 2.6.

## 1.7 Évaluation des Critères de Sortie et Reporting

La documentation et la communication pour le suivi et le contrôle de l'avancement des tests sont décrites en détail dans la Section 2.6.

Du point de vue du processus de test, il est important de s'assurer que des processus efficaces soient en place pour fournir la source d'information nécessaire à l'évaluation des critères de sortie et du reporting.

La définition des exigences relatives à l'information et aux méthodes de collecte des données font partie de la gestion des tests, de son suivi et de son contrôle. Pendant l'analyse des tests, la conception des tests, l'implémentation et l'exécution des tests, le Test Manager doit s'assurer que tous les membres de l'équipe de test responsables de ces activités obtiennent en temps et en heure l'information exacte requise, afin de faciliter une évaluation et une communication efficaces.

La fréquence et le niveau de détail requis pour le reporting sont dépendants du projet et de l'organisation. Cela devrait être négocié pendant la phase de planification des tests et devrait inclure la consultation des principales parties prenantes.

## 1.8 Activités de Clôture des Tests

Lorsque l'exécution des tests est considérée terminée, les résultats principaux devraient être récupérés et, soit passés à la personne appropriée, soit archivés. Globalement, il s'agit des activités de clôture des tests. Les activités de clôture des tests se répartissent en quatre groupes principaux:

1. Vérification de la complétude des tests - s'assurer que tout le travail lié aux tests est effectivement terminé. Par exemple, tous les tests planifiés devront avoir été joués ou délibérément sautés, et toutes les anomalies connues devraient soit avoir été résolues et la correction confirmée correcte, soit reportées à une date ultérieure, soit acceptées comme restrictions permanentes.



2. Transmission des artefacts de test - fournir les livrables nécessaires à ceux qui en ont besoin. Par exemple, toute anomalie acceptée comme limitation, ou dont la correction est repoussée, devrait être communiquée à ceux qui utiliseront le système et à ceux qui supporteront le système. Les tests et les environnements de test devraient être transmis à ceux qui seront responsables des tests de maintenance. L'ensemble des tests de non-régression (automatisés ou manuels) devrait être documenté et livré à l'équipe de maintenance.
3. Leçons apprises – organiser ou participer à des réunions de rétrospective (retours d'expérience) où d'importantes leçons (issues à la fois du projet et de tout le cycle de développement du logiciel) peuvent être documentées. Lors de ces réunions, des plans d'actions sont établis afin de s'assurer que les bonnes pratiques peuvent être reproduites et que les mauvaises pratiques seront soit évitées, soit – si elles ne peuvent être évitées – gérées dans le plan de projet. Parmi les domaines à prendre en compte nous avons les suivants :
  - a. Lors des sessions d'analyse des risques qualité, la représentation des utilisateurs a-t-elle été suffisamment représentative? Par exemple, vu la découverte tardive et non anticipée d'agrégats de défauts, l'équipe peut avoir découvert que les utilisateurs devraient être mieux représentés lors des projets futurs.
  - b. Les estimations faites étaient-elles correctes? Par exemple, des estimations peuvent avoir été significativement mal évaluées et par conséquent les futures activités d'estimation devront prendre ceci en compte, y compris les raisons sous-jacentes, par ex., le test a-t-il été inefficace ou les estimations inférieures à ce qu'elles auraient dû être?
  - c. Quelles sont les tendances et les résultats de l'analyse des causes et effets des défauts? Par exemple, évaluer si des demandes d'évolution tardives ont affecté la qualité de l'analyse et du développement, chercher des tendances mettant en évidence de mauvaises pratiques, par ex., sauter un niveau de test qui aurait dû identifier des défauts plus tôt et de façon plus rentable, au prétexte d'avoir l'impression d'économiser du temps. Regarder si les tendances des défauts pourraient être liées à des domaines comme de nouvelles technologies, des changements de personnel, ou à un manque de compétences.
  - d. Y a-t-il des opportunités potentielles d'amélioration de processus?
  - e. Y a-t-il eu des variations non anticipées par rapport au plan qui devraient être prises en compte dans les prochaines planifications?
4. Archiver les résultats, les logs, les rapports et les autres documents ou livrables dans le système de gestion de configuration. Par exemple, le plan de test et le plan de projet devraient être stockés dans l'archive de planification, et rattachés clairement au système et à la version où ils ont été utilisés.

Ces tâches sont importantes, souvent non réalisées et devraient être explicitement incluses dans le plan de test.

Il est fréquent qu'une ou plusieurs de ces tâches soient oubliées, généralement à cause d'une réaffectation prématurée ou un renvoi des membres de l'équipe projet, de pressions en termes de ressources ou de délais venant de projets suivants, ou d'une saturation de l'équipe. Lors de projets menés sur la base d'un contrat (sous-traités), comme pour des développements, le contrat doit spécifier les tâches requises.

## 2. Gestion des Tests – 750 mins.

### Mots clés

plan de test de niveau, plan de test maître, risque produit, risque projet, risque qualité, risque, analyse de risque, évaluation des risques, identification des risques, niveau de risque, gestion des risques, réduction des risques, test basé sur les risques, approche de test, conditions de test, contrôle des tests, directeur de test, estimation des tests, responsable de tests, niveau de test, Gestion des Tests, contrôle des tests, plan de test, politique de test, stratégie de test, Delphi à Large Bande

### Objectifs d'Apprentissage pour la Gestion des Tests

#### 2.2 Gestion des Tests en Situation

- TM-2.2.1 (K4) Analyser les parties prenantes, le contexte et les besoins d'un projet logiciel ou d'un programme, y compris le modèle de cycle de développement logiciel, et identifier les activités de test optimales
- TM-2.2.2 (K2) Comprendre comment les activités et les livrables du cycle de développement affectent le test, et comment le test affecte les activités du cycle de développement des logiciels et les livrables
- TM-2.2.3 (K2) Expliquer différentes façons de gérer les problèmes de Gestion des Tests par rapport au test basé sur l'expérience et au test non fonctionnel

#### 2.3 Test Basé sur les Risques et autres Approches pour la Priorisation des Tests et l'Allocation de l'Effort

- TM-2.3.1 (K2) Expliquer différentes façons par lesquelles le test basé sur les risques répond aux risques
- TM-2.3.2 (K2) Expliquer, en donnant des exemples, différentes techniques pour l'analyse des risques produit
- TM-2.3.3 (K4) Analyser, identifier et évaluer les risques qualité produit, en synthétisant les risques et leur niveau d'évaluation, sur la base du point de vue des principales parties prenantes du projet
- TM-2.3.4 (K2) Décrire comment des risques qualité produit peuvent être réduits et gérés, selon leur niveau, tout au long du cycle de vie et du processus de test
- TM-2.3.5 (K2) Donner des exemples de différentes façons d'effectuer la sélection des tests, la priorisation des tests et l'allocation de l'effort de test

#### 2.4 Documentation des Tests et Autres Livrables

- TM-2.4.1 (K4) Analyser des exemples de politiques de test et stratégies de test, et créer les plans de test maîtres, plans de test de niveau et les autres livrables complets et cohérents par rapport à ces documents
- TM-2.4.2 (K4) Pour un projet donné, analyser les risques projet et sélectionner les bonnes options de gestion des risques (par ex., réduction, contingence, transfert, et/ou acceptation)
- TM-2.4.3 (K2) Décrire, en donnant des exemples, comment les stratégies de test affectent les activités de test
- TM-2.4.4 (K3) Définir des normes et des modèles de documentation pour les livrables du test, qui seront adaptés à l'organisation, au cycle de vie, aux besoins du projet, en adaptant des modèles issus de standards lorsque cela est possible

#### 2.5 Estimation des Tests



- TM-2.5.1 (K3) Pour un projet donné, réaliser une estimation pour toutes les activités du processus de test, en utilisant toutes les techniques d'estimation applicables
- TM-2.5.2 (K2) Comprendre et donner des exemples de facteurs pouvant influencer les estimations de test

## 2.6 Définition et Utilisation de Métriques de Test

- TM-2.6.1 (K2) Décrire et comparer les principales métriques relatives au test
- TM-2.6.2 (K2) Comparer les différentes dimensions du contrôle de l'avancement des tests
- TM-2.6.3 (K4) Analyser et documenter les résultats de test par rapport au risque résiduel, au statut des défauts, au statut de l'exécution des tests, au statut de la couverture des tests, et être en mesure de fournir des aperçus et des recommandations qui permettront aux parties prenantes du projet de prendre les décisions sur la livraison

## 2.7 Valeur Financière du Test

- TM-2.7.1 (K2) Donner des exemples pour chacune des quatre catégories déterminant le coût de la qualité
- TM-2.7.2 (K3) Estimer la valeur du test par rapport au coût de la qualité, sur la base de considérations quantitatives et qualitatives, et communiquer la valeur estimée aux parties prenantes

## 2.8 Tests Distribués, Externalisés et Internalisés

- TM-2.8.1 (K2) Comprendre les facteurs déterminants pour une mise en œuvre réussie de stratégies d'équipes de test, distribuées, externalisées et internalisées

## 2.9 Gérer l'Application de Standards Industriels

- TM-2.9.1 (K2) Résumer les principales sources et utilisations de standards pour le test logiciel



## 2.1 Introduction

Au Niveau Avancé, les professionnels du test commencent à se spécialiser dans leur carrière. Ce chapitre se concentre sur les domaines de connaissance nécessaires aux professionnels du test se dirigeant vers les postes de Responsable de Tests, Test Manager, et Directeur de test. Dans ce syllabus, ces professionnels du test sont appelés Test Managers, mais il est entendu que des organisations différentes auront des libellés de poste et des niveaux de responsabilité différents pour ces personnes.

## 2.2 Gestion des Tests en Pratique

Une des principales responsabilités d'un manager est d'obtenir et d'utiliser des ressources (personnes, logiciels, matériels, infrastructure, etc.) pour exécuter des processus apportant une valeur ajoutée. Pour les responsables logiciels et informatiques, les processus font souvent partie d'un projet ou programme destiné à fournir du logiciel ou un programme pour un usage interne ou externe. Pour les Test Managers, les processus sont ceux en lien avec le test, en particulier les activités du processus de test fondamental décrites dans le syllabus Niveau Fondation et dans le Chapitre 1 de ce syllabus. Comme les processus de test n'apportent de la valeur qu'en contribuant au succès global du projet ou du programme (ou en évitant un type de défaillance grave), le Test Manager doit gérer et contrôler les processus de test en conséquence. En d'autres termes, le Test Manager doit organiser les processus de test, y compris les activités et livrables associés, selon les besoins et les circonstances des autres parties prenantes, leurs activités (par ex. le cycle de développement logiciel dans lequel se produisent les tests), et leurs livrables (par ex. spécifications d'exigences)

### 2.2.1 Comprendre les Parties-prenantes du Test

Une personne est partie prenante du test lorsqu'elle a un intérêt dans les activités de test, les livrables du test, ou la qualité du système ou du livrable final. L'intérêt d'une partie prenante peut se caractériser par une implication directe ou indirecte dans les activités de test, la réception directe ou indirecte de livrables du test ou être affecté directement ou indirectement par la qualité des livrables produits par le projet ou programme.

Bien que les parties prenantes du test varient, selon le projet, le produit, l'organisation, et d'autres facteurs, elles incluent notamment les rôles suivants:

- Développeurs, responsables de développement, et chefs du développement. Ces parties prenantes implémentent le logiciel à tester, reçoivent les résultats de test, et sont souvent amenées à prendre des actions sur la base de ces résultats (par ex., faire corriger des défauts rapportés)
- Les architectes de bases de données, architectes système et concepteurs. Ces parties prenantes conçoivent le logiciel, reçoivent les résultats de test et sont souvent amenées à prendre des actions sur la base de ces résultats.
- Les analystes métier et Marketing. Ces parties prenantes déterminent les caractéristiques qui doivent être présentes dans le logiciel et leur niveau de qualité. Elles sont aussi souvent impliquées dans la définition de la couverture de test nécessaire, en révisant les résultats de test, et en prenant des décisions basées sur ces résultats.
- Les responsables Senior, les responsables de produit et sponsors de projet. Ces parties prenantes sont souvent impliquées dans la définition de la couverture de test nécessaire, en révisant les résultats de test, et en prenant des décisions basées sur ces résultats.
- Les gestionnaires de projet. Ces parties prenantes sont responsables de la gestion de leur projet pour en assurer le succès, ce qui demande des arbitrages entre la qualité, les délais, les fonctionnalités et les priorités budgétaires. Elles fournissent souvent les ressources

nécessaires aux activités de test et collaborent avec le Test Manager dans la planification et le contrôle des tests.

- Le support technique, le support client et l'équipe de soutien. Ces parties prenantes aident les utilisateurs et clients bénéficiant des fonctionnalités et de la qualité du logiciel développé.
- Les utilisateurs directs et indirects. Ces parties prenantes utilisent directement le logiciel (c.à.d, elles en sont les utilisateurs finaux), ou reçoivent des sorties ou services produits ou supportés par le logiciel.

Pour davantage d'information sur les parties prenantes du test, voir le chapitre 2 de [Goucher09].

Cette liste de parties prenantes n'est pas exhaustive. Les Test Managers doivent identifier les parties prenantes du test pour leur projet ou programme. Le Test Manager doit aussi comprendre la nature précise de la relation des parties prenantes avec les tests et comment l'équipe de test sert les besoins des parties prenantes. En plus d'identifier les parties prenantes du test comme décrit ci-dessus, le Test Manager devrait identifier les autres activités et livrables du cycle de développement logiciel qui affectent le test et/ou sont affectés par le test. Sans cela, le processus de test pourrait ne pas atteindre son efficacité et son rendement optimaux (Voir Section 2.2.3).

## 2.2.2 Autres Activités du Cycle de Vie du Développement Logiciel et Livrables

Comme le test de logiciels est une évaluation de la qualité d'un ou de plusieurs livrables produits en dehors des activités de test, il existe généralement dans le contexte plus large des activités du cycle de développement de logiciels. Le Test Manager doit organiser et conduire les activités de test en comprenant comment ces autres activités affectent le test, comme cela a été évoqué dans le syllabus Niveau Fondation, et comment le test affecte ces autres activités et leurs livrables.

Par exemple, dans les organisations utilisant les pratiques de développement Agile, les développeurs réalisent souvent du développement piloté par les tests, créant des tests unitaires automatisés, et intégrant continuellement le code (ainsi que les tests pour ce code) dans le système de gestion de configuration. Le Test Manager devrait travailler avec le responsable du développement pour s'assurer que les testeurs sont intégrés à ces activités et en phase avec elles. Les Testeurs peuvent réviser les tests unitaires à la fois pour émettre des suggestions visant à améliorer la couverture et l'efficacité de ces tests et pour obtenir une compréhension approfondie du logiciel et de son implémentation. Les testeurs peuvent évaluer des façons d'intégrer leurs propres tests automatisés, notamment les tests de non-régression fonctionnelle, dans le système de gestion de configuration. [Crispin09]

Bien que la relation particulière entre les activités de test, les autres parties prenantes du test, les activités du cycle de développement logiciel et les livrables, varie selon le projet, le cycle de développement logiciel choisi, et une variété d'autres facteurs, le test est étroitement interconnecté et lié aux processus suivants:

- Ingénierie et Gestion des Exigences. Durant la période de cadrage et d'estimation de l'effort de test, le Test Manager doit prendre en compte les exigences et également rester conscient de la possibilité de changements dans les exigences, et effectuer les actions de contrôle du test permettant l'adaptation à ces changements. Les Analystes Techniques de Test et les Analystes de Test devraient participer aux revues des exigences.
- Gestion de Projet. Le Test Manager, en travaillant avec des Analystes de Test et des Analystes Techniques de Test, doit fournir au Chef de Projet des exigences relatives au planning et aux ressources. Le Test Manager doit travailler avec le Chef de Projet pour comprendre les changements dans le plan de projet et effectuer les actions de contrôle du test permettant l'adaptation à ces changements.
- Gestion de Configuration, des Livraisons et du Changement. Le Test Manager, en travaillant avec l'équipe de test, doit établir les processus et mécanismes de livraison des objets de test, et les documenter dans le plan de test. Le Test Manager peut demander aux Analystes de



Test et aux Analystes Techniques de Test de créer des tests de vérification de la construction logicielle (build) et de s'assurer du contrôle des versions durant l'exécution des tests.

- Développement et Maintenance du logiciel. Le Test Manager devrait travailler avec les responsables de développement pour organiser la livraison des objets de test, incluant le contenu et les dates de chaque livraison en test, et également participer à la gestion des défauts (voir Chapitre 4).
- Support Technique. Le Test Manager devrait travailler avec le responsable du support technique pour garantir une livraison adéquate des résultats de test lors de la clôture des tests afin que les personnes impliquées dans le support du produit après sa livraison soient au courant des défaillances et contournements connus. En plus, le Test Manager devrait travailler avec le responsable du support technique pour analyser les défaillances survenues en production dans le but d'améliorer les processus de test.
- Production de documentation technique. Le Test Manager devrait travailler avec le responsable de la documentation technique pour garantir une livraison de la documentation pour le test dans les délais, et également la gestion des défauts trouvés dans ces documents.

En plus d'identifier les parties prenantes du test, comme décrit ci-dessus, le Test Manager doit identifier les autres activités et livrables du cycle de développement logiciel qui affectent le test et/ou sont affectés par le test. Sans cela, le processus de test peut ne pas atteindre son efficacité et son rendement optimaux.

### 2.2.3 Alignement des Activités de Test et autres Activités du Cycle de Vie

Le test devrait faire partie intégrante du projet, quels que soient les modèles de développement utilisés. Cela comprend:

- Les modèles séquentiels, tels que le modèle en cascade, le modèle en V et le modèle en W. Dans un modèle séquentiel, tous les livrables et activités d'une phase donnée (par ex. exigences, conception, implémentation, tests unitaires, tests d'intégration, tests système et tests d'acceptation) sont terminés avant le démarrage de la phase suivante. La planification des tests, l'analyse des tests, la conception des tests et l'implémentation des tests se déroulent de façon concurrente avec la gestion de projet, l'analyse métier/exigences, la conception du logiciel et de la base de données, et la programmation, avec un possible chevauchement dont la nature précise dépend du niveau de test considéré. L'exécution des tests se déroule de façon séquentielle selon les niveaux de test présentés dans le syllabus niveau Fondation et dans ce syllabus.
- Les modèles itératifs ou incrémentaux, comme RAD, le développement rapide d'applications (Rapid Application Development) ou RUP, le processus unifié Rational (Rational Unified Process). Dans un modèle itératif ou incrémental, les fonctionnalités à implémenter sont regroupées (par ex. selon la priorité métier ou le risque), et ensuite, les différentes phases de projet, y compris leurs livrables et activités, sont effectuées pour chaque groupe de fonctionnalités. Les phases peuvent se dérouler séquentiellement, ou avec un chevauchement, et, les itérations elles-mêmes peuvent être séquentielles ou se chevaucher. Lors du démarrage du projet, une planification des tests à haut niveau et l'analyse des tests se déroulent en parallèle de la planification du projet et de l'analyse métier/exigences. La planification détaillée des tests, l'analyse des tests, la conception des tests et l'implémentation des tests se produisent en parallèle au début de chaque itération. L'exécution des tests implique souvent des niveaux de test qui se recouvrent. Chaque niveau de test commence aussi tôt que possible et peut se prolonger après que des niveaux de test ultérieurs, supérieurs aient commencé.
- Les modèles Agiles, tels que SCRUM et XP (Extreme Programming). Ce sont des cycles de vie itératifs où les itérations sont très courtes (en général deux à quatre semaines). Les

livrables et activités de chaque itération sont terminés avant le début de l'itération suivante (c.à.d, les itérations sont séquentielles). Le Test se déroule de la même façon qu'avec des modèles itératifs mais avec un degré de chevauchement plus important des différentes activités de test avec les activités de développement, notamment un chevauchement très important de l'exécution des tests (à différents niveaux) avec les activités de développement. Toutes les activités d'une itération, y compris les activités de test, devraient être terminées avant le démarrage de l'itération suivante. Dans un projet Agile, le rôle de Test Manager passe généralement d'un rôle hiérarchique direct à un rôle d'autorité/conseil technique.

- Spirale. Dans un modèle en spirale, des prototypes sont utilisés tôt dans le projet pour confirmer la faisabilité et essayer des choix de conception et d'implémentation, en utilisant les niveaux de priorité métier et de risque technique pour déterminer l'ordre dans lequel les essais de prototypage seront effectués. Ces prototypes sont testés pour déterminer quels aspects des problèmes techniques demeurent non résolus. Une fois les principaux problèmes techniques résolus, le projet se déroule selon un modèle soit séquentiel, soit itératif.

Afin d'aligner correctement les activités de test dans le cycle de vie, le Test Manager doit avoir une bonne compréhension des modèles de cycles de vie utilisés dans son organisation. Par exemple, avec un modèle en V, les processus de test fondamentaux de l'ISTQB appliqués au niveau système pourraient se dérouler de la façon suivante:

- Les activités de planification du test système se déroulent en parallèle des activités de planification de projet, et le contrôle des tests continue jusqu'à la fin de l'exécution des tests systèmes et la clôture.
- Les activités d'analyse et conception des tests système se déroulent en parallèle de la spécification des exigences, de l'architecture système et de la conception de haut niveau du système, et de la spécification de la conception des composants (bas niveau).
- Les activités d'implémentation des tests système pourraient commencer pendant la conception du système, bien que l'essentiel de ces activités se produira en général de façon concurrente avec le codage et le test de composant, avec des activités d'implémentation des tests système se poursuivant souvent jusqu'à quelques jours avant le démarrage de l'exécution des tests système.
- Les activités d'exécution des tests système commencent lorsque les critères d'entrée pour le test système sont tous satisfaits (ou exemptés), ce qui signifie généralement qu'au moins le test de composant et souvent aussi les tests d'intégration de composant sont terminés. L'exécution des tests système se poursuit jusqu'à ce que les critères de sortie des tests système soient satisfaits.
- L'évaluation des critères de sortie et le reporting des résultats des tests système se font tout au long de l'exécution des tests système, généralement avec une fréquence et un degré d'urgence plus élevé à l'approche de la date de fin du projet.
- Les activités de clôture des tests système se produisent une fois que les critères de sortie des tests système sont satisfaits et que l'exécution des tests système est déclarée terminée, même si elles peuvent parfois être retardées jusqu'à ce que les tests d'acceptation et toutes les activités du projet soient terminés.

Dans un cycle de vie itératif ou incrémental, les mêmes tâches doivent être réalisées mais leur timing et leur ampleur peuvent varier. Par exemple, plutôt qu'être en mesure de mettre en œuvre un environnement de test complet au début du projet, il peut être plus efficace de ne mettre en place que la partie nécessaire à l'itération en cours. Avec un modèle de développement itératif ou incrémental, au plus tôt le planning a lieu, au plus on peut étendre le périmètre du processus de test fondamental.

En plus des phases de planification qui ont lieu sur chaque projet, l'exécution des tests et le reporting peuvent aussi être influencés par le cycle de vie utilisé par l'équipe. Par exemple, avec un cycle de vie itératif, il peut être utile de produire des rapports complets et de mener des revues de fin d'itération avant le démarrage de l'itération suivante. En considérant chaque itération comme un mini-projet,

l'équipe a l'opportunité de faire des corrections et des ajustements à partir de ce qui s'est passé lors de la précédente itération. Comme les itérations peuvent être courtes et bornées dans le temps, cela peut valoir la peine de diminuer la durée et l'effort mis sur ce reporting et cette évaluation, mais les tâches devraient être menées d'une façon permettant de suivre l'avancement global du test et d'identifier aussi vite que possible toute zone problématique. Les problèmes de processus identifiés sur une itération pourront facilement affecter voire même se reproduire sur l'itération suivante si des mesures correctives ne sont pas prises.

Des informations générales sur la façon d'aligner les activités de test avec les autres activités du cycle de vie peuvent être saisies dans la stratégie de test (voir Section 2.4.2). Le Test Manager devrait effectuer les ajustements spécifiques au projet, pour chaque niveau de test et pour toute combinaison retenue de cycle de développement logiciel et de processus de test, durant la planification des tests et/ou la planification du projet.

Selon les besoins de l'organisation, du projet et du produit, des niveaux de test supplémentaires à ceux définis dans le syllabus Niveau Fondation peuvent être requis, par exemple:

- Tests d'intégration matériel-logiciel
- Tests d'intégration système
- Tests d'interaction de fonctionnalité
- Tests d'intégration des produits client

Chaque niveau de test devrait avoir les éléments suivants clairement définis:

- Objectifs de test, avec des buts atteignables
- Périmètre du test et éléments de test
- Base de test, avec un moyen de mesurer la couverture de cette base (c.à.d., la traçabilité)
- Critères d'entrée et de sortie
- Livrables du test, incluant le rapport des résultats de test
- Techniques de test applicables, ainsi qu'une manière pour assurer un degré de couverture avec ces techniques
- Mesures et métriques adaptées aux objectifs de test, aux critères d'entrée et de sortie, et reporting des résultats (y compris mesures de couverture)
- Outils de test à appliquer pour certaines tâches de test (si et où applicable)
- Ressources (par ex., environnements de test)
- Personnes et groupes responsables, à la fois au sein et hors de l'équipe de test
- Conformité avec les standard de l'organisation, les standards légaux et d'autres standards (lorsque cela est possible)

Comme abordé plus loin dans ce chapitre, la meilleure pratique consiste à définir ces éléments de façon cohérente pour tous les niveaux de test pour éviter des pertes et écarts dangereux entre les différents niveaux de tests similaires.

## 2.2.4 Gérer les Tests Non Fonctionnels

Ne pas gérer correctement les tests non-fonctionnels peut aboutir à la découverte de problèmes qualité sérieux, parfois désastreux, dans le système après livraison. Cependant, de nombreux types de tests non-fonctionnels sont coûteux et le Test Manager doit sélectionner quels tests non-fonctionnels exécuter en fonction des risques et des contraintes. De plus, il y a de nombreux types de tests non-fonctionnels et certains peuvent ne pas être adaptés à une application donnée.

Comme le Test Manager peut ne pas avoir l'expertise suffisante pour traiter toutes les questions de planification, le Test Manager peut avoir besoin de déléguer une partie des responsabilités de planification des tests aux Analystes Techniques de Test (et parfois aux Analystes de Test) affectés

aux activités de test non-fonctionnel. Le Test Manager devrait demander aux analystes de considérer les facteurs généraux suivants:

- Exigences des parties prenantes
- Outillage nécessaire
- Environnement de test
- Facteurs organisationnels
- Sécurité

Pour plus de détails, voir le Chapitre 4 du syllabus avancé Analyste Technique de Test [ISTQB ATTA SYL].

Une autre considération importante pour le Test Manager est comment intégrer les tests non-fonctionnels dans le cycle de développement logiciel. Une erreur commune consiste à attendre que tous les tests fonctionnels soient terminés avant de commencer les tests non-fonctionnels, ce qui peut résulter en la découverte tardive de défauts non-fonctionnels critiques. Au lieu de ça, les tests non-fonctionnels devraient être priorisés et ordonnés selon le risque. Il y a souvent des façons de réduire les risques non-fonctionnels dans les niveaux de test précoces ou même durant le développement. Par exemple, des revues d'utilisabilité de prototypes d'interface utilisateur peuvent être très utiles pour découvrir des défauts d'utilisabilité qui créeraient des problèmes de planning significatifs s'ils étaient découverts vers la fin des tests système.

Dans les cycles de vie itératifs, le rythme du changement et des itérations peuvent rendre difficile de se focaliser sur certains tests non-fonctionnels demandant la construction d'un cadre de test sophistiqué. Les activités de conception et d'implémentation des tests prenant plus de temps qu'une itération devraient être organisées comme des activités séparées, hors de ces itérations.

## 2.2.5 Gérer les Tests Basés sur l'Expérience

Bien que les tests basés sur l'expérience fournissent des avantages en trouvant efficacement des défauts que d'autres techniques pourraient manquer et servent à vérifier la complétude de ces techniques, ils apportent des challenges pour la gestion des tests. Le Test Manager devrait être conscient des risques autant que des bénéfices des techniques basées sur l'expérience, en particulier le test exploratoire. Il n'est pas évident de déterminer la couverture atteinte durant de tels tests, en raison de la documentation (logging) limitée et d'une préparation réduite des tests. La reproductibilité des résultats de test demande une attention particulière du Test Manager, en particulier quand plusieurs testeurs sont impliqués.

Une façon de gérer les tests basés sur l'expérience, spécialement les tests exploratoires, est de diviser le travail de test en petites périodes de 30 à 120 minutes parfois appelées sessions de test. Ce découpage limite et concentre le travail à faire dans une session et permet un niveau de contrôle et d'organisation. Chaque session couvre une « charte » qui est communiquée par écrit ou verbalement au testeur par le Test Manager. La charte de test donne la (les) condition(s) de test à couvrir durant la session de test, ce qui permet de mieux focaliser les tests et d'éviter des recouvrements si plusieurs personnes effectuent du test exploratoire simultanément.

Une autre technique permettant de gérer les tests basés sur l'expérience est d'intégrer cette manière spontanée et autodirigée de tester à des sessions préconçues de tests plus traditionnelles. Par exemple, les testeurs peuvent avoir la permission (et le temps) pour explorer au-delà des étapes explicites, des données d'entrée et des résultats attendus de leurs tests prédéfinis. Les testeurs peuvent aussi se voir affecter des sessions autodirigées dans le cadre de leurs activités journalières, avant, pendant ou après une journée d'exécution de tests prédéfinis. Si de telles sessions de test permettent de trouver des défauts ou des endroits à tester davantage, alors les tests prédéfinis peuvent être mis à jour.

Au début de la session exploratoire, le testeur s'assure et exécute les tâches de mise en place nécessaire aux tests. Pendant la session, le testeur apprend l'application à tester, conçoit et exécute des tests selon la technique appliquée et selon ce qui a été appris sur l'application, le testeur analyse les défauts, et capture les résultats des tests dans un log (s'il est nécessaire de pouvoir répéter les tests, le testeur devrait aussi documenter les entrées, les actions et les événements). Après la session, un debriefing peut avoir lieu, ce qui permet de définir la direction des sessions suivantes.

## 2.3 Test Basé sur les Risques et Autres Approches pour la Priorisation des Tests et l'Allocation de l'Effort

Un challenge universel de la Gestion des Tests est la bonne sélection, allocation et priorisation des tests. C'est à dire que, à partir d'un nombre pratiquement infini de conditions de test et de combinaisons de conditions qui devraient être couvertes, l'équipe de test doit sélectionner un ensemble fini de conditions, déterminer la quantité appropriée d'effort à affecter de façon à couvrir chaque condition avec des cas de tests, et ordonner les cas de test résultant selon un ordre de priorisation qui optimise l'efficacité et la rentabilité du travail de test à faire. L'identification et l'analyse des risques, ainsi que d'autres facteurs, peuvent être utilisées par le Test Manager pour aider à résoudre ce problème, bien que de nombreuses contraintes interagissent et que des variables puissent nécessiter une solution de compromis.

### 2.3.1 Test Basé sur les Risques

Le risque est la possibilité d'un événement négatif ou non désiré. Les risques existent chaque fois qu'un quelconque problème peut survenir et dégrader la perception de qualité du produit ou le succès du projet pour un client, un utilisateur ou une autre partie prenante. Lorsque l'effet initial d'un problème potentiel est sur la qualité du produit, alors on parle de risques qualité, de risque produit, ou de risque qualité du produit. Lorsque l'effet initial d'un problème potentiel impacte le succès du projet, on parle de risque projet ou de risque de planning.

Dans le test basé sur les risques, les risques qualité sont identifiés et évalués lors d'une analyse des risques qualité du produit avec les parties prenantes. Ensuite, l'équipe de test conçoit, implémente et exécute les tests pour mitiger les risques qualité. La qualité inclut la totalité des fonctionnalités, comportements, caractéristiques et attributs affectant les clients, les utilisateurs et la satisfaction des parties prenantes. Ainsi, un risque qualité est une situation potentielle où des problèmes qualité pourraient exister dans un produit. Des exemples de risques qualité pour un système incluent, des erreurs de calcul dans des rapports (un risque fonctionnel lié à la précision), des temps de réponse lents suite à une saisie utilisateur (un risque non-fonctionnel lié à l'efficacité et au temps de réponse), et une difficulté à comprendre des écrans ou des champs de saisie (un risque non fonctionnel lié à l'utilisabilité et à la facilité de compréhension). Quand des tests découvrent des défauts, les tests auront réduit les risques qualité en les découvrant et en permettant de les traiter avant la livraison. Si les tests ne découvrent pas de défauts, le test aura mitigé les risques qualité en assurant que, dans les conditions du test, le système fonctionne correctement.

Le test basé sur les risques utilise les risques qualité pour sélectionner les conditions de test, allouer l'effort de test pour ces conditions, et prioriser les cas de test correspondants. Différentes techniques existent pour le test basé sur les risques, avec des différences significatives à la fois dans le type et le niveau de documentation collectée et dans le niveau de formalisme appliqué. Que ce soit de façon explicite ou de façon implicite, le test basé sur les risques a pour objectif d'utiliser les tests pour réduire le niveau de risque qualité global, et plus précisément, de réduire ce niveau de risque jusqu'à un niveau acceptable.

Les tests basés sur les risques consistent en quatre activités principales:

- L'identification des risques
- L'évaluation des risques
- La mitigation des risques
- La gestion des risques

Ces activités se recouvrent. Les sous-sections suivantes décriront chacune d'elles.

Pour être le plus efficace, l'identification et l'évaluation des risques devraient impliquer des représentants de tous les groupes de parties prenantes du projet et du produit, cependant parfois les réalités du projet font que certaines parties prenantes agissent comme substitut d'autres parties prenantes. Par exemple, dans un développement logiciel grand public, un petit échantillon de clients potentiels peut se voir demander d'identifier des défauts potentiels qui impacteraient le plus leur utilisation du logiciel ; dans ce cas, l'échantillon de clients potentiels sert de substitut à l'ensemble de la base clients. Du fait de leur expertise particulière avec les risques qualité du produit et les défaillances, les testeurs devraient être activement impliqués dans le processus d'identification et évaluation des risques.

#### **2.3.1.1 Identification des Risques**

Les parties prenantes peuvent identifier les risques via une ou plusieurs des techniques suivantes:

- Entretiens avec des experts
- Évaluations indépendantes
- Utilisation de templates de risque
- Rétrospectives de projets
- Groupe de travail sur les risques
- Brainstorming
- Check-lists
- Rappel d'expérience passées

En impliquant un échantillon le plus large possible de parties prenantes, le processus d'identification des risques aura le plus de chance de trouver les risques qualité de produit les plus significatifs.

L'identification des risques produit souvent d'autres livrables, p.ex., l'identification de problèmes qui ne sont pas des risques qualité de produit. Cela peut inclure des questions générales ou des soucis sur le produit ou le projet, ou des problèmes dans les documents référencés comme les exigences ou les spécifications de conception. Des risques projet sont aussi souvent identifiés lors de l'identification des risques qualité mais ne sont pas l'objet principal des tests basés sur les risques. Cependant, la gestion des risques projet est importante pour tout le test, pas seulement pour les tests basés sur les risques, et ceci est discuté plus avant dans la Section 2.4.

#### **2.3.1.2 Evaluation des Risques**

Une fois les risques identifiés, l'évaluation des risques, qui correspond à l'étude de ces risques, peut commencer. Spécifiquement, l'analyse des risques implique la catégorisation de chaque risque et la détermination de sa probabilité et de son impact. L'évaluation des risques peut aussi impliquer l'évaluation ou l'affectation de différentes propriétés à chaque risque, comme le propriétaire du risque.

La catégorisation des risques signifie affecter à chaque risque un type approprié tel que performances, fiabilité, fonctionnalité et ainsi de suite. Certaines organisations utilisent les caractéristiques qualité du standard ISO 9126 [ISO9126] (qui est en cours de remplacement par le standard [ISO25000]), mais beaucoup d'organisations utilisent d'autres schémas de catégorisation. La check-list d'identification des risques est souvent utilisée pour catégoriser ces risques. Il existe des check-lists génériques pour les risques qualité et beaucoup d'organisations les adaptent. Lorsque des

check-lists sont utilisées comme base pour identifier les risques, la catégorisation intervient souvent durant l'identification.

Déterminer le niveau de risque implique généralement l'évaluation, pour chacun des risques, de la probabilité d'occurrence et de l'impact lié à l'occurrence. La probabilité d'occurrence est la probabilité que le problème potentiel existe dans le système en cours de test. En d'autres termes, la probabilité est l'évaluation du niveau du risque technique. On peut citer comme facteurs influençant la probabilité pour les risques produit et projet:

- La complexité de la technologie et des équipes
- Les questions de personnel et de formation entre les analystes métier, les concepteurs et programmeurs
- Les conflits au sein de l'équipe
- Les problèmes contractuels avec les fournisseurs
- La répartition géographique d'une équipe distribuée
- Les anciens systèmes contre les nouvelles approches
- Les outils et les technologies
- Une mauvaise gestion technique ou managériale
- Des pressions liées aux délais, aux ressources et au budget
- L'absence d'assurance qualité précoce
- Un fort taux de changement
- Un taux de défauts précoces élevé
- Les questions d'interface et d'intégration

L'impact en cas d'occurrence est la sévérité de l'effet sur les utilisateurs, clients ou autres parties prenantes. Parmi les facteurs influençant les risques projet et produit, on peut citer:

- La fréquence d'utilisation de la fonctionnalité affectée
- La criticité de la caractéristique dans l'accomplissement de l'objectif métier
- La dégradation de la réputation
- La perte commerciale
- Les pertes financières, écologiques, sociales ou juridiques potentielles
- Des sanctions légales, civiles ou pénales
- La perte de licence
- L'absence de solutions de contournement acceptables
- La visibilité de la défaillance générant une contre publicité
- La sûreté et la sécurité

Le niveau de risque peut être évalué soit quantitativement, soit qualitativement. Si probabilité et impact peuvent être évalués quantitativement, les deux valeurs peuvent être multipliées l'une par l'autre pour calculer une priorité quantitative du risque. Typiquement cependant, le niveau de risque ne peut être évalué que de façon qualitative. Ceci dit, il est possible de parler de priorité très haute, haute, moyenne, faible ou très faible, mais une priorité ne peut être exprimée par un pourcentage précis; de même, il est possible de parler d'impact très haut, haut, moyen, faible ou très faible, mais l'impact ne peut être exprimé en termes financiers de façon précise et complète. Cette évaluation qualitative des niveaux de risque ne devrait pas être considérée comme moins valable que les approches quantitatives; en effet, si l'évaluation quantitative des niveaux de risque n'est pas utilisée de façon appropriée, les résultats tromperont les parties prenantes concernant la façon de comprendre et de gérer les risques. Les évaluations qualitatives des niveaux de risque sont souvent combinées, par des multiplications ou des additions pour créer un score de risque agrégé. Ce score de risque agrégé peut être exprimé comme un numéro de priorité du risque mais devrait être interprété comme une valeur qualitative par rapport à une échelle ordinale.

De plus, à moins que l'analyse de risque ne soit basée sur des données de risques extensives et statistiquement valides, l'analyse de risques devra se baser sur l'appréciation subjective des parties prenantes en ce qui concerne la probabilité et l'impact. Les chefs de projet, programmeurs, utilisateurs, analystes métiers, architectes et testeurs ont typiquement des perceptions différentes et ainsi probablement des options différentes sur le niveau de risque à attribuer à chaque risque. Le processus d'analyse de risque devrait inclure une manière d'atteindre un consensus ou, dans le pire des cas, d'établir un niveau de risque convenu (par ex., selon une consigne de la direction ou en prenant la moyenne, ou la médiane du niveau proposé pour le risque). De plus, les niveaux de risque devraient être vérifiés pour s'assurer d'une bonne distribution sur leur échelle de valeurs et pour assurer que les taux de risques apportent des informations utiles en termes d'ordonnement, de priorisation et d'allocation de l'effort de test. Sinon, les niveaux de risque ne pourront pas être utilisés comme guides lors des activités de réduction des risques.

### 2.3.1.3 Mitigation des Risques

Le test basé sur les risques commence par une analyse des risques qualité (identification et évaluation des risques qualité du produit). Cette analyse est la base du plan de test maître et des autres plans de test. Comme indiqué dans le ou les plan(s), les tests sont conçus, implémentés et exécutés de façon à couvrir les risques. L'effort associé au développement et à l'exécution d'un test est proportionnel au niveau de risque, ce qui signifie que des techniques de test plus méticuleuses (comme le test par paires) sont utilisées pour les risques les plus élevés, alors que des techniques moins méticuleuses (telles que les partitions d'équivalence ou le test exploratoire par périodes de temps) seront utilisées pour des risques moins élevés. De plus, la priorité du développement et de l'exécution du test est basée sur le niveau de risque. Certains standards liés à la sûreté (par ex., FAA DO-178B/ED 12B, IEC 61508), prescrivent les techniques de test et le degré de couverture en fonction du niveau de risque. De plus, le niveau de risque doit influencer les décisions comme l'utilisation de revues des livrables du projet (incluant les tests), le niveau d'indépendance, le niveau d'expérience des testeurs et le degré de test de confirmation (re-test) et de test de non-régression à réaliser.

Durant le projet, l'équipe de test devrait rester attentive à toute information complémentaire pouvant modifier l'ensemble des risques qualité et/ou le niveau de risque associé aux risques qualité connus. Des ajustements périodiques de l'analyse des risques qualité, entraînant des ajustements dans les tests, devraient être effectués. Ces ajustements devraient avoir lieu au minimum lors de chaque jalon majeur du projet. Les ajustements incluent l'identification de nouveaux risques, la réévaluation du niveau des risques existants et l'évaluation de l'efficacité des activités de mitigation des risques. Pour prendre un exemple, si une session d'identification et évaluation des risques a eu lieu sur la base des spécifications d'exigences, une fois la spécification finalisée, une réévaluation des risques devrait avoir lieu. Pour prendre un autre exemple, si pendant les tests un composant est identifié comme contenant considérablement plus d'erreurs que prévu, on peut conclure que la probabilité de défauts à cet endroit a été plus élevée qu'anticipé et par conséquent ajuster à la hausse la probabilité et le niveau de risque global. Cela pourrait résulter en un accroissement de l'effort de tests à exécuter pour ce composant.

Les risques qualité du produit peuvent aussi être mitigés avant le début de l'exécution des tests. Par exemple, si des problèmes avec les exigences sont identifiés pendant l'identification des risques, l'équipe projet peut implémenter des revues de spécifications d'exigence comme action de mitigation. Cela peut réduire le niveau de risque, ce qui pourra signifier que moins de tests seront nécessaires pour mitiger les risques qualité restants.

### 2.3.1.4 Gestion des Risques dans le Cycle de Vie

Idéalement, la gestion des risques à lieu tout au long du cycle de vie. Si une organisation a un document de politique de test et/ou de stratégie de test, alors ceux-ci devraient décrire le processus général de gestion des risques produit et des risques projet par le test, et montrer comment la gestion des risques est intégrée et affecte toutes les étapes du test.



Dans une organisation mature où la sensibilisation aux risques imprègne l'équipe de projet, la gestion des risques est présente à différents niveaux et pas seulement pendant le test. Les risques importants ne sont pas seulement traités plus tôt lors d'un niveau de test particulier, mais aussi lors de niveaux de test précédents (Par exemple, si les performances sont perçues comme un fort risque qualité, les tests de performance apparaîtront au début des tests système mais des tests de performance seront exécutés pendant les tests unitaires et les tests d'intégration). Les organisations matures non seulement identifient les risques, mais aussi les sources de risques et les conséquences de ces risques s'ils devaient se produire. Pour les défauts qui apparaissent, une analyse des causes racines est utilisée pour mieux comprendre les sources de risques et implémenter des améliorations de processus qui préviendront l'apparition des défauts. La mitigation existe tout au long du cycle de développement logiciel. L'analyse de risque est complètement informée, prenant en compte le travail associé, l'analyse du comportement du système, l'évaluation des risques basée sur les coûts, l'analyse des risques produit, l'analyse des risques des utilisateurs finaux et l'analyse des responsabilités. L'analyse des risques transcende le test, avec l'équipe de test participant à et influençant une analyse de risque au niveau du programme.

La plupart des méthodes du test basé sur les risques comprennent aussi des techniques pour utiliser le niveau de risque pour ordonnancer et prioriser le test, assurant ainsi une couverture précoce des zones les plus importantes et une découverte des défauts les plus importants pendant l'exécution des tests. Dans certains cas, tous les tests correspondant à un risque de haut niveau sont exécutés avant les tests couvrant des niveaux de risques plus faibles, et les tests sont exécutés strictement dans l'ordre du niveau de risque (souvent appelé "parcours en profondeur"); dans d'autres cas, une méthode d'échantillonnage est utilisée pour sélectionner un échantillon de tests parmi tous ceux identifiés en utilisant les risques pour pondérer la sélection tout en s'assurant en même temps que chaque élément de risque est couvert au moins une fois (souvent appelé "parcours en largeur").

Que le test basé sur les risques effectue un parcours en profondeur ou un parcours en largeur, il est possible que le temps alloué au test soit épuisé avant que tous les tests soient exécutés. Le test basé sur les risques permet aux testeurs d'informer leur hiérarchie en terme de niveau de risque résiduel à ce moment, et permet à la hiérarchie de décider de prolonger le test ou de transférer les risques restants aux utilisateurs, clients, équipes de support/soutien technique et/ou opérationnelles.

Pendant l'exécution des tests, les techniques de test basé sur les risques les plus sophistiquées — qui n'ont pas besoin d'être les plus formelles ou les plus lourdes — permettent aux participants du projet, aux responsables projet et produit, à la hiérarchie, aux responsables sénior et aux parties prenantes du projet, de mesurer et contrôler le cycle de développement logiciel, en prenant les décisions de livraison sur la base du niveau de risque résiduel. Cela demande à ce que le Test Manager communique les résultats des tests en termes de risques de façon compréhensible par chacune des parties prenantes.

### 2.3.2 Test Basé sur les Risques Techniques

Il y a un grand nombre de techniques pour le test basé sur les risques. Certaines sont très informelles. Il s'agit par exemple des approches où les testeurs analysent les risques qualité pendant du test exploratoire [Whittaker09]. Cela peut aider à orienter le test mais peut aboutir à une focalisation excessive portée à la probabilité des défauts, plutôt qu'à leur impact, et n'inclut pas les entrées des parties prenantes aux profils fonctionnels différents. De plus, de telles approches sont subjectives et dépendent des compétences, de l'expérience et des préférences de chaque testeur. Ainsi, ces approches n'atteignent que rarement l'ensemble des bénéfices du test basé sur les risques.

De manière à capturer les bénéfices du test basé sur les risques tout en minimisant les coûts, beaucoup de testeurs emploient des approches allégées de test basées sur les risques. Celles-ci mélangent la réactivité et la flexibilité des approches informelles avec la puissance et l'intérêt des



aspects consensuels des approches plus formelles. Parmi les approches allégées, on peut citer: Analyse et Gestion Pragmatique des Risques (Pragmatic Risk Analysis and Management ou PRAM) [Black09], Test Logiciel Systématique (Systematic Software Testing ou SST) [Craig02], et la Gestion du Risque Produit (Product Risk Management ou PRisMa) [vanVeenendaal12]. En plus des attributs habituels du test basé sur les risques, ces techniques présentent typiquement les attributs suivants:

- Evolution au cours du temps sur la base d'une expérience industrielle du test basé sur les risques, particulièrement dans les industries où les questions d'efficacité sont importantes
- Prédicatives, par une implication extensive des équipes fonctionnelles transverses de parties prenantes, représentant des perspectives à la fois métier et techniques, pendant la phase d'identification initiale des risques et de leur évaluation
- Optimisées quand introduites lors des premières phases d'un projet, quand les options de mitigation de risques qualité sont maximisées et quand les principaux livrables et produits dérivés de l'analyse de risque peuvent aider à influencer les spécifications et l'implémentation du produit de façon à minimiser les risques
- Utilisent les sorties générées (matrice de risques ou tableau d'analyse des risques) comme base pour le plan de test et les conditions de test, et par conséquent pour toutes les autres activités de gestion et analyse des tests
- Supportent la fourniture de rapports d'exécution des tests, en termes de risques résiduels, à tous les niveaux de parties prenantes.

Certaines de ces techniques (par ex., SST) nécessitent les spécifications d'exigences en entrée de l'analyse de risque et ne peuvent être mises en œuvre si ces spécifications ne sont pas fournies. D'autres techniques (par ex., PRisMa et PRAM) encouragent l'utilisation d'une stratégie mélangeant les risques et les exigences, en utilisant les exigences et/ou autres spécifications en entrée des analyses de risques, mais peuvent fonctionner entièrement sur base des éléments fournis par les parties prenantes. L'utilisation des exigences comme entrant aide à assurer une bonne couverture des exigences, mais le Test Manager doit s'assurer que les risques importants non suggérés par les exigences – spécialement dans les zones non-fonctionnelles – ne soient pas oubliés. Lorsque des exigences bien formulées et priorisées sont fournies en entrée, on voit typiquement une forte corrélation entre les niveaux de risque et la priorité des exigences.

Beaucoup de ces techniques encouragent aussi l'utilisation de l'identification et de l'évaluation des risques comme moyen pour créer un consensus entre les parties prenantes sur approche de test. Ceci est un bénéfice important, mais requiert des parties prenantes du temps pour participer soit à des groupes de réflexion, soit à des entretiens individuels. Une participation insuffisante des parties prenantes aboutira à des trous dans l'analyse de risque. Bien sûr, les parties prenantes ne sont pas toujours d'accord avec le niveau de risque, donc la personne dirigeant l'analyse des risques qualité doit travailler de façon créative et positive avec les parties prenantes pour atteindre le meilleur degré d'entente possible. Toutes les compétences d'un modérateur formé à l'animation de réunions de revue sont applicables pour la personne responsable de l'analyse des risques qualité.

Comme les techniques plus formelles, les techniques allégées permettent l'utilisation de facteurs de pondération de la probabilité et de l'impact pour mettre en avant des risques métier ou techniques (selon le niveau de test, par exemple). Cependant, à la différence de techniques plus formelles, les techniques allégées:

- utilisent seulement deux facteurs, probabilité et impact, et
- utilisent des jugements et échelles simples et qualitatifs.

La nature légère de ces approches apporte de la flexibilité, les rend applicables dans plusieurs domaines et accessibles au sein des équipes de tous niveaux d'expérience et de compétences (même avec des personnes junior et non techniques).

Du côté formel, et plus lourd des méthodes, le Test Manager a à sa disposition de nombreuses options:

- Analyse des dangers, qui étend le processus analytique vers l'amont afin d'identifier les dangers sous-jacents à chaque risque.
- Coût d'exposition, où le processus d'évaluation des risques implique de déterminer, pour chaque risque qualité, trois facteurs: 1) La probabilité (exprimée en pourcentage) d'une défaillance associée à un risque, 2) le coût de la perte (exprimé avec un montant financier) associée aux défaillances provoquées par un risque qui se produirait en production; et, 3) le coût du test de telles défaillances.
- Analyse des Modes de Défaillances et Effets (AMDE) (Failure Mode and Effect Analysis ou FMEA) et ses variantes [Stamatis03], où les risques qualité, avec leur causes et effets potentiels, sont identifiés et où les regroupements en termes de sévérité, priorité et détection sont renseignés.
- Déploiement Qualité de Fonctions (DQF) (Quality Function Deployment ou QFD), qui est une technique de gestion des risques qualité impliquant le test, et traitant en particulier des risques qualité résultants d'une compréhension incorrecte ou déficiente des exigences client ou utilisateur.
- Analyse par Arbre de Défaillance (Fault Tree Analysis ou FTA), où différentes défaillances observées (en test ou en production) ou potentielles (risques qualité), font l'objet d'une analyse des causes racines commençant avec les défauts pouvant provoquer la défaillance, puis les erreurs ou défauts pouvant causer ces défauts, et ceci jusqu'à ce que les différentes causes racines soient identifiées.

Les techniques particulières à utiliser pour le test basé sur les risques, et le degré de formalisme de chaque technique dépendent du projet, du processus et du produit. Par exemple, une approche informelle comme la technique exploratoire de Whittaker pourrait s'appliquer à un patch ou un correctif. Dans des projets Agile, l'analyse des risques qualité est entièrement intégrée au début de chaque période ou sprint et la documentation des risques est associée au suivi des user stories. Les Systèmes de Systèmes nécessitent une analyse de risque par système mais aussi pour le Système de Système complet. Des projets critiques en termes de sécurité ou de mission demanderont des niveaux plus élevés de formalisme et de documentation.

Les entrées, processus et sorties pour du test basé sur les risques tendront en général à être déterminés par la technique sélectionnée. Les entrées habituelles incluent le point de vue des parties prenantes, les spécifications et des données historiques. Les processus habituels sont l'identification des risques, l'évaluation des risques, et le contrôle des risques. Les sorties habituelles sont une liste de risques qualité (avec un niveau de risque associé et l'effort de test recommandé), les défauts identifiés dans les documents d'entrée tels que les spécifications, des questions ou soucis liés aux éléments de risques, et les risques projet affectant le test ou le projet dans son ensemble.

Généralement, le facteur de succès critique pour du test basé sur les risques est l'implication d'une bonne équipe de parties prenantes dans l'identification et l'évaluation des risques. Toutes les parties prenantes ont leurs propres compréhensions de ce qui constitue la qualité pour le produit, et leurs propres priorités et inquiétudes concernant la qualité. Cependant, les parties prenantes tendent à se ranger en deux grandes catégories: les parties prenantes métier et les parties prenantes techniques.

Les parties prenantes métier incluent, entre autres, les clients, les utilisateurs, l'équipe opérationnelle, le support client et le support technique. Ces parties prenantes comprennent les clients et les utilisateurs et sont donc en mesure d'identifier les risques et d'évaluer leurs impacts du point de vue métier.

Les parties prenantes techniques incluent, entre autres, les développeurs, les architectes, les administrateurs de bases de données et les administrateurs réseau. Ces parties prenantes

comprennent les manières par lesquelles le logiciel peut défaillir et ainsi identifier les risques et estimer leur probabilité d'un point de vue technique.

Certaines parties prenantes ont une vision à la fois métier et technique. Par exemple, des experts d'un sujet, qui ont des rôles de testeurs ou analystes métier auront souvent une vision plus large des risques grâce à leur expertise technique et métier.

Le processus d'identification des éléments à risque générera une liste substantielle de risques. Il n'est pas nécessaire pour les parties prenantes de débattre au sujet des risques; dès qu'une partie prenante perçoit quelque chose comme un risque pour la qualité du système, c'est un élément à risque. Cependant, il est important que les parties prenantes arrivent à un consensus sur la valorisation du niveau de risque. Par exemple, dans des approches allégées utilisant la probabilité et l'impact comme facteurs de valorisation, une partie du processus doit inclure l'identification d'un schéma de valorisation commun des probabilités et des impacts. Toutes les parties prenantes, y compris l'équipe de test, doivent utiliser cette même échelle de valorisation et être capables de converger vers une même valorisation de la probabilité et de l'impact pour chaque élément de risque qualité.

Si du test basé sur les risques est utilisé à long terme, alors le Test Manager doit défendre et initier du test basé sur les risques auprès des parties prenantes. Ce groupe fonctionnel transverse doit voir la valeur de l'analyse de risque pour assurer une utilisation continue de la technique. Ceci exige que le Test Manager comprenne les besoins et les attentes des parties prenantes ainsi que leur disponibilité pour participer au processus.

Un bon engagement de la part des parties prenantes dans le processus d'analyse des risques offrira un bénéfice important au Test Manager. Ce bénéfice est que, sur des projets sous-spécifiés ou ayant des exigences imprécises ou manquantes, les parties prenantes peuvent encore identifier les risques quand elles sont guidées par une check-list adaptée. Ce bénéfice peut être vu quand, après l'implémentation de test basé sur les risques, l'efficacité de la détection des défauts par l'équipe de test s'améliore. Ceci est possible parce qu'une base de test plus complète – dans ce cas la liste des risques qualité – est utilisée.

Durant les activités de clôture des tests pour du test basé sur les risques, les équipes de test devraient mesurer dans quelle mesure ils ont obtenu les bénéfices attendus. Dans de nombreux cas, cela implique de répondre, par le biais de métriques et en consultation avec l'équipe, à certaines des questions suivantes, ou à toutes :

- L'équipe de test a-t-elle détecté un pourcentage plus élevé de défauts importants que de défauts peu importants?
- L'équipe de test a-t-elle trouvé les défauts les plus importants au début de la période d'exécution des tests?
- L'équipe de test a-t-elle été en mesure d'expliquer aux parties prenantes les résultats de test en termes de risques?
- Les tests sautés par l'équipe de test (s'il y en a) avaient-ils un niveau de risque associé plus bas que ceux exécutés?

Dans la plupart des cas, un test basé sur les risques mené avec succès aboutit à des réponses positives pour chacune de ces quatre questions. Sur le long terme, le Test Manager devrait définir des objectifs d'amélioration de processus pour ces métriques et veiller à améliorer l'efficacité du processus d'analyse des risques qualité. Bien sûr, d'autres métriques et critères de succès peuvent être appliqués aux tests basés sur les risques, et le Test Manager devrait considérer attentivement les relations entre ces métriques, les objectifs stratégiques servis par l'équipe de test, et les comportements qui résulteront d'une gestion basée sur l'usage d'un ensemble particulier de métriques et de critères de succès.

### 2.3.3 Autres Techniques de Sélection des Tests

Bien que beaucoup de Test Managers utilisent du test basé sur les risques comme un élément de leur stratégie de test, beaucoup utilisent aussi d'autres techniques.

L'une des principales techniques alternatives pour développer et prioriser les conditions de test est le test basé sur les exigences. Le test basé sur les exigences peut utiliser différentes techniques, telles que les revues d'ambiguïté, l'analyse des conditions de test, et les graphes cause à effet. Les revues d'ambiguïté identifient et éliminent les ambiguïtés dans les exigences (qui servent de base de test), souvent en utilisant une check-list des défauts communs dans les exigences (voir [Wiegers03]).

Comme décrit dans [Craig02], l'analyse des conditions de test implique une lecture attentive de la spécification des exigences pour identifier les conditions de test à couvrir. Si une priorité a été affectée à ces exigences, elle peut être utilisée pour allouer l'effort et prioriser les cas de tests. En l'absence d'une affectation de priorité, il est difficile de déterminer l'effort approprié et l'ordre des tests sans fusionner le test basé sur les exigences avec le test basé sur les risques.

Les graphes de cause à effet sont discutés dans le module Analyste de Test Avancé dans un contexte de couverture des combinaisons de conditions de test lors de l'analyse de test. Cependant, ils ont un usage plus large en ce qu'ils permettent de réduire un problème de test très étendu en un nombre gérable de cas de test, tout en fournissant 100% de couverture fonctionnelle de la base de test. Les graphes de cause à effet identifient aussi les lacunes dans les bases de test lors de la conception des tests, ce qui permet d'identifier des défauts tôt dans le cycle de développement logiciel quand la conception des tests débute à partir d'exigences non finalisées. Un obstacle majeur à l'adoption des graphes de cause à effet est la complexité de leur création. Il existe des outils qui supportent cette méthode laquelle peut être compliquée à faire manuellement.

Un obstacle courant aux tests basés sur les exigences est que les spécifications d'exigences sont souvent ambiguës, non testables, incomplètes ou inexistantes. Toutes les organisations ne sont pas motivées pour résoudre ces problèmes, donc les testeurs confrontés à de telles situations doivent sélectionner une autre stratégie.

Une autre méthode parfois utilisée pour augmenter l'utilisation des exigences disponibles repose sur la création de profils opérationnels ou d'usage, une approche basée sur les modèles, qui utilise un mélange de cas d'utilisation, d'utilisateurs (parfois appelés personnes), d'entrées et de sorties, afin de représenter l'utilisation réelle du système. Cela permet le test non seulement de fonctionnalités mais aussi d'utilisabilité, d'interopérabilité, de fiabilité, de sécurité et de performance.

Pendant l'analyse et la planification des tests, l'équipe de test identifie les profils d'utilisation et essaie de les couvrir avec des cas de test. Le profil d'utilisation est une estimation, basée sur les informations disponibles, de l'usage réaliste du logiciel. Cela signifie que, comme avec le test basé sur les risques, les profils d'utilisation peuvent ne pas modéliser parfaitement la réalité. Cependant, si suffisamment d'informations et d'entrées utilisateurs sont disponibles, le modèle sera adéquat (voir [Musa04]).

Certains Test Managers appliquent aussi des approches méthodiques comme les check-lists pour déterminer quoi tester, en quelle quantité et dans quel ordre. Pour des produits très stables, une check-list des principaux domaines fonctionnels et non-fonctionnels à tester, combinée à un référentiel de cas de test existant peut être suffisant. Les check-lists fournissent des heuristiques pour l'allocation de l'effort et de l'ordonnement des tests, généralement basés sur le type et le volume de changements réalisés. Ces approches tendent à être moins valides quand utilisées pour tester plus que des changements mineurs.

Enfin, une autre méthode commune est d'utiliser une approche réactive. Dans une approche réactive, peu de tâches d'analyse, de conception ou d'implémentation se produisent avant l'exécution des tests. L'équipe de test se focalise sur la réaction au produit tel qu'il a été livré. Des agrégats de défauts, selon leur détection, feront l'objet d'un test approfondi. La priorisation et l'allocation sont complètement dynamiques. L'approche réactive peut fonctionner en complément à d'autres approches, mais quand elle est utilisée seule, elle tend à manquer des domaines majeurs de l'application qui sont importants mais qui ne sont pas touchés par un nombre important de défauts.

### 2.3.4 Priorisation des Tests et Allocation de l'Effort dans le Processus de Test

Quelle que soit la technique — ou, mieux, la combinaison de techniques — qu'un Test Manager utilise, le test Manager doit intégrer cette technique dans le projet et le processus de test. Par exemple, dans un cycle de développement séquentiel (par ex., cycle en V), l'équipe de test sélectionne les tests, alloue l'effort de test et priorise initialement les tests pendant la phase d'exigences, avec des ajustements périodiques, alors que des cycles de développement itératifs ou Agiles demandent une approche itération par itération. La Planification et le Contrôle des tests doivent considérer le degré d'évolution des risques, des exigences et/ou profils d'utilisation et y répondre adéquatement.

Pendant l'analyse, de la conception et de l'implémentation des tests, l'allocation et la priorisation déterminées pendant la planification des tests doivent être appliquées. C'est une défaillance fréquente du processus de test que de mener une analyse et/ou une modélisation soigneuse, sans utiliser cette information pour guider l'avancement du processus de test. Cette défaillance se produit en général pendant la conception et l'implémentation.

Lors de l'exécution des tests, la priorisation déterminée lors de la planification des tests devrait être mise en place, même s'il est important de la mettre à jour périodiquement sur base des informations obtenues après l'écriture initiale du plan de test. Lors de l'évaluation et du reporting des résultats de test et de l'état des critères de sortie, le Test Manager doit aussi évaluer et communiquer en termes de risques, exigences, profils d'utilisation, check-lists et autres guides utilisés pour sélectionner et prioriser les tests. Si nécessaire, le triage des tests devrait être organisé à partir du schéma de priorisation.

Inclus dans le reporting des résultats et l'évaluation des critères de sortie, le Test Manager peut mesurer le degré de complétude des tests. Cela devrait inclure la traçabilité des cas de test et des défauts découverts jusqu'aux bases de test correspondantes. Par exemple, dans un test basé sur les risques, alors que les tests sont exécutés et les défauts découverts, les testeurs peuvent évaluer le niveau de risque résiduel. Cela supporte l'utilisation du test basé sur les risques pour déterminer le bon moment pour livrer. Le reporting de test devrait traiter des risques couverts et des risques encore ouverts, de même que les bénéfices atteints ou non atteints. Pour un exemple de reporting des résultats de test basé sur la couverture du risque, voir [Black03].

Finalement, pendant la clôture des tests, le Test Manager devrait évaluer les métriques et les critères de succès qui sont pertinents au regard des besoins et attentes des parties prenantes du test, y compris les besoins et attentes des clients et utilisateurs en termes de qualité. C'est uniquement quand le test couvre ces besoins et ces attentes qu'une équipe de test peut prétendre être réellement efficace.

## 2.4 Documentation des Tests et autres Livrables

La documentation est souvent produite dans le cadre des activités de Gestion des tests. Même si les noms précis des documents de Gestion des Tests et le périmètre de chacun des documents ont

tendance à varier, les types de documents suivants sont en général trouvés dans les organisations et les projets:

- Politique de test – décrit les objectifs et les finalités du test pour l'organisation
- Stratégie de test – décrit les méthodes de test générales et indépendantes des projets au sein de l'organisation
- Plan de test maître (ou plan de test projet) – décrit l'implémentation de la stratégie de test sur un projet particulier
- Plan de test de niveau (ou plan de test de phase) – décrit les activités précises à mettre en œuvre pour chaque niveau de test

L'organisation physique de ces différents types de documents peut varier d'un contexte à un autre. Dans certaines organisations et sur certains projets, ils peuvent être combinés en un seul document; dans d'autres, on peut trouver des documents séparés; et dans certains cas leur contenu peut s'avérer intuitif, non documenté, ou correspondre à une méthodologie traditionnelle de test. Des organisations et projets plus grands et plus formels tendent à avoir tous ces types de livrables sous forme de livrables écrits, alors que des organisations et projets plus petits et moins formels tendent à avoir en général moins de livrables documentés ainsi. Ce syllabus décrit séparément chacun de ces types de documents, cependant en pratique le contexte de l'organisation ou du projet détermine la bonne utilisation de chacun.

### 2.4.1 Politique de Test

Le politique de test décrit pourquoi l'organisation teste. Cela définit les objectifs généraux que l'organisation veut atteindre pour le test. Cette politique devrait être développée par une personne expérimentée (sénior) en Gestion des Tests au sein de l'organisation, en collaboration avec des responsables seniors appartenant aux différentes parties prenantes impliquées.

Dans certains cas, la politique de test viendra compléter ou fera partie d'une politique qualité plus large. Cette politique qualité décrit les valeurs et objectifs de la hiérarchie associés à la qualité.

Quand une politique de test écrite existe, il peut s'agir d'un document court, de haut niveau qui:

- Résume la valeur que l'organisation dégage du test
- Définit les objectifs du test, tels qu'acquiescer de la confiance dans le logiciel, détecter des défauts dans le logiciel, et réduire le niveau de risque qualité (voir Section 2.3.1)
- Décrit comment évaluer l'efficacité et la rentabilité des tests pour atteindre ces objectifs
- Décrit le processus de test typique, éventuellement en utilisant comme base les processus de test fondamentaux de l'ISTQB
- Spécifie comment l'organisation améliorera ses processus de test (voir Chapitre 5)

La politique de test devrait s'appliquer aux activités de test pour un nouveau développement comme pour les activités de maintenance. Elle peut aussi se référer à des standards internes et/ou externes pour les livrables du test et la terminologie à utiliser dans l'organisation.

### 2.4.2 Stratégie de Test

La stratégie de test décrit la méthodologie générale de l'organisation. Cela inclut la façon d'utiliser le test pour gérer les risques produit et les risques projet, la séparation en niveaux de test et les activités de haut niveau associées au test. (Une même organisation peut avoir différentes stratégies pour différentes situations, telles que différents cycles de développement logiciel, différents niveaux de risques, ou différentes exigences réglementaires). La stratégie de test, et les processus et activités qu'elle décrit, devraient être en accord avec la politique de test. Elle devrait fournir les critères d'entrée et de sortie génériques pour l'organisation ou pour un ou plusieurs programmes.



Comme indiqué ci-dessus, les stratégies de test décrivent des méthodologies de test habituelles qui incluent typiquement:

- Les stratégies analytiques, telles que le test basé sur les risques, où l'équipe de test analyse la base de test pour identifier les conditions de test à couvrir. Par exemple, avec le test basé sur les exigences, l'analyse des tests dérive les conditions de test à partir des exigences, des tests sont alors conçus et implémentés pour couvrir ces conditions. Les tests sont ensuite exécutés, en utilisant souvent la priorité des exigences couvertes par chaque test pour déterminer l'ordre d'exécution des tests. Les résultats de test sont rapportés en termes de statut des exigences, p.ex., exigences testées avec succès, exigences testées en échec, exigences partiellement testées, exigences dont le test est bloqué, etc.
- Les stratégies basées sur les modèles, comme le profiling opérationnel, où l'équipe de test développe un modèle (basé sur des situations réelles ou anticipées) de l'environnement dans lequel le système existe, des entrées et conditions auxquelles le système est soumis, et comment le système devrait se comporter. Par exemple, pour du test de performances basé sur des modèles d'une application mobile grandissant rapidement, on pourrait développer des modèles de trafic réseau entrant et sortant, d'utilisateurs actifs ou inactifs, et des charges de traitement associées, basées sur l'utilisation actuelle et la croissance du projet dans le temps. De plus, des modèles pourraient être développés en considérant l'environnement de production matériel et logiciel actuel, ses données, son réseau et son infrastructure. Des modèles peuvent aussi être développés pour des taux de transmission, des temps de réponses et des allocations de ressources, idéaux, attendus ou minimum.
- Les stratégies méthodiques, telles que celles basées sur les caractéristiques qualité, où l'équipe de test utilise un ensemble prédéterminé de conditions de test, comme un standard qualité (p. ex., ISO 25000 [ISO25000] qui est en train de remplacer l'ISO 9126 [ISO9126]), une check-list ou une collection de conditions de test logiques pouvant correspondre à un domaine particulier, à une application particulière ou à un type de test particulier (p. ex., test de sécurité), et qui utilise cet ensemble de conditions de test d'une itération à la suivante ou d'une release à la suivante. Par exemple, dans du test de maintenance d'un site de e-commerce stable, les testeurs pourraient utiliser une check-list identifiant les fonctions principales, les attributs et les liens pour chaque page. Les testeurs couvriraient chaque élément pertinent de cette check-list pour chaque modification faite sur le site.
- Les stratégies basées sur un processus ou un standard, telles que pour des systèmes médicaux soumis aux standards de la "Food and Drug Administration" américaine, où l'équipe de test suit un ensemble de processus définis par un comité de standardisation ou un autre panel d'experts, où les processus traitent la documentation, la bonne identification et utilisation de la base de test et des oracles de test, ainsi que l'organisation de l'équipe de test. Par exemple, avec des projets suivant les techniques de management Scrum de l'approche agile, à chaque itération les testeurs analysent les user stories décrivant des fonctionnalités particulières, estiment l'effort de test pour chaque fonctionnalité lors du processus de planification de l'itération, identifient pour chaque user story les conditions de test (souvent appelées critères d'acceptation), exécutent les tests couvrant ces conditions, et documentent le statut de chaque user story (non testée, en échec, passée) pendant l'exécution des tests.
- Les stratégies réactives, telles que les attaques basées sur des défauts, où l'équipe de test attend la livraison du logiciel avant de commencer à concevoir et implémenter les tests, réagissant au système réellement en cours de test. Par exemple, en utilisant du test exploratoire sur une application à base de menus, un ensemble de chartes ou agréments peuvent être développés, correspondant aux particularités, aux sélections dans le menu, et aux différents écrans. Chaque testeur se voit affecté un ensemble de chartes de test lesquelles sont utilisées ensuite pour structurer leurs sessions de test exploratoire. Les testeurs rapportent régulièrement les résultats de leurs sessions de test au Test Manager, qui peut ajuster les chartes de test en conséquence.
- Les stratégies consultatives, telles que du test dirigé par les utilisateurs, où l'équipe de test s'appuie sur les entrées d'une ou de plusieurs des principales parties prenantes pour

déterminer les conditions de test à couvrir. Par exemple, lors du test, sous-traité, de compatibilité pour une application web, une société peut donner au sous-traitant en charge du test externalisé une liste priorisée de versions de navigateurs, d'anti-virus, de systèmes d'exploitation, de types de connexion, et autres options de configuration à évaluer pour son application. Le fournisseur de service de test peut alors utiliser des techniques telles que le test par paires (pour les options les plus prioritaires) et les partitions d'équivalence (pour les options les moins prioritaires) pour générer des tests.

- Les stratégies anti-régression, telles que l'automatisation extensive, où l'équipe de test utilise différentes techniques pour gérer le risque de régression, en particulier l'automatisation de tests de non-régression fonctionnelle et/ou non-fonctionnelle à un ou plusieurs niveaux. Par exemple, pour le test de non-régression d'une application web, les testeurs peuvent utiliser un outil d'automatisation basé sur l'interface utilisateur pour automatiser les principaux cas d'utilisation typiques et les cas d'utilisation en exception pour l'application. Ces tests sont alors exécutés à chaque fois que l'application est modifiée.

Diverses stratégies peuvent être combinées. Les stratégies choisies devraient être appropriées pour les besoins et moyens de l'organisation, et les organisations peuvent adapter leurs stratégies pour correspondre à des opérations ou projets particuliers.

La stratégie de test peut décrire les niveaux de test à mettre en œuvre. Elle doit alors préciser les critères d'entrée et sortie à chaque niveau et les relations entre ces niveaux (par ex., la répartition des objectifs de couverture de test).

La stratégie de test peut aussi décrire les points suivants:

- Procédures d'intégration
- Techniques de spécification des tests
- Indépendance du test (qui peut varier selon le niveau)
- Standards obligatoires ou optionnels
- Environnements de test
- Automatisation des tests
- Outils de test
- Réutilisabilité des livrables de développement et des livrables de test
- Test de confirmation (re-test) et test de non-régression
- Contrôle des tests et reporting
- Mesure du test et métriques
- Gestion des défauts
- Approche de gestion en configuration des articles de test
- Rôles et responsabilités

Des stratégies de test différentes peuvent être nécessaires pour le court terme et pour le long terme. Différentes stratégies de test sont nécessaires selon les organisations et les projets. Par exemple, lorsque des applications critiques au niveau de la sécurité et de la sûreté sont impliquées, une stratégie plus intensive peut être mieux adaptée que dans d'autres cas. De plus, la stratégie de test diffère aussi selon les modèles de développements.

### 2.4.3 Plan de Test Maître

Le plan de test maître couvre tout le travail de test à faire sur un projet particulier, y compris les niveaux de test à mettre en œuvre, les relations entre ces niveaux et entre les niveaux de test et les activités de développement correspondantes. Le plan de test maître devrait aborder la façon de mettre en œuvre la stratégie de test pour ce projet (c.à.d., l'approche de test). Le plan de test maître devrait être consistant avec la politique et la stratégie de test, et, si ce n'est pas le cas, il devrait expliquer ces divergences et exceptions, y compris tout impact potentiel résultant de ces divergences.



Par exemple, si la stratégie de test de l'organisation préconise d'exécuter une passe complète de tests de non-régression sur un système inchangé juste avant une livraison, mais le projet en cours n'aura pas de test de non-régression, le plan de test devrait expliquer pourquoi c'est planifié ainsi et ce qui sera fait pour mitiger les risques associés à cette variation par rapport à la stratégie usuelle. Le plan de test devrait également comprendre une explication de tout autre effet attendu de cette différence. Par exemple, sauter les tests de non-régression pourrait nécessiter de prévoir une version de maintenance un mois après la livraison initiale du projet. Le plan de test maître devrait compléter le plan de projet ou le guide opérationnel en ce qu'il devrait décrire l'effort de test faisant partie du projet ou d'une opération plus large.

Bien que le contenu et la structure d'un plan de test maître varient selon l'organisation, ses standards de documentation et du formalisme du projet, les points typiques d'un plan de test maître incluent:

- Les éléments à tester et à ne pas tester
- Les attributs qualité à tester et à ne pas tester
- Le calendrier des tests et le budget (qui devrait être aligné sur le budget du projet ou opérationnel)
- Les cycles d'exécution des tests et leurs liens avec le plan de livraison du logiciel
- Les relations et livrables au sein de l'équipe de test et entre le test et les autres personnes ou départements
- La définition de quels articles de test sont dans le périmètre ou hors du périmètre de chaque niveau de test décrit
- Les critères spécifiques d'entrée, les critères de continuation (suspension/reprise) et les critères de sortie pour chaque niveau et les relations entre les niveaux
- Les risques du projet de test
- La gouvernance globale de l'effort de test
- Les responsabilités pour l'exécution de chacun des niveaux de test
- Les entrées et sorties de chaque niveau de test

Sur des projets ou des opérations plus petits, où un seul niveau de test est formalisé, le plan de test maître et le plan de test correspondant seront souvent combinés en un seul document. Par exemple, si le test système est le seul niveau formalisé, avec des tests de composant et d'intégration réalisés de façon informelle par des développeurs et des tests d'acceptation informels réalisés par des clients dans le cadre d'un processus de beta test, alors le plan de test système pourra inclure les éléments mentionnés dans cette section.

De plus, le test dépend typiquement d'autres activités dans le projet. Dans le cas où ces activités n'étaient pas suffisamment documentées, notamment en termes de relations et d'influences vis à vis du test, les aspects liés à ces activités peuvent être couverts par le plan de test maître (ou dans le plan de test de niveau approprié). Par exemple, si le processus de gestion de configuration n'est pas documenté, le plan de test devrait spécifier comment les objets de test doivent être livrés à l'équipe de test.

#### 2.4.4 Plan de Test de Niveau

Les plans de test de niveau décrivent les activités particulières à exécuter dans chaque niveau de test ou, dans certains cas, pour chaque type de test. Les plans de test de niveau développent, lorsque cela est nécessaire, le plan de test maître pour le niveau ou type de test particulier documenté. Ils précisent des détails sur le planning, les tâches, les jalons pas forcément couverts dans le plan de test maître. De plus, lorsque des standards ou templates particuliers s'appliquent à la spécification des tests à différents niveaux, les détails peuvent être précisés dans les plans de test de niveau.

Sur des projets ou opérations moins formels, un plan de test unique est souvent le seul document de Gestion des Tests rédigé. Dans ce cas, une partie des éléments informatifs mentionnés plus haut peut être couverte dans ce plan de test.

Pour les projets Agiles des plans de test de sprint ou d'itération peuvent prendre la place de plans de test de niveau.

## 2.4.5 Gestion des Risques Projet

Une partie importante d'une bonne planification inclut la gestion des risques projet. Les risques projets peuvent être identifiés avec des processus similaires à ceux décrit en section 2.3. Quand des risques projet sont identifiés, ils devraient être communiqués au et traités par le chef de projet (test manager). De tels risques ne sont pas toujours de la responsabilité de l'organisation de test. Cependant, certains risques projet peuvent et devraient être mitigés avec succès par le Test Manager, notamment:

- Disponibilité des environnements et des outils de test
- Disponibilité et compétence de l'équipe de test
- Manque de standards, de règles et de techniques pour l'effort de test

Les approches de gestion des risques projet incluent la préparation des articles de test, le pré-test des environnements de test, le pré-test de versions précoces du produit, l'application de critères d'entrée du test plus stricts, l'implémentation d'exigences de testabilité, la participation à des revues des premiers livrables du projet, la participation à la gestion du changement et la mesure de la qualité et de l'avancement du projet.

Une fois un risque identifié et analysé, il existe quatre options principales pour gérer ce risque:

1. Mitiger le risque par des mesures préventives pour diminuer sa probabilité et/ou son impact
2. Faire des plans de contingence pour réduire l'impact si le risque se produit
3. Transférer la gestion du risque à une autre entité
4. Ignorer ou accepter le risque

Sélectionner la meilleure option dépend des bénéfices et opportunités créés par cette option, ainsi que du coût et, potentiellement, des risques additionnels associés à cette option. Quand un plan de contingence est identifié pour un risque projet, la meilleure pratique consiste à identifier un déclencheur (qui déterminera quand et comment le plan de contingence est invoqué) ainsi qu'un porteur (qui exécutera le plan de contingence).

## 2.4.6 Autres Livrables du Test

Le test implique la création d'autres livrables, tels que les rapports d'anomalies, les spécifications de cas de test et les logs de test. La plupart de ces livrables sont produits par des Analystes de Test et des Analystes Techniques de Test. Dans le syllabus Analyste de Test, la façon de produire et documenter ces livrables est discutée. Le Test Manager doit assurer la cohérence et la qualité de ces livrables par les activités suivantes:

- Établir et mesurer les métriques qui mesurent la qualité de ces livrables, telle que le pourcentage de rapports d'anomalie rejetés
- Travailler avec les Analystes de Test et les Analystes Techniques de Test pour sélectionner et adapter les templates à utiliser pour ces livrables
- Travailler avec les Analystes de Test et les Analystes Techniques de Test pour établir des standards pour ces livrables de tests, tels que le niveau de détail requis pour les tests, les logs et les rapports
- Revoir les livrables du test en utilisant les techniques appropriées et en impliquant les parties prenantes et les participants appropriés.

L'étendue, le type et la spécificité de la documentation de test peuvent être influencés par les choix de cycle de développement logiciel, de standards et de règlement applicables, de qualité du produit et de risques projets associés au système en cours de développement, entre autres.

Il y a de nombreux templates pour les livrables de test, telles que l'IEEE829. Il est important pour le Test Manager de se souvenir que les documents IEEE829 sont conçus pour être utilisés quel que soit le type d'industrie. Ainsi, les templates contiennent un grand nombre de détails qui peuvent être applicables ou non à une organisation particulière. C'est une pratique recommandée d'adapter les documents de l'IEEE829 pour créer des templates standard pour une organisation. L'utilisation consistante des templates réduit les besoins de formation et aide à unifier le processus au sein de l'organisation.

Le test implique aussi la création de rapport de résultats de test, qui sont typiquement produits par le Test Manager et sont décrits plus loin dans ce chapitre.

## 2.5 Estimation des Tests

L'estimation, en tant qu'activité de gestion, est la création d'une cible approximative pour les coûts et dates de réalisation associés à un projet particulier ou à une opération particulière. Les meilleures estimations:

- Représentent la sagesse collective d'un ensemble de praticiens expérimentés et ont le soutien des participants impliqués
- Fournissent un catalogue spécifique et détaillé des coûts, ressources, tâches et personnes impliquées
- Présentent, pour chaque activité évaluée, les valeurs les plus probables pour le coût, l'effort et la durée

L'estimation de l'ingénierie logicielle et système a longtemps été connue pour être semée d'embûches, à la fois techniques et politiques, même si les bonnes pratiques d'estimation de projet sont établies. L'estimation des tests est l'application de ces bonnes pratiques aux activités de test associées à un projet ou une opération.

L'estimation des tests devrait couvrir toutes les activités impliquées dans le processus de test comme décrit au Chapitre 1. L'estimation du coût, de l'effort et spécialement de la durée d'exécution des tests est généralement d'un grand intérêt pour le management, car l'exécution des tests est typiquement sur le chemin critique du projet. Cependant, les estimations d'exécution de tests tendent à être difficiles à produire et peu fiables quand la qualité globale du logiciel est basse ou inconnue. De plus, la familiarisation et l'expérience avec le système affecteront en général la qualité des estimations. Une pratique commune consiste à estimer le nombre de cas de test devant être exécutés, mais ceci ne fonctionne que si l'on peut supposer que le nombre de défauts dans le logiciel à tester est faible. Les suppositions faites devraient toujours être documentées et faire partie de l'estimation.

L'estimation des tests devrait intégrer tous les facteurs pouvant influencer le coût, l'effort et la durée des activités de test. Parmi ces facteurs, on peut citer (liste non limitative):

- Le niveau de qualité requis du système
- La taille du système à tester
- Des données historiques sur le test de projets précédents, qui peuvent être complétées par des données ou benchmark de l'industrie ou d'autres organisations
- Des facteurs liés au processus, comme la stratégie de test, le cycle de développement ou de maintenance, la maturité des processus ainsi que l'exactitude des estimations du projet



- Des facteurs matériels, incluant l'automatisation des tests et les outils, le ou les environnement(s) de test, les données de test, le ou les environnement(s) de développement, la documentation du projet (par ex., exigences, conceptions, etc.), et des livrables de test réutilisables
- Des facteurs humains, incluant les managers et responsables techniques, les engagements et les attentes de la direction exécutive et de la hiérarchie, les compétences, expériences et comportements dans l'équipe projet, la stabilité de l'équipe projet, les relations au sein de l'équipe, le support affecté à l'environnement de test et au débogage, la disponibilité de sous-traitants ou consultants compétents, et la connaissance du domaine
- La complexité du processus, de la technologie, de l'organisation, du nombre de parties prenantes du test, la composition et la localisation des sous-équipes
- Les besoins de prise de connaissance, de formations et d'orientations
- L'assimilation ou le développement de nouveaux outils, de nouvelles technologies, de nouveaux processus, de nouvelles techniques, de matériel spécifique, ou d'une grande quantité d'articles de test
- Les exigences pour un haut niveau d'informations détaillées pour la spécification des tests, spécialement pour satisfaire à standard de documentation non familier
- Un séquençement complexe dans l'arrivée des composants, spécialement pour les tests d'intégration et le développement des tests
- Des données de test fragiles (par ex., des données sensibles au temps)

La qualité du logiciel livré au test est aussi un facteur déterminant que les Test Managers devraient prendre en compte dans leurs estimations. Par exemple, si les développeurs ont adopté de bonnes pratiques comme le test unitaire automatisé et l'intégration continue, alors jusqu'à 50% des défauts seront supprimés avant la livraison du code à l'équipe de test. (voir [Jones11] pour en savoir plus sur l'efficacité de la suppression des défauts avec ces pratiques). Certains auteurs ont indiqué que les méthodologies Agile, incluant le développement piloté par les tests, résultent en un niveau encore supérieur de qualité de ce qui est livré au test.

L'estimation peut se faire de bas en haut ou de haut en bas. Les techniques suivantes peuvent être utilisées pour l'estimation de test :

- Intuition, supposition et expérience passée
- Structure de la répartition du travail (WBS Work-breakdown-structure)
- Sessions d'estimation en équipe (par ex., Delphi à Large Bande)
- Règles et normes de la société
- Pourcentages de l'effort pour l'ensemble du projet ou du niveau des effectifs (par exemple, les ratios testeurs / développeurs)
- Historique et métriques de l'organisation, y compris les modèles dérivés de métriques qui estiment le nombre de défauts, le nombre de cycles de test, le nombre de cas de test, l'effort moyen estimé de chaque test, et le nombre de cycles de non-régression impliqués
- Des moyennes issues de l'industrie et des modèles prédictifs tels que les points de fonction, les lignes de code, l'effort de développement estimé, ou d'autres paramètres du projet (par ex., voir [Jones07]).

Dans la plupart des cas, l'estimation, une fois établie, doit être fournie à la direction, avec des justificatifs (voir Section 2.7). Fréquemment, certaines négociations s'en suivent, résultant souvent en une refonte de l'estimation. Idéalement, l'estimation finale de la charge des tests représente l'équilibre optimal possible entre les objectifs du projet et ceux de l'organisation dans les domaines de la qualité, des délais, du budget, et des caractéristiques.

Il est important de se souvenir que toute estimation est basée sur l'information disponible au moment de sa préparation. Au début du projet, l'information peut être fort limitée. De plus, de l'information

disponible tôt dans le projet peut changer avec le temps. Afin de rester exactes, les estimations devraient être mises à jour pour refléter les informations nouvelles ou modifiées.

## 2.6 Définition et Utilisation de Métriques de Test

Un cliché de gestion dit que tout ce qui est mesuré est fait. De plus ce qui n'est pas mesuré n'est pas fait parce que ce qui n'est pas mesuré est facile à ignorer. En conséquence, il est important qu'un ensemble adapté de métriques soit défini pour toute activité, y compris le test.

Les métriques de test peuvent être classifiées comme appartenant à une ou plusieurs des catégories suivantes:

- Métriques projet, qui mesurent le progrès au regard des critères de sortie du projet, telles que les pourcentages de cas de test exécutés, passés avec succès et en échec
- Métriques produit, qui mesurent certains attributs du produit, telles que l'étendue de ce qui a été couvert par des tests ou la densité des défauts
- Métriques processus, qui mesurent la capacité des processus de développement ou de test, telles que le pourcentage de défauts découverts par le test
- Métriques liées aux personnes, qui mesurent la capacité des individus ou groupes, telles que l'implémentation des cas de test selon un planning donné

Une métrique donnée peut appartenir à deux, trois, ou même quatre catégories. Par exemple, un graphe de tendance montrant le taux de défauts détectés par jour peut être associé à un critère de sortie (aucun nouveau défaut trouvé pendant une semaine), à la qualité du produit (le test ne peut pas trouver davantage de défauts) et à la capacité du processus de test (le test trouve un grand nombre de défauts au début de la période d'exécution des tests).

Les métriques liées aux personnes sont particulièrement délicates. Les managers confondent parfois des métriques processus avec des métriques liées aux personnes, ce qui a des conséquences désastreuses lorsque les personnes agissent pour fausser les métriques d'une façon qui leur est favorable. La bonne manière de motiver et d'évaluer les personnes impliquées dans le test est discutée au chapitre 7 de ce syllabus et dans le syllabus Expert Test Manager [ISTQB ETL SYL].

Au niveau avancé, nous sommes principalement concernés par l'utilisation de métriques pour mesurer l'avancement du test, c'est à dire, des métriques projet. Certaines de ces métriques projet utilisées pour la mesure de l'avancement du test sont liées aussi au produit et au processus. Le syllabus Expert Test Manager donne plus d'information sur la façon pour un manager d'utiliser les métriques processus. Le syllabus Expert Amélioration des Processus de Test [ISTQB ITP SYL] donne plus d'information sur l'utilisation de métriques processus.

L'utilisation de métriques permet aux testeurs de rapporter des résultats de façon consistante et permet un suivi cohérent de l'avancement dans le temps. Il est souvent demandé aux Test Managers de présenter des métriques à diverses réunions auxquelles peuvent participer plusieurs niveaux de parties prenantes, allant de l'équipe technique à la direction générale. Parce que les métriques sont souvent utilisées pour déterminer le succès global d'un projet, une grande attention devrait être portée sur ce qu'il faut mesurer, la fréquence à laquelle communiquer, et la méthode utilisée pour présenter l'information. Un Test Manager doit notamment considérer les points suivants:

- Définition des métriques. Un ensemble limité de métriques utiles devrait être défini. Les métriques devraient être définies sur la base d'objectifs spécifiques au projet, au processus et/ou au produit. Les métriques devraient être définies de façon équilibrée, une unique métrique pouvant induire une mauvaise interprétation d'un statut ou de tendances. Une fois ces métriques définies, l'interprétation doit être acceptée par toutes les parties prenantes afin



d'éviter toute confusion quand les mesures seront discutées. On a souvent tendance à définir trop de métriques au lieu de se concentrer sur les plus pertinentes.

- Suivi des métriques. Le suivi et la consolidation des métriques devraient être aussi automatisés que possible pour réduire le temps passé à prendre et à traiter des mesures. Des variations dans le temps pour une métrique spécifique, peuvent refléter une information différente de l'interprétation prévue initialement. Le Test Manager devrait être préparé à analyser soigneusement les divergences possibles entre ce qui est mesuré et ce qui était attendu, et les raisons de ces divergences.
- Reporting des métriques. L'objectif est de fournir une compréhension immédiate de l'information pour les besoins de gestion. Des présentations peuvent montrer un aperçu des métriques à un instant t ou montrer leur évolution dans le temps pour évaluer des tendances.
- Validité des métriques. Un Test Manager doit aussi vérifier l'information qui est rapportée. Les mesures d'une métrique peuvent ne pas refléter le statut réel d'un projet ou peuvent fournir une tendance trop positive ou négative. Avant toute présentation de résultat, le Test Manager doit les revoir attentivement, à la fois en termes d'exactitude et des messages qui sont susceptibles d'être transmis.

L'avancement est principalement mesuré selon cinq dimensions:

- Risques produit (qualité)
- Défauts
- Tests
- Couverture
- Confiance

Les risques produit, les défauts, les tests et la couverture peuvent être et sont souvent rapportés de façon spécifique au long du projet ou de l'opération. Si ces mesures sont liées à des critères de sortie définis dans le plan de test, elles peuvent fournir un standard objectif par lequel juger la complétude de l'effort de test. La confiance est mesurable par des sondages ou en utilisant la couverture comme métrique de substitution, cependant, la confiance est souvent aussi rapportée de façon subjective.

Les métriques liées au risque produit incluent:

- Pourcentage de risques entièrement couverts par des tests passés avec succès
- Pourcentage de risques pour lesquels certains ou tous les tests sont en échec
- Pourcentage de risques n'étant pas encore complètement testés
- Pourcentage des risques couverts, triés par catégories de risques
- Pourcentage des risques identifiés après l'analyse de risque qualité initiale

Les métriques liées aux défauts incluent:

- Nombre cumulé de défauts rapportés (trouvés) versus nombre total de défauts résolus (corrigés)
- Temps moyen entre les défaillances ou taux de nouvelles défaillances
- Répartition du nombre ou du pourcentage de défauts classés par:
  - Élément ou composant de test particulier
  - Causes racines
  - Source du défaut (par ex., spécification d'exigences, nouvelle fonctionnalité, régression, etc.)
  - Version de livraison en test
  - Phase d'introduction, de détection et de résolution du défaut
  - Priorité/sévérité
  - Rapports d'anomalie rejetés ou en doublon
- Tendances en termes de durée moyenne entre découverte et résolution des défauts

- Nombre de défauts corrigés ayant introduit de nouveaux défauts (parfois appelés défauts dépendants)

Les métriques liées aux tests incluent:

- Nombre total de tests planifiés, spécifiés (implémentés), exécutés, passés avec succès, en échec, bloqués, et sautés
- Statut des tests de non-régression et de confirmation, y compris les tendances et les volumétries relatives aux défaillances lors des tests de non-régression et de confirmation
- Nombre d'heures de test prévues par jour versus nombre d'heures réellement passées à tester
- Disponibilité de l'environnement de test (pourcentage d'heures de disponibilité de l'environnement de test pour l'équipe de test)

Les métriques liées à la couverture incluent:

- Couverture des exigences et des éléments de conception
- Couverture du risque
- Couverture des environnements et des configurations
- Couverture du code

Il est important que les Test Managers comprennent comment interpréter et utiliser les métriques de couverture de manière à comprendre et informer sur le statut des tests. Pour les niveaux plus élevés de test, tels que le test des systèmes, l'intégration des systèmes, et les tests d'acceptation, les bases de test initiales sont souvent des livrables tels que des spécifications d'exigences, des spécifications de conception, des cas d'utilisation, des user stories, des risques produit, les environnements et configurations supportés. Les métriques de couverture de la structure du code s'appliquent plus à des niveaux de test inférieurs (p.ex. couverture des instructions et des branches) et l'intégration des composants (p.ex. couverture des interfaces). Bien que les Test Managers peuvent utiliser les métriques de couverture de code pour mesurer le niveau atteint de couverture des structures du système en cours de test, le reporting de résultats de tests de plus haut niveau ne devrait typiquement pas inclure des métriques de couverture de code. De plus, les Test Managers devraient comprendre que, même si le test unitaire et le test d'intégration des composants atteignent 100% de leurs objectifs de couverture structurelle, des défauts et des risques qualité restent à traiter dans des niveaux de test supérieurs.

Les métriques peuvent aussi être liées aux activités des processus de test fondamentaux (décrits dans le syllabus niveau fondation et dans ce syllabus). De cette façon, les métriques peuvent être utilisées tout au long du processus de test pour mesurer le processus lui-même ainsi que l'avancement par rapport aux objectifs du projet.

Les métriques pour mesurer la planification et le suivi des tests peuvent inclure:

- La couverture des risques, des exigences et des autres éléments de la base de test
- La découverte de défauts
- Le nombre d'heures prévues versus passées pour développer le testware et exécuter les cas de test

Les métriques pour mesurer les activités d'analyse de test peuvent inclure:

- Le nombre de conditions de test identifiées
- Le nombre de défauts trouvés lors de l'analyse des tests (par ex., en identifiant les risques ou les autres conditions de test en utilisant la base de test)

Les métriques pour mesurer les activités de conception de tests peuvent inclure:

- Pourcentage des conditions de test couvertes par des cas de tests

- Nombre de défauts trouvés lors de la conception des tests (par ex., en développant des tests à partir de la base de test)

Les métriques pour mesurer les activités d'implémentation des tests peuvent inclure:

- Pourcentage d'environnements de test configurés
- Pourcentage de données de test chargées
- Pourcentage de cas de tests automatisés

Les métriques pour mesurer les activités d'exécution des tests peuvent inclure:

- Pourcentage de cas de tests exécutés, passés avec succès et en échec
- Pourcentage de conditions de test couvertes par des cas de test exécutés (et/ou passés avec succès)
- Nombre prévu versus réel de défauts trouvés/résolus
- Couverture prévue versus réalisée.

Les métriques contrôlant l'avancement du test et la complétude des activités de test incluront une correspondance (mapping) entre les jalons, les critères d'entrée, et les critères de sortie (définis et approuvés lors de la planification des tests), qui peuvent inclure les éléments suivants:

- Nombre de conditions de test, cas de tests ou spécifications de test prévus et ceux ayant été exécutés, répartis par statut (passés ou en échec)
- Nombre total de défauts, souvent classés par sévérité, priorité, état actuel, sous-système concerné, ou autre classification (voir Chapitre 4)
- Nombre de changements demandés, acceptés, implémentés et testés
- Coût prévu versus réel
- Durée prévue versus réelle
- Dates de passage des jalons de test prévues versus réelles
- Dates de passage prévues versus dates réelles pour les jalons du projet ayant un lien avec le test (par ex., fin des développements)
- Statut des risques produit (qualité), souvent classés par, risques mitigés versus non-mitigés, domaines de risques, nouveaux risques découverts après l'analyse des tests, etc.
- Pourcentage d'effort de test, de coût ou de temps perdus suite à des événements bloquants ou des changements de planning
- Statuts des tests de confirmation et de non-régression

Les métriques pour mesurer les activités de clôture des tests peuvent inclure:

- Le pourcentage de cas de test exécutés, passés avec succès, en échec, bloqués, et sautés pendant l'exécution des tests
- Le pourcentage de cas de test identifiés comme réutilisables dans le référentiel de test
- Le pourcentage de cas de test devant être automatisés versus le pourcentage de cas de test réellement automatisés
- Le pourcentage de cas de test intégrés aux tests de non-régression
- Le pourcentage de rapports de défauts résolus/non résolus
- Le pourcentage de livrables du test identifiés et archivés

De plus, les techniques de gestion de projet classiques, telles que le découpage du travail en tâches et sous-tâches (Work Breakdown Structure) sont souvent utilisées pour piloter le processus de test. Dans les équipes agiles, le test fait partie de l'avancement des user stories dans le graphe du réalisé (burn-down chart). Lorsque les techniques Lean sont utilisées, l'avancement du test, sur la base du cas par cas (story by story), est souvent piloté par le déplacement des user stories d'une colonne à une autre dans le tableau de Kanban .



Sur la base d'un ensemble donné de métriques, les mesures peuvent être rapportées de façon textuelle, avec des tableaux de chiffres ou avec des graphiques. Les mesures peuvent être utilisées avec différentes finalités, comme:

- L'analyse, pour découvrir les tendances et causes identifiables par le biais des résultats de test
- Le reporting, pour communiquer sur les découvertes des tests aux participants du projet et aux parties prenantes intéressées
- Le contrôle, pour modifier les tests ou le projet dans son ensemble et pour mesurer les résultats de ce changement d'orientation

Les bonnes manières de collecter, d'analyser et de communiquer ces mesures dépendent des besoins spécifiques en termes d'information, des objectifs et des compétences des personnes qui utiliseront ces mesures. De plus, le contenu des rapports de test devrait être adapté en fonction des destinataires.

Pour le contrôle des tests, il est essentiel que des métriques, tout au long du processus de test (une fois la planification des tests terminée) apportent au Test Manager l'information requise pour orienter l'effort de test vers l'accomplissement de la mission, des stratégies et des objectifs du test. Par conséquent, la planification des tests doit considérer ces besoins d'informations et la mesure des tests doit inclure la collecte des métriques nécessaires. Le volume d'information requis et l'effort nécessaire à cette collecte dépendent de différents facteurs du projet comme la taille, la complexité et le risque.

Le contrôle du test doit répondre à l'information générée par le test de même qu'aux changements des conditions dans lesquelles un projet ou une opération existe. Par exemple, si du test dynamique révèle des agrégats de défauts dans des domaines qui avaient été considérés comme ne devant pas contenir beaucoup de défauts, ou si la période d'exécution des tests est raccourcie à la suite d'un retard dans le démarrage des tests, l'analyse de risque et le plan devront être révisés. Cela pourrait aboutir à une re-priorisation des tests et à une réallocation de l'effort d'exécution des tests.

Si des divergences par rapport au plan de test sont découvertes par les rapports de test, le contrôle des tests devrait être réalisé. Le contrôle des tests a pour objectif de réorienter le projet et/ou le test dans une meilleure direction. Quand les résultats des tests sont utilisés pour influencer ou mesurer les efforts de contrôle d'un projet, les options suivantes devraient être envisagées:

- Révision de l'analyse des risques qualité, des priorités de test et/ou des plans de test
- Ajout des ressources ou augmenter l'effort de test ou l'effort du projet
- Retarder la date de livraison
- Assouplir ou renforcer les critères de sortie de test
- Modifier le périmètre (fonctionnel et/ou non-fonctionnel) du projet

Implémenter de telles options demande un consensus entre les parties prenantes du projet ou de l'opération et l'approbation des responsables du projet ou de l'opération.

L'information livrée dans un rapport de test devrait dépendre largement des besoins d'information de l'audience cible, par ex., responsables projet ou responsables métier. Pour un chef de projet, une information détaillée sur les défauts sera probablement utile, pour le responsable métier, le statut des risques produits pourrait être l'élément clé du rapport.

## 2.7 Valeur Financière du Test

Le Test Manager doit travailler à optimiser les tests de façon à livrer la meilleure valeur ajoutée au métier. Tester de façon excessive n'apportera pas une bonne valeur métier, parce que le test imposera des délais et coûts supérieurs à ce qu'il permettrait d'économiser. Ne pas tester suffisamment, n'apportera pas une bonne valeur au métier car trop de défauts seront livrés aux utilisateurs. L'optimum se situe entre ces deux extrêmes. Le Test Manager doit aider les parties prenantes du test à comprendre cet optimum et la valeur apportée par le test.

Même si beaucoup d'organisations sont conscientes de la valeur du test, peu de managers, y compris des Test Managers, peuvent quantifier, décrire ou exprimer clairement cette valeur. De plus, beaucoup de Test Managers, responsables de test et testeurs se focalisent sur les détails tactiques du test (aspects spécifiques à une tâche ou un niveau de test), tout en ignorant les problèmes stratégiques plus larges (d'un niveau supérieur) qui importent aux autres participants, notamment les managers.

Le test fournit de la valeur à l'organisation, au projet et/ou à l'opération à la fois de façon quantitative et de façon qualitative:

- Les valeurs quantitatives incluent la découverte de défauts qui sont prévenus ou corrigés avant la livraison, la découverte de défauts qui sont connus avant la livraison (non corrigés mais documentés, si possible avec des solutions de contournements), la réduction du risque par l'exécution de tests, et la fourniture d'information sur le projet, le processus, et le statut du produit
- Les valeurs qualitatives incluent l'amélioration de la réputation en termes de qualité, la maîtrise des livraisons, une confiance accrue, la protection d'un point de vue légal, et la réduction de risques de pertes de missions, voire de vies humaines

Les Test Managers devraient identifier lesquelles de ces valeurs s'appliquent à leur organisation, projet, et/ou opération, et devraient être capables de communiquer sur les tests selon ces valeurs.

Une méthode établie pour mesurer la valeur quantitative et l'efficacité des tests est appelée le coût de la qualité (ou, parfois, coût de la non qualité). Le coût de la qualité implique la classification des coûts du projet et des opérations selon quatre catégories relatives au coût des défauts du produit.

- Coûts de prévention, par ex., formation des développeurs pour écrire un code plus facilement maintenable ou plus sécurisé
- Coûts de détection, par ex., écrire des cas de tests, configurer les environnements de test, et revoir les exigences
- Coûts internes des défaillances, par ex., correction des défauts trouvés pendant le test ou les revues, avant la livraison
- Coûts externes des défaillances, par ex., coûts du support associé à la livraison d'un logiciel défaillant aux clients

Une partie du budget de test est un coût de détection (c.à.d., argent dépensé même si les testeurs ne découvrent pas de défauts, tels que l'argent dépensé pour développer les tests) alors que le reste est un coût interne de défaillance (c.à.d., le coût associé aux défauts trouvés). La somme des coûts de détection et des coûts internes de défaillance est généralement bien inférieure au coût externe de défaillance, ce qui rend le test très rentable. En déterminant les coûts dans ces quatre catégories, le Test Manager peut créer une étude financière convaincante en faveur du test.

Des informations supplémentaires sur la valeur financière du test sont données dans [Black03].

## 2.8 Tests Distribués, Externalisés et Internalisés

Dans de nombreux cas, une partie ou même tout l'effort de test, est réalisé par des personnes situées à différents endroits, employées par différentes sociétés, ou séparées de l'équipe projet. Si l'effort de test est produit à différents endroits, alors il est distribué. Si l'effort de test est mené à un ou plusieurs endroits par des personnes n'étant pas des employés de la société et n'étant pas situées avec l'équipe projet, l'effort de test est externalisé. Si l'effort de test est réalisé par des personnes situées avec l'équipe projet mais n'appartenant pas à la même société, l'effort de test est internalisé.

Un point commun à tous ces types d'effort de test est le besoin de canaux de communication clairs et d'attentes bien définies pour les missions, les tâches et les livrables. L'équipe projet devrait s'appuyer moins sur des canaux de communication informels tels que les conversations de couloir pour les périodes de détente entre collègues. Il est important d'avoir des façons définies pour communiquer, incluant comment escalader les problèmes, les types d'informations à communiquer et les méthodes de communication à utiliser. Chacun, des différents bords des relations d'équipes, doit avoir une compréhension claire de ses rôles et responsabilités de même qu'une compréhension claire des rôles et responsabilités des autres parties, pour éviter des incompréhensions et des attentes irréalistes. Les différences de lieu géographique, de fuseau horaire, de culture et de langage font que les problèmes de communication ou d'attentes irréalistes sont plus probables.

Un autre point commun à ces efforts de test est le besoin d'aligner les méthodologies. Alors que de mauvais alignements de méthodologies peuvent survenir sur chaque projet, ils ont plus de chance de survenir dans des situations où le travail est distribué et/ou effectué par des entités externes. Si deux groupes de test utilisent des méthodologies différentes ou si le groupe de test utilise une méthodologie différente de celle utilisée par le développement ou la gestion de projet, ceci peut résulter en de réels problèmes, en particulier pendant l'exécution des tests. Par exemple, si un client utilise un développement agile alors que le fournisseur de services de tests a une méthodologie de test prédéfinie basée sur un cycle de vie séquentiel, alors la planification et la nature des livraisons des articles de test au fournisseur des services de test sera un point de contention.

Pour les tests distribués, la division du travail de test entre les différents lieux devra être explicite et décidée intelligemment. Sans cela, même le groupe le plus compétent peut ne pas réussir un travail de test correspondant pourtant à ses qualifications. De plus, si chaque équipe de test ne sait pas de quoi elle est responsable, elle pourrait ne pas faire ce qu'elle est supposée faire. Les attentes de chacune des équipes doivent être clairement communiquées. Sans une gestion minutieuse, le travail de test dans son ensemble risque de souffrir de lacunes (qui augmenteront les risques qualité sur la livraison) et de recouvrements (qui réduiront l'efficacité).

Finalement, pour tous ces efforts de test, il est critique que l'ensemble de l'équipe projet développe et maintienne une confiance telle que tous les membres de l'équipe de test exécuteront leurs rôles efficacement quels que soient les potentiels problèmes organisationnels, culturels, de langage ou géographiques. Un manque de confiance entraîne une inefficacité et des délais liés à la vérification des activités, au report de blâme sur les autres, et aux jeux politiques au sein des organisations.

## 2.9 Gérer l'Application de Standards Industriels

Dans les syllabi niveau Fondation et Avancé, de nombreux standards sont référencés. Ces standards référencés couvrent le cycle de développement logiciel, le test logiciel, les caractéristiques qualité logicielle, les revues, et la gestion des anomalies. Les Test Managers devraient être au courant des standards, de la politique de leur organisation vis à vis des standards, et si les standards sont ou non exigés ou nécessaires pour leur travail.



Il existe différentes sources de standards, telles que :

- Internationaux ou avec des objectifs internationaux
- Nationaux, comme l'application nationale de standards internationaux
- Spécifiques à un domaine, comme les standards nationaux ou internationaux adaptés à un domaine, ou développés pour des domaines particuliers

Les organismes de standardisation incluent l'ISO et l'IEEE. ISO est l'Organisation Internationale des Standards (International Standards Organization), aussi appelée IOS, l'International Organization for Standardization. Elle est composée de membres représentants, pour leur pays, le comité national le plus pertinent pour le domaine en cours de standardisation. Cet organisme international a publié un nombre de standards utiles aux testeurs de logiciels comme, ISO 9126 (en cours de remplacement par ISO 25000), ISO 12207 [ISO 12207], et ISO 15504 [ISO 15504].

L'IEEE est l'Institut des Ingénieurs en Electricité et Electronique (Institute of Electrical and Electronics Engineers), une organisation professionnelle basée aux États-Unis, mais ayant des représentants nationaux dans plus de cent pays. Cette organisation a proposé un nombre de standards utiles aux testeurs de logiciel, tels que l'IEEE 829 [IEEE829] et l'IEEE 1028 .

Beaucoup de pays ont leurs propres standards nationaux. Certains de ces standards sont utiles pour le test logiciel. Un exemple est le standard anglais BS-7925-2 [BS7925-2], qui fournit des informations relatives à de nombreuses techniques de conception des tests décrites dans les syllabi avancés Analyste de Test et Analyste Technique de Test.

Certains standards sont spécifiques à des domaines particuliers, et certains parmi eux ont des implications sur le test, la qualité et le développement des logiciels. Par exemple, dans le domaine de l'aviation, le standard DO-178B de la U.S. Federal Aviation Administration (et son équivalent européen ED 12B) s'appliquent aux logiciels utilisés dans des avions civils. Ce standard prescrit certains critères de niveaux de couverture structurelle selon le niveau de criticité du logiciel testé.

Un autre exemple de standard spécifique à un domaine, est présent dans le domaine médical avec la U.S. Food and Drug Administration et le document 21 CFR Part 820 [FDA21]. Ce standard recommande certaines techniques de test structurelles et fonctionnelles. Ce standard recommande aussi des stratégies de test et des principes consistants avec les syllabi ISTQB.

Dans certains cas, le test est influencé par des standards ou méthodologies largement répandus qui ne sont pas concernés initialement par le test, mais qui influencent fortement le contexte du processus logiciel dans lequel s'exécutent les tests. Le modèle d'amélioration des processus logiciels CMMI® en est un exemple. Il contient deux domaines de processus clé, vérification et validation, qui sont souvent assimilés à des niveaux de test (comme le test système et le test d'acceptation respectivement). Il a aussi des implications en termes de stratégie de test, souvent interprétée comme nécessitant du test analytique basé sur les exigences comme partie de la stratégie de test.

Trois autres exemples importants sont le PMBOK de PMI , PRINCE2®, et ITIL®. PMI et PRINCE2 sont des référentiels de gestion de projet largement utilisés respectivement en Amérique du Nord et Europe. ITIL est un référentiel destiné à s'assurer qu'un groupe des technologies de l'information délivre des services de valeur à l'organisation à laquelle il appartient. La terminologie et les activités spécifiées dans ces référentiels diffèrent fortement des syllabi et du glossaire ISTQB. Quand il travaille dans des organisations utilisant PMBOK, PRINCE2, et/ou ITIL, le Test Manager doit comprendre les référentiels choisis, leur implémentation, et leur terminologie suffisamment pour être efficace.

Quels que soient les standards ou méthodologies adoptés, il est important de se souvenir qu'ils ont été créés par des groupes de professionnels. Les standards reflètent l'expérience et la sagesse



collective, mais aussi leurs faiblesses. Le Test Manager devrait être conscient des standards qui s'appliquent à son contexte et à son environnement, qu'il s'agisse de standards formels (internationaux, nationaux ou spécifiques à un domaine) ou de standards maison et de pratiques recommandées.

En considérant l'utilisation de multiples standards, il faut garder à l'esprit que certains standards sont inconsistants, voire contradictoires, avec d'autres standards. Le Test Manager devrait déterminer l'utilité des différents standards pour le contexte particulier dans lequel se déroule le test. L'information mentionnée dans un standard peut être utile au projet ou au contraire lui être nuisible. Cependant, les standards peuvent fournir des références à de bonnes pratiques, et fournir une base pour l'organisation du processus de test.

Dans certains cas, la conformité à des standards est requise et à des implications sur le test. Le Test Manager doit être conscient de telles exigences de respect de standards et s'assurer que la conformité nécessaire est maintenue.



## 3. Revues – 180 mins.

### Mots clés

audit, revue informelle, inspection, revue de gestion, modérateur, revue, plan de revue, relecteur, revue technique, relecture technique

### Objectifs d'Apprentissage pour les revues

#### 3.2 Revues de Gestion et Audits

TM-3.2.1 (K2) Comprendre les caractéristiques principales des revues de gestion et audits

#### 3.3 Gérer des Revues

TM-3.3.1 (K4) Analyser un projet pour sélectionner les types de revues appropriés et pour définir un plan de pilotage des revues, afin d'assurer l'efficacité de la mise en œuvre, du suivi et des responsabilités

TM-3.3.2 (K2) Comprendre les facteurs, les compétences et le temps requis pour participer aux revues

#### 3.4 Métriques pour les Revues

TM-3.4.1 (K3) Définir le processus et les métriques produit à utiliser dans les revues

#### 3.5 Gestion de Revues Formelles

TM-3.5.1 (K2) Expliquer, en utilisant des exemples, les caractéristiques d'une revue formelle

## 3.1 Introduction

Les revues furent introduites dans le Syllabus ISTQB niveau fondation comme activités de test statique de produits. Les audits et les revues de gestion portent davantage sur le processus que sur les livrables logiciels.

Les revues étant une forme de test statique, le Test Manager peut être responsable de leur succès global, en particulier concernant les livrables produits. Cependant, dans le contexte plus large des projets logiciels cette responsabilité devrait être identifiée dans la politique de l'organisation. Vu l'application possible à grande échelle de revues formelles pour de nombreuses disciplines, à la fois avant et pendant un projet logiciel, le responsable peut être un Test Manager, ou un responsable qualité, ou un coordinateur de revues formé. Dans ce syllabus, le responsable (quel qu'il soit) est appelé responsable de revue.

Le responsable de revue devrait s'assurer de l'existence d'un environnement permettant l'implémentation des facteurs de succès définis dans le Syllabus ISTQB niveau Fondation. De plus, le responsable de revue devrait concevoir un plan de mesure des revues pour assurer que les revues apportent une valeur réelle.

Comme les testeurs ont une bonne compréhension du comportement opérationnel et des caractéristiques requises par le système logiciel, leur implication dans le processus de revue est importante.

Les participants aux revues devraient être formés pour mieux comprendre leurs rôles respectifs dans un processus de revue. Tous les participants doivent s'engager à contribuer au bénéfice d'une revue bien menée.

Lorsqu'elles sont menées proprement, les revues sont le contributeur le plus important et le plus efficace pour une qualité globale livrée. Il est donc essentiel que les responsables de revue soient capables d'implémenter efficacement les revues sur leurs projets et d'en démontrer les bénéfices.

Les revues possibles sur un projet incluent:

- Les revues contractuelles, initialisées au démarrage du projet et aux jalons majeurs
- Les revues d'exigences, initialisées lorsque les exigences sont disponibles pour être revues, ce qui idéalement couvre les exigences fonctionnelles et non fonctionnelles
- Les revues de conception de haut niveau, initiées quand la conception d'architecture est disponible pour revue
- Les revues de conception détaillée, initiées lorsque la conception détaillée est disponible
- Les revues de code, effectuées lorsque les modules individuels sont créés, et pouvant inclure les tests unitaires et leurs résultats de même que le code lui-même
- Les revues de livrables de test, qui peuvent inclure le ou les plan(s) de test, les conditions de test, les résultats de l'analyse des risques qualité (produit), les tests, les données de test, les environnements de test, et les résultats de test
- Les revues de démarrage des tests (disponibilité des tests) et les revues de sortie des tests pour chaque niveau de test, qui vérifient respectivement les critères d'entrée des tests avant de commencer leur exécution et les critères de sortie avant d'arrêter le test
- Les revues d'acceptation, utilisées pour obtenir l'acceptation d'un système par un client ou une partie prenante

En plus d'appliquer divers types de revues à un produit, il est important pour un responsable de revue de se souvenir que même si les revues peuvent trouver des défauts dans un document statique, elles devraient être complétées par d'autres formes de tests statiques (telles que l'analyse statique) et par

du test dynamique du code. Utiliser une combinaison de ces techniques améliore la couverture des tests et identifiera plus de défauts.

Chaque technique a un objectif différent. Par exemple, une revue peut éliminer un problème au niveau des exigences, avant qu'il ne se retrouve dans le code. L'analyse statique peut aider à implémenter des standards de codage et identifier des problèmes qui seraient trop difficiles à trouver par un simple examen du livrable. Les inspections peuvent non seulement aboutir à la découverte et à la suppression de défauts, mais aussi apprendre aux auteurs à éviter de créer des défauts dans leurs livrables.

Le syllabus ISTQB niveau fondation présente les types de revues suivants:

- Revue informelle
- Relecture technique
- Revue technique
- Inspection

Le Test Manager peut aussi être impliqué dans des:

- Revues de gestion
- Audits

## 3.2 Revues de Gestion et Audits

Les revues de gestion sont utilisées pour suivre l'avancement, évaluer le statut, et prendre des décisions sur des actions futures. Ces revues aident à la prise de décision sur le devenir du projet, comme l'adaptation du niveau de ressources, l'implémentation d'actions correctives ou la modification du périmètre du projet.

Les caractéristiques suivantes sont des éléments clés des revues de gestion:

- Piloté par ou pour des managers ayant une responsabilité directe sur le projet ou le système
- Piloté par ou pour une partie prenante ou un décideur, par ex., un manager hiérarchique supérieur ou un directeur
- Recherche de la cohérence et des écarts par rapport aux plans
- Recherche l'adéquation des procédures de gestion
- Estime les risques projet
- Évalue l'impact des actions et comment mesurer ces impacts
- Produit une liste d'actions, de problèmes à résoudre et de décisions prises.

Les revues de gestion des processus, comme les rétrospectives projet (leçons apprises), sont une partie intégrante des activités d'amélioration de processus.

Le Test Manager devrait participer et peut initier des revues de gestion traitant de l'avancement des tests.

Les audits sont généralement organisés pour démontrer la conformité à un ensemble défini de critères, le plus souvent un standard applicable, une contrainte réglementaire ou une obligation contractuelle. Les audits ont ainsi pour objet de fournir une évaluation indépendante de la conformité aux processus, réglementations, standards, etc. On peut citer comme éléments clés des audits:

- Conduits et modérés par un auditeur en chef
- Collecte de preuves de conformité par des interviews, l'observation et l'examen de documents
- Des résultats documentés incluant des observations, des recommandations, des actions correctives et une évaluation de passage ou d'échec

### 3.3 Gérer les Revues

Les revues devraient être organisées à des moments naturels ou jalons du projet logiciel. typiquement, les revues devraient être organisées après la définition des exigences et la définition de la conception, avec des revues associées aux objectifs métiers et descendant jusqu'au plus bas niveau de conception. Les revues de gestion devraient être organisées aux principaux jalons du projet, souvent dans le cadre de l'activité de vérification, avant, pendant et après l'exécution des tests et des autres phases importantes du projet. La stratégie de revue doit être coordonnée avec la politique de test et la stratégie de test globale.

Avant de définir un plan de revue complet au niveau projet, le responsable de revue (qui peut être un Test Manager) devrait prendre en compte:

- Ce qui doit être revu (produit et processus)
- Qui devrait être impliqué dans chaque revue spécifique
- Quels facteurs de risque importants couvrir

Tôt dans la phase de planification du projet, le responsable des revues devrait identifier les éléments à réviser et sélectionner les types de revue appropriés (revue informelle, relecture technique, revue technique ou inspection, ou un mélange de deux ou trois types de revues) ainsi que le niveau de formalisme. C'est à ce moment que des formations aux revues peuvent être recommandées. A partir de ce moment, un budget (temps de ressources) peut être alloué au processus de revue. La définition de ce budget devrait se baser sur une évaluation des risques et un calcul du retour sur investissement.

Le retour sur investissement des revues est la différence entre le coût de la mise en place de la revue, et le coût de traitement de ces défauts s'ils étaient trouvés (ou non trouvés) à une étape ultérieure si les revues n'avaient pas été faites. Le calcul du coût de la qualité expliqué dans la section 2.7 peut être utilisé pour déterminer cette valeur.

Déterminer le temps optimal pour exécuter une revue, dépend de :

- La disponibilité des éléments à revoir dans un état suffisamment avancé
- La disponibilité des bonnes personnes pour la revue
- Le moment où la version finale de l'élément devrait être disponible
- Le temps nécessaire pour le processus de revue de cet élément particulier

Des métriques adéquats pour l'évaluation des revues devraient être définies par le responsable des revues lors de la planification des tests. Si des inspections sont utilisées, alors des inspections partielles devraient être menées à la demande de l'auteur, selon la disponibilité des fragments de documents (par ex., des exigences individuelles ou sections particulières)

Les objectifs du processus de revue doivent être définis durant la planification des tests. Cela inclut la mise en place de revues rentables et efficaces ainsi que l'atteinte de décisions consensuelles concernant les retours des revues.

Des revues de projets sont souvent tenues pour le système global et peuvent aussi être nécessaires pour des sous-systèmes et même des composants logiciels individuels. Le nombre de revues, le type des revues, l'organisation des revues et les personnes impliquées dépendent de la taille du projet, de sa complexité et des risques produits.

Pour être efficaces, les participants aux revues doivent avoir le bon niveau de connaissance, à la fois sur la technique et procédurale. La complétude et l'attention aux détails sont parmi les autres

compétences nécessaires pour les réviseurs, de façon à avoir des revues efficaces. La clarté et la bonne priorisation sont des attributs à rechercher dans de bons commentaires de revues. Le besoin de connaissances sur les processus peut justifier que des formations soient nécessaires pour s'assurer que les réviseurs comprennent leurs rôles et responsabilités dans le processus de revue.

La planification des revues devrait traiter les risques associés aux facteurs techniques, organisationnels, et humains lors de l'exécution des revues. La disponibilité de réviseurs avec les compétences techniques suffisantes est critique au succès d'une revue. Toutes les équipes du projet devraient être impliquées dans la planification des revues, ce qui devrait garantir l'engagement de chaque équipe par rapport au succès du processus de revue. La planification doit s'assurer que chaque partie prenante alloue suffisamment de temps aux réviseurs désignés pour leur permettre de se préparer et de participer aux revues aux bons moments dans le déroulement du projet. Du temps devrait aussi être prévu pour toute formation, aux techniques et/ou processus de revue, des réviseurs. Des réviseurs de remplacement devraient être identifiés au cas où certains des réviseurs clé deviendraient indisponibles suite à des modifications personnelles ou professionnelles.

Lors du déroulement de revues formelles, le responsable de la revue doit s'assurer que:

- Des mesures appropriées sont fournies par les participants aux revues pour permettre d'évaluer l'efficacité d'une revue
- Des check-lists sont créées et maintenues pour améliorer les revues futures
- L'évaluation de la sévérité et de la priorité des défauts est définie pour utilisation par la gestion des anomalies trouvées durant les revues (voir Chapitre 4)

Après chaque revue, le responsable de revues devrait:

- Collecter les métriques de revue et s'assurer que les problèmes identifiés sont traités pour atteindre les objectifs de la revue
- Utiliser des métriques de revues comme entrées pour déterminer le retour sur investissement (ROI) des revues
- Fournir des informations en retour aux parties prenantes pertinentes
- Fournir des informations en retour aux participants aux revues

Pour évaluer l'efficacité des revues, le Test Manager peut comparer les résultats trouvés lors des tests subséquents aux revues (c.à.d., après les revues) avec les résultats des rapports de revues. Dans le cas où un livrable est revu, approuvé sur la base de la revue, mais jugé défectueux par la suite, le responsable de revues devrait considérer les manières par lesquelles le processus de revue a permis à ces défauts de passer. On peut citer comme causes probables des problèmes avec le processus de revue (par ex., des critères d'entrée/sortie mal définis), une mauvaise composition de l'équipe de revue, des outils de revue inadaptés (check-lists, etc.), une formation et une expérience insuffisantes chez le relecteur, et un temps de préparation de revue insuffisant.

Une typologie particulière de défauts non trouvés et répétés sur plusieurs projets (en particulier des défauts majeurs) indiquera qu'il y a des problèmes significatifs dans la conduite des revues. Dans une telle situation, le responsable de revue a besoin de réviser le processus et de prendre les mesures appropriées. Il est également possible que, pour diverses raisons, les revues puissent perdre de leur efficacité au cours du temps. Un tel effet sera identifié lors de rétrospectives projet par une réduction de l'efficacité de détection des défauts lors des revues. Ici encore, le responsable de revue doit rechercher et résoudre les causes. Dans tous les cas, les métriques des revues ne devraient pas être utilisées pour punir ou récompenser des individus (relecteur ou auteur), mais devraient se focaliser sur le processus de revue en lui-même.

## 3.4 Métriques pour les Revues

Les responsables de revues (qui peuvent être, comme indiqué dans les sections précédentes, le Test Manager) doivent garantir la disponibilité de métriques pour :

- Évaluer la qualité de l'élément revu
- Évaluer le coût de réalisation de la revue
- Évaluer le bénéfice engendré par la conduite de revue

Les responsables de revue peuvent utiliser des mesures pour déterminer le retour sur investissement et l'efficacité des revues. Ces métriques peuvent aussi être utilisées pour fournir des informations pour les activités d'amélioration des processus.

Pour chaque livrable revu, les métriques suivantes peuvent être mesurées et utilisées pour l'évaluation du produit :

- Taille des livrables (pages, lignes de codes, etc.)
- Temps de préparation nécessaire (avant la revue)
- Temps nécessaire pour mener la revue
- Temps nécessaire pour corriger les défauts
- Durée du processus de revue
- Nombre et sévérité des défauts trouvés
- Identification des agrégats de défauts (c.à.d., parties ayant une plus forte densité de défauts)
- Type de revue (revue informelle, relecture technique, revue technique ou inspection)
- Densité de défauts moyenne (par ex., défauts par page ou par millier de lignes de code)
- Estimation du nombre de défauts résiduels (densité de défauts résiduelle)

Pour chaque livrable revu, les métriques suivantes peuvent être mesurées et utilisées pour l'évaluation du processus :

- Efficacité de détection des défauts (prenant en compte les défauts trouvés ultérieurement dans le cycle de vie)
- Amélioration de la charge et du temps alloués au processus de revue
- Pourcentage de couverture des livrables prévus
- Types et sévérité des défauts détectés
- Enquêtes réalisées auprès des participants sur la rentabilité et l'efficacité du processus de revue
- Métrique sur le coût de la qualité pour les défauts identifiés lors de revues, versus coût des défauts trouvés lors des tests dynamiques et des défauts trouvés en production
- Corrélation sur l'efficacité des revues (type de revues versus efficacité de la détection de défauts)
- Nombre de relecteurs
- Nombre de défauts trouvés par heure consacrée à la revue
- Estimation du temps économisé pour le projet
- Effort moyen par défaut (c.à.d., temps total consacré à la détection et à la résolution des défauts divisé par le nombre de défauts)

De plus, les métriques indiquées ci-dessus pour l'évaluation du produit sont aussi utiles à l'évaluation du processus.

## 3.5 Gérer des Revues Formelles



Le syllabus ISTQB niveau fondation décrit les différentes phases d'une revue formelle: planification, lancement, préparation individuelle, réunion de revue, re-travail et suivi. Pour mettre en œuvre correctement les revues formelles, les responsables de revues doivent s'assurer que toutes les étapes du processus de revue sont suivies.

Les principales caractéristiques des revues formelles sont:

- Critères d'entrée et de sortie définis
- Check-lists utilisées par les relecteurs
- Livrables tels que des rapports, des notes d'évaluation et tout autre document résumant les résultats de la revue
- Métriques pour un reporting sur la rentabilité, l'efficacité et la progression des revues

Avant d'initier une revue formelle, la satisfaction des prérequis à la revue (tels que définis dans la procédure ou dans les critères d'entrée) devrait être confirmée par le responsable de la revue.

Si les prérequis pour la revue formelle ne sont pas satisfaits, le responsable de la revue peut proposer l'une des alternatives suivantes au décideur:

- Redéfinition de la revue avec des objectifs différents
- Mise en place d'actions correctives pour permettre de tenir la revue
- Report de la revue

Dans le cadre du contrôle de revues formelles, ces revues sont suivies dans le cadre du programme global (de plus haut niveau) et sont associées à des activités d'assurance qualité du projet. Le contrôle de revues formelles inclut le recueil de feedback utilisant des métriques produit et processus.



## 4. Gestion des Anomalies – 150 mins.

### Mots clé

anomalie, défaut, comité de tri des défauts, défaillance, faux-négatif, faux-positif, maîtrise de la phase, priorité, cause racine, sévérité

### Objectifs d'Apprentissage pour Gestion des Anomalies

#### 4.2 Cycle de Vie des Anomalies et Cycle de développement logiciel

- TM-4.2.1 (K3) Développer un processus de gestion des défauts pour une organisation de test, incluant un cycle de vie des défauts, pouvant être utilisé pour piloter et contrôler les défauts d'un projet tout au long du cycle de vie du test
- TM-4.2.2 (K2) Expliquer le processus et les participants requis pour une gestion efficace des défauts.

#### 4.3 Données d'un Rapport d'Anomalie

- TM-4.3.1 (K3) Définir les données et les informations de classification devant être collectées durant le processus de gestion des défauts

TM-

#### 4.4 Évaluer l'Efficacité d'un Processus avec l'Information des Rapports d'Anomalie

- TM-4.4.1 (K2) Expliquer comment des statistiques sur les rapports de défauts peuvent être utilisées pour évaluer l'efficacité des processus de développement et test logiciels

## 4.1 Introduction

Le processus de gestion des défauts d'une organisation, et l'outil utilisé pour gérer ce travail sont d'une importance capitale, non seulement pour l'équipe de test mais pour toutes les équipes impliquées dans le développement du logiciel. L'information collectée avec une gestion efficace des défauts permet au Test Manager et aux autres parties prenantes du projet d'avoir un aperçu sur l'état du projet tout au long du cycle de développement, et, en collectant et en analysant les données au cours du temps, peut aider à localiser des domaines d'amélioration pour les processus de développement et de test.

En plus de comprendre le processus général de gestion des défauts et comment il peut être utilisé pour mesurer et contrôler à la fois les processus de test et de développement logiciel, le Test Manager doit aussi être familiarisé avec les données qu'il est critique de collecter et doit savoir expliquer l'usage adéquat du processus et de l'outil sélectionné pour la gestion des anomalies.

## 4.2 Cycle de Vie des Anomalies et Cycle de développement logiciel

Comme expliqué dans le syllabus niveau fondation, les défauts sont introduits quand une personne commet une erreur lors de la création d'un livrable. Ce livrable peut être une spécification d'exigence, une user story, un document technique, un cas de test, le code du programme, ou tout autre livrable produit durant le processus de développement ou de maintenance du logiciel.

Les défauts peuvent être introduits à tout moment dans le cycle de développement logiciel et dans tout livrable associé au logiciel. Par conséquent, chaque phase du cycle de développement logiciel devrait intégrer des activités pour détecter et supprimer les défauts potentiels. Par exemple, des techniques de test statique (c.à.d., revues et analyse statique) peuvent être utilisées sur les spécifications de conception, les spécifications d'exigences, et le code, avant leur livraison aux activités suivantes. Plus tôt un défaut est détecté et supprimé, plus bas sera le coût global de la qualité du système ; le coût de la qualité pour un niveau donné de défauts sera minimisé quand un défaut est supprimé dans la phase durant laquelle il a été introduit (c.à.d., quand le processus logiciel atteint une maîtrise parfaite de la phase). De plus, comme indiqué dans le syllabus niveau fondation, le test statique trouve directement des défauts, et non des défaillances, et donc, le coût de suppression du défaut est plus bas car les activités de débogage ne sont pas requises pour isoler le défaut.

Pendant des activités de test dynamique comme le test de composant, le test d'intégration, et le test système, la présence d'un défaut est révélée quand il cause une défaillance qui résulte en une divergence entre le résultat obtenu et le résultat attendu pour un test (c.à.d., une anomalie). Dans certains cas, un faux-négatif arrive lorsque le testeur n'observe pas l'anomalie. Si le testeur observe l'anomalie, on se trouve dans une situation où une investigation ultérieure est nécessaire. Cette investigation commence avec le remplissage d'un rapport d'anomalie.

Avec le développement dirigé par les tests, des tests unitaires automatisés sont utilisés comme une forme de spécifications de conception. Au fur et à mesure que le code est développé il est immédiatement exécuté par ces tests. Tant que le développement des composants n'est pas terminé, un ou plusieurs tests vont échouer. Ainsi les défaillances de tels tests ne constituent pas un réel défaut et ne sont typiquement pas suivies.

## 4.2.1 Cycle d'une Anomalie et États

La plupart des organisations de test utilisent un outil pour gérer les rapports d'anomalie tout au long du cycle de vie des défauts. Un rapport d'anomalie évolue typiquement selon une séquence d'états au fur et à mesure qu'il progresse dans le cycle de vie de l'anomalie. Pour la plupart de ces états, un seul responsable du cycle du défaut détient le rapport et est responsable d'une tâche qui, une fois terminée, permettra au rapport d'anomalie de passer à l'état suivant (et d'être affecté au responsable suivant). Dans les derniers états, le rapport ne désigne pas nécessairement un porteur puisque il n'y a plus d'action sur le défaut. C'est le cas par exemple, lorsque le rapport d'anomalie est clôturé (ce qui signifie que le défaut concerné est corrigé et que la correction a été vérifiée par un test de confirmation), annulé (ce qui signifie en général que le rapport d'anomalie n'est pas valide), non-reproductible (ce qui signifie en générale qu'il n'est pas possible d'observer l'anomalie), ou reporté (ce qui signifie en général que l'anomalie fait référence à un vrai défaut mais que ce défaut ne sera pas corrigé pendant le projet).

Pour les défauts découverts par les testeurs pendant les tests, il y a trois états particuliers où les actions sont avec l'équipe de test:

- L'état "initial"
  - Dans cet état, un ou plusieurs testeurs collectent l'information qui sera nécessaire à la personne responsable de la résolution du défaut pour reproduire l'anomalie (voir section 4.3 pour l'information à inclure dans le rapport d'anomalie).
  - Cet état peut aussi être appelé "ouvert" ou "nouveau".
- L'état "retourné"
  - Dans cet état, le destinataire du rapport a rejeté ce rapport ou a demandé au testeur de fournir davantage d'informations. Cet état peut indiquer un manque dans le processus initial de recueil d'informations ou dans le test lui-même, et les Test Managers devraient mesurer les taux de retour excessifs. Le(s) testeur(s) doi(ven)t fournir l'information complémentaire, ou confirmer que le rapport devrait en effet être rejeté.
  - Cet état peut aussi être appelé "rejeté" ou "clarification".
- L'état "test de confirmation"
  - Dans cet état, le testeur va exécuter un test de confirmation (généralement, en suivant les étapes indiquées dans le rapport d'anomalie pour reproduire la défaillance) pour déterminer si la correction a bien résolu le problème. Si le test de confirmation indique que le défaut est réparé, le testeur devrait clôturer le rapport. Si le test de confirmation indique que le défaut n'est pas réparé, le testeur devrait réouvrir le rapport, déclenchant sa réaffectation au porteur précédent, qui pourra alors compléter le travail nécessaire pour réparer le défaut.
  - Cet état peut aussi être appelé "résolu" ou "vérification".

## 4.2.2 Gérer des Rapports d'Anomalie Invalides ou Dupliqués

Dans certains cas, une anomalie n'apparaît pas comme le symptôme d'un défaut, mais à cause d'un problème avec l'environnement de test, les données de test, d'autres éléments liés aux articles de test, ou une mauvaise compréhension de la part du testeur lui-même. Si le testeur ouvre un rapport d'anomalie qui se révèle ne pas être lié à un défaut dans le produit en cours de test, c'est un faux-positif. De tels rapports sont en général annulés ou clôturés comme invalides. De plus, dans certains cas un défaut peut montrer des symptômes différents qui peuvent sembler pour le testeur totalement indépendants. Si plusieurs rapports d'anomalie s'avèrent être liés à la même cause racine, l'un d'entre eux pourra être conservé et les autres clôturés en tant que rapports d'anomalie dupliqués.

Bien que les rapports invalides ou dupliqués représentent un certain niveau d'inefficacité, un certain nombre de tels rapports est inévitable et devrait être accepté comme tel par le Test Manager. Quand des managers essaient d'éliminer tous les rapports d'anomalie invalides ou dupliqués, le nombre de

faux-négatifs a tendance à augmenter, vu que les testeurs sont découragés de remplir des rapports d'anomalies. Cela fait baisser l'efficacité de la détection des défauts de l'organisation, qui est un objectif essentiel de l'organisation de test.

#### 4.2.3 Gestion Fonctionnelle Transverse des Défauts

Bien que l'organisation de test et le Test Manager sont typiquement responsables du processus global de gestion des anomalies et des outils de gestion des anomalies, une équipe fonctionnelle transverse est généralement responsable de la gestion du reporting des défauts pour un projet donné. En plus du Test Manager, les participants au comité de gestion des anomalies (ou comité de tri des défauts) incluent généralement le développement, la gestion du projet, la gestion du produit et d'autres parties prenantes ayant un intérêt dans le logiciel en développement.

Au fur et à mesure de la découverte et de l'introduction des anomalies dans l'outil de gestion des anomalies, le comité de gestion des anomalies devrait se rencontrer pour déterminer si chaque rapport d'anomalie représente réellement un défaut, et si celui-ci devrait être corrigé ou repoussé. Cette décision implique l'étude, par le comité de gestion des anomalies, des bénéfices, risques et coûts associés à la résolution ou non du défaut. Si le défaut doit être corrigé, l'équipe devrait définir la priorité de correction par rapport aux autres tâches du projet. Le Test Manager et l'équipe de test peuvent être consultés au sujet de l'importance relative d'une anomalie et devraient fournir toute information objective disponible.

Un outil de suivi des défauts ne devrait pas être utilisé comme substitut d'une bonne communication, de même que les réunions du comité de gestion des défauts ne devraient pas se substituer à l'usage d'un bon outil de suivi des défauts. Une bonne communication, un support d'outil adapté, un cycle de vie d'anomalie bien défini, et un comité de gestion des anomalies actif sont tous des éléments nécessaires à une gestion des défauts rentable et efficace.

### 4.3 Données d'un Rapport d'Anomalie

Quand un défaut est détecté (dans le cadre de tests statiques) ou une défaillance observée (via du test dynamique), les données devraient être réunies par la ou les personne(s) impliquée(s) et intégrées au rapport d'anomalie. L'information doit permettre trois choses:

- La gestion de l'anomalie selon le cycle de vie des anomalies
- Une évaluation du statut du projet, spécifiquement en termes de qualité produit et avancement du test
- L'évaluation de la capacité du processus (comme discuté dans la section 4.4 ci-dessous)

Les données nécessaires à la gestion des rapports d'anomalies et du statut du projet peuvent varier selon le moment auquel le défaut est détecté dans le cycle de vie, avec notamment moins d'information nécessaire au début (par ex., revues d'exigences et tests unitaires). Cependant, l'information principale collectée devrait être cohérente tout au long du cycle de vie aussi, idéalement, entre tous les projets pour permettre une comparaison significative du processus de gestion des anomalies sur l'ensemble d'un projet mais aussi entre tous les projets.

Les données collectées sur les défauts peuvent aider au contrôle de l'avancement des tests et à l'évaluation des critères de sortie. Par exemple, l'information sur les défauts devrait permettre d'aider l'analyse de la densité des défauts, l'analyse de tendances des défauts détectés et résolus, le temps moyen entre la détection et la correction du défaut, et l'intensité des défaillances (par ex., analyse MTBF: temps moyen entre deux défaillances).

Parmi les données collectées sur les anomalies, on peut citer:



- Le nom de la personne ayant découvert le défaut
- Le rôle de la personne (par ex., utilisateur final, analyste métier, développeur, personne du support technique)
- Le type de test réalisé (par ex., utilisabilité, performance, non-régression)
- Un résumé du problème
- Une description détaillée du problème
- Les étapes nécessaires pour reproduire la défaillance (pour un défaut), ainsi que les résultats attendus et obtenus (mettant en évidence l'anomalie), intégrant des copies d'écran, des extraits de bases de données et de logs là où c'est possible
- Les phases du cycle de vie durant lesquelles le défaut a été introduit, détecté et supprimé, y compris le niveau de test si c'est applicable
- Le livrable dans lequel le défaut a été introduit
- La sévérité de l'impact sur le système et/ou les parties prenantes produit (en général déterminé par le comportement technique du système)
- La priorité de correction du problème (généralement déterminé par l'impact métier de la défaillance)
- Le sous-système ou composant auquel le défaut est attaché (pour l'analyse des agrégats de défauts)
- L'activité en cours sur le projet lorsque le problème a été détecté
- La méthode d'identification qui a révélé le problème (par ex., revue, analyse statique, test dynamique, utilisation en production)
- Le type de défaut (correspondant en général à la taxonomie utilisée)
- La caractéristique qualité affectée par le défaut
- L'environnement de test dans lequel le défaut a été observé (pour le test dynamique)
- Le projet et produit dans lesquels le problème se trouve
- Le propriétaire actuel; c.à.d., la personne actuellement affectée au problème, à condition que le rapport ne soit pas dans un état final
- L'état courant du rapport (en général géré par l'outil de suivi des anomalies dans le cycle de vie de l'anomalie)
- Les livrables spécifiques (par ex., éléments de test avec leur numéro de version) dans lesquels le problème a été observé, ainsi que le livrable spécifique où le problème a été définitivement résolu
- L'impact sur les intérêts des parties prenantes du projet et du produit
- Les conclusions, recommandations et approbations des actions prises ou non pour résoudre le problème
- Les risques, coûts, opportunités, et bénéfices associés à la correction ou non du défaut
- Les dates de transitions d'état du défaut dans son cycle de vie, le propriétaire du défaut au moment de chaque transition, et les actions prises par l'équipe projet pour isoler, réparer et vérifier la correction
- Une description de la façon de résoudre le défaut et des recommandations pour le test de la correction (si le défaut a été résolu par un changement dans le logiciel)
- D'autres références, comme le test qui a mis en évidence le défaut et le risque, l'exigence, ou tout autre élément relié au défaut (pour le test dynamique)

Différents standards et documents, tels que ISO 9126 (en cours de remplacement par ISO25000), IEEE 829 [IEEE829], IEEE 1044 [IEEE1044], et Orthogonal Defect Classification (Classification Orthogonale des Défauts), existent et peuvent aider le Test Manager à déterminer l'information devant être recueillie pour le reporting.

Quelle que soit l'information définie comme nécessaire pour les rapports d'anomalies, il est essentiel que les testeurs la saisissent à temps de façon complète, concise, précise, objective et pertinente. Même quand une intervention manuelle ou des discussions directes peuvent permettre de surmonter

des problèmes dus au contenu des rapports d'anomalies en termes de résolution d'un défaut particulier, les problèmes avec les rapports d'anomalie peuvent engendrer des obstacles insurmontables à la réalisation d'une évaluation fiable du statut d'un projet, de l'avancement du test et de capacité des processus.

#### 4.4 Évaluer l'Efficacité d'un Processus avec l'Information des Rapports d'Anomalie

Comme discuté au Chapitre 2, les rapports d'anomalies peuvent être utilisés au suivi et au contrôle du statut du projet. Bien que l'implication des métriques dans les processus est principalement traitée dans le syllabus de Test Manager niveau Expert [ISTQB ETM SYL], au niveau Avancé les Test Managers devraient être conscients de ce que les rapports d'anomalies indiquent en termes d'évaluation de la capacité des processus de tests et de développement des logiciels.

En plus des éléments de mesure de l'avancement des tests mentionnés au chapitre 2 et dans la section 4.3, les informations relatives aux anomalies doivent supporter les initiatives d'amélioration des processus. On peut citer comme exemples:

- Utiliser l'information sur la phase d'introduction, de détection, et de suppression, pour évaluer, phase par phase le confinement par phase et suggérer des manières d'améliorer l'efficacité de détection des défauts dans chaque phase
- Utiliser l'information sur la phase d'introduction pour une analyse, selon la loi de Pareto, des phases dans lesquelles le plus grand nombre de défauts sont introduits, afin d'identifier des améliorations ciblées visant à réduire le nombre total de défauts
- Utiliser les informations de cause racine des défauts de l'analyse des causes racines des défauts pour déterminer les raisons sous-jacentes à l'introduction d'un défaut afin d'améliorer le processus pour réduire le nombre total de défauts
- Utiliser l'information sur les phases d'introduction, de détection, et de suppression, pour réaliser une analyse du coût de la qualité afin de minimiser le coût associé aux défauts
- Utiliser l'information sur les composants défaillants pour effectuer une analyse des agrégats de défauts afin de mieux comprendre les risques techniques (pour du test basé sur les risques) et permettre une ré-ingénierie des composants problématiques

L'utilisation de métriques pour évaluer la rentabilité et l'efficacité du processus de test est traitée dans le syllabus Test Manager niveau Expert.

Dans certains cas, les équipes peuvent choisir de ne pas suivre les défauts trouvés à certains moments ou tout au long du cycle de développement logiciel. Même si ceci est effectué au nom de l'efficacité et dans le but de réduire les surcoûts projet, en réalité, cela réduit considérablement la visibilité sur l'efficacité des processus de test et de développement logiciels. Cela rend les améliorations suggérées plus haut difficiles à mettre en œuvre par manque de données fiables.



## 5. Améliorer le Processus de Test – 135 mins.

### Mots clé

Capability Maturity Model Integration (CMMI), Critical Testing Processes (CTP), Systematic Test and Evaluation Process (STEP), Test Maturity Model integration (TMMi), TPI Next

### Objectifs d'Apprentissage pour Améliorer le Processus de Test

#### 5.2 Processus d'Amélioration des Tests

TM-5.2.1 (K2) Expliquer, en utilisant des exemples, pourquoi il est important d'améliorer le processus de test

#### 5.3 Améliorer le Processus de Test

TM-5.3.1 (K3) Définir un plan d'amélioration du processus de test en utilisant le modèle IDEAL

#### 5.4 Améliorer le Processus de Test avec TMMi

TM-5.4.1 (K2) Résumer l'historique, le périmètre et les objectifs du modèle d'amélioration de processus TMMi

#### 5.5 Améliorer le Processus de Test avec TPI Next

TM-5.5.1 (K2) Résumer l'historique, le périmètre et les objectifs du modèle d'amélioration de processus TPI Next

#### 5.6 Améliorer le Processus de Test avec CTP

TM-5.6.1 (K2) Résumer l'historique, le périmètre et les objectifs du modèle d'amélioration de processus CTP

#### 5.7 Améliorer le Processus de Test avec STEP

TM-5.7.1 (K2) Résumer l'historique, le périmètre et les objectifs du modèle d'amélioration de processus STEP



## 5.1 Introduction

Une fois en place, le processus de test global de l'organisation devrait s'améliorer de façon continue. Ce chapitre couvre d'abord les problématiques de l'amélioration générique puis introduit certains modèles spécifiques qui peuvent être utilisés pour l'amélioration de processus de test. Le Test Manager devrait supposer que ces éléments seront le moteur de tout changement ou amélioration et devrait donc être familiarisé avec les techniques acceptées dans l'industrie discutées dans ce chapitre. Davantage d'information sur les processus d'amélioration du test est discuté dans le syllabus Niveau Expert : Améliorer le Processus de Test.

## 5.2 Processus d'Amélioration des Tests

De même que les organisations utilisent le test pour améliorer le logiciel, les techniques d'amélioration de processus peuvent être sélectionnées et utilisées pour améliorer le processus de développement logiciel et les livrables logiciels associés. L'amélioration de processus peut aussi être appliquée au processus de test. Différentes façons de faire et méthodes sont disponibles pour améliorer le test des logiciels et des systèmes contenant du logiciel. Ces méthodes ont pour objectif d'améliorer le processus, et par conséquent les livrables, en fournissant des recommandations et des axes d'amélioration.

Les tests représentent souvent une part importante du coût total d'un projet. Cependant, les modèles d'amélioration de processus n'y accordent qu'une faible part, comme avec CMMI® (voir ci-dessous pour plus de détails).

Les modèles d'amélioration du test, tels que Test Maturity Model integration (TMMi®), Systematic Test and Evaluation Process (STEP), Critical Testing Processes (CTP) et TPI Next® ont été développés pour adresser ce manque d'attention au test dans la plupart des modèles d'amélioration de processus. Bien utilisés, ces modèles peuvent fournir des métriques transverses à l'organisation utiles à la réalisation de comparaisons ou benchmarks.

Les modèles présentés dans ce syllabus ne doivent pas être pris pour des recommandations d'utilisation, mais sont présentés pour fournir une vue représentative de comment ces modèles fonctionnent et de ce qu'ils incluent.

### 5.2.1 Introduction à l'amélioration de processus

Les améliorations de processus sont utiles au processus de développement logiciel et au processus de test. Apprendre de ses erreurs rend possible l'amélioration des processus utilisés par une organisation pour développer et tester du logiciel. Le cycle d'amélioration de Deming: Planifier (Plan), Faire (Do), Vérifier (Check), Agir (Act), est utilisé depuis des décennies, et reste pertinent quand les testeurs doivent améliorer le processus en vigueur.

Un des postulats de l'amélioration de processus est de croire que la qualité d'un système est fortement influencée par la qualité du processus utilisé pour développer le logiciel. Améliorer la qualité dans l'industrie logicielle diminuera le besoin en ressources pour maintenir le logiciel et laissera donc plus de temps pour créer plus de solutions de meilleure qualité dans le futur. Les modèles de processus constituent un point de départ à l'amélioration, en mesurant la capacité des processus d'une organisation par rapport à un modèle. Les modèles fournissent aussi un cadre d'amélioration des processus de l'organisation s'appuyant sur les résultats d'une évaluation.

Une évaluation des processus permet de déterminer l'efficacité d'un processus, ce qui motive l'amélioration des processus. Cela peut impliquer une évaluation subséquente des processus, pour mesurer l'effet de l'amélioration.

## 5.2.2 Types de processus d'amélioration

L'utilisation de modèles d'évaluation est une méthode classique qui assure une approche standardisée à l'amélioration de processus, sur la base de pratiques éprouvées et reconnues.

Les modèles d'amélioration de processus se classent en deux types:

1. Le modèle de référence de processus qui fournit une mesure de la maturité dans le cadre de l'évaluation pour évaluer la capacité de l'organisation par rapport au modèle, pour évaluer l'organisation au sein du modèle, et pour fournir un plan d'amélioration des processus
2. Le modèle de référence de contenu qui fournit des évaluations orientées métier d'opportunités d'amélioration de l'entreprise incluant, dans certains cas, des comparaisons par rapport à des moyennes constatées dans l'industrie, sur la base de mesures objectives. Cette évaluation peut être utilisée pour créer un plan d'amélioration des processus.

L'amélioration du processus de test peut aussi être réalisée sans modèle en utilisant, par exemple, des approches analytiques et des réunions de rétrospective projets.

## 5.3 Améliorer le Processus de Test

L'industrie informatique peut travailler avec des modèles d'amélioration de processus pour atteindre un niveau supérieur de maturité et de professionnalisme. Des modèles de standards de l'industrie aident à développer des métriques et mesures transverses à l'organisation pouvant être utilisées pour comparer. Vu le besoin d'amélioration des processus dans l'industrie des tests, plusieurs ensembles de processus recommandés ont vu le jour. Ceux-ci incluent STEP, TMMi, TPI Next et CTP. Les modèles étagés tels que TMMi et CMMI, fournissent des standards permettant la comparaison entre diverses organisations et sociétés. Les modèles continus, comme CTP, STEP et TPI Next, permettent à une organisation de traiter ses priorités les plus hautes avec plus de liberté dans l'ordre de mise en œuvre. Ceux-ci sont discutés dans ce chapitre.

Tous ces modèles permettent à une organisation de déterminer où elle se situe en termes de ses processus de test actuels. Une fois l'évaluation faite, TMMi et TPI Next suggèrent un plan d'amélioration du processus de test. Alternativement, STEP et CTP fournissent à l'organisation les moyens de déterminer où les meilleurs retours sur investissement en amélioration des processus arriveront et laissent à l'organisation choix de construire un plan approprié.

Une fois qu'il a été accepté que les processus de test devraient être revus et améliorés, les étapes de mise en œuvre d'une amélioration de processus pouvant être adoptées pour cette activité pourraient correspondre à celles définies dans le modèle IDEAL<sup>SM</sup> [IDEAL96]:

- Initiating: Initier le processus d'amélioration
- Diagnosing: diagnostiquer la situation actuelle
- Establishing: établir un plan d'amélioration de processus
- Acting: agir pour mettre en œuvre l'amélioration
- Learning: apprendre du programme d'amélioration

### **Initiating: Initier le processus d'amélioration**

Avant le début des activités d'amélioration de processus, les objectifs, les buts, le périmètre et la couverture des améliorations de processus doivent être convenus par les parties prenantes. Le choix

du modèle d'amélioration de processus est aussi fait à ce moment. Le modèle peut soit être choisi parmi ceux du domaine public (tels que CTP, STEP, TMMi, et TPI Next) soit développé en interne. De plus, les critères de succès devraient être définis et la méthode à utiliser pour les mesurer tout au long de l'activité d'amélioration devrait être déterminée.

#### **Diagnosing: diagnostiquer la situation actuelle**

L'approche d'évaluation convenue est mise en œuvre et un rapport d'évaluation est créé lequel contient une estimation des pratiques de test actuelles et une liste d'améliorations possibles des processus.

#### **Establishing: établir un plan d'amélioration de processus**

Une liste d'améliorations possible des processus est priorisée. La priorisation pourrait être basée sur le retour sur investissement, les risques, le respect d'une stratégie de l'organisation, et/ou des bénéfices qualitatifs ou quantitatifs mesurables. Une fois établi l'ordre de priorité, un plan de mise en œuvre des améliorations est développé.

#### **Acting: agir pour mettre en œuvre l'amélioration**

Le plan d'amélioration des processus de test est implémenté. Cela pourrait inclure toutes les formations ou coaching nécessaires, le pilotage des processus et leur déploiement complet.

#### **Learning: apprendre du programme d'amélioration**

Une fois les améliorations de processus entièrement déployées, il est essentiel de vérifier quels bénéfices ont été atteints (en considérant les bénéfices escomptés initialement mais aussi les éventuels bénéfices non-prévus). Il est aussi important de regarder quels critères de succès ont été atteints en termes d'activités d'amélioration de processus.

Selon le modèle de processus utilisé, cette étape du processus est où commence la mesure du prochain niveau de maturité et où est prise la décision d'arrêter l'activité d'amélioration ou de recommencer le processus d'amélioration à nouveau.

## 5.4 Améliorer le Processus de Test avec TMMi

Le Testing Maturity Model integration (TMMi) est composé de cinq niveaux de maturité et se veut un complément à CMMI. Chaque niveau de maturité contient des domaines de processus devant être atteint à 85% en atteignant des objectifs spécifiques et génériques avant que l'organisation ne puisse avancer au niveau suivant.

Les niveaux de maturité TMMi sont:

- Niveau 1: Initial  
Le niveau initial correspond à un état où le processus de test n'est ni documenté ni formellement structuré. Les tests sont développés de façon ad hoc après le codage, et test et débogage sont considérés sans distinction. Le test est simplement perçu comme quelque chose dont l'objectif est de prouver que le logiciel fonctionne.
- Niveau 2: Géré  
Le second niveau est atteint lorsque les processus de test sont clairement séparés du débogage. Il peut être atteint en définissant une politique de test et des objectifs, en introduisant les étapes constitutives du processus de test fondamental (par ex., planification des tests), et en implémentant les techniques et méthodes de test de base.
- Niveau 3: Défini  
Le troisième niveau est atteint quand un processus de test est intégré dans le cycle de développement logiciel, et documenté selon des standards, procédures et méthodes

formelles. Des revues sont organisées et une activité de test logiciel distincte, pouvant être contrôlée et pilotée, doit exister.

- Niveau 4: Mesuré  
Le niveau quatre est atteint quand le processus de test peut être mesuré efficacement et géré au niveau de l'organisation afin d'en faire bénéficier chaque projet.
- Niveau 5: Optimisé  
Le niveau final correspond à l'état de maturité où les données issues du processus de test peuvent être utilisées pour aider à prévenir les défauts, et le focus est sur l'optimisation du processus en place.

Pour plus d'information sur TMMi, voir [vanVeenendaal11] et [www.tmmi.org].

## 5.5 Améliorer le Processus de Test avec TPI Next

Le modèle TPI définit 16 domaines clé, couvrant chacun un aspect spécifique du processus de test, comme la stratégie de test, les métriques, les outils et les environnements de test.

Le modèle définit quatre niveaux de maturité:

- Initial
- Contrôlé
- Efficace
- Optimisé

Des points de contrôle spécifiques sont définis pour évaluer chaque domaine clé à chaque niveau de maturité. L'information recueillie est synthétisée et représentée via une matrice de maturité couvrant tous les domaines clé. La définition des objectifs d'amélioration et leur implémentation peuvent être ajustés selon les besoins et les capacités de l'organisation de test.

L'approche générique rend TPI Next indépendant de tout modèle d'amélioration de processus. Cela couvre à la fois les aspects d'ingénierie des tests et les aspects liés au support à la prise de décision de gestion. [deVries09].

Pour plus d'information sur TPI Next, voir [www.tpinext.com]

## 5.6 Améliorer le Processus de Test avec CTP

Le postulat de base du modèle d'évaluation Critical Testing Processes (CTP) est que certains processus de test sont critiques. Ces processus de test critiques, s'ils sont bien mis en œuvre, contribueront au succès des équipes de test. A l'inverse, si ces activités ne sont pas mises en œuvre correctement, même des testeurs et Test Managers brillants auront peu de chance de réussir. Le modèle identifie douze processus critiques. CTP est essentiellement un modèle de référence de contenu.

Le modèle CTP est une approche dépendant du contexte qui permet d'ajuster le modèle avec:

- L'identification de challenges spécifiques
- La reconnaissance des attributs de bons processus
- La sélection d'un ordre et de l'importance d'implémentation des améliorations de processus

Le modèle CTP est adaptable au contexte de tout modèle de développement logiciel.

En plus d'interviews de participants, le modèle CTP inclut l'usage de métriques pour comparer les organisations et les meilleures pratiques de l'industrie.



Pour plus d'information sur CTP, voir [Black03].

## 5.7 Améliorer le Processus de Test avec STEP

STEP (Systematic Test and Evaluation Process), comme CTP, et à la différence de TMMi et TPI Next, n'impose pas un ordre spécifique dans la mise en œuvre des améliorations.

STEP est principalement un modèle de référence de contenu basé sur l'idée que le test est une activité du cycle de vie qui commence avec la formulation des exigences et continue jusqu'au retrait du système. La méthodologie STEP met en avant le "tester puis coder" en utilisant une stratégie de test basée sur les exigences pour s'assurer que la création de cas de test précoces valide les spécifications d'exigences avant le démarrage de la conception et du codage.

Les bases de la méthodologie comprennent:

- Une stratégie de test basée sur les exigences
- Le commencement du test au début du cycle de vie
- Les tests sont utilisés comme modèles d'exigences et d'usage
- La conception des articles de test dirige la conception du logiciel
- Les défauts sont détectés précocement ou totalement évités
- Les défauts sont systématiquement analysés
- Testeurs et développeurs travaillent ensemble

Dans certains cas, le modèle d'évaluation STEP est mélangé avec le modèle de maturité TPI Next.

Pour plus d'information sur STEP, voir [Craig02].



## 6. Outils de Test et Automatisation – 135 min

### Mots clé

Outil open-source, outil sur-mesure

### Objectifs d'Apprentissage pour Outils de Test et Automatisation

#### 6.2 Sélection d'un Outil

- TM-6.2.1 (K2) Décrire les problèmes de gestion lors de la sélection d'un outil open-source
- TM-6.2.2 (K2) Décrire les problèmes de gestion lors du choix d'un outil sur-mesure
- TM-6.2.3 (K4) Evaluer une situation particulière afin d'établir un plan pour la sélection des outils, intégrant les risques, les coûts et les bénéfices

#### 6.3 Cycle de vie d'un Outil

- TM-6.3.1 (K2) Expliquer les différentes phases dans le cycle de vie d'un outil

#### 6.4 Métriques sur les Outils

- TM-6.4.1 (K2) Décrire comment le recueil de métriques et les évaluations peuvent être améliorés par l'utilisation d'outils

## 6.1 Introduction

Cette partie étend le syllabus niveau Fondation en couvrant un nombre de concepts généraux que le Test Manager doit considérer en ce qui concerne les outils et l'automatisation des tests.

## 6.2 Sélection d'un Outil

Il y a différents aspects à prendre en compte par le Test Manager lors du choix d'outils de test

Le choix le plus fréquent, historiquement, est d'acheter un outil à un éditeur de logiciel. Dans certains cas, cela peut constituer le seul choix viable. Cependant, il y a d'autres possibilités, comme le choix d'outils open-source et d'outils sur-mesure, qui sont des options faisables.

Indépendamment du type d'outil, un Test Manager doit soigneusement investiguer le coût total de possession d'un outil sur toute la durée de son utilisation prévue, en effectuant une analyse des coûts et bénéfices. Ce sujet est couvert plus bas dans la partie retour sur investissement (ROI, Return On Investment).

### 6.2.1 Outils Open-Source

Des outils open-source existent pour presque toutes les facettes du processus de test, de la gestion des cas de test au suivi des anomalies ou à l'automatisation des cas de tests pour n'en citer que quelques-uns. Une distinction importante des outils open-source est que si l'outil en lui-même n'a généralement pas un coût d'achat élevé, il peut ne pas y avoir de support formel disponible pour cet outil. Beaucoup d'outils open-source, cependant, ont un suivi dédié qui est d'accord pour fournir un support non-traditionnel ou informel aux utilisateurs.

De plus, beaucoup d'outils open-source ont été créés à l'origine pour résoudre un problème spécifique ou répondre à un problème particulier; par conséquent, l'outil peut ne pas apporter toutes les fonctions que l'on trouverait dans l'outil équivalent chez un éditeur d'outils. A cause de cela, une analyse soignée des besoins réels de l'équipe de test devrait être effectuée avant la sélection d'un outil open-source.

L'un des bénéfices de l'utilisation d'outils open-source est qu'ils peuvent en général être modifiés ou améliorés par leurs utilisateurs. Si l'organisation de test a les compétences nécessaires, l'outil peut être modifié pour fonctionner avec d'autres outils ou être modifié pour satisfaire les besoins de l'équipe de test. Plusieurs outils peuvent être combinés pour traiter des problèmes que des outils commerciaux ne peuvent traiter. Bien sûr, au plus les outils utilisés sont nombreux et au plus les modifications sont importantes, au plus cela ajoutera de la complexité et des coûts. Un Test Manager doit s'assurer que l'équipe de test ne commence pas à utiliser des outils open-source uniquement pour le plaisir de les utiliser; comme pour les autres outils, l'effort doit rester ciblé sur l'obtention d'un ROI positif.

Un Test Manager doit comprendre le mode de licence de l'outil sélectionné. Beaucoup d'outils open-source ont une licence basée sur GNU General Public License qui précise que la distribution du logiciel doit toujours être sous les mêmes termes que sa réception. Si l'équipe de test effectue des changements dans l'outil pour faciliter ses tests, ces changements peuvent devoir être rendus disponibles pour tous les utilisateurs externes de cet outil selon cette licence. Un Test Manager devrait vérifier les ramifications légales de redistribution de cet outil pour sa société.



Les organisations qui développent des logiciels à sécurité critique ou à missions critiques, ou qui sont soumises à des conformités réglementaires, peuvent avoir des problèmes avec l'utilisation d'outils open-source. Bien que beaucoup d'outils open-source sont de très bonne qualité, l'exactitude d'un outil open-source donné n'a probablement pas été certifiée. Les vendeurs d'outils sont souvent certifiés par rapport à leur exactitude et pertinence pour une tâche particulière (par ex., DO-178B). Bien que l'outil open-source puisse être aussi bon, la certification peut devenir la responsabilité du groupe qui l'utilise, constituant ainsi un surcroît de travail.

## 6.2.2 Outils Maison

L'organisation de test peut identifier qu'elle a un besoin spécifique pour lequel aucun outil commercial ou open-source ne répond. Cela peut être dû à une plateforme matérielle propriétaire, un environnement particulier ou encore un processus modifié d'une façon inhabituelle. Dans ce cas, si les compétences nécessaires existent dans l'équipe, le Test Manager peut souhaiter considérer le développement d'un outil sur-mesure.

Les bénéfices du développement d'un outil sur-mesure sont que l'outil peut répondre précisément aux besoins de l'équipe et peut être opérationnel dans le contexte requis par l'équipe. L'outil peut être écrit pour s'interfacer avec d'autres outils utilisés et pour générer des données exactement dans le format attendu par l'équipe. De plus, l'outil peut avoir des applications dans l'organisation, au-delà du projet en cours. Cependant, avant de prévoir la mise à disposition de l'outil pour d'autres projets, il est important de considérer l'objectif, les buts, bénéfices et inconvénients potentiels.

Un Test Manager qui envisage de créer des outils sur-mesure doit aussi prendre en compte les problèmes potentiels. Les outils sur mesure sont souvent dépendants des personnes qui les ont créés. Par conséquent, les outils sur-mesure doivent être documentés adéquatement afin de pouvoir être maintenus par d'autres. Si ceci n'est pas fait, ils peuvent se retrouver abandonnés et inutilisés dès que leur créateur quitte le projet. Dans le temps, le périmètre d'un outil sur-mesure peut être étendu au-delà du périmètre envisagé, ce qui peut causer des problèmes de qualité dans l'outil, aboutissant à des rapport d'anomalies pour de faux-positifs ou à la création de données inexactes. Le Test Manager doit garder à l'esprit qu'un outil sur-mesure n'est autre qu'un produit logiciel et donc, à ce titre, sujet aux mêmes problématiques de développement que tout autre produit logiciel. Les outils sur-mesure devraient être conçus et testés pour s'assurer qu'ils fonctionnent comme attendu.

## 6.2.3 Retour sur Investissement (ROI)

Il est de la responsabilité du Test Manager de s'assurer que tous les outils introduits dans l'organisation de test ajoutent de la valeur au travail de l'équipe et peuvent montrer un ROI positif pour l'organisation. Pour assurer qu'un outil apportera des bénéfices réels et durables, une analyse coûts-bénéfices devrait être effectuée avant l'achat ou la conception d'un outil. Dans cette analyse, le calcul du ROI devrait prendre en compte à la fois les coûts récurrents et non-récurrents, certains se chiffrant en argent et d'autres sous forme de ressources ou de temps, et les risques pouvant réduire la valeur de l'outil.

Les coûts non-récurrents incluent :

- Définition des exigences de l'outil pour atteindre les objectifs et buts
- Évaluation et sélection du bon outil et du bon fournisseur
- Achat, adaptation ou développement de l'outil
- Réalisation de la formation initiale à l'outil
- Intégration de l'outil avec d'autres outils
- Acquisition du matériel/logiciel nécessaire au support de l'outil

Les coûts récurrents incluent :

- Possession de l'outil

- Frais de licences et de support
- Coûts de maintenance pour l'outil en lui-même
- Maintenance des artefacts créés par l'outil
- Coûts de formation continue et de conseil ou coaching
- Portage de l'outil sur les différents environnements
- Adaptation de l'outil aux nouveaux besoins
- Amélioration de la qualité et des processus pour assurer une utilisation optimale des outils sélectionnés

Le Test Manager doit aussi prendre en compte les coûts inhérents à tout outil. Le temps passé à acquérir, administrer, former et utiliser l'outil pourrait avoir été consacré à de réelles tâches de test; par conséquent des ressources supplémentaires en test peuvent être nécessaires jusqu'à ce que l'outil soit opérationnel.

Il y a de nombreux risques à utiliser un outil ; tous les outils ne fournissent pas des avantages supérieurs à leurs inconvénients. Les risques liés aux outils ont été discutés dans le syllabus niveau fondation. Le Test Manager devrait aussi prendre en compte les risques suivant dans son étude de ROI:

- Immaturité de l'organisation (elle n'est pas prête à utiliser l'outil)
- Les artefacts créés par l'outil peuvent être difficiles à maintenir, nécessitant des révisions multiples lors des modifications du logiciel testé
- La réduction de l'implication des Analystes de Test dans les tâches de test peut diminuer la valeur du test (par ex., l'efficacité de détection de défauts peut être réduite si seulement des scripts automatisés sont exécutés)

Enfin, un Test Manager doit regarder les avantages qui découlent de l'utilisation de l'outil. Les avantages de l'introduction et de l'usage d'un outil, peuvent inclure:

- Réduction du travail répétitif
- Réduction de la durée des cycles de test (par ex., en utilisant des tests de non-régression automatisés)
- Réduction des coûts d'exécution des tests
- Augmentation de certains types de test (par ex., tests de non-régression)
- Réduction des erreurs humaines dans les différentes phases de test, par exemple:
  - Les données de test peuvent être meilleures avec des outils de génération de données
  - La comparaison de résultats de test peut être plus précise avec des outils de comparaison
  - Les données de test en entrée peuvent être meilleures si elles sont entrées avec un outil de scripting
- Réduction de l'effort nécessaire pour accéder à l'information sur les tests, par exemple:
  - Rapports et métriques générés par les outils
  - Réutilisation d'éléments de test comme les cas de tests, scripts de test et données de test
- Augmentation du test impossible sans outils (par ex., tests de performance, tests de charge)
- Amélioration du statut des testeurs qui automatisent des tests et de l'organisation dans son ensemble par la démonstration d'une compréhension et d'une utilisation d'outils sophistiqués.

En général, il est rare que le groupe de test n'utilise qu'en seul outil, le ROI global qu'obtiendra l'équipe de test est fréquemment fonction de tous les outils utilisés. Les outils ont besoin de partager de l'information et de travailler ensemble de façon coopérative. Une stratégie complète, à long terme, est conseillée.

## 6.2.4 Processus de Sélection

Les outils de test constituent des investissements sur le long terme qui couvriront probablement plusieurs itérations d'un même projet et/ou applicables sur plusieurs projets. Un Test Manager doit considérer un outil potentiel selon divers points de vue.

- Pour le métier, un ROI positif est nécessaire. Afin d'obtenir de la valeur ajoutée de leurs investissements, l'organisation devrait s'assurer que les outils – ce qui peut inclure à la fois des outils de test et des outils non dédiés aux tests – puissent coopérer. Dans certains cas, les processus et la connectivité pour l'utilisation des outils doivent être améliorés pour permettre cette interopérabilité, et ceci peut demander un certain temps.
- Pour le projet, l'outil doit être efficace (par ex., pour éviter des erreurs durant le test manuel, telles que des fautes de frappe lors de la saisie de données en entrée). L'outil peut demander un temps appréciable avant d'apporter un ROI positif. Dans bien des cas, le ROI se produira lors de la deuxième livraison ou lors de la maintenance, plutôt que lors du projet initial pendant la mise en place de l'automatisation. Le Test Manager devrait prendre en compte le cycle de vie complet de l'application.
- Pour les utilisateurs de l'outil, celui-ci doit les aider à réaliser leurs tâches de façon plus efficace et rentable. La courbe d'apprentissage doit être prise en compte pour s'assurer que les utilisateurs sauront apprendre à utiliser l'outil rapidement et avec le moins de stress possible. Lors de leur première introduction, les outils de test demandent de la formation et de l'accompagnement pour les utilisateurs.

Afin de s'assurer que tous les points de vue soient pris en compte, il est important de créer une roadmap pour l'introduction des outils de test.

Le processus de sélection pour un outil de test a été abordé dans le syllabus niveau Fondation comme suit:

- Évaluer la maturité de l'organisation
- Identifier les exigences pour les outils
- Évaluer les outils
- Évaluer le fournisseur ou le support (outils open-source, outils sur-mesure)
- Identifier les exigences internes pour la formation et l'accompagnement à l'usage des outils
- Évaluer les besoins en formation en prenant en compte les compétences actuelles de l'équipe de test en automatisation
- Estimer les coûts-bénéfices (comme discuté Section 6.2.3 ROI)

Pour chaque type d'outil, quelle que soit la phase de test dans laquelle il est utilisé, un Test Manager devrait prendre en compte les capacités suivantes:

- Analyse
  - L'outil pourra-t-il "comprendre" les éléments qui lui seront donnés en entrée?
  - L'outil est-il adapté au besoin?
- Conception
  - L'outil aidera-t-il à concevoir les articles de test sur la base d'information existantes (par ex., outils de conception de tests qui créent des cas de test à partir des exigences)?
  - La conception peut-elle être générée automatiquement?
  - Le code de l'article de test peut-il être généré, ou partiellement généré, dans un format maintenable et utilisable?
  - Les données de test nécessaires peuvent-elles être générées automatiquement (par ex., données générées à partir d'une analyse de code)?
- Sélection des données et des tests
  - Comment l'outil sélectionne-t-il les données dont il a besoin? (par ex., quels cas de test exécuter avec quelles données)?

- L'outil peut-il accepter des critères de sélection entrés soit manuellement soit automatiquement?
- L'outil peut-il déterminer comment "nettoyer" des données de production sur la base d'entrées précises?
- L'outil peut-il déterminer quels tests sont nécessaires sur base de critères de couverture (par ex., pour un ensemble donné d'exigences, l'outil peut-il utiliser la traçabilité pour déterminer quels cas de test sont nécessaires à l'exécution)?
- Exécution
  - L'outil pourra-t-il s'exécuter automatiquement ou bien une intervention manuelle sera-t-elle nécessaire?
  - Comment l'outil s'arrête-t-il et redémarre-t-il?
  - L'outil devrait-il être capable d' "écouter" certains événements (par ex., un outil de Gestion des Tests devrait-il automatiquement mettre à jour le statut des cas de test si un défaut lié au cas de test est clôturé?)
- Évaluation
  - Comment l'outil déterminera-t-il s'il a reçu un résultat correct (par ex., l'outil va-t-il utiliser un oracle de test pour déterminer la bonne réponse?)?
  - Quel type de fonctions de récupération d'erreurs l'outil possède-t-il?
  - L'outil fournit-il les logs et le reporting adaptés??

### 6.3 Cycle de vie d'un Outil

Le cycle de vie d'un outil est constitué des quatre étapes principales suivantes que le Test Manager doit gérer:

1. Acquisition. L'outil doit être acquis comme discuté ci-dessus (voir Section 6.2 Sélection d'un Outil). Une fois prise la décision d'introduire l'outil, le Test Manager devrait affecter quelqu'un (souvent un Analyste de Test ou un Analyste Technique de Test) pour en être l'administrateur. Cette personne devrait prendre des décisions relatives à l'usage de l'outil comme, comment et quand l'utiliser, où enregistrer les artefacts créés, quelles conventions de nommage adopter, etc. Prendre ces décisions à l'avance, plutôt que de les laisser arriver de façon ad hoc peut faire une différence importante au niveau du ROI de l'outil. Des formations seront sans doute nécessaires pour les utilisateurs de l'outil.
2. Support et maintenance. Des processus continus de support et de maintenance de l'outil seront nécessaires. La responsabilité de la maintenance de l'outil peut être confiée à l'administrateur de l'outil, ou peut être confiée à un groupe d'utilisateurs particuliers de l'outil. Si l'outil doit fonctionner avec d'autres outils, alors les échanges de données et les processus de coopération devront être considérés. Des décisions pour les sauvegardes et restauration de l'outil et de ses livrables devront être prises.
3. Évolution. Des changements doivent être envisagés. A mesure que le temps passe, l'environnement, les besoins métier, ou des problèmes d'éditeur peuvent nécessiter des changements majeurs dans l'outil ou dans son utilisation. Par exemple, l'éditeur de l'outil peut exiger une mise à jour de l'outil qui cause des problèmes avec les outils interfacés. Un changement d'environnement nécessaire pour des raisons métier peut causer des problèmes à l'outil. Au plus l'environnement d'exécution d'un outil est complexe, au plus les changements dus à des évolutions pourront perturber son utilisation. A ce sujet, selon le rôle joué par l'outil dans le test, un Test Manager doit s'assurer que l'organisation dispose d'un moyen de s'assurer de la continuité du service.
4. Retrait. Un moment viendra où l'outil aura dépassé sa durée d'utilisation utile. L'outil devra alors être retiré de façon adaptée. La fonctionnalité fournie par l'outil devra être remplacée et les données devront être préservées et archivées. Ceci peut arriver parce que l'outil arrive en fin de vie, ou simplement parce qu'il a atteint un point où les bénéfices et opportunités d'une migration vers un nouvel outil dépassent les coûts et risques.



Tout au long du cycle de vie de l'outil, il est de la responsabilité du Test Manager de s'assurer du maintien d'un fonctionnement continu et régulier des services offerts par l'outil à l'équipe de test.

## 6.4 Métriques sur les Outils

Un Test Manager peut concevoir et collecter des métriques objectives à partir des outils utilisés par des Analystes Techniques de Test et des Analystes de Test. Différents outils peuvent capturer de précieuses données en temps réel et peuvent réduire l'effort de collecte des données. Ces données peuvent être utilisées par le Test Manager pour gérer l'effort de test global.

Différents outils se focalisent sur la collecte de différents types de données, par exemple:

- Les outils de Gestion de Tests peuvent fournir une variété de métriques. La traçabilité depuis les exigences vers les cas de test et les scripts automatisés permet d'obtenir des métriques de couverture. A tout instant, il est possible d'obtenir un aperçu des tests disponibles ou planifiés, et le statut actuel d'exécution des tests (exécuté avec succès, en échec, bloqué, en attente).
- Les outils de gestion d'anomalies peuvent apporter une mine d'information sur les défauts, y compris leur statut actuel, leur sévérité et priorité, la distribution des défauts au sein du système, etc. D'autres données utiles, telles que la phase dans laquelle le défaut a été introduit, la phase où le défaut a été trouvé, le taux de défauts manqués, etc. aident le Test Manager à diriger l'amélioration de processus.
- Les outils d'analyse statique peuvent aider à détecter et à rapporter des problèmes de maintenabilité.
- Les outils de performance peuvent fournir de l'information utile sur l'évolutivité du système.
- Les outils de couverture peuvent aider le Test Manager à comprendre dans quelle mesure le système a réellement été sollicité lors des tests.

Les exigences de reporting des outils devraient être définies lors du processus de sélection de l'outil. Ces exigences doivent aussi être implémentées correctement pendant la configuration de l'outil pour assurer que l'information suivie par les outils soit rapportée de façon compréhensible par les parties prenantes.



## **7. Compétences – Composition de l'Équipe – 210 mins.**

### **Mots clé**

Indépendance du test

### **Objectifs d'Apprentissage pour Compétences – Composition de l'Équipe**

#### **7.2 Compétences Individuelles**

- TM-7.2.1 (K4) Utilisant un tableau d'évaluation des compétences, analysez les forces et faiblesses des membres de l'équipe par rapport à l'utilisation des systèmes logiciels, à la connaissance du domaine et du métier, aux différents domaines du développement logiciel, aux compétences relationnelles et en test de logiciels
- TM-7.2.2 (K4) Analysez une évaluation des compétences donnée pour une équipe afin de définir un plan de formation et de développement de compétences

#### **7.3 Dynamique de l'Équipe de Test**

- TM-7.3.1 (K2) Pour une situation donnée, discutez des compétences fermes et relationnelles nécessaires pour diriger une équipe de test

#### **7.4 Introduire le Test dans une Organisation**

- TM-7.4.1 (K2) Expliquez différentes options d'indépendance des tests

#### **7.5 Motivation**

- TM-7.5.1 (K2) Fournissez des exemples de facteurs motivants et démotivants pour les testeurs

#### **7.6 Communication**

- TM-7.6.1 (K2) Expliquez les facteurs influençant l'efficacité de la communication au sein d'une équipe de test et envers ses parties prenantes

## 7.1 Introduction

Un Test Manager brillant recrute, emploie et maintient des équipes avec un mélange adéquat de compétences. Les exigences relatives aux compétences peuvent changer au cours du temps, donc, en plus de recruter initialement les bonnes personnes, fournir des formations adéquates et des opportunités d'évolution est important pour retenir les membres de l'équipe et la maintenir à un haut niveau de performance. En plus des compétences de l'équipe, le Test Manager doit aussi maintenir un ensemble de compétences qui permettra un fonctionnement efficace dans un environnement évoluant rapidement et sous haute pression.

Ce chapitre se focalise sur la façon d'évaluer les compétences, de combler les lacunes pour créer une équipe pleine de synergie qui sera soudée et efficace dans l'organisation, et aussi à la façon de motiver cette équipe et de communiquer efficacement.

## 7.2 Compétences Individuelles

La capacité d'un individu à tester des logiciels peut s'acquérir par l'expérience ou par l'enseignement et la formation. Chacun des éléments suivants contribue à alimenter les connaissances des testeurs:

- Utilisation de systèmes logiciels
- Connaissance du domaine ou du métier
- Participation à différentes phases des activités du processus de développement logiciel, y compris l'analyse, le développement et le support technique
- Participation à des activités de test logiciel

Les utilisateurs finaux de systèmes logiciels ont une bonne compréhension du fonctionnement du système, des parties où les défaillances auront le plus d'impact, et de la réaction que le système devrait avoir à différentes situations. Des utilisateurs ayant une expertise du domaine, connaissent les parties les plus importantes pour le métier et comment ces parties affecteront la capacité du métier à atteindre ses exigences. Cette connaissance peut être utilisée pour aider à prioriser les activités de test, créer des données de test et de cas de test réalistes, et pour créer ou vérifier les cas d'utilisation.

La connaissance du processus de développement logiciel (analyse des exigences, architecture, conception et codage) aide à comprendre comment des erreurs amènent à l'introduction de défauts, où les défauts peuvent être détectés et comment prévenir l'introduction de défauts. L'expérience dans le support technique fournit une connaissance des expériences utilisateurs, de leurs attentes et ses exigences d'utilisabilité. Une expérience en développement logiciel est importante pour l'utilisation des outils de test demandant une expertise en programmation et conception, et pour participer à l'analyse statique de code, aux revues de code, au test unitaire et aux tests d'intégration techniques.

Les compétences spécifiques en test de logiciels incluent celles abordées dans les syllabi Fondation, Analyste de Test Avancé, et Analyste Technique de Test Avancé, telles que la capacité à analyser une spécification, à participer à une analyse de risques, à concevoir des cas de tests, et la capacité à exécuter des tests et à enregistrer des résultats de test.

Spécifiquement pour les Test Managers, disposer de connaissances, de compétences et d'expérience en gestion de projet est particulièrement important, puisque la Gestion des Tests inclut beaucoup d'activités de gestion de projet, par ex., faire un plan, suivre l'avancement et fournir un reporting aux parties prenantes. En cas d'absence d'un chef de projet, le Test Manager peut prendre les deux rôles de Test Manager et chef de projet, en particulier lors des dernières étapes du projet. Ces compétences s'ajoutent aux capacités discutées dans le syllabus niveau Fondation et dans ce syllabus.



En plus des compétences techniques, des compétences relationnelles, comme apporter et recevoir des critiques constructives, influencer et négocier, sont toutes importantes pour le rôle du test. Des testeurs compétents techniquement sont susceptibles d'échouer s'ils ne possèdent et n'utilisent les compétences relationnelles nécessaires. En plus de réussir à travailler efficacement avec d'autres, un professionnel du test efficace doit aussi être bien organisé, attentif au détail et posséder d'importantes compétences en communication orale et écrite.

L'équipe de test idéale aura un mélange de différents niveaux de compétence et expérience, et les membres de l'équipe devraient avoir la volonté et la capacité d'enseigner à, et d'apprendre de, leurs pairs. Dans certains environnements, certaines compétences seront plus importantes ou plus respectées que d'autres. Par exemple, dans un environnement de test technique, où des compétences en test d'API et en programmation sont requises, des compétences techniques pourront être mieux valorisées que des compétences métier. Dans un environnement de test boîte noire, l'expertise métier pourra être plus valorisée. Il est important de se souvenir que les environnements et les projets changent.

Lors de la création d'un tableau d'évaluation des compétences, le Test Manager devrait lister toutes les compétences importantes pour le travail et adaptées à la position. Une fois ces compétences détaillées, chaque personne de l'équipe peut être évaluée via un système de notation (par ex., une note de 1 à 5 avec 5 étant le plus haut niveau de compétence attendu pour ce domaine). Les individus peuvent être évalués pour déterminer leurs forces et leurs faiblesses et, sur la base de cette information, des plans de formation individuels ou collectifs peuvent être créés. Le Test Manager peut fixer un objectif de performance aux individus visant à améliorer leurs compétences dans un domaine particulier et devrait définir des critères spécifiques qui seront utilisés pour mesurer les compétences d'une personne.

Les personnes doivent être recrutées pour du long terme et pas seulement pour leur contribution à un unique projet. Lorsqu'un Test Manager investit dans des testeurs et crée un environnement propice à l'apprentissage continu, les membres de l'équipe seront motivés pour améliorer leurs compétences et leurs connaissances de façon à être mieux préparés pour l'opportunité suivante.

Un Test Manager sera rarement capable de recruter les membres d'équipes parfaits. Et même si les membres de l'équipe sont parfaits pour le projet actuel, ils peuvent ne pas l'être pour le projet suivant. Le Test Manager devrait recruter des personnes intelligentes, curieuses, capables, désireuses de travailler, pouvant travailler efficacement en équipe, et ayant le souhait et la capacité d'apprendre. Même si cet ensemble parfait d'individus est probablement indisponible, une équipe forte peut être construite en équilibrant les forces et faiblesses de ses membres.

En utilisant un tableau d'évaluation des compétences le Test Manager peut identifier où l'équipe est forte et où elle est faible. Cette information constitue la base d'un programme de formation et de développement des compétences. En commençant par les faiblesses ayant l'impact le plus important sur l'efficacité et la rentabilité de l'équipe, le Test Manager devrait décider comment aborder les différents domaines. Une approche repose sur la formation, par ex., envoyer les personnes à une formation, organiser des sessions de formation en interne, développer des formations sur mesure ou utiliser des cours en ligne. Une autre approche est le travail personnel, par ex., des livres, des webinars ou des ressources sur Internet. Une approche encore différente est la formation croisée, par ex., affecter à une tâche nécessitant une compétence particulière, une personne devant apprendre cette compétence avec une personne la possédant déjà, faire donner des présentations par des experts locaux sur un domaine d'expertise, etc. (L'accompagnement est une approche similaire où une personne ayant un nouveau rôle est assistée par une personne senior ayant eu ce rôle, et la personne senior agit comme une ressource en fournissant régulièrement du conseil et de l'assistance). En plus de s'occuper des faiblesses, le Test Manager devrait se souvenir d'utiliser les forces identifiées lors de l'évaluation des compétences dans le cadre du programme de formation et

de développement des compétences. Pour davantage d'information sur les plans de développement de l'équipe de test, voir [McKay07].

## 7.3 Dynamique de l'Équipe de Test

Pour construire la meilleure équipe possible, la sélection des membres est une des fonctions les plus importantes pour un rôle de management dans l'organisation. Beaucoup d'éléments doivent être pris en compte en plus des compétences individuelles spécifiques requises pour le travail. Lors de la sélection d'une personne pour intégrer l'équipe, la dynamique de l'équipe doit être considérée. Cette personne viendra-t-elle compléter les compétences et les types de personnalités déjà présents dans l'équipe? Il est important de considérer les avantages à avoir différents types de personnalités dans l'équipe, ainsi qu'un mélange de compétences techniques. Une équipe forte est capable d'aborder de multiples projets, de complexité variable, tout en gérant efficacement les relations interpersonnelles avec les membres d'autres équipes projet.

Le test est souvent une activité sous haute pression. Les plannings de développement logiciel sont souvent compressés et même irréalistes. Les parties prenantes ont des attentes élevées envers l'équipe de test, parfois déraisonnables. Le Test Manager doit recruter des personnes qui géreront bien les situations de forte pression, qui détourneront la frustration et qui peuvent se concentrer sur le travail même quand le planning semble impossible à tenir. C'est au Test Manager de gérer les problèmes relatifs aux plannings et aux attentes, mais le Test Manager doit aussi comprendre que ces pressions sont ressenties aussi par les membres de l'équipe. Quand le Test Manager recrute des personnes pour l'équipe, il est important de considérer l'environnement de travail et d'accorder les types de personnalités avec cet environnement. Dans un environnement où il y a plus de travail que de temps, le Test Manager devrait identifier des personnes finissant une tâche et se demandant quoi faire ensuite.

Les individus travailleront plus dur et seront plus attentifs s'ils se sentent valorisés et utiles. Chaque individu doit comprendre qu'il est un membre important de l'équipe et que sa contribution est vitale pour le succès de l'équipe dans son ensemble. Une fois cette dynamique créée, la formation transverse se fera de façon informelle, la répartition de la charge de travail sera faite par les membres de l'équipe eux-mêmes et le Test Manager aura plus de temps pour gérer les problèmes extérieurs.

Les nouveaux arrivants dans l'équipe doivent être rapidement intégrés dans l'équipe et obtenir l'encadrement et le soutien nécessaire. Chacun devrait avoir un rôle bien défini dans l'équipe. Cela peut se baser sur un processus d'évaluation individuelle. Le but est d'assurer le succès individuel, tout en contribuant au succès global de l'équipe. Ceci se fait principalement en accordant les types de personnalités aux rôles dans l'équipe et en s'appuyant sur les compétences innées des personnes ainsi qu'en augmentant leur portefeuille de compétences.

Lors de l'identification des personnes à recruter ou à ajouter à l'équipe, une évaluation objective des compétences peut être utile. Cela peut se faire par des entretiens ou des tests du candidat, la revue d'exemples de ses réalisations et en vérifiant ses références. Les compétences qui devraient être évaluées incluent :

Compétences techniques pouvant être démontrées par:

- La création de cas de tests à partir d'un document d'exigences
- La revue de documentation technique (exigences, code, etc.)
- La formulation de commentaires de revue clairs, compréhensibles et objectifs
- L'application adéquate de diverses techniques de test selon un scénario donné (par ex., utiliser les tables de décision pour tester un ensemble de règles métier)
- L'évaluation d'une défaillance et sa documentation de façon précise

- La démonstration d'une bonne compréhension sur la classification d'informations d'anomalies
- La démonstration d'une bonne compréhension des causes racines des défauts
- L'utilisation d'un outil pour tester une API donnée, avec des tests positifs et négatifs
- L'utilisation de SQL pour trouver et modifier de l'information de bases de données pour tester un scénario donné
- La conception d'un harnais de test automatisé qui exécutera plusieurs tests et collectera les résultats
- L'exécution de tests automatisés et la correction des défaillances
- L'écriture des spécifications/plans de test
- L'écriture d'un rapport de synthèse de tests incluant une évaluation des résultats de test

Compétences relationnelles pouvant être démontrées par:

- Présentation d'informations relatives à un projet de test en retard
- Explication d'un rapport d'anomalie à un développeur qui pense qu'il n'y a pas de défaut
- Formation d'un collègue
- Présentation d'un problème aux responsables hiérarchiques concernant un processus qui n'est pas rentable
- Révision d'un cas de test créé par un collègue et présentation à cette personne des commentaires
- Interview d'un collaborateur
- Complimenter un développeur

Même si cette liste est incomplète et même si les compétences nécessaires varient entre les environnements et les organisations, elle correspond à des compétences fréquemment requises. En créant une liste de questions d'interview pertinentes et en permettant la démonstration de compétences, le Test Manager peut évaluer les compétences du candidat et déterminer ses forces et faiblesses. Une fois qu'un bon système d'évaluation est en place, il devrait aussi être appliqué à l'ensemble du personnel pour déterminer les axes de progression et de formation.

En plus des compétences nécessaires aux contributeurs individuels, le Test Manager doit aussi posséder de très bonnes compétences en communication et en diplomatie. Le Test Manager doit être capable de dénouer des situations controversées, doit connaître les bons média à utiliser pour la communication nécessaire et doit être capable de se concentrer sur l'établissement et le maintien des relations dans l'organisation.

## 7.4 Introduire le Test dans une Organisation

Les organisations ont différentes manières d'introduire le test dans la structure de l'organisation. Même si la qualité est de la responsabilité de tous tout au long du cycle de développement logiciel, une équipe de test indépendante peut contribuer de façon significative à la qualité du produit. L'indépendance de la fonction de test varie grandement en pratique, comme le montre la liste suivante, triée par degré d'indépendance croissant:

- Pas de testeurs indépendants
  - Dans ce cas il n'y a aucune indépendance et le code est testé par le développeur qui l'a écrit
  - Le développeur, en fonction du temps alloué au test, déterminera si le code fonctionne comme prévu, ce qui peut ou non couvrir les exigences réelles
  - Le développeur peut corriger rapidement tout défaut trouvé
- Le test est fait par un développeur autre que celui qui a écrit le code
  - Il y a peu d'indépendance entre le développeur et le testeur



- Un développeur testant le code d'un autre développeur peut être réticent à rapporter les défauts
- La mentalité du développeur vis à vis des tests est en général focalisée sur les cas de test positifs
- Le test est fait par un testeur (ou une équipe de test) faisant partie de l'équipe de développement
  - Le testeur (ou l'équipe de test) peut rapporter au chef de projet ou au responsable des développements
  - La mentalité du testeur est focalisée sur la vérification du respect des exigences
  - Comme le testeur appartient à l'équipe de développement, il peut avoir des responsabilités de développement en plus du test
  - Le responsable du testeur peut être davantage préoccupé par la tenue des délais que par l'atteinte des objectifs qualité
  - Le test peut être dévalorisé par rapport au développement dans l'équipe
  - Le testeur n'a pas forcément d'autorité pour influencer les objectifs qualité et le respect de ces objectifs
- Le test est fait par des spécialistes en test appartenant à l'organisation métier, à la communauté des utilisateurs ou à d'autres organisations techniques ne faisant pas de développement
  - L'information concernant les résultats de test est communiquée de façon objective aux parties prenantes
  - La qualité est la préoccupation principale de cette équipe
  - Le développement des compétences et la formation sont axés sur les tests
  - Le test est vu comme un parcours professionnel
  - Une hiérarchie spécifique est consacrée aux groupes de test avec une focalisation sur la qualité
- Des spécialistes de test externes réalisent différents types de test
  - L'expertise est appliquée à des domaines spécifiques sur lesquels du test classique ne sera a priori pas suffisant
  - Les tests peuvent être de type utilisabilité, sécurité, performance ou tout autre domaine nécessitant une spécialisation
  - La qualité devrait être la préoccupation principale de ces personnes mais leur vision est limitée à leur domaine d'expertise. Un produit peut bien fonctionner mais ne pas satisfaire à ses exigences fonctionnelles mais cela peut ne pas être décelé par des spécialistes des performances.
- Le test est réalisé par une organisation externe à la société
  - Ce modèle assure le maximum d'indépendance entre testeurs et développeurs
  - Le transfert de connaissance peut ne pas être suffisant pour permettre au testeur d'être compétent
  - Des exigences claires et une structure de communication clairement définie seront nécessaires
  - La qualité de l'organisation externe doit être auditée régulièrement.

Cette liste est basée sur des préoccupations typiques des individus, mais peut ne pas être applicable dans toutes les organisations. Un poste et un titre ne déterminent pas nécessairement le travail réel de la personne. Les responsables de développement peuvent être orientés qualité, comme peut l'être un bon Test Manager. Les Test Managers indépendants peuvent rapporter à une hiérarchie focalisée sur les délais et ainsi devenir plus focalisés sur les délais que sur la qualité. Pour déterminer le meilleur emplacement pour l'équipe de test au sein d'une organisation, le Test Manager doit comprendre les objectifs de l'organisation.

Il y a divers degrés d'indépendance entre les organisations de développement et de test. Il est important de comprendre qu'il peut exister un compromis où plus d'indépendance résulte en plus

d'isolation et, moins de transfert de connaissances. Un niveau d'indépendance plus faible peut accroître la connaissance mais aussi introduire des conflits d'objectifs. Le niveau d'indépendance sera aussi déterminé par le modèle de développement logiciel utilisé, par ex., avec un développement Agile les testeurs sont plus souvent intégrés à l'équipe de développement.

Chacune des options ci-dessus peut être présente dans une organisation. Du test peut être effectué au sein de l'organisation de développement comme par une organisation de test indépendante, et il peut y avoir une certification finale donnée par un organisme externe. Il est important de comprendre les responsabilités et attentes de chaque phase de test et de définir ces exigences de façon à maximiser la qualité du produit final tout en respectant les contraintes de délais et de budget.

## 7.5 Motivation

Il y a plusieurs façons de motiver un individu travaillant comme testeur. Celles-ci incluent:

- La reconnaissance du travail accompli
- L'approbation par la hiérarchie
- Le respect au sein du projet et parmi ses pairs
- Des responsabilités et une autonomie croissantes
- Des récompenses adaptées au travail accompli (incluant le salaire, des responsabilités, la reconnaissance)

Il peut y avoir des influences du projet qui rendent difficilement applicables ces facteurs de motivation. Par exemple, un testeur peut travailler très dur sur un projet ayant une deadline impossible à atteindre. Le testeur aura beau faire tout son possible pour pousser la qualité au sein de l'équipe, faire des heures supplémentaires, et malgré tout, le produit pourra être livré avant qu'il ne devrait l'être suite à des influences externes. Le résultat sera un produit de mauvaise qualité malgré les meilleurs efforts du testeur. Cela peut être démotivant si la contribution du testeur n'est pas comprise et mesurée, indépendamment du succès final du produit.

L'équipe de test doit s'assurer de suivre les bonnes métriques pour prouver qu'un bon travail a été fait pour accomplir les tests, mitiger les risques et enregistrer les résultats avec précision. Si ces données ne sont pas collectées et publiées, l'équipe pourra facilement devenir démotivée quand elle ne reçoit pas la reconnaissance attendue pour un travail bien fait.

La reconnaissance ne doit pas se limiter à des aspects intangibles comme du respect et de l'approbation mais doit aussi se concrétiser par des opportunités de promotions, une augmentation de responsabilités et des augmentations au mérite. Si l'équipe de test n'est pas respectée, ces opportunités pourront ne pas être disponibles.

La reconnaissance et le respect sont acquis lorsqu'il est clair que le testeur contribue à augmenter la valeur du projet. Dans un projet indépendant, cela est atteint en impliquant le testeur au début du projet et en l'impliquant tout au long du cycle de vie. Au fil du temps, les testeurs gagneront le respect des autres parties prenantes par leurs contributions positives au projet. Cette contribution devrait aussi être quantifiée en termes de diminution du coût de la qualité et de réduction des risques.

Le Test Manager joue un rôle important dans la motivation des individus d'une équipe de test mais aussi comme représentant de l'équipe de test au sein d'une organisation plus large. Le Test Manager devrait reconnaître les réussites de chaque testeur, et doit aussi évaluer avec justesse et honnêteté les erreurs. Un Test Manager juste et consistant motivera les membres de son équipe par l'exemple.

## 7.6 Communication

La communication de l'équipe de test prend principalement les formes suivantes:

- La documentation des livrables du test - stratégie de test, plan de test, cas de tests, rapports de synthèse de tests, rapports de défauts, etc.
- Les retours fournis sur les documents revus - exigences, spécifications fonctionnelles, cas d'utilisation, documentation du test des composants, etc.
- La collecte et la diffusion d'information – interactions avec les développeurs, les autres membres de l'équipe de test, la hiérarchie, etc.

La communication entre les testeurs et les autres parties prenantes doit être professionnelle, objective et efficace afin d'établir et de maintenir le respect pour l'équipe de test. Diplomatie et objectivité sont nécessaires pour émettre des critiques, en particulier des critiques constructives, sur les livrables des autres. De plus, la communication devrait se focaliser sur l'atteinte des objectifs de test et sur l'amélioration de la qualité à la fois des produits et des processus utilisés pour produire les systèmes logiciels.

Le Test Manager communique avec une large audience, incluant les utilisateurs, les membres de l'équipe du projet, la hiérarchie, des groupes de test externes et les clients. La communication doit être efficace pour l'audience cible. Par exemple, un rapport produit pour l'équipe de développement montrant des tendances et des évolutions du nombre et de la sévérité des défauts pourra être trop détaillé pour convenir à une réunion de direction. Avec toute communication, mais particulièrement pendant des présentations, il est important de comprendre le message qui est envoyé, comment un message peut être reçu et l'explication nécessaire pour créer un environnement où ce message pourra être accepté. Comme le Test Manager est souvent en position de présenter des informations relatives au statut du projet, il est important que ces informations aient le niveau de détail approprié (par ex., les managers veulent en général voir les tendances relatives aux défauts plutôt que les défauts eux-mêmes), et soient présentées de façon claire et facilement compréhensible (par ex., des diagrammes simples et des graphiques en couleurs). Communiquer efficacement aide à garder l'attention de l'audience tout en véhiculant le bon message. Chaque présentation devrait être considérée par le Test Manager comme une opportunité de promouvoir la qualité et des processus de qualité.

Un Test Manager ne communique pas seulement avec les personnes extérieures à son département (communication externe). Une part importante du travail d'un Test Manager est de communiquer efficacement à l'intérieur du groupe de test (communication interne) pour diffuser des nouvelles, des instructions, des changements de priorité et toute autre information standard communiquée dans un processus de test normal. Un Test Manager peut aussi communiquer vers des individus particuliers, soit vers le haut de l'organigramme de la société (communication montante) soit vers le bas (communication descendante). Indépendamment de la direction de la communication, les mêmes règles s'appliquent, la communication devrait être adaptée à l'audience, le message devrait être envoyé efficacement et sa compréhension devrait être confirmée.

Les Test Managers doivent parfaitement maîtriser les différents moyens de communication. Beaucoup d'informations sont communiquées par courriel, oralement, lors de réunions formelles ou informelles, via des rapports formels ou informels et même par l'intermédiaire d'outils de Gestion des Tests comme les outils de gestion d'anomalies. Toute communication doit être professionnelle et objective. La relecture, à la fois pour la qualité et le contenu est une étape nécessaire, même pour les communications les plus urgentes. La durée de vie d'une communication écrite dépasse souvent celle du projet en cours. Il est important que le Test Manager produise une documentation de qualité à l'image d'une organisation qui promeut la qualité.



## 8. Références

### 8.1 Standards

- [BS7925-2] BS 7925-2, Software Component Testing Standard.  
Chapitre 2
- [FDA21] FDA Title 21 CFR Part 820, Food and Drug Administration, USA  
Chapitre 2
- [IEEE829] IEEE Standard for Software and System Test Documentation  
Chapitres 2 et 4
- [IEEE1028] IEEE Standard for Software Reviews and Audits  
Chapitre 2
- [IEEE1044] IEEE Standard Classification for Software Anomalies  
Chapitre 4
- [ISO9126] ISO/IEC 9126-1:2001, Software Engineering - Software Product Quality,  
Chapitres 2 et 4
- [ISO12207] ISO 12207, Systems and Software Engineering, Software Life Cycle Processes  
Chapitre 2
- [ISO15504] ISO/IEC 15504, Information Technology - Process Assessment  
Chapitre 2
- [ISO25000] ISO/IEC 25000:2005, Software Engineering - Software Product Quality  
Requirements and Evaluation (SQuaRE)  
Chapitres 2 et 4
- [RTCA DO-178B/ED-12B]: Software Considerations in Airborne Systems and Equipment  
Certification, RTCA/EUROCAE ED12B.1992.  
Chapitre 2

### 8.2 Documents ISTQB

- [ISTQB\_AL\_OVIEW] ISTQB Advanced Level Overview, Version 1.0
- [ISTQB\_ALTM\_SYL] ISTQB Advanced Level Test Manager Syllabus, Version 1.0
- [ISTQB\_ALTTA\_SYL] ISTQB Advanced Level Technical Test Analyst Syllabus, Version 1.0
- [ISTQB\_ETM\_SYL] ISTQB Expert Test Management Syllabus, Version 1.0
- [ISTQB\_FL\_SYL] ISTQB Foundation Level Syllabus, Version 2011
- [ISTQB\_GLOSSARY] Standard glossary of terms used in Software Testing, Version 2.2,  
2012
- [ISTQB\_ITP\_SYL] ISTQB Expert Improving the Test Process Syllabus, Version 1.0



## 8.3 Marques

Les marques déposées et marques de service suivantes sont utilisées dans ce document:

CMMI®, IDEAL<sup>SM</sup>, ISTQB®, ITIL®, PRINCE2®, TMMi®, TPI Next®

CMMI est enregistrée au “U.S. Patent and Trademark Office” par Carnegie Mellon University.

IDEAL est une marque de service du “Software Engineering Institute” (SEI), Carnegie Mellon University

ISTQB® est une marque déposée de l’International Software Testing Qualifications Board

ITIL est une marque déposée, et une communauté de marque enregistrée du “Office of Government Commerce”, et est déposée au “U.S. Patent and Trademark Office”

PRINCE2 est une marque déposée du “Cabinet Office”

TMMi est une marque déposée de “TMMi Foundation”

TPI Next est une marque déposée de “Sogeti Nederland B.V”.

## 8.4 Ouvrages

[Black03]: Rex Black, “Critical Testing Processes,” Addison-Wesley, 2003, ISBN 0-201-74868-1

[Black09]: Rex Black, “Managing the Testing Process, third edition,” John Wiley & Sons: New York, 2009, ISBN 0-471-22398-0

[Black11]: Rex Black, Erik van Veenendaal, Dorothy Graham, “Foundations of Software Testing,” Thomson Learning, 2011, ISBN 978-1-84480-355-2

[Craig02]: Rick Craig, Stefan Jaskiel, “Systematic Software Testing,” Artech House, 2002, ISBN 1-580-53508-9

[Crispin09]: Lisa Crispin, Janet Gregory, “Agile Testing: A Practical Guide for Testers and Agile Teams,” Addison-Wesley, 2009, ISBN 0-321-53446-8

[de Vries09]: Gerrit de Vries, et al., “TPI Next”, UTN Publishers, 2009, ISBN 90-72194-97-7

[Goucher09]: Adam Goucher, Tim Riley (editors), “Beautiful Testing,” O’Reilly, 2009, ISBN 978-0596159818

[IDEAL96]: Bob McFeeley, “IDEAL: A User’s Guide for Software Process Improvement,” Software Engineering Institute (SEI), 1996, CMU/SEI-96-HB-001

[Jones11]: Capers Jones et Olivier Bonsignour, “Economics of Software Quality,” Pearson, 2011, ISBN 978-0132582209

[McKay07]: Judy McKay, “Managing the Test People,” Rocky Nook, 2007, ISBN 978-1933952123

[Musa04]: John Musa, “Software Reliability Engineering, second edition,” Author House, 2004, ISBN 978-1418493882



- 
- [Pol01]: Martin Pol, Ruud Teunissen, Erik van Veenendaal, "Software Testing: A Guide to the T-Map Approach," Addison-Wesley, 2001, ISBN 0-201-74571-2
- [Stamatis03]: D.H. Stamatis, "Failure Mode and Effect Analysis," ASQC Quality Press, 2003, ISBN 0-873-89300
- [vanVeenendaal11] Erik van Veenendaal, "Le Little TMMi," UTN Publishers, 2011, ISBN 9-490-986038
- [vanVeenendaal12] Erik van Veenendaal, "Practical Risk-based Testing," UTN Publishers, 2012, ISBN 978-9490986070
- [Whittaker09]: James Whittaker, "Exploratory Testing," Addison-Wesley, 2009, ISBN 978-0321636416
- [Wiegers03]: Karl Wiegers, "Software Requirements 2," Microsoft Press, 2003, ISBN 978-0735618794



## 8.5 Autres Références

Les références suivantes pointent vers de l'information disponible sur internet. Ces références ont été vérifiées lors de la publication du syllabus niveau avancé.

<http://www.istqb.org>

<http://www.sei.cmu.edu/cmmi/>

<http://www.tmmi.org/>

<http://www.tpinext.com/>

## 9. Index

- activités de clôture des tests, 15
- agile, 10, 14, 20, 31, 34, 36, 39, 41, 45, 48, 83
- améliorer le processus de test, 66
- améliorer le processus de test avec CTP, 68
- améliorer le processus de test avec STEP, 69
- améliorer le processus de test avec TMMi, 67
- améliorer le processus de test avec TPI Next, 68
- analyse de la densité des défauts, 61
- analyse de risque, 28
- analyse de risque, 17
- analyse de risque qualité, 25
- analyse des conditions de test, 33
- analyse des dangers, 31
- Analyse des Modes de Défaillances et Effets (AMDE), 31
- analyse des risques, 26
- analyse des tests, 11
- Analyse par Arbre de Défaillance, 31
- anomalie, 58, 59
- approche de test, 17, 37
- approche réactive, 34
- approches méthodiques, 33
- audit, 51
- audits, 53
- base de test, 11
- cas de test, 8, 13
- cause racine, 58
- causes racines, 63
- charte de test, 24
- clôture des tests, 8
- CMMI, 49, 66
- comité de tri des défauts, 58
- communication, 84
- communication descendante, 84
- communication externe, 84
- communication interne, 84
- communication montante, 84
- compétences, 77
- compétences individuelles, 78
- compétences relationnelles, 79
- compétences relationnelles, 81
- compétences techniques, 80
- composition de l'équipe, 77
- conception de tests, 8
- conception des tests, 13
- condition de test, 8, 11, 12
- conditions de test, 17
- contrôle des tests, 8, 11, 17, 46
- coût d'exposition, 31
- coût de prévention, 47
- coûts de detection, 47
- coûts externes des défaillances, 47
- coûts internes des défaillances, 47
- critère de sortie, 8
- critères d'entrée et de sortie, 23
- Critical Testing Processes (CTP), 64, 68
- CTP, 65, 66, 68
- cycle d'amélioration de Deming, 65
- cycle de vie
  - agile, 21
  - cascade, 21
  - itératif, 21
  - modèle en V, 21
  - modèle en W, 21
- cycle de vie d'un outil, 75
- cycle de vie du défaut, 60
- défaillance, 58, 59
- défaut, 58, 59
- Delphi à Large Bande, 17, 41
- démotivation, 83
- Déploiement Qualité de Fonction (DQF), 31
- directeur de test, 17, 19
- dynamique de l'équipe de test, 80
- erreur, 59
- estimation des tests, 17, 40
- évaluation des compétences, 79
- évaluation des critères de sortie et reporting, 15
- évaluation des risques, 17
- exécution des tests, 8, 14
- faux-négatif, 58
- faux-négatif, 59
- faux-positif, 58, 60
- gérer des revues formelles, 56
- gérer les revues, 54
- gestion des risques, 17
- gestion des risques, 17
- gestion des risques, 26
- gestion des risques, 28
- Gestion des Tests, 17, 19
- GNU General Public License, 71
- graphes cause à effet, 33
- IDEAL, 66
- identification des risques, 17, 26



IEEE, 49  
IEEE 1028, 49  
IEEE 829, 49  
impact, 26  
implémentation des tests, 8, 13  
indépendance du test, 77  
inspection, 51  
introduire le test dans une organisation, 81  
ISO, 49  
ITIL, 49  
Kanban  
    tableau de Kanban, 45  
Lean, 45  
livrable, 23  
maîtrise de la phase, 58, 59  
métriques, 44, 45  
métriques, 43  
métriques, 23  
métriques, 9  
métriques, 44  
métriques, 83  
métriques de test, 42  
métriques liées aux personnes, 42  
métriques pour les revues, 56  
métriques processus, 42  
métriques produit, 42  
métriques projet, 42  
métriques sur les outils, 76  
mission, 9  
mitigation des risques, 28  
modérateur, 51  
motivation, 83  
niveau de risque, 27  
niveau de risque, 17, 27  
niveau de test, 17, 38  
open-source, 71  
outil open-source, 70  
outil sur-mesure, 70  
outils de test, 23  
parcours en largeur, 29  
parcours en profondeur, 29  
partie prenante, 19  
parties prenantes métier, 31  
parties prenantes techniques, 31  
plan de revue, 51  
plan de test, 17, 20  
plan de test de niveau, 17, 38  
plan de test maître, 17, 28, 37  
planification des tests, 8, 9  
planification, suivi et contrôle des tests, 9  
planning risque, 25  
PMI, 49  
politique de test, 17, 35  
PRINCE2, 49  
priorité, 58  
priorité de risque, 27  
probabilité, 26  
procédure de test, 8  
processus d'amélioration des tests, 65  
processus fondamental de tes, 9  
profil d'utilisation d'utilisation, 33  
profiling opérationnel, 36  
projet gestion des risques, 39  
rapport, 23  
rapport de synthèse de tests, 8  
réduction des risques, 17  
registre de test, 8  
relecteur, 51  
relecture technique, 51  
responsable de tests, 17, 19  
retour sur investissement (ROI), 72  
revue, 51  
revue de gestion, 51, 53  
revue informelle, 51  
revue technique, 51  
revues, 52  
revues d'ambiguïté, 33  
risque, 17  
risque, 25  
risque produit, 9  
risque produit, 17  
risque produit, 25  
risque projet, 17, 25, 39  
risque qualité, 17, 25  
risque qualité du produit, 25  
risque technique, 27  
score de risque agrégé, 27  
script de test, 8  
service level agreements (SLAs), 10  
sessions de test, 24  
sévérité, 58  
spécialistes de test externes, 82  
standards  
    BS-7925-2, 49  
    CMMI, 64  
    DO-178B, 28  
    ED-12B, 28, 49  
    IEC 61508, 28  
    US FDA Title 21 CFR Part 820, 49  
STEP, 65, 66, 69  
stratégie de test, 9, 17, 35, 37  
stratégies  
    consultatives, 36  
stratégies  
    basées sur les modèles, 36  
    réactives, 36



---

stratégies	test d'interaction de fonctionnalité, 23
analytiques, 36	test de confirmation, 28
stratégies	test de non-régression, 28
méthodiques, 36	test distribué, 48
stratégies	test exploratoire, 24
basées sur un processus ou un standard,	test externalisé, 48
36	test internalisé, 48
stratégies	Test Manager, 19
anti-régression, 37	Testing Maturity Model integration (TMMi),
Structure de la répartition du travail, 41	67
Systematic Test and Evaluation Process	tests basés sur l'expérience, 24
(STEP), 64	tests non-fonctionnels, 23
techniques, 23	TMMi, 65, 66
test basé sur les exigences, 33	TPI Next, 64, 65, 66, 68
test basé sur les risques, 9, 10, 17, 25, 26,	traçabilité, 11
29, 30, 31, 32, 33, 34, 36, 63	types de processus d'amélioration, 66
test d'intégration de produit client, 23	valeurs qualitatives, 47
test d'intégration matériel-logiciel, 23	valeurs quantitatives, 47
test d'intégration système, 23	Work Breakdown Structures (WBS), 41