Syllabus Testeur Certifié de Niveau Avancé Management des tests

Version 3.0

International Software Testing Qualifications Board

Notice de copyright

Historique de révision

Version	Date	Remarques
ISEB v1.1	2001/09/04	Syllabus ISEB Practitioner.
ISTQB 1.2E	2003/09	Syllabus ISTQB du niveau avancé de EOQ-SG.
V2007	2007/10/12	Syllabus Testeur certifié de niveau Avancé version 2007.
D100626	2010/06/10	Incorporation des changements acceptés en 2009, séparation de chaque chapitre pour les modules séparés.
D101227	2010/12/10	Acceptation des changements de format et des corrections qui n'ont pas d'impact sur le sens des phrases.
D2011	2011/10/31	Modification du syllabus en deux parties, remaniement des objectifs d'apprentissage et modification du texte pour qu'il corresponde aux objectifs d'apprentissage. Ajout de BOs.
Alpha 2012	2012/02/09	Incorporation de tous les commentaires des Membres reçus depuis la release d'octobre.
Beta 2012	2012/03/26	Incorporation des commentaires des Membres reçus à temps à partir de la release Alpha.
Beta 2012	2012/04/07	Version bêta soumise à l'AG.
Beta 2012	2012/06/08	Version contrôlée livrée aux Membres.
Beta 2012	2012/06/27	Intégration des commentaires des groupes de travail Examens et Glossaire.
RC 2012	2012/08/15	Version candidate à la release - modifications finales des Membres incluses.
Beta v3.0	2023/10/31	Incorporation de tous les commentaires reçus des Membres pour toutes les sections (BOIncs) de la revue Alpha.
POST Beta v3.0	2024/01/31	Incorporation de tous les commentaires des Membres reçus pour toutes les sections (BOIncs) de la revue Beta.
POST Beta v3.0	2024/02/29	Modifications mineures suite à la relecture technique.
RC v3.0	2024/03/28	Version candidate à la release - derniers changements de templates formels proposés par le PWG.
Version 3.0	2024/07/01	Version française

Table des matières

Remerciements

Ce document a été officiellement livré à l'Assemblée générale de l'ISTQB® le 3 mai 2024.

Il a été réalisé par une équipe de l'International Software Testing Qualifications Board: Horst Pohlmann (Product Owner, Vice Chair AELWG), Tauhida Parveen, Francis Fenner, Laura Albert, Matthias Hamburg, Maud Schlich, Tanja Tremmel, Ralf Bongard, Erik van Veenendaal, Jan Giessen, Bernd Freimut, Andreas Neumeister, Georg Sehl, Rabi Arabi, Therese Kuhfuß, Ecaterina Irina Manole, Veronica Belcher, Kenji Onishi, Pushparajan Balasubramanian, Meile Postuma et Miroslav Renda.

L'équipe remercie Gary Mogyorodi pour sa revue technique (en version bêta), Julia Sabatine pour sa relecture, l'équipe de révision et les membres pour leurs suggestions et leurs contributions.

Les personnes suivantes ont participé à la revue, aux commentaires et au vote de ce syllabus :

Revues alpha: Benjamin Timmermans, Mattijs Kemmink, Rik Marselis, Jean-Francois Riverin, Gary Mogyorodi, Ralf Bongard, Ingvar Nordström, Yaron Tsubery, Imre Mészáros, Mattijs Kemmink, Ádám Bíró, Ramit M Kaul, Chinthaka Indikadahena, Darvay Tamás Béla, Beata Karpinska, Young jae Choi, Stuart Reid, Tal Pe'er, Meile Posthuma, Daniel van der Zwan, Klaudia Dussa-Zieger, Jörn Münzel, Ralf Bongard, Petr Neugebauer, Derk-Jan de Grood, Rik Kochuyt, Andreas Hetz, Laura Albert, Eszter Sebestyeni, Tamás Szőke, Henriett Braunné Bokor, Ágota Horváth, Péter Sótér, Ferenc Hamori, Darvay Tamás Béla, Paul Weymouth, Lloyd Roden, Kevin Chen, Huang qin, Pushparajan Balasubramanian, Szilard Szell, Tamas Stöckert, Lucjan Stapp, Adam Roman, Anna Miazek, Márton Siska, Erhardt Wunderlich, László Kvintovics, Murian Song, Mette Bruhn-Pedersen, Petra Schneider, Michael Stahl, Ramit M Kaul, Imre Mészáros, Dilhan Jayakody, Francisca Cano Ortiz, Johan Klintin, Liang Ren, Ole Chr. Hansen, Zsolt Hargitai, Tamás Rakamazi, Kenji Onishi, Arnika Hryzszko, Rabih Arabi, Veronica Belcher, et Vignesh Balasubramanian.

Revues bêta: Maria-Therese Teichmann, Dominik Weber, Thomas Puffler, Peter Kunit, Martin Klonk, Michaël Pilaeten, Wim Decoutere, Arda Ender Torçuk, Piet de Roo, Rik Marselis, Jakub Platek, Ding Guofu, Zheng Dandan, Liang Ren, Yifan Chen, Hallur Helmsdal, Ole Chr. Hansen, Klaus Skafte, Gitte Ottosen, Tanzeela Gulzar, Arne Becher, Klaudia Dussa-Zieger, Jan Giesen, Florian Fieber, Carsten Weise, Arnd Pehl, Matthias Hamburg, Stephanie Ulrich, Jürgen Beniermann, Márton Siska, Sterbinszky Ádám, Ágnes Srancsik, Marton Matyas, Tamas Stöckert, Csilla Varga, Zsolt Hargitai, Bíró Ádám, Horváth Ágota, Sebestyéni Eszter, Szilárd Széll, Péter Sótér, Giancarlo Tomasig, Nicola de Rosa, Kaiwalya Katyarmal, Pradeep Tiwari, Sreeja Padmakumari, Seunghee Choi, Stuart Reid, Dmitrij Nikolajev, Mantas Aniulis, Monika Stoecklein-Olsen, Adam Roman, Mahmoud Khalaili, Ingvar Nordström, Beata Karpinska, Armin Born, Ferdinand Gramsamer, Mergole Kuaté, Thomas Letzkus, Nishan Portoyan, Ainsley Rood, Lloyd Roden, Sarah Ireton.

Le syllabus Test Manager de niveau Avancé 2010-2012 a été réalisée par le sous-groupe Niveau Avancé Test Manager du groupe de travail Niveau Avancé de
l'International Software Testing Qualifications Board: Rex Black (Chair), Judy McKay (Vice Chair), Graham Bath, Debra Friedenberg, Bernard Homès, Paul Jorgensen, Kenji Onishi, Mike Smith, Geoff Thompson, Erik van Veenendaal, Tsuyoshi Yumoto.

L'équipe principale remercie l'équipe de réviseurs et les Membres pour leurs suggestions et leur contribution.

Au moment où le syllabus de niveau Avancé a été achevé, le groupe de travail pour le niveau Avancé était composé des membres suivants (par ordre alphabétique):

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

Les personnes suivantes ont participé à la revue, aux commentaires et au vote de ce syllabus :

Chris van Bael, Graham Bath, Kimmo Hakala, Rob Hendriks, 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.

0 Introduction

0.1 Objectif de ce syllabus

Ce syllabus constitue la base de la qualification internationale de test du logiciel au niveau Avancé pour le management des tests. L'ISTQB® fournit ce syllabus comme suit :

1. Aux Membres, pour qu'ils le traduisent dans leur langue locale et accréditent les organismes de formation. Les Membres peuvent adapter le syllabus à leurs besoins linguistiques particuliers et modifier les références pour les adapter à leurs publications locales.
2. Aux organismes de certification, afin d'élaborer des questions d'examen dans leur langue locale, adaptées aux objectifs d'apprentissage de ce syllabus.
3. Aux organismes de formation accrédités par l'ISTQB®, afin de produire des supports de cours et de déterminer les méthodes d'enseignement appropriées.
4. Aux candidats à la certification, pour se préparer à l'examen de certification (l'ISTQB® recommande de suivre une formation accréditée par l'ISTQB® avant de se présenter à un examen de niveau avancé de l'ISTQB®).
5. À la communauté internationale de l'ingénierie des logiciels et des systèmes, pour faire progresser la profession de testeur de logiciels et de systèmes, et comme base de tests pour des livres et des articles

0.2 Le Testeur Certifié de niveau Avancé en Management des Tests dans les Tests

Cette certification de niveau avancé s'adresse à toute personne impliquée dans le management des tests de logiciels. Cela inclut les personnes dans des rôles tels que les testeurs, les consultants en test, les Test Managers, les testeurs d'acceptation utilisateurs, les scrum masters, les gestionnaires de projet ou les Product Owners. Cette certification Management des tests de niveau Avancé est également appropriée pour toute personne souhaitant une compréhension avancée des tests de logiciels, comme les chefs de projet, les responsables qualité, les responsables du développement de logiciels, les analystes métier, les directeurs informatiques et les consultants en management. Les titulaires de la certification de niveau Avancé en management des tests pourront accéder aux certifications de test du logiciel de niveau Expert de l'ISTQB®. Le certificat ISTQB® Certified Tester Advanced Level - Test Manager ou Management des tests est valable à vie et n'a pas besoin d'être renouvelé. Le certificat est reconnu au niveau international et démontre la compétence professionnelle et la crédibilité des candidats en matière de management des tests.

0.3 Chemin de carrière pour les testeurs

Le programme de l'ISTQB® offre un soutien aux professionnels du test à tous les stades de leur carrière. Les personnes qui obtiennent la certification ISTQB® testeur certifié de niveau Avancé Management des tests peuvent également être intéressées par les autres certifications Core de niveau Avancé (i.e., Analyste de Test et Analyste Technique de Test) et par la suite par les certifications ISTQB® de niveau

Expert (i.e., Management des Tests ou Amélioration du processus de test). Toute personne cherchant à développer ses compétences en matière de test dans le cadre du développement logiciel Agile peut envisager les certifications Testeur Technique Agile ou conduite des tests en mode Agile à l'échelle (ATLaS). La filière Spécialiste offre une plongée en profondeur dans des domaines qui ont des approches de test et des activités de test spécifiques (par exemple, dans l'automatisation des tests, les tests d'IA ou les tests d'applications mobiles), ou un savoir-faire en matière de tests groupés pour certains domaines industriels (par exemple, l'automobile ou les jeux). Veuillez consulter le site www.istqb.org pour obtenir les dernières informations du programme de testeur certifié de l'ISTQB®.

0.4 Objectifs métier (BO)

Cette section énumère les onze objectifs métier attendus d'un candidat ayant obtenu la certification de niveau Avancé en management des tests.

Un testeur certifié en Management des tests de niveau Avancé peut …

TM_01	Gérer les tests dans le cadre de divers projets de développement de logiciels en appliquant les processus de gestion des tests établis pour l'équipe de projet ou l'organisation des tests.
TM_02	Identifier les parties prenantes des tests et les modèles de cycle de vie du développement logiciel qui sont pertinents dans un contexte donné.
TM_03	Organiser des sessions d'identification des risques et d'évaluation des risques dans le cadre d'un cycle de vie du développement logiciel et utiliser les résultats pour orienter les tests afin d'atteindre les objectifs du test.
TM_04	Définir une stratégie de test cohérente avec la stratégie de test de l'organisation et le contexte du projet.
TM_05	Suivre et contrôler en continu les tests pour atteindre les objectifs du projet.
TM_06	Évaluer et rapporter l'avancement des tests aux parties prenantes du projet.
TM_07	Identifier les compétences nécessaires et développer ces compétences au sein de son équipe.
TM_08	Préparer et présenter un cas métier pour tester dans différents contextes en soulignant les coûts et les bénéfices attendus.
TM_09	Diriger les activités d'amélioration du processus de test dans le cadre de projets ou de flux de produits de développement de logiciels et contribuer aux initiatives organisationnelles d'amélioration du processus de test.
TM_10	Planifier les activités de test, y compris l'infrastructure de test requise, et estimer les efforts nécessaires pour tester.
TM_11	Créer des rapports de défaut et un cycle de vie des défauts adapté à un cycle de vie du développement logiciel.

0.5 Objectifs d'apprentissage examinables et niveau cognitif de connaissance

Les objectifs d'apprentissage soutiennent les objectifs métier et sont utilisés pour créer les examens de Testeur certifié de niveau Avancé Management des tests.

En général, tout le contenu de ce syllabus est examinable au niveau K1, à l'exception de l'Introduction, des Références, de l'Epilogue et des Annexes. Cela signifie qu'il peut être demandé au candidat de reconnaître, de se souvenir ou de rappeler un mot-clé ou un concept mentionné dans l'un des trois chapitres de ce syllabus. Les objectifs d'apprentissage spécifiques et les niveaux correspondants sont indiqués au début de chaque chapitre et classés comme suit :

- K2 : Comprendre
- K3 : Appliquer
- K4 : Analyser

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

Tous les termes listés comme mots-clés sous les titres des chapitres doivent être mémorisés (K1), même s'ils ne sont pas explicitement mentionnés dans les objectifs d'apprentissage.

0.6 L'examen de certification de niveau Avancé en Management des tests

L'examen du certificat de niveau Avancé en Management des tests est basé sur ce syllabus. Les réponses aux questions de l'examen peuvent nécessiter l'utilisation d'exigences basées sur plus d'une section de ce syllabus. Toutes les sections du syllabus sont examinables, à l'exception de l'introduction, des annexes et des références. Les normes et les livres sont inclus en tant que références (chapitre 5), mais leur contenu, au-delà de ce qui est résumé dans le syllabus lui-même à partir de ces normes et livres, n'est pas examinable

Pour plus de détails, voir le document "Structures et règles d'examen 1.1 compatibles avec les niveaux Fondation et Avancé du syllabus et les modules Spécialiste".

- Les critères d'admission à l'examen du certificat de management des tests de niveau avancé sont les suivants :
  - Les candidats sont titulaires du certificat de niveau Fondation de l'ISTQB® avant de passer l'examen de certification de niveau avancé en management des tests. Cependant, il est fortement recommandé que le candidat ait au moins une expérience minimale dans le développement ou le test de logiciels, par exemple six mois d'expérience en tant que testeur ou développeur de logiciels.

0.7 Accréditation

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

Les directives d'accréditation pour ce syllabus sont les directives d'accréditation génériques publiées par le groupe de travail Processus et Conformité de l'ISTQB®.

0.8 Gestion des normes

Certaines normes sont référencées dans le syllabus Management des tests de niveau Avancé (par exemple, ISO, IEC et IEEE). Le but de ces références est de fournir un framework ou de fournir une source d'information supplémentaire si le lecteur le souhaite. Veuillez noter que le syllabus utilise ces normes comme références. Les normes ne sont pas destinées à être examinées. Pour plus d'informations sur les normes, reportez-vous au chapitre 5 "Références".

0.9 Niveau de détail

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

- Des objectifs pédagogiques généraux décrivant l'intention du syllabus de management des tests de niveau avancé.
- D'une liste de mots-clés dont les étudiants doivent être capables de se rappeler.
- Des objectifs d'apprentissage pour chaque domaine de connaissance, décrivant le résultat cognitif à atteindre.
- D'une description des concepts clés, y compris des références à des sources telles que la littérature acceptée ou les normes.

Le contenu du syllabus n'est pas une description de l'ensemble du domaine de connaissance des tests de logiciels ; il reflète le niveau de détail à couvrir dans les cours de formation de niveau Avancé en Management des tests. Il se concentre sur les concepts et les techniques de test qui peuvent être appliqués à tous les projets logiciels.

0.10 Organisation de ce syllabus

Il y a trois chapitres dont le contenu peut faire l'objet d'un examen. L'en-tête de chaque chapitre précise l'exigence minimale pour les formations accréditées de couvrir le contenu du chapitre ; le temps n'est pas indiqué au-dessous du niveau du chapitre. Pour les cours accrédités, le syllabus exige un minimum de 22,75 heures d'enseignement, réparties sur les trois chapitres comme suit :

- Chapitre 1 : Management des activités de test (750 minutes).
  - Le participant apprend à expliquer les activités de management des tests (c-à-d. planification des tests, suivi des tests, contrôle des tests, et clôture des tests).
  - Le participant apprend à définir une stratégie de test de projet incluant les objectifs du test et la sélection de l'approche de test appropriée en cohérence avec la stratégie de test de l'organisation et le contexte du projet.
  - Le participant apprend à gérer des projets dans différents contextes.
  - Le participant apprend à appliquer le test basé sur les risques afin de concentrer les tests

sur les risques identifiés.

- Le participant apprend à mener les activités d'amélioration du processus de test en conduisant une rétrospective du projet ou de l'itération.
- Le participant apprend à améliorer le soutien des outils pour les tests en prenant en compte les risques, les coûts et les bénéfices du soutien des outils.

- Chapitre 2 : Management du produit (390 minutes)
  - Le participant apprend à suivre et à contrôler les tests pour atteindre les objectifs des tests en utilisant des métriques de test et à évaluer et rapporter l'avancement des tests.
  - Le participant apprend à sélectionner les techniques d'estimation de test appropriées à différentes filières suivant différents modèles de développement de logiciel.
  - Le participant apprend à définir un cycle de vie des défauts dans le cadre de la gestion des défauts pour s'adapter aux modèles de développement logiciel séquentiel, Agile et hybride.
- Chapitre 3 : Management de l'équipe (225 minutes)
  - Le participant apprend à analyser le contexte d'un projet donné pour identifier les exigences de l'équipe de test.
  - Le participant apprend à gérer une équipe selon l'approche de l'équipe intégrée.
  - Le participant apprend à définir un business case pour les activités de test dans le projet.

Note : Pour chaque objectif d'apprentissage, il existe dans le syllabus actuel une sous-section correspondante avec son contenu (par exemple, pour l'objectif d'apprentissage 1.2.3, il existe une sous-section 1.2.3).

0.11 Quelles sont les hypothèses fondamentales de ce syllabus ?

Ce syllabus est destiné à tous ceux qui souhaitent atteindre un niveau de compétence avancé en management des tests, tels que les test managers, les analystes de tests, les ingénieurs de tests, les consultants en tests, les coordinateurs de tests, les leaders de tests et les chefs de projets. Le syllabus est aligné sur le syllabus de niveau Fondation V.4 de l'ISTQB®, qui fournit les connaissances et la compréhension de base des tests de logiciels.

Ce syllabus couvre deux rôles principaux dans les tests : le rôle de management des tests et le rôle de testeur. Le rôle de management des tests est également connu sous le nom de Test Manager dans le contexte du modèle de développement séquentiel, où le test manager est typiquement un rôle séparé du chef de projet ou du Product Owner. Le rôle de management des tests est responsable du processus de test global, de l'équipe de test et du management des tests. Cela inclut la définition de la stratégie de test, la planification des activités de test, le suivi et le contrôle de l'avancement des tests, le reporting des résultats des tests et la gestion des risques et des problèmes liés aux tests. Le management des tests permet également de s'assurer que les objectifs de test sont alignés sur les besoins du métier et des parties prenantes, et que les activités de test sont coordonnées avec les autres parties prenantes du projet.

Le rôle de test s'occupe également de l'évaluation des tests, de la gestion des défauts et de la clôture des tests. Le testeur utilise diverses techniques de test pour garantir la qualité et la fiabilité des artefacts

de test et du système sous test. Il utilise également des outils de test et d'automatisation pour soutenir le processus de test et améliorer l'efficience et l'efficacité des tests. Les activités et les tâches assignées à ces rôles peuvent varier en fonction du contexte, tel que le projet, le produit, les compétences et l'organisation. (voir le syllabus du niveau Fondation de l'ISTQB® V.4).

Le terme *membre de l'équipe de test* est utilisé dans ce syllabus pour désigner toute personne dans un rôle de management des tests ou de test qui effectue des tests, indépendamment du contexte organisationnel et des autres rôles. Les équipes de test sont composées d'individus ayant des aptitudes et des compétences différentes. Les membres de l'équipe peuvent également avoir différents niveaux d'expérience et de certification, tels que le niveau Fondation, le niveau Avancé ou le niveau Expert. Les membres de l'équipe de test peuvent également avoir différents rôles et responsabilités en fonction de l'approche de test et du modèle de processus de test utilisé, comme les tests Agile, les tests basés sur des modèles, le test basé sur les risques, etc.

Un point important concernant la perspective que le syllabus fournit est le fait qu'il se concentre sur le management des tests au niveau du projet et non sur le management des tests au niveau de l'organisation. Par conséquent, ce syllabus est conforme et contient des informations qui peuvent être utilisées pour les activités de management des tests au niveau du projet, mais moins pour le management des tests au niveau organisationnel.

Le développement logiciel hybride est utilisé dans ce syllabus pour faire référence à une approche de développement logiciel qui combine des éléments de différents modèles de cycle de vie logiciel, tels que le modèle en V, itératif, incrémental et Agile. Le développement de logiciels hybrides vise à tirer parti des forces et à atténuer les faiblesses de chaque modèle, en fonction du contexte et des besoins du projet. Par exemple, une approche de développement logiciel hybride peut utiliser un modèle en V pour les phases initiales de planification et d'analyse des exigences, suivi d'un modèle Agile pour les phases de conception, de développement et de test. Par ailleurs, une approche de développement logiciel hybride peut utiliser un modèle itératif pour la gestion globale du projet, tout en appliquant un modèle incrémental pour chaque itération, et un modèle Agile pour chaque incrément. Le développement de logiciels hybrides exige un degré élevé de flexibilité, de communication et de collaboration entre les parties prenantes, ainsi qu'une compréhension claire des objectifs, des risques et des contraintes de chaque phase et de chaque modèle.

Selon ce syllabus et le glossaire, une stratégie de test est une description de la manière dont les tests seront effectués afin d'atteindre les objectifs du test dans des circonstances données. Une stratégie de test définit le périmètre, l'approche et les ressources pour tester un système ou un produit. Elle est généralement documentée dans un plan de test ou dans le cadre d'autres documents, en fonction du contexte du test. Une stratégie de test est influencée par la stratégie de test organisationnelle, qui est une stratégie de test de haut niveau décrivant comment les tests sont effectués dans une organisation. Une stratégie de test peut également exister pour un niveau de test unique ou un type de test, qui sont des aspects spécifiques du test qui se concentrent sur des objectifs, des cibles et des critères différents. Le terme générique "stratégie de test" peut être utilisé dans n'importe quel contexte (projet, organisation, produit). Une approche de test est la manière dont les tâches de test sont implémentées, en particulier la sélection et la combinaison des niveaux de test, des types de test et des techniques de test pour les tests statiques et dynamiques, ainsi que d'autres pratiques de test telles que les tests scriptés, les tests manuels, les tests dos-à-dos, etc. L'approche de test choisie par le rôle de management des tests est une décision clé dans la formulation d'une stratégie de test appropriée pour un contexte donné.

1 Management des activités de test – 750 minutes

Mots-clés

test basé sur l'expérience, test fonctionnel, modèle de développement logiciel hybride, modèle de développement incrémental, modèle de développement itératif, test non fonctionnel, risque produit, risque qualité, rétrospective, analyse des risques, évaluation des risques, identification des risques, impact des risques, niveau des risques, probabilité des risques, gestion des risques, atténuation des risques, suivi des risques, test basé sur les risques, modèle de développement séquentiel, méthodologie des objectifs S.M.A.R.T., cycle de vie du développement logiciel, clôture des tests, contrôle des tests, niveau de tests, Test Maturity Model integration (TMMi), suivi des tests, objectif de test, plan de test, amélioration du processus de test, stratégie de test, type de test, TPI NEXT

Mots-clés spécifiques au domaine

Objectif-question-métrique (GQM), IDEAL, indicateur, mesure, métrique

Objectifs d'apprentissage pour le chapitre 1:

1.1 Le processus de test

TM-1.1.1 (K2) Résumer la planification des tests.

TM-1.1.2 (K2) Résumer le suivi des tests et le contrôle des tests.

TM-1.1.3 (K2) Résumer la clôture des tests.

1.2 Le contexte des tests

TM-1.2.1 (K2) Comparer les raisons pour lesquelles les différentes parties prenantes s'intéressent aux tests.

TM-1.2.2 (K2) Expliquer pourquoi la connaissance des parties prenantes est importante dans le management des tests.

TM-1.2.3 (K2) Expliquer les tests dans un modèle de développement logiciel hybride.

TM-1.2.4 (K2) Résumer les activités de management des tests pour les différents cycles de vie du développement logiciel.

TM-1.2.5 (K2) Comparer les activités de management des tests à différents niveaux de test.

TM-1.2.6 (K2) Comparer les activités de management des tests pour différents types de tests.

TM-1.2.7 (K4) Analyser un projet donné et déterminer les activités de management des tests qui mettent l'accent sur la planification des tests, le suivi des tests et le contrôle des tests.

1.3 Test basé sur les risques

TM-1.3.1 (K2) Expliquer les différentes mesures prises par le test basé sur les risques pour répondre aux risques.

TM-1.3.2 (K2) Donnez des exemples de différentes techniques qu'un test Manager peut utiliser pour identifier les risques liés à la qualité du produit.

TM-1.3.3 (K2) Résumez les facteurs qui déterminent les niveaux de risque liés à la qualité des produits.

TM-1.3.4 (K4) Sélectionner les activités de test appropriées pour atténuer les risques en fonction de leur niveau dans un contexte donné.

TM-1.3.5 (K2) Distinguer des exemples de techniques de tests basés sur les risques, lourdes ou légères

TM-1.3.6 (K2) Donnez des exemples de métriques de réussite et de difficultés associées au test basé sur les risques.

1.4 Stratégie de test d'un projet

TM-1.4.1 (K2) Expliquer les choix typiques d'une approche de test.

TM-1.4.2 (K4) Analyser la stratégie de test d'une organisation et le contexte du projet pour sélectionner l'approche de test appropriée.

TM-1.4.3 (K3) Utiliser la méthodologie S.M.A.R.T. (méthodologie des objectifs) pour définir des objectifs de test mesurables et des critères de sortie.

1.5 Amélioration du processus de test

TM-1.5.1 (K2) Expliquer comment utiliser le modèle IDEAL pour l'amélioration du processus de test sur un projet donné.

TM-1.5.2 (K2) Résumer l'approche de l'amélioration basée sur les modèles sur le processus de test et comprendre comment l'appliquer dans le cadre d'un projet.

TM-1.5.3 (K2) Résumer l'approche de l'amélioration du processus de test basée sur l'analyse et comprendre comment l'appliquer dans le cadre d'un projet.

TM-1.5.4 (K3) Implémenter une rétrospective du projet ou de l'itération pour évaluer les processus de test et découvrir les domaines de test à améliorer.

1.6 Outils de test

TM-1.6.1 (K2) Résumer les meilleures pratiques pour l'introduction d'un outil.

TM-1.6.2 (K2) Expliquer l'impact des différents aspects techniques et métier lors du choix d'un type d'outil.

TM-1.6.3 (K4) Analyser une situation donnée afin d'élaborer un plan de sélection des outils, en tenant compte des risques, des coûts et des avantages.

TM-1.6.4 (K2) Différencier les étapes du cycle de vie de l'outil.

TM-1.6.5 (K2) Donner des exemples de collecte et d'évaluation de métriques à l'aide d'outils.

1.1 Le processus de test

Introduction

Le syllabus ISTQB® Testeur Certifié de niveau Fondation V.4 décrit un processus de test qui comprend les activités suivantes : planification des tests, pilotage et contrôle des tests, analyse de test, conception des tests, implémentation des tests, exécution des tests et clôture des tests.

Le syllabus de niveau Fondation V.4 de l'ISTQB® précise que ces activités du processus de test sont souvent implémentées de manière itérative ou en parallèle, en fonction du modèle du cycle de vie du développement logiciel (SDLC) et du contexte du projet. L'adaptation de ces activités au contexte du produit et du projet est généralement une exigence.

Dans ce syllabus, l'accent est mis sur les activités clés de management des tests suivantes :

- **Planification des tests :** définition des objectifs du test, de l'approche du test, du périmètre du test, des ressources du test, du calendrier du test, des livrables du test et des participants au test (parties prenantes du test).
- **Pilotage et contrôle des tests :** suivi de l'avancement des tests, des résultats des tests et des déviations des tests ; prise de mesures correctives si nécessaire ; rapport des tests et leurs résultats aux parties prenantes concernées.
- **Clôture des tests :** finalisation et archivage des artefacts de test, évaluation du processus de test et du produit de test, identification des actions d'amélioration du processus de test et communication de la clôture des tests aux parties prenantes concernées.

La norme ISO/IEC/IEEE 29119-2 définit les processus de management des tests qui couvrent ces activités de management des tests. Ces processus de management des tests peuvent être appliqués à différents niveaux de test tels que le projet, le programme ou le portefeuille. Chaque niveau de test peut avoir son propre plan de test qui s'aligne sur le plan de test du niveau supérieur.

1.1.1 Activités de planification des tests

Cette section se concentre sur les activités de planification des tests pour différents périmètres tels que l'ensemble du projet, un niveau de test, un type de test ou une release/itération/sprint dans Agile. En fonction du périmètre, la planification des tests peut commencer et se terminer à différents moments du processus de développement. La planification des tests est une activité qui consiste à identifier les activités et les ressources nécessaires pour atteindre les objectifs du test identifiés dans la politique de test. La planification des tests doit être initiée le plus tôt possible dans le processus de développement, de préférence avant l'identification des exigences, et doit être mise à jour au fur et à mesure de l'avancement du projet. La planification des tests est souvent un processus itératif qui nécessite une nouvelle planification au cours du projet pour tenir compte des changements et du retour d'information.

Les tâches suivantes font partie de la planification des tests (semblables à celles constatées dans la norme ISO/IEC/IEEE 29119-2) :

- **Comprendre le contexte et organiser la planification des tests**
  La compréhension du contexte de l'organisation (par exemple, la politique de test et la stratégie de test de l'organisation), du périmètre du test et de l'élément du test (c'est-à-dire le produit d'activités testé) est essentielle à la planification des tests. (voir section 1.2., Le contexte du test). Cela implique également toutes les activités nécessaires pour organiser le développement du

plan de test et pour obtenir l'approbation de ces activités et du calendrier par les parties prenantes (par exemple, le Product Owner, le chef de projet ou le manager de l'équipe de développement).

- **Identifier et analyser les risques produit**
  L'analyse des risques implique l'identification et l'évaluation de l'impact potentiel et de la probabilité des risques produit dans le cadre de la planification des tests. Voir la section 1.3 de ce syllabus, Tests basés sur le risque, pour plus de détails sur les risques produits.
- **Identifier les approches de traitement du risque**
  Sur la base de l'analyse des risques, les approches appropriées de traitement des risques sont sélectionnées et documentées dans le plan de test. Il peut s'agir d'actions préventives, correctives ou d'atténuation des risques identifiés.
- **Définir l'approche de test et estimer et allouer les ressources de test**
  Sur la base de la stratégie de test de l'organisation, de normes réglementaires, de toutes les contraintes données par le projet, et des approches de traitement des risques, l'approche de test est définie pour le périmètre de test actuel (voir Section 1.4, La stratégie de test du projet). Lorsqu'une approche de test est définie, il est important d'estimer les ressources de test nécessaires, telles que le personnel de test, les outils de test, les environnements de test et les données de test, et d'allouer ces ressources aux activités de test.
- **Établir le plan de test**
  Le plan de test doit être accepté par toutes les parties prenantes et, par conséquent, les désaccords entre elles doivent être résolus.

1.1.2 Activités de pilotage et contrôle des tests

Pour que le Management des tests puisse fournir un contrôle des tests efficient, un calendrier des tests et un framework de pilotage doivent être établis pour permettre de suivre l'état et l'avancement des tests. Ce framework doit inclure les mesures et les objectifs détaillés qui sont requis pour relier l'état des produits d'activités et des ressources de test au plan et aux objectifs stratégiques.

Pour les projets de petite taille et moins complexes, il peut être relativement facile d'établir un lien entre les produits d'activités de test et le plan et les objectifs stratégiques, mais il faut généralement définir des objectifs plus détaillés pour y parvenir. Cela peut inclure la définition de mesures et d'objectifs qui sont nécessaires pour atteindre les objectifs de test et la couverture de la base de test.

Il est particulièrement important de décrire l'état d'avancement des produits d'activités testés d'une manière compréhensible et pertinente pour les parties prenantes du projet et du métier.

Le pilotage et le contrôle des tests sont des activités permanentes.

Le contrôle des tests compare les progrès réels par rapport au plan de test et implémente des actions correctives si nécessaire. Il guide les tests pour qu'ils répondent aux stratégies et aux objectifs de test (voir section 1.4, La stratégie de test du projet), et revoit les activités de planification des tests lorsque cela s'avère nécessaire. Les données de contrôle nécessitent des informations détaillées sur la planification des tests pour pouvoir réagir de manière appropriée. Cette activité implique:

- L'implémentation du plan de test et des directives de contrôle.
- Le management des écarts par rapport aux tests planifiés.
- Le traitement des risques nouvellement identifiés et modifiés.

- La définition de conditions pour commencer les tests.
- L'accord et l'obtention l'approbation de la clôture des tests sur la base des critères de sortie.

Le pilotage des tests implique la collecte et l'enregistrement des résultats des tests, l'identification des écarts par rapport aux tests planifiés, l'identification et l'analyse des nouveaux risques qui nécessitent des tests, et le suivi des changements pour les risques identifiés.

1.1.3 Activités de clôture des tests

La clôture des tests intervient généralement lors des jalons du projet (par exemple, une release, la fin d'une itération ou la clôture d'un niveau de test). Pour tout défaut non résolu, des demandes de changement ou des éléments du product backlog sont créés. Voir le syllabus de niveau Fondation de l'ISTQB® V.4. Une fois que les critères de sortie sont remplis, les principaux résultats doivent être capturés, archivés et fournis aux parties prenantes concernées. La clôture des tests requiert les tâches suivantes :

- **Créer et approuver le rapport de clôture des tests.**
  Cette tâche permet de s'assurer que tous les tests ont été effectués et que tous les objectifs du test ont été atteints. Cette tâche implique la collecte d'informations pertinentes à partir de divers testware tels que les plans de test, les résultats de test, les rapports d'avancement des tests, les rapports de clôture de test et les rapports de défaut. Les informations collectées sont évaluées et résumées dans le rapport de clôture des tests. Le rapport de clôture des tests est approuvé et communiqué aux parties prenantes concernées.
- **Archiver le testware**
  Cette tâche permet d'identifier les testware qui peuvent être utiles à l'avenir ou qui devraient être réutilisés ; généralement des cas de test. Elle les rend accessibles et faciles à comprendre en vue d'une réutilisation future. En outre, les résultats des tests, les logs de test, les rapports de test et les autres testware doivent être temporairement archivés dans le système de gestion de configuration.
- **Transférer testware**
  Cette tâche permet de fournir des produits d'activités utiles à ceux qui en ont besoin. Par exemple, les défauts connus reportés ou acceptés doivent être communiqués à ceux qui utiliseront ou soutiendront l'utilisation du testware.
- **Effectuer toutes les tâches nécessaires pour nettoyer l'environnement de test et le remettre dans un état prédéfini**
  Cette tâche permet de s'assurer que l'environnement de test est prêt pour le cycle de test ou le projet suivant. Elle consiste à retirer de l'environnement de test les données de test, les outils de test, les pilotes de test, les bouchons de test et les scripts de test. Elle consiste également à remettre l'environnement de test dans son état d'origine ou dans l'état souhaité.
- **Réaliser/collecter/documenter les leçons apprises**
  Cette tâche est effectuée dans le cadre de rétrospectives au cours desquelles les leçons apprises tirées du processus de test sont discutées et documentées. Cela peut inclure des constatations pour l'ensemble du cycle de vie du développement logiciel (SDLC). Les leçons apprises peuvent être utilisées pour l'amélioration du processus de test, comme décrit dans la section 1.5. de ce syllabus, Amélioration du processus de test.

1.2 Le contexte du test

Introduction

Le contexte du test englobe les conditions et les contraintes uniques qui influencent le processus de test, façonnant les décisions et les stratégies de planification, de conception et d'exécution des tests. Il est vital pour les Test Managers de comprendre ce contexte afin d'aligner les tests sur les besoins et les objectifs spécifiques du projet de développement logiciel. Ce contexte peut varier en fonction du type de produit, de l'industrie, des exigences réglementaires et, surtout, du cycle de développement logiciel (SDLC) utilisé.

Les Test Managers ont pour mission d'appliquer les stratégies de test établies et de choisir les techniques de test plutôt que de les développer. Ils jouent un rôle clé dans la formulation de plans de test adaptés au contexte du projet. En comprenant et en prenant en compte ces différents facteurs, les Test Managers peuvent s'assurer que les tests sont pertinents, efficaces et efficients pour atteindre les objectifs du test.

1.2.1 Les parties prenantes du test

Les parties prenantes du test sont des individus ou des groupes ayant un intérêt direct ou indirect dans la qualité du produit. Voici une liste type de parties prenantes potentielles, modifiée pour refléter la diversité de leurs intérêts dans les tests:

- **Les développeurs, les responsables du développement et les managers du développement :** Outre l'implémentation du système sous test et l'action sur les résultats des tests, ces parties prenantes sont également impliquées dans les tests unitaires et contribuent au processus de test.
- **Les testeurs, les test leads et les Test Managers :** Ces personnes préparent les testware, ce qui implique de développer des plans de test et de contribuer au processus de test par des activités telles que l'analyse des exigences, la conception des tests, l'exécution des tests, le suivi et le rapport des défauts, l'automatisation des tests et le reporting des tests.
- **Les chefs de projet, les Product Owners et les utilisateurs métier :** Ils spécifient les exigences, définissent le niveau de qualité requis et recommandent la couverture nécessaire basée sur les risques perçus. Ils révisent également les produits de travail, participent aux tests d'acceptation utilisateurs (UAT) et prennent des décisions sur la base des résultats des tests.
- **L'équipe d'exploitation :** Engagée dans les tests d'acceptation opérationnelle, elle veille à ce que le système soit prêt pour la production et contribue à la définition des exigences non fonctionnelles.
- **Clients et utilisateurs :** Les clients achètent le produit, tandis que les utilisateurs s'en servent directement. Tous deux jouent un rôle clé dans la définition des exigences et devraient participer aux tests d'acceptation utilisateurs (UAT) afin de valider que le produit répond à leurs besoins.

Cette liste ne comprend pas toutes les parties prenantes potentielles. Les Test Managers doivent effectuer une analyse des parties prenantes dans le cadre de la création de la stratégie de test et du plan de test, en tenant compte du périmètre du test pour identifier les parties prenantes spécifiques à leur projet.

1.2.2 Importance de la connaissance des parties prenantes dans le management des tests

Dans le cadre du management des tests, il est crucial de prendre en compte les perspectives et l'influence des différentes parties prenantes. La matrice des parties prenantes, souvent appelée matrice pouvoir-intérêt, aide les Test Managers à hiérarchiser l'engagement des parties prenantes et à gérer les attentes de manière efficiente. La matrice des parties prenantes est un outil de gestion des tests stratégique qui permet :

- D'utiliser l'expertise des parties prenantes, les utilisateurs finaux et les équipes techniques fournissant un retour d'information et des perspectives sur la performance et la sécurité.
- De soutenir la gestion du risque en mettant en évidence les intérêts et l'influence des parties prenantes, encourageant ainsi les efforts d'atténuation proactifs.
- De valoriser la diversité des points de vue en fournissant un retour d'information précieux.

La matrice des parties prenantes est composée de quatre quadrants :

- **Promoteurs (influence élevée, intérêt élevé) :** collaborateurs clés ayant une grande influence et un grand intérêt, essentiels pour élaborer la stratégie de test et le plan.
- **Latents (influence élevée, faible intérêt) :** Bien qu'elles ne s'intéressent pas forcément aux tâches quotidiennes, leurs décisions sont essentielles pour l'affectation des ressources et l'orientation du projet à haut niveau.
- **Défenseurs (faible influence, intérêt élevé) :** Elles fournissent souvent un retour d'information qualitatif et peuvent être maintenues engagées grâce à des mises à jour régulières et à leur participation à des discussions spécifiques.
- **Apathiques (faible influence, faible intérêt) :** Bien qu'elles ne soient pas étroitement impliquées, le fait de les tenir au courant des principaux jalons et de leur demander leur avis sur des questions particulières peut permettre d'obtenir des informations uniques.

Le rôle du Test Manager consiste notamment à dresser une liste détaillée des parties prenantes et à comprendre la connectivité de chacune d'entre elles avec les activités de test, en utilisant la matrice des parties prenantes pour améliorer l'efficacité des pratiques de management des tests.

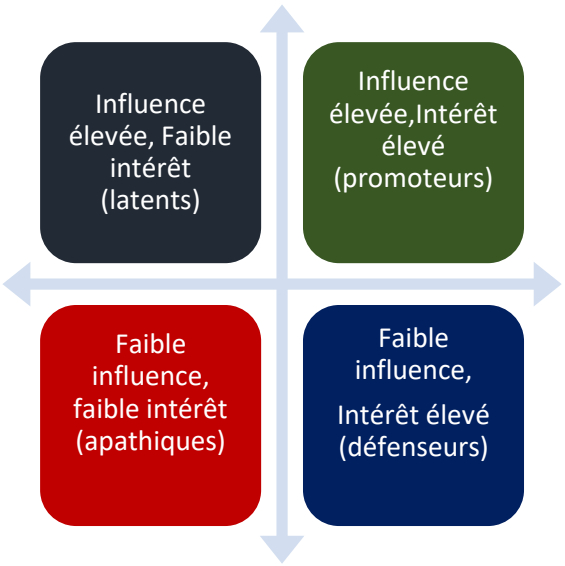

Figure 1. Différents types de parties prenantes

1.2.3 Le Management des tests dans un modèle de développement logiciel hybride

Les modèles de développement logiciel hybride intègrent des éléments provenant à la fois des approches séquentielles traditionnelles et des pratiques Agile afin de répondre aux besoins spécifiques d'un projet ou aux transitions organisationnelles. Les raisons suivantes sont couramment évoquées pour utiliser un modèle de développement logiciel hybride, bien qu'en fonction de l'organisation et du projet, il puisse y en avoir d'autres également :

- **L'hybride comme transition vers l'Agile :** la transition des méthodes traditionnelles vers l'Agile peut s'avérer difficile en raison des changements fondamentaux dans les processus, la culture et la dynamique de l'équipe. Les modèles hybrides offrent une approche équilibrée qui facilite cette transition en combinant la structure des méthodes traditionnelles avec la flexibilité des pratiques Agile.
- **L'hybride comme adapté à l'objectif :** certaines organisations ou certains projets peuvent ne pas être en mesure de passer à l'Agile. Les projets à haut risque peuvent nécessiter des tâches séquentielles pour certaines choses et des pratiques Agile pour d'autres. Ils peuvent utiliser un modèle hybride en fonction de leur objectif.

Dans un cadre hybride, les activités de management des tests peuvent comprendre les éléments suivants :

- Évaluer la compréhension et la facilité de l'équipe à effectuer une transition transparente entre les méthodologies traditionnelles et Agile.
- Identifier les forces et les faiblesses dans l'adaptation à une approche hybride.
- S'assurer que l'équipe sait combiner des processus structurés avec la flexibilité Agile.
- Améliorer la collaboration entre l'équipe de test et les parties prenantes afin de mieux gérer les tests au sein des sprints et des phases de test traditionnelles.

- Participer à des efforts coordonnés, tels que le scrum-of-scrums pour les testeurs, afin de maintenir l'attention sur les tests tout en contribuant aux objectifs globaux de développement.
- Suivre et revoir les efforts et les cas de test au sein des sprints pour s'assurer qu'ils s'alignent sur les pratiques Agile.

D'autres informations peuvent être trouvées dans (Fowler, 2010).

1.2.4 Activités de management des tests pour différents modèles de cycle de vie du développement logiciel

Pour aligner correctement les tests dans le modèle SDLC, un test manager doit comprendre les différents modèles SDLC utilisés dans son organisation et utiliser cette connaissance pour aligner correctement les tests avec les activités de développement. Le tableau ci-dessous compare les différentes activités de management des tests dans des modèles de SDLC différents :

Aspect	**Modèle de développement séquentiel** **p.ex. Modèle en V**	**Modèle de développement itératif** **p.ex. SCRUM**
Estimation	Estimation détaillée précoce pour chaque niveau de test.	Estimation itérative, partie de la planification des Stories (story planning) par itération.
Testware	Comprend la stratégie, le plan, les cas, le calendrier et les rapports.	Se concentre sur les critères d'acceptation et la Definition of Done ; documentation minimale.
Rôles	Le Test Manager supervise les décisions et la gestion de l'équipe.	Les rôles sont intégrés ; le facilitateur ou le coach remplace le traditionnel Test Manager.
Outils	Principalement des outils de management des tests adaptés aux tests par phase.	Les outils pour CI/CD et l'automatisation sont centraux, soutenant les tests en continu.
Approche de test	Planifiée à l'avance, correspondant aux phases du projet.	Intégrée dans les itérations, avec un accent sur l'adaptabilité et le retour d'information.
Automatisation des tests	Implémentée de manière stratégique, peut intervenir à différents stades.	Intégrée dès le début, en mettant l'accent sur la régression automatisée dans CI/CD.
Suivi et reporting	Rapports basés sur des jalons, avec en option des tableaux de bord automatisés.	Reporting continu avec des tableaux de bord en temps réel et des mises à jour quotidiennes de l'état d'avancement.
Métriques	Se concentre sur les métriques de test traditionnelles et la gestion des défauts. (par exemple, exécution du test, taux de défaut).	Inclut des métriques Agile pour le suivi des itérations en plus des métriques traditionnelles. (par exemple, la vélocité de l'équipe, les burndown charts).

Tableau 1 : Activités de management des tests pour différents modèles de SDLC

1.2.5 Activités de management des tests à différents niveaux de test

Tests de composants:

- Définir le périmètre, les objectifs et les critères de clôture des tests de composants (tests unitaires).
- Impliquer les testeurs dans des activités allant au-delà des rôles traditionnels des testeurs, telles que les revues de code, où leurs compétences analytiques apportent une valeur ajoutée.
- Coordonner avec l'équipe de développeurs la résolution des problèmes et la contribution aux tests unitaires.

Tests d'intégration des composants:

- Déterminer les séquences d'intégration et les combinaisons de tests en collaboration avec l'équipe de développeurs, en tenant compte du modèle, des outils et des processus du SDLC.
- Superviser l'avancement des travaux pour s'assurer qu'ils sont conformes aux stratégies de test du système et d'acceptation.
- Gérer cette phase en collaboration avec les développeurs, en tenant aussi compte des tests d'intégration des composants (unitaires).

Tests d'intégration des systèmes:

- Veiller à ce que le périmètre et les objectifs des tests d'intégration des systèmes soient clairs et adaptés à l'évaluation des risques et aux objectifs de qualité.
- Maintenir la supervision des progrès, des résultats et de la gestion des problèmes pendant les tests d'intégration des systèmes.

Tests système:

- Adapter la planification au modèle SDLC, en allouant soigneusement les ressources, en sélectionnant les outils et en établissant un calendrier.
    - Pour les projets Agile, intégrer les tests systèmes avec les tests itératifs des User Stories, en évitant les phases de test distinctes et en veillant à ce que les tests soient continus et intégrés. Alors que dans les modèles séquentiels, les tests peuvent suivre des étapes planifiées.

Tests d'acceptation:

- Collaborer avec les parties prenantes pour revoir et confirmer le respect des critères d'acceptation et planifier les activités de test, notamment en gérant les tests des utilisateurs dans le cadre de l'UAT.
- Coordonner la logistique des tests d'acceptation, en facilitant les tests sur le site du client, afin de s'assurer que le produit répond aux besoins du métier et aux normes de qualité en dehors de l'environnement de développement.
- Faciliter la résolution de tout problème lié à l'UAT et guider les parties prenantes tout au long du processus d'approbation du produit pour qu'il réponde aux critères d'acceptation.

1.2.6 Activités de management des tests pour différents types de tests

Un management des tests efficace nécessite une approche intégrée qui prend en compte les exigences uniques des tests fonctionnels, non fonctionnels, boîte noire et boîte blanche. Pour les managers chargés des tests fonctionnels, il s'agit de s'assurer que toutes les fonctionnalités sont testées de manière approfondie et qu'elles répondent aux exigences définies. La gestion des tests non fonctionnels consiste à vérifier les attributs du système tels que la performance et la sécurité. Le management des tests boîte noire consiste à s'assurer que les tests sont centrés sur l'utilisateur et que toutes les interactions externes possibles sont couvertes. Le management des tests boîte blanche met l'accent sur la compréhension de la structure du code et veille à ce que les tests couvrent intégralement la logique interne.

Management des tests fonctionnels :

- Planification stratégique et suivi des progrès : élaboration d'une stratégie de test détaillée qui s'aligne sur les exigences fonctionnelles et les objectifs du projet, de même qu'un suivi des progrès.
- Coordination des ressources : Allocation efficiente des ressources humaines et techniques pour couvrir tous les aspects fonctionnels du système.

Management des tests non-fonctionnels :

- Benchmarking des performances : Établir des benchmarks de performance et gérer les activités de test qui évaluent le système par rapport à ces critères.
- Vérification de la conformité : Superviser les tests qui garantissent que le système répond aux normes non fonctionnelles telles que la sécurité, l'utilisabilité et la fiabilité.

Management des tests boîte noire :

- Analyse de la couverture des tests : S'assurer que les tests boîte noire couvrent tous les scénarios d'utilisation et toutes les exigences métier.
- Incorporation du retour d'information : Manager le processus de collecte des retours d'expérience des parties prenantes afin de remanier les approches de test boîte noire et la correction des défauts.

Management des tests boîte blanche :

- Optimisation de la couverture du code : Superviser l'utilisation d'outils de couverture de code pour identifier les lacunes dans les tests boîte blanche et orienter les ressources pour y remédier.
- Intégration des connaissances techniques : Gérer l'incorporation des connaissances techniques dans le processus de planification des tests, en veillant à ce que les tests soient conçus avec une compréhension du fonctionnement interne de l'application.

1.2.7 Activités de management des tests pour la planification, le pilotage et le contrôle

Un management des tests efficace est la pierre angulaire de tout effort de test réussi. Il englobe un large éventail d'activités qui nécessitent une planification minutieuse, un pilotage vigilant et un contrôle stratégique. Les Test Managers jouent un rôle essentiel en veillant à ce que le processus de test soit non seulement efficace et efficient, mais aussi adapté aux exigences spécifiques du projet en cours.

Planification des tests:

- **Définition complète du périmètre :** Un plan de test doit être méticuleusement élaboré, en intégrant une définition complète du périmètre. Il s'agit notamment d'identifier toutes les exigences fonctionnelles et non fonctionnelles afin d'assurer une couverture complète des tests. Il s'agit également de prendre en compte les implications des méthodologies de test boîte noire et boîte blanche, en veillant à ce que les cas de test développés soient capables de valider le système sous test sous tous les angles.
- **Évaluation des risques et plan d'atténuation des risques :** Un framework de gestion des risques robuste fait partie intégrante du plan de test. Les Test Managers doivent entreprendre une analyse d'impact détaillée, en identifiant les vulnérabilités et les défis potentiels qui pourraient avoir un impact à la fois sur le flux de travail du projet et sur le produit final. Le développement de stratégies d'atténuation est crucial, impliquant une planification préventive pour contourner ou minimiser ces risques de manière efficace.
- **Stratégie d'allocation des ressources :** La planification des ressources est un autre élément essentiel. Elle va au-delà de la simple allocation et consiste à définir la structure de l'équipe, à délimiter les rôles et à établir des protocoles de communication. Dans les environnements où les équipes sont réparties, comme c'est le cas avec les modèles sur site/hors site, cet aspect devient particulièrement important pour maintenir la synchronisation et assurer une collaboration sans faille.

Pilotage des tests:

- **Suivi de l'exécution :** Le pilotage joue un rôle central dans le processus de management des tests. Il implique une revue continue de l'exécution des tests par rapport au plan établi, le suivi de l'avancement des cas de test et la gestion des défauts qui surviennent. L'ajustement des priorités basé sur l'évaluation des risques et les développements en temps réel garantit que les tests restent concentrés et alignés sur les domaines les plus critiques.
- **Optimisation des outils et de l'environnement :** La sélection et l'utilisation judicieuses des outils et des environnements de test sont cruciales pour soutenir la stratégie de test. Le suivi continu permet de s'assurer qu'ils sont efficacement intégrés dans le pipeline CI/CD, ce qui facilite les tests continus et les boucles de rétroaction immédiates qui sont vitales pour le processus de développement Agile.
- **Collaboration en matière de développement :** La maintenabilité d'une relation de travail étroite avec l'équipe de développeurs est essentielle pour obtenir de bons résultats en matière de tests. Cette collaboration doit soutenir une approche globale des tests, en tirant parti des perspectives boîte blanche et boîte noire pour traiter de manière préventive les problèmes potentiels.

Contrôle des tests:

- **Gestion adaptable des processus :** Le contrôle des tests consiste à ajuster dynamiquement le processus de test en réponse aux nouvelles connaissances, aux défis et à l'évolution de la dynamique du projet. Cela exige du test Manager qu'il soit réactif et flexible, capable d'implémenter des changements dans l'approche des tests qui reflètent l'état actuel du projet.
- **Gestion des points de contrôle qualité :** Une approche structurée de la gestion des points de contrôle qualité est fondamentale. Il s'agit notamment de définir ce qui constitue un point de contrôle de la qualité dans le cycle de vie des tests et de prendre des décisions éclairées sur la progression de la phase de test, ce qui est essentiel pour maintenir l'intégrité du produit.

En se concentrant sur ces activités spécifiques dans le cadre de la planification des tests, du pilotage des tests et du contrôle des tests, les Test Managers peuvent s'assurer que le processus de test est bien défini, qu'il est adaptable aux changements du projet et qu'il aboutit à un produit qui répond à la fois aux exigences du projet et aux attentes des parties prenantes.

1.3 Test basé sur les risques

Introduction

Le test basé sur les risques implique l'identification, l'évaluation, le suivi et l'atténuation des risques pour piloter les tests. Ces risques sont identifiés par diverses parties prenantes et peuvent être utilisés pour sélectionner et hiérarchiser les tests. Plus le niveau de risque est élevé, plus les tests doivent commencer tôt et plus l'effort de test doit être intense et prolongé.

1.3.1 Les tests en tant qu'activité d'atténuation des risques

Un risque produit est une situation potentielle où des problèmes de qualité peuvent exister dans un produit. Lorsque les tests révèlent des défauts, ils ont contribué à atténuer le risque produit en permettant de prendre conscience des défauts et de les traiter avant la release. Lorsque les tests ne constatent pas de défauts, ils indiquent que le niveau de risque produit est plus faible que prévu.

Entre autres choses, le Test Manager a la responsabilité de fournir une évaluation correcte et fiable de la qualité du produit. Cela nécessite une participation active à la gestion des risques projet, en mettant l'accent sur les risques liés à l'assurance qualité (par exemple, des exigences ambiguës qui entraînent des problèmes majeurs lors d'une validation tardive, des environnements de test insuffisants qui entravent l'exécution des tests).

Le test basé sur les risques base les tests sur les risques qualité. Il suit le processus générique de gestion du risque, qui comprend les principales activités suivantes :

- L'analyse des risques, qui comprend l'identification des risques et leur évaluation.
- Le contrôle des risques, qui consiste à suivre et atténuer des risques.

Ces activités principales sont organisées logiquement dans un ordre séquentiel, mais elles peuvent se chevaucher.

Pour concentrer les tests sur les risques qualité, les risques qualité doivent être identifiés et évalués. Pour être la plus efficace possible, l'analyse des risques doit inclure diverses parties prenantes. En tant que partie prenante principale de l'analyse des risques qualité, le Test Manager devrait comprendre et suivre ces activités et être capable de les modérer.

Le suivi des tests doit inclure le suivi des risques. Outre le suivi de l'évolution des risques qualité connus, il devrait inclure l'analyse de tout nouveau risque qualité et l'ajustement du registre des risques.

Le Test Manager est l'une des nombreuses personnes qui pilotent l'atténuation des risques qualité. L'atténuation des risques est répartie sur plusieurs activités de test. Par exemple, les résultats de l'analyse des risques qualité sont utilisés dans la planification des tests pour concentrer les tests sur les bons domaines en utilisant les bonnes techniques. Dans l'analyse des tests, les niveaux de risque guident la sélection des conditions de test à couvrir. Dans l'exécution des tests, la hiérarchisation basée sur les risques régit la séquence d'exécution des tests.

1.3.2 Identification des risques qualité

La tâche du Test Manager consiste à recueillir les risques auprès des parties prenantes. Les parties prenantes peuvent identifier les risques qualité par le biais d'une ou plusieurs des techniques suivantes :

- Entretiens avec des experts

- Evaluations indépendantes
- Rétrospectives
- Ateliers sur les risques
- Brainstorming
- Checklists
- Références à l'expérience passée

En impliquant l'échantillon le plus large possible de parties prenantes, le processus d'identification des risques permet souvent d'identifier la plupart des risques produit importants. Il est très important de déterminer les parties prenantes qui participeront à ce stade. Il est essentiel de veiller à ce que la liste des parties prenantes participantes soit complète et convenue avec le chef de projet. Une identification des risques qui ne tient pas compte des parties prenantes clés peut s'avérer très problématique. L'essentiel est de veiller à ce que toutes les parties prenantes concernées aient une chance de participer. Si elles ne peuvent pas participer, elles doivent au moins avoir la possibilité de déléguer la tâche. Lorsque des parties prenantes clés risquent de ne pas être représentées, une réunion de lancement peut être utilisée pour déterminer si elles manquent à l'appel.

Dans le test basé sur les risques, il est important de comprendre que le risque n'est pas uniformément réparti dans les objets du test. Par exemple, les composants d'une application en libre-service orientés vers le client peuvent présenter des risques d'utilisabilité très différents de ceux des composants d'administration. L'identification des risques individuels des différents éléments du test est une tâche importante dans la planification des tests.

L'identification des risques donne souvent lieu à des sous-produits (c'est-à-dire à l'identification de problèmes qui ne sont pas des risques produit). Il s'agit par exemple de questions ou de problèmes généraux concernant le produit ou le projet, ou de problèmes dans les documents de référence tels que les exigences et les spécifications de conception. Les risques projet sont également souvent identifiés comme un sous-produit de l'identification des risques qualité, mais ne sont pas au centre du test basé sur les risques. Le Test Manager peut souvent jouer un rôle important en mettant en évidence ces sous-produits et en faisant comprendre que la qualité est l'affaire de tous. Des exigences faibles ou manquantes sont souvent une indication d'un problème plus fondamental dans la planification et la préparation, car l'assurance qualité est impliquée tout au long du SDLC.

1.3.3 Évaluation des risques qualité

Une fois les risques identifiés, ils peuvent être évalués. Cette évaluation des risques qualité comprend la catégorisation des risques par type (risque produit ou risque projet) et par caractéristiques de qualité impactées.

La détermination du niveau de risque implique généralement une évaluation, pour chaque élément de risque, de la probabilité du risque et de l'impact du risque en cas de survenance. Les facteurs qui influencent la probabilité du risque qualité sont les suivants :

- Complexité de la technologie, des outils ou de l'architecture du système
- Maturité de l'organisation

- Problèmes de personnel liés aux compétences, à la disponibilité, à la motivation ou au travail autonome, y compris la connaissance du SDLC utilisé
- Conflit au sein de l'équipe
- Problèmes contractuels avec les fournisseurs
- Equipes géographiquement dispersées
- Faiblesse du leadership managérial ou technique
- Pression liée au temps, aux ressources, au budget et au management
- Absence d'activités d'assurance qualité à un stade précoce
- Taux élevés de changement de la base de test, du produit ou du personnel

Les facteurs influençant l'impact du risque sont les suivants :

- Fréquence d'utilisation de la fonctionnalité affectée
- Criticité de la fonctionnalité affectée
- Criticité de l'objectif métier concerné
- Atteinte à la réputation
- Perte de revenus pour le métier
- Pertes financières, écologiques ou sociales potentielles, ou responsabilité (juridique)
- Sanctions civiles ou pénales
- Problèmes d'interface et d'intégrité
- Absence de solutions de rechange raisonnables
- Besoins en matière de sûreté

Le Test Manager combine la probabilité du risque et l'impact du risque pour déterminer le niveau de risque.

Si l'analyse des risques est basée sur des données étendues et statistiquement valables, une évaluation quantitative est appropriée. Par exemple, la probabilité du risque peut être exprimée en pourcentage et l'impact du risque en montant. Dans ce cas, le niveau de risque peut être calculé comme le produit de ces deux facteurs. En général, cependant, la probabilité du risque et l'impact du risque ne peuvent être déterminés que qualitativement sur des échelles ordinales, par exemple très élevé, élevé, moyen, faible ou très faible. Les valeurs de probabilité du risque et d'impact du risque sont ensuite combinées dans une matrice de risque pour créer un niveau de risque global. Ce niveau de risque global doit être interprété comme une évaluation qualitative et relative sur une échelle ordinale.

À moins que l'analyse des risques ne soit basée sur des données exhaustives et statistiquement valables, l'analyse des risques sera qualitative, basée sur les perceptions subjectives des parties prenantes de la probabilité du risque et de son impact.

1.3.4 Atténuation des risques qualité par des tests appropriés

Dans le cadre du développement de logiciels, tester est l'activité d'atténuation des risques qualité la plus importante et permet de réduire la probabilité des défaillances. Parmi les autres mesures possibles d'atténuation des risques, on peut citer un plan d'urgence (par exemple, en prévoyant des solutions de contournement), le transfert des risques à un tiers (par exemple, le vendeur d'un composant) ou l'acceptation des risques.

Dans la **planification des tests**, le temps et les efforts associés au développement et à l'exécution d'un test doivent être proportionnels au niveau de risque : les tests pour les niveaux de risque élevés doivent commencer tôt et utiliser des techniques de test plus rigoureuses, tandis que les tests pour les niveaux de risque plus faibles peuvent commencer plus tard et utiliser des techniques de test moins rigoureuses. Pour atténuer au mieux le risque global grâce aux tests, le Test Manager doit analyser les facteurs contextuels suivants et sélectionner une approche de test appropriée :

- **Les éléments du test :** Différents éléments du test au sein d'un objet du test peuvent présenter différents niveaux d'un même type de risque, de sorte qu'un objet du test n'a pas besoin d'être testé avec une rigueur uniforme.
- **Les caractéristiques de qualité :** Les risques affectant des caractéristiques de qualité spécifiques doivent être atténués par des types de tests associés qui nécessitent un effort de test, des environnements de test et des compétences de test spécifiques.
- **Les niveaux de test et les types de test :** Certains risques ne peuvent être testés que de manière dynamique à des niveaux de test particuliers, d'autres par des tests statiques (par exemple, analyse statique et revues de code pour la maintenabilité), ou par une combinaison des deux (par exemple, par un réviseur de l'architecture et des tests dynamiques du système intégré pour les vulnérabilités de sécurité). Le fait de tester chaque élément du test le plus tôt possible atténue le risque de constater des défauts critiques à un stade avancé du cycle de vie, ce qui entraînerait des coûts de défaillance internes plus élevés et des retards.
- **Le SDLC :** Les activités de test ont leurs propres critères d'entrée spécifiques. Les différents SDLC les remplissent à des moments différents.
- **L'équipe de test :** Les personnes les plus qualifiées doivent tester les éléments du test présentant les niveaux de risque les plus élevés.
- **Les exigences réglementaires :** Certaines normes relatives à la sûreté (par exemple, la norme IEC 61508) prescrivent les techniques de test et la couverture requise en fonction du niveau d'intégrité. Le Test Manager doit s'assurer que ces normes sont respectées.

En outre, le niveau de risque devrait influencer les décisions en matière de contrôle de la qualité, telles que le recours à des revues des produits d'activités comme les cas de test, le niveau d'indépendance des tests par rapport au développement et l'étendue des tests de régression effectués.

Lors du **pilotage et** du **contrôle des tests**, le test basé sur les risques permet d'établir des rapports sur l'avancement des tests en fonction du niveau de risque résiduel à tout moment. Cela soutient l'équipe de développement et les parties prenantes dans le pilotage et le contrôle du développement de logiciels, y compris la prise de décisions de livraison, basées sur les risques résiduels. Cela exige de rapporter les résultats des tests en termes de risques d'une manière compréhensible pour les parties prenantes.

Lors de l'**implémentation des tests**, la priorisation des tests est basée sur les niveaux de risque. Pendant l'exécution du test, cela permet d'assurer une couverture précoce des domaines les plus critiques et d'atténuer les risques les plus élevés.

- Dans certains cas, les tests sont exécutés par ordre de priorité strictement décroissant des niveaux de risque qu'ils couvrent, en commençant par le plus élevé. Cette approche, appelée "profondeur d'abord", est appropriée lorsqu'il est important d'atténuer les niveaux de risques les plus élevés le plus tôt possible.
- Une autre solution consiste à attribuer la plus haute priorité à au moins un test pour chaque risque. Tous les autres tests sont classés par ordre de priorité en fonction de leur niveau de risque couvert. Cette approche, appelée "largeur d'abord", est appropriée lorsque les parties prenantes souhaitent avoir une vue d'ensemble de la qualité du produit le plus tôt possible. Dans la pratique, les tests commencent souvent par l'approche pilotée par la profondeur, mais lorsque le temps devient plus limité, ils passent à l'approche pilotée par la largeur, en testant tous les éléments de risque restants au moins une fois.

Que le test basé sur le risque soit effectué en profondeur, en largeur ou de manière combinée, le temps alloué aux tests peut être épuisé sans que tous les tests planifiés aient été exécutés. À ce stade, le test basé sur les risques facilitet la formulation d'une recommandation fondée sur les risques à l'intention du management, qu'il s'agisse de prolonger les tests ou d'accepter le risque restant.

1.3.5 Techniques pour le test basé sur les risques

Il existe un certain nombre de techniques spécifiques, plus ou moins formelles, pour implémenter le test basé sur les risques. L'adéquation d'une technique dépend de considérations relatives au projet, au processus et au produit. Il existe deux types de techniques de base : les techniques lourdes et les techniques légères. Dans les systèmes critiques en termes de sécurité, les techniques lourdes sont très souvent utilisées. Dans les applications non critiques en termes de sécurité, les techniques légères sont généralement employées.

Les techniques lourdes sont formelles et font appel à des procédures définies et à une documentation détaillée. Elles impliquent de larges groupes de parties prenantes. L'évaluation des risques dans le cadre des techniques lourdes fait appel à des facteurs détaillés de probabilité du risque et d'impact du risque, ainsi qu'à des formules mathématiques pour calculer la probabilité du risque et l'impact du risque à partir de ces facteurs. Voici quelques exemples de techniques lourdes :

- **Analyse des dangers :** Prolonge le processus analytique en amont en tentant d'identifier les dangers qui sous-tendent chaque risque.
- **Coût de l'exposition :** Détermine, pour chaque élément de risque qualité, la probabilité d'une défaillance, le coût d'une perte associée à une défaillance typique et le coût des tests pour de telles défaillances.
- **Analyse des modes de défaillance et de leurs effets (AMDE) et ses variantes :** Identifie les risques qualité, leurs causes potentielles et leurs effets probables, puis leur attribue une sévérité, une priorité et un taux de détection.
- **Analyse des arbres de défaillance :** Relie les défaillances potentielles aux défauts qui peuvent provoquer la défaillance, puis aux erreurs qui peuvent provoquer ces défauts, en continuant jusqu'à ce que les différentes causes racines soient identifiées.

En revanche, les techniques légères sont moins approfondies et nécessitent moins d'effort de la part de l'équipe de test et des parties prenantes. Comme les techniques lourdes, elles sont également basées sur l'implication des parties prenantes, en utilisant les résultats de l'analyse des risques comme base de la planification des tests et des conditions de test. Cependant, le groupe de parties prenantes peut ne pas être très large, et les facteurs de risque sont généralement réduits à l'impact du risque et à la probabilité

du risque sur une échelle ordinale. Certaines de ces techniques, comme les tests systématiques de logiciels (SST) (Craig & Jaskiel, 2002), ne peuvent être utilisées que lorsque les spécifications des exigences sont fournies. D'autres techniques, notamment l'analyse et la gestion pragmatiques des risques (PRAM) (Black, 2009) et la gestion des risques produit (PRISMA) (van Veenendaal, 2012), utilisent les exigences et/ou d'autres spécifications comme données d'entrée de l'analyse des risques, mais peuvent fonctionner entièrement sur la base des données fournies par les parties prenantes.

1.3.6 Métriques de succès et difficultés associées au test basé sur les risques

Dans une rétrospective, l'équipe de test devrait mesurer le degré de réalisation des bénéfices du test basé sur les risques. Dans de nombreux cas, cela implique de répondre à tout ou partie des questions suivantes par le biais de métriques et de consultations :

- Les parties prenantes concernées ont-elles été impliquées ou représentées dans l'analyse des risques ?
- La participation des parties prenantes à l'analyse des risques était-elle appropriée ?
- S'il y a eu des incidents critiques en production indiquant que des défauts critiques se sont échappés, ont-ils été résolus ?
- La plupart des défauts de priorité élevée ont-ils été constatés au début de l'exécution du test ?
- L'équipe de test a-t-elle été en mesure d'expliquer les résultats du test aux parties prenantes en termes de risque ?
- Les tests ignorés présentaient-ils un niveau de risque associé inférieur à celui des tests exécutés ?

Dans la plupart des cas, le test basé sur les risques aboutit à une réponse affirmative à toutes ces questions. À long terme, il convient de fixer des objectifs d'amélioration des processus pour les métriques de réussite, tout en s'efforçant d'améliorer l'efficience du processus d'analyse des risques qualité.

La gestion du risque se heurte souvent à des difficultés inattendues dues à des complexités souvent négligées.

- **Difficulté à évaluer le niveau de risque :** L'estimation de l'impact du risque et de la probabilité du risque peut s'avérer très difficile. Solution : utiliser des données historiques et demander aux principales parties prenantes du projet leur évaluation.
- **Des débuts prometteurs :** La mise en place et la maintenabilité d'une approche de test basé sur les risques appropriée, sont souvent négligées face à la forte pression de réussite à court terme. Solution : suivre et signaler régulièrement les risques aux parties prenantes.
- **Déjà vu :** les mêmes risques sont soulevés pour chaque projet, ce qui conduit à une certaine complaisance à l'égard des risques. Solution : faire participer les bonnes personnes à l'identification des risques et n'atténuer que les risques jugés importants.
- **Des risques importants ne sont pas identifiés :** La cause racine de ce problème est généralement due à l'implication de personnes inexpérimentées ou inappropriées dans le processus. Solution : impliquer les personnes appropriées et les former.
- **Les parties prenantes changent :** Les parties prenantes peuvent changer au fil du temps et de nouveaux risques peuvent apparaître. L'analyse des risques est donc une activité continue et itérative qui ne doit pas être réalisée une seule fois au début du projet.

1.4 La stratégie de test du projet

Introduction

Tout au long de ce syllabus, la stratégie de test organisationnelle est considérée comme donnée. Le développement et la maintenance d'une stratégie de test organisationnelle sont considérés dans le contexte de l'ISO/IEC/IEEE 29119-3 (où il est fait référence à une "pratique de test organisationnelle") et les syllabus pour testeurs certifiés de l'ISTQB® de niveau Expert – Management des tests, ou encore Conduite des tests en mode Agile à l'échelle (ATLaS).

Si une stratégie de test organisationnelle n'existe pas ou ne couvre pas les aspects requis, le management des tests doit chercher à clarifier les détails manquants avec les parties prenantes concernées.

Dans le contexte de cette section, la définition d'une stratégie de test de projet est un exemple de tout type de stratégie de test détaillée pour un projet, une release, un produit, ou tout autre type d'initiative de développement ou d'acquisition de systèmes. Une stratégie de test de projet (appelée "stratégie de test" dans la norme ISO/IEC/IEEE 29119-3) spécifie l'approche de test dans un contexte spécifique afin que les objectifs de l'organisation puissent être atteints, en particulier ceux relatifs à la qualité du produit et aux activités de test. Une stratégie de test peut également exister pour un seul niveau de test ou un seul type de test.

La stratégie de test du projet est le principal résultat de la planification des tests pour un projet et est généralement documentée dans un plan de test ou dans le cadre d'autres documents. La documentation de la stratégie de test est recommandée, mais pas nécessairement sous la forme d'un plan de test formel. Le besoin de documentation dépend du contexte du test (voir Section 1.2, Le contexte du test). Lorsqu'un projet suit un modèle de développement séquentiel, la stratégie de test du projet est généralement documentée, de préférence dans le plan de test (voir ISO/IEC/IEEE 29119-3). La documentation est également souvent exigée par les contrats, les accords, les organismes de réglementation ou les lois.

1.4.1 Choisir une approche de test

La stratégie de test du projet guide toutes les activités de test au sein d'un projet et détaille les objectifs, les ressources, les calendriers et les responsabilités. Cette stratégie doit être adaptée aux exigences particulières du projet. Les décisions clés comprennent la sélection des niveaux de test, des types de test et des techniques de test pour les tests statiques et dynamiques et d'autres pratiques de test (par exemple, les tests scripté, les tests manuels, les tests dos-à-dos).

En théorie, tous les types de tests peuvent être effectués à n'importe quel niveau de test, et n'importe quelle technique de test peut être appliquée à n'importe quel type de test et à n'importe quel niveau de test. En pratique, la sélection et la combinaison appropriées de ces choix ont un impact significatif sur l'efficacité et l'efficience des tests. Par exemple, la maintenabilité du code peut souvent être évaluée de manière plus efficace et efficiente en utilisant l'analyse statique du code ou la revue du code. D'autre part, l'efficience de la performance peut être mieux évaluée par des tests système scriptés en raison de l'interaction des composants internes, ou l'utilité d'une fonctionnalité peut être mieux validée avec les utilisateurs par des tests d'acceptation manuels développés en collaboration. Le choix de la meilleure approche pour une stratégie de test peut être un processus complexe qui peut être influencé par la stratégie de test de l'organisation, le contexte du projet et d'autres aspects.

La sélection et la combinaison des niveaux de test, des types de test et des techniques de test sont donc essentielles à une stratégie de test de projet efficace, car elles influencent de manière significative l'efficience et l'efficacité des tests.

1.4.2 Analyser la stratégie de test de l'organisation, le contexte du projet et d'autres aspects

La stratégie de test de l'organisation, le contexte du projet et les facteurs ou contraintes supplémentaires liés aux tests doivent être parfaitement compris pour permettre le développement d'une stratégie de test du projet.

Pour choisir une approche de test appropriée, les facteurs suivants doivent généralement être analysés :

- **Domaine :** Le domaine pour lequel le produit sera créé ou modifié. Toute réglementation, norme ou pratique spécifique à un domaine peut modifier la rigueur des tests, la documentation requise, ainsi que son niveau de détail. Par exemple, dans le domaine des produits pharmaceutiques et de la médecine, l'approche de test met souvent l'accent sur des tests intensifs d'acceptation par les utilisateurs, axés sur les risques pour la santé des patients, en utilisant des cas de test basés sur les exigences fonctionnelles des utilisateurs; alors que les tests d'acceptation par les utilisateurs pour les applications d'assurance en ligne peuvent être axés sur l'utilisabilité et l'augmentation de la probabilité de nouveaux contrats d'assurance par le biais de tests A/B**.**
- **Objectifs de l'organisation et caractéristiques de qualité primordiales :** Les objectifs organisationnels peuvent inclure la nécessité de démontrer la valeur des tests et d'augmenter le degré d'automatisation des tests ou les caractéristiques de qualité du processus de test telles que le niveau de maturité des tests ou l'efficience de la détection des défauts. Ces caractéristiques peuvent déterminer les niveaux de test et les types de test qui doivent être respectés.

- **Les objectifs et le type de projet :** Les objectifs du projet (par exemple, concernant le budget, le temps et la qualité) et le type de projet (par exemple, le développement d'un produit spécifique au client ou orienté vers le marché) contiennent typiquement des contraintes et des risques, ainsi que des opportunités qui affectent les tests. Par exemple, un budget serré et des contraintes de temps peuvent exiger l'utilisation rigoureuse du test basé sur les risques pour prioriser les cas de test pour l'exécution des tests, alors que le développement d'un produit spécifiquement pour un client peut exiger des tests qui couvrent des critères d'acceptation contractuels prédéfinis.
- **Ressources de test :** Toute contrainte concernant la disponibilité des ressources de test, y compris les outils de test, l'infrastructure de test, la technologie et l'environnement de développement utilisés dans le projet, ainsi que le personnel de test disponible et ses compétences, doit être prise en compte. (voir section 3.1, L'équipe de test). Par exemple, les tests basés sur l'expérience nécessitent des testeurs ayant une bonne connaissance du domaine ; les applications mobiles doivent généralement être testées sur un nombre typiquement limité de dispositifs différents ; l'utilisation des outils de test peut être limitée par le nombre de licences disponibles.

- **Le modèle de cycle de vie du développement logiciel utilisé pour le projet :** Pour déterminer les niveaux de test appropriés, l'effort de test, les critères d'entrée et les critères de sortie appropriés, voir le syllabus de niveau Fondation de l'ISTQB® V.4, section 2.2 et 5.1. Un cycle de vie du logiciel avec intégration continue exige davantage de tests automatisés qu'un

développement ponctuel utilisant un modèle en cascade et, par conséquent, différents types de tests et techniques de test peuvent être utilisés.

- **Interfaces avec d'autres systèmes :** Dans un système de systèmes, il est essentiel d'aligner les tests avec d'autres équipes ou projets et de sélectionner des niveaux de test appropriés, en particulier pour les tests d'intégration des systèmes. Par exemple, les tests basés sur les risques permettent de hiérarchiser et d'échelonner les tests d'intégration des systèmes.
- **Disponibilité des données de test :** Il faut tenir compte des contraintes liées à la disponibilité des données de test, comme la nécessité de disposer de données de test anonymisées provenant de la production, ou la création de données de test spécifiques qui peuvent être difficiles à fournir et qui doivent être validées, comme les données pour les tests d'IA. Par exemple, les tests basés sur des modèles peuvent soutenir la création de données de test et la gestion des données de test.

Le Test Manager doit déterminer quelle combinaison de techniques de test, de niveaux de test et de types de test doit être utilisée comme la meilleure approche pour satisfaire la stratégie de test de l'organisation, le contexte du projet et les facteurs ou contraintes supplémentaires liés aux tests.

1.4.3 Définition des objectifs de test

Un plan de test doit être défini pour chaque projet de test et doit contenir, entre autres, le périmètre du test, les objectifs de test et les critères de sortie. Le plan de test peut être établi au niveau de la release, en tant que plan de test du projet (également appelé plan de test maître) et, si nécessaire, en tant que plan de test de niveau pour les différents niveaux de test. En outre, il est possible de définir des plans de test pour les différentes caractéristiques de qualité, comme un plan de test de sécurité ou un plan de test de performance. Dans les projets de développement logiciel en Agile et les projets de développement logiciel hybride, un plan de test par itération peut être convenu. Pour chaque release et itération, le périmètre des fonctionnalités et de leurs caractéristiques non fonctionnelles à livrer est défini dans le plan de test et approuvé par les parties prenantes.

Les objectifs du test du projet et les critères de sortie doivent être définis en relation avec les fonctionnalités livrées pour être testées dans le cadre d'un projet. Cela peut être fait en utilisant la méthodologie S.M.A.R.T orientée sur les objectifs :

- S = spécifique. L'objectif du test d'un projet et le critère de sortie doivent être clairs et sans ambiguïté.
- M = mesurable. Il doit être quantifiable et comporter des critères spécifiques permettant de mesurer les progrès accomplis afin de déterminer s'il a été atteint.
- A = atteignable. Il doit être réalisable compte tenu des ressources, du calendrier et des capacités disponibles.
- R = réaliste. Il doit être aligné sur les objectifs généraux du projet.
- T = temporel. Il doit être assorti d'un calendrier précis et d'une date limite d'achèvement.

Les objectifs du test du projet doivent porter sur tous les aspects ciblés de la qualité et de la quantité, pour autant qu'ils soient mesurables ou évaluables. Voici quelques exemples d'objectifs de test de projet :

- Atteindre les critères de sortie spécifiés dans le délai défini.

- Atteindre les objectifs de qualité de l'organisation (par exemple, mesuré comme un indicateur de performance clé pour le nombre de réclamations de clients pour un produit).
- Se conformer aux règles et réglementations de l'industrie concernée.
- Garantir la disponibilité des données aux seuls utilisateurs autorisés (par exemple, par des droits d'accès).
- Vérifier la complétude fonctionnelle, l'exactitude fonctionnelle, l'efficience, l'efficacité, la portabilité et la sécurité de la migration des données.
- Améliorer le niveau d'automatisation des tests (par exemple, pour les tests de régression ou de performance, selon un pourcentage défini).
- Refactoriser le code avec succès et montrer qu'il n'a pas introduit de nouveaux défauts (par exemple, pour supprimer un code source mal structuré ou une dette technique tout en maintenant la fonctionnalité existante, ce qui est testé par un test de régression).
- Prouver la sécurité des interfaces (par exemple, en validant les messages XML (Extensible Markup Language) par rapport à leur définition de schéma XML pour garantir le rejet des données malveillantes).
- Vérifier l'utilisabilité d'une interface utilisateur et atteindre un certain degré de sous-caractéristique (par exemple, en mesurant le temps nécessaire à l'accomplissement d'une tâche spécifique dans une boutique en ligne).

Outre le comptage et la mesure des objectifs de test du projet, l'évaluation du niveau de qualité par les experts du domaine et les parties prenantes doit être pris en compte.
En fonction du contexte du projet et des objectifs de test, il est parfois nécessaire de disposer de plusieurs environnements de test avec les ressources et/ou les outils de test disponibles. Les environnements de test peuvent ne pas être tous disponibles en même temps. Il convient d'en tenir compte lors de la formulation d'objectifs du test et de critères de sortie réalisables.

En fonction du contexte du projet, des facteurs supplémentaires doivent être pris en compte dans la définition des objectifs de test et du périmètre de test du projet, comme décrit dans la section 1.2 de ce syllabus, Le contexte du test.

1.5 Améliorer le processus de test

Introduction

Les tests constituent une part importante du développement de logiciels et représentent souvent au moins 30 à 40 % du coût total du projet. Outre les nombreux défis (techniques) auxquels les projets de logiciels sont confrontés (complexité et taille croissantes, nouvelles technologies, grande variété d'appareils et de systèmes d'exploitation, vulnérabilités en matière de sécurité), il est nécessaire d'optimiser l'efficacité et l'efficience des tests ainsi que d'améliorer les processus de test en conséquence. Apprendre des meilleures pratiques existantes et de ses propres erreurs permet d'améliorer le processus de test et de rendre les projets plus fructueux.

Un processus d'amélioration au niveau de l'organisation est généralement plus utile qu'un processus d'amélioration au niveau d'un projet ou d'une équipe. Cependant, il est également possible et bénéfique d'appliquer l'amélioration des processus au niveau d'un projet ou d'une équipe, mais elle doit être adaptée aux besoins du projet ou de l'équipe. Une amélioration des tests peut être initiée, par exemple, par une insatisfaction des résultats des tests actuels, des défauts inattendus, un changement de circonstances, un résultat de benchmark ou un manque de communication. Différentes techniques sont disponibles pour améliorer les tests (Bath & van Veenendaal, 2014). Certaines de ces techniques sont décrites ci-dessous. Les techniques décrites dans ce syllabus peuvent être appliquées à la fois aux modèles de développement séquentiel et aux modèles de développement logiciel Agile/incrémental. Le syllabus « Improving the Test Process » (Améliorer le Processus de Test), de niveau Expert de l'ISTQB® offre un aperçu plus approfondi.

1.5.1 Processus d'amélioration des tests (IDEAL)

Une fois qu'il a été convenu que les processus de test devaient être améliorés, les activités d'implémentation de l'amélioration du processus à adopter pour cette activité peuvent être définies comme dans le modèle IDEAL, qui est basé sur des idées similaires au cycle bien connu plan-do-check-act (planifier, exécuter, vérifier, agir) (PDCA).
IDEAL est un acronyme qui signifie: Initiating, Diagnosing, Establishing, Acting and Learning (Initier, Diagnostiquer, Etablir, Agir et apprEndre.

Bien qu'IDEAL ait été défini à l'origine pour soutenir les activités d'amélioration à un niveau organisationnel, il peut également être appliqué au niveau d'un projet ou d'une équipe de développement logiciel en mode Agile. Dans le cadre d'un projet, les objectifs des activités (voir ci-après) doivent encore être atteints. La principale différence réside probablement dans la phase d'initiation, qui est beaucoup moins importante au niveau d'un projet ou d'une équipe qu'au niveau de l'organisation. Le diagnostic par le biais d'une rétrospective et l'établissement d'un plan seront probablement beaucoup moins importants qu'au niveau organisationnel. L'action et l'apprentissage seront également pertinents au niveau d'un projet ou d'une équipe.

Initier le processus d'amélioration

Au début du processus d'amélioration, les objectifs et le périmètre des améliorations du processus sont convenus par les parties prenantes.

Diagnostiquer la situation actuelle

Le processus de test actuel est évalué afin d'identifier les améliorations possibles. L'évaluation est généralement réalisée par rapport à un framework standard dans le cas d'une amélioration du processus de test basée sur un modèle (voir Section 1.5.2, Amélioration du processus de test basée sur un modèle) ou peut être basée sur une analyse de métriques spécifiques dans le cas d'une approche analytique du

processus de test. (Voir section 1.5.3, Approche de l'amélioration du processus de test basée sur l'analyse).

Établissement d'un plan d'amélioration du processus de test

Un plan d'amélioration du processus de test peut être un document formel qui énumère toutes les actions détaillées qui doivent être réalisées pour obtenir des améliorations. Selon le contexte, le plan peut être très informel et très léger. La liste des améliorations possibles du processus doit être classée par ordre de priorité. L'ordre de priorité peut être basé sur le retour sur investissement (ROI), les risques, l'alignement avec les stratégies du projet ou de l'équipe, et/ou les avantages quantitatifs ou qualitatifs mesurables qui doivent être accomplis.

Agir pour implémenter l'amélioration du processus de test

Le plan d'amélioration du processus test pour la mise en œuvre des améliorations est implémenté. Cela comprend généralement la formation et le pilotage des processus modifiés et leur déploiement complet dans le projet ou l'équipe.

Apprendre du programme d'amélioration

Après avoir pleinement déployé les améliorations du processus, il est essentiel de vérifier quels avantages, prévus ou inattendus, ont été obtenus. Ayant appris ce qui a fonctionné et ce qui n'a pas fonctionné, nous devons agir sur la base de ces informations et ce n'est qu'ensuite que le prochain cycle d'amélioration peut commencer.

1.5.2 Amélioration du processus de test basés sur des modèles

L'une des prémisses de l'amélioration des processus de test basés sur des modèles et de l'amélioration basée sur l'analyse est l'hypothèse selon laquelle la qualité du produit est fortement influencée par la qualité des processus utilisés et appliqués. Lors de l'application de l'amélioration du processus de test basée sur des modèles, on utilise un modèle d'amélioration du test. Les modèles d'amélioration du test sont basés sur les meilleures pratiques de test et organisent l'amélioration du test de manière progressive.

Plusieurs modèles de processus recommandés ont émergé pour soutenir l'amélioration du processus de test. Il s'agit notamment du Tests Maturity Model integration (TMMi®) et de TPI NEXT®.

L'amélioration basée sur un modèle peut également être appliquée au niveau d'un projet. Dans ce cas, l'évaluation et le processus d'amélioration sont spécifiquement axés sur les processus de test ou les domaines clés définis dans le modèle qui se rapportent aux activités au niveau du projet (par exemple, la planification des tests et la conception des tests) et omettent souvent en grande partie ceux qui se situent au niveau de l'organisation (par exemple, la politique de test et l'organisation des tests). Il est également possible d'adapter les pratiques au niveau de l'organisation au contexte du projet.

Pour plus d'informations sur l'amélioration du processus de test basée sur des modèles, voir le syllabus ISTQB® de niveau Expert "Improving the Test Process" (Amélioration du Processus de Test).

Test Maturity Model integration

Le TMMi® (van Veenendaal & Cannegieter, 2011) (van Veenendaal, 2020) est composé de cinq niveaux de maturité. Chaque niveau de maturité, à l'exception du TMMi® niveau 1, contient des domaines de processus de test et des objectifs d'amélioration. En outre, pour faciliter et soutenir son implémentation,

TMMi® contient des pratiques, des sous-pratiques et des exemples. Le TMMi® a été initialement développé pour compléter le Capability Maturity Model Integration (CMMI®), mais il est aujourd'hui largement utilisé indépendamment du CMMI®.

Pour faciliter et soutenir la mise à jour du TMMi® dans le développement logiciel Agile, une ligne directrice spécifique a été développée qui explique comment le TMMi® peut être utilisé et appliqué de manière bénéfique dans le développement logiciel Agile.

Pour plus d'informations sur le TMMi®, voir www.tmmi.org et www.cftl.fr pour sa documentation en français.

TPI NEXT®

Le modèle TPI NEXT® (van Ewijk, 2013) définit 16 domaines clés, chacun couvrant un aspect spécifique du processus de test (par exemple, la stratégie de test, les métriques de test, les outils de test et l'environnement de test). Quatre niveaux de maturité sont définis dans le modèle pour chacun des 16 domaines clés.

Des points de contrôle spécifiques sont définis pour évaluer chaque domaine clé à chacun des niveaux de maturité. Les résultats de l'évaluation sont résumés et visualisés au moyen d'une matrice de maturité qui couvre tous les domaines clés.

Pour plus d'informations sur TPI NEXT® voir www.tmap.net.

1.5.3 Approche analytique de l'amélioration du processus de test

Dans le cadre d'une approche d'amélioration basée sur des modèles, telle que décrite dans la section précédente, des améliorations sont introduites en comparant l'approche test d'un projet ou d'une équipe aux meilleures pratiques externes. Les approches analytiques identifient les problèmes sur la base de données provenant du projet ou de l'équipe elle-même. Les améliorations appropriées peuvent être dérivées d'une analyse de ces problèmes. Les approches analytiques peuvent être utilisées conjointement avec une approche basée sur un modèle pour vérifier les résultats et apporter de la diversité.

Les problèmes peuvent être identifiés à l'aide de données quantitatives et qualitatives. La section 1.5.3 de ce syllabus, Approche analytique de l'amélioration du processus de test, présente les approches analytiques qui utilisent principalement des données quantitatives du processus de test et des données de défauts pour évaluer l'approche actuelle. La section 1.5.4 de ce syllabus, Rétrospectives, présente les rétrospectives, dans lesquelles des données qualitatives concernant ce qui fonctionne bien et ce qui ne fonctionne pas bien sont collectées auprès des membres des équipes de développement et de test.

L'analyse des données est importante pour l'amélioration objective du processus de test et constitue un soutien précieux aux évaluations purement qualitatives, qui peuvent sinon aboutir à des recommandations imprécises qui ne sont pas étayées par des données. L'application d'une approche d'amélioration analytique implique le plus souvent une analyse quantitative du processus de test afin d'identifier les domaines problématiques et de fixer des objectifs spécifiques au projet. La définition et la mesure de paramètres clés sont nécessaires pour évaluer le processus de test et déterminer si les améliorations sont fructueuses.

Voici quelques exemples d'approches analytiques :

- Analyse des causes racines
- Analyse à l'aide de mesures, de métriques et d'indicateurs

- L'approche GQM (Goal-Question-Metric) / (NDT: But-Question-Métrique)

L'analyse des causes racines est l'étude des problèmes afin d'en identifier les causes racines. Cela permet d'identifier des solutions qui éliminent les causes des problèmes plutôt que de se contenter de traiter les symptômes immédiats et évidents. Une procédure d'analyse possible consisterait à sélectionner un ensemble approprié de défauts, à identifier des regroupements dans ces données et à utiliser des diagrammes de cause à effet (également appelés diagrammes d'Ishikawa ou diagrammes en arêtes de poisson) pour identifier les causes racines des regroupements de défauts importants. Des améliorations sont ensuite apportées afin d'éviter que des défauts similaires ne se produisent.

Les mesures, métriques et indicateurs sont utilisés de manière quantitative pour évaluer le processus de test au sein du projet ou de l'équipe. Les attributs clés du processus de test à prendre en compte sont l'efficacité, l'efficience et la prévisibilité. Pour chacun de ces attributs, une ou plusieurs métriques peuvent être sélectionnées. La collecte et l'analyse des données correspondantes permettent d'identifier les domaines clés nécessitant une amélioration.

L'approche GQM (Basili, et al., 2014) (van Solingen & Berghout, 1999) fournit un framework permettant de définir et d'analyser des métriques adaptées aux besoins d'information des parties prenantes du projet. Les objectifs de mesure définissent un aspect de la qualité d'un objet qui doit être mesuré dans un but, une perspective et un contexte particuliers. Ces objectifs sont remaniés en questions qui définissent l'aspect de la qualité du point de vue des parties prenantes. On sélectionne ensuite les métriques qui fournissent les informations nécessaires pour répondre à la question. Les données collectées pour les métriques répondent aux questions, afin d'évaluer l'objectif de la mesure et de satisfaire les besoins d'information des parties prenantes.

Plus d'informations sur ces approches d'amélioration du processus de test basées sur l'analyse peuvent être trouvées dans le syllabus ISTQB® de niveau Expert "Improving the Test Process".

1.5.4 Rétrospectives

Les rétrospectives sont des réunions au cours desquelles une équipe révise ses méthodes et sa collaboration, tire des leçons (bonnes et mauvaises) et décide des changements et des actions à entreprendre pour obtenir des améliorations (à la fois pour les questions de test et les autres questions). Les rétrospectives abordent des sujets tels que le processus, les personnes, l'organisation, la collaboration et les outils.

Les rétrospectives sont utilisées à la fois dans les modèles de développement séquentiel et dans le développement logiciel en mode Agile. Dans les modèles de développement séquentiel, elles font partie de la clôture des tests. Dans ce contexte, les rétrospectives visent à générer des leçons apprises afin de mieux gérer les projets futurs. Dans le développement logiciel en mode Agile, les rétrospectives sont généralement organisées à la fin de chaque itération pour discuter de ce qui a été réussi et de ce qui doit être amélioré, et de la manière dont ces améliorations peuvent être intégrées dans l'itération suivante. Les rétrospectives sont réalisées par l'ensemble de l'équipe. Elles soutiennent donc l'approche de l'équipe intégrée et favorisent l'amélioration continue. Il convient de noter que des rétrospectives spécifiques sont parfois nécessaires pour traiter les problèmes liés aux tests.

Une rétrospective typique comprend les étapes suivantes :

Introduction : L'objectif et le programme de la rétrospective sont revus et une atmosphère de confiance mutuelle est créée afin que les problèmes puissent être discutés sans blâme ni jugement.

Collecter des données : Des données sont collectées sur ce qui s'est passé pendant l'itération ou le projet. Il est possible de recueillir des données qualitatives, telles qu'une chronologie des événements

clés qui identifie les problèmes et indique ce que chaque membre de l'équipe pense de ces problèmes. En outre, des données quantitatives provenant de métriques peuvent être présentées, par exemple, les données relatives à la progression des tests, à la détection des défauts, à l'efficacité des tests, à l'efficience des tests et à la prévisibilité peuvent fournir un aperçu objectif des tests du projet ou de l'itération.

Identifier des améliorations : Les données collectées sont analysées pour comprendre la situation actuelle et générer des idées d'amélioration. Par exemple, l'analyse des causes racines peut être appliquée pour identifier les causes racines des problèmes identifiés et une session de brainstorming peut être organisée afin de générer des idées sur la façon de résoudre les causes racines.

Décider des actions d'amélioration : Les actions visant à implémenter les idées d'amélioration sont dérivées et classées par ordre de priorité. Un plan d'amélioration et des responsabilités sont définis. Des objectifs et des métriques associées peuvent être définis pour évaluer l'impact des actions sur les problèmes identifiés. L'implémentation d'un trop grand nombre d'améliorations à la fois est difficile à gérer avec des étapes vérifiables.

Clôturer la rétrospective : Lors de cette dernière étape, la rétrospective elle-même est revue afin d'identifier les points forts et les améliorations à apporter au processus de rétrospective. Une rétrospective est effectuée régulièrement, en particulier dans le cadre du développement logiciel en mode Agile. L'amélioration continue est également appliquée à la rétrospective elle-même.

Il est important de documenter de manière appropriée les résultats d'une rétrospective. Dans un modèle de développement séquentiel, les constatations, les conclusions et les recommandations doivent être distribuées et communiquées de manière compréhensible aux membres de l'organisation. Dans le développement logiciel en mode Agile, les problèmes et les actions doivent également être documentés pour permettre la revue des actions et leur impact potentiel sur les problèmes dans l'itération suivante.

Les testeurs, qui font partie de l'équipe (de projet), apportent leur point de vue unique. Ils peuvent soulever des problèmes liés aux tests (et d'autres) et inciter l'équipe à réfléchir aux améliorations possibles.

Des informations complémentaires peuvent être trouvées dans (Derby & Larsen, 2006).

1.6 Outils de test

Introduction

Il existe trois types d'outils métier :

- Outils commerciaux
- Outils Open-Source
- Outils personnalisés (NDT: maison)

Lors de la sélection d'un outil commercial, il faut tenir compte de toutes les exigences et réglementations de l'organisation et des parties prenantes.

Il existe également des outils techniques tels que les outils d'automatisation des tests, les outils de management des tests et bien d'autres encore.

Des exemples d'utilisation d'outils de test peuvent être trouvés dans le Syllabus de niveau Fondation de l'ISTQB® V.4.

1.6.1 Bonnes pratiques pour l'introduction des outils

Cette section contient les étapes nécessaires à l'évaluation et à l'introduction d'un outil de test.

Un Test Manager peut être impliqué dans l'introduction d'un outil ou peut encourager ou faciliter le processus d'introduction. Les Test Managers sont généralement responsables d'un outil de test dédié, ou de tout autre outil lié aux tests, tel qu'un outil de gestion des exigences, de gestion des défauts ou de pilotage.

Il existe des bonnes pratiques génériques et des considérations lors de l'évaluation et de la sélection d'un outil de test. Ces pratiques et considérations sont notamment les suivantes :

- Identifier les possibilités d'amélioration des processus, avec le soutien d'outils appropriés.
- Comprendre les technologies utilisées dans une organisation et choisir un outil compatible avec ces technologies.
- Comprendre comment un outil est techniquement et organisationnellement intégré dans le cycle de vie du développement (SDLC).
- Évaluer l'outil en fonction d'exigences claires et de critères objectifs.
- Évaluer le fournisseur si vous envisagez d'utiliser un outil commercial. Évaluer le soutien apporté aux outils non commerciaux (par exemple, les outils Open-Source).
- Identifier les exigences internes en matière d'accompagnement, de mentorat ou de formation à l'utilisation de l'outil.
- Examiner les avantages et les inconvénients des différents modèles de licence.
- Pour finir, effectuez une évaluation de validation du concept (NDT: proof of concept).

Les bonnes pratiques génériques pour l'adoption et le déploiement d'un outil sont les suivantes :

- Mener un projet pilote afin de valider les critères de sélection et les exigences et d'évaluer la manière dont l'outil s'intègre aux processus et pratiques existants.

- Adapter et améliorer les processus pour qu'ils correspondent à l'utilisation de l'outil, et adapter l'outil aux processus existants, si nécessaire.
- Définir des lignes directrices pour l'utilisation de l'outil.
- Assurer la formation, l'accompagnement et le mentorat des utilisateurs de l'outil.
- Introduire l'outil dans l'organisation de manière incrémentale.
- Implémenter un moyen de recueillir des informations à partir de l'utilisation réelle de l'outil en vue d'améliorations ultérieures.
- Définir qui est responsable de l'outil.

1.6.2 Aspects techniques et métiers pour les décisions relatives aux outils

De multiples facteurs influencent la décision concernant l'implémentation et l'utilisation d'un outil. Pour un Test Manager, il est important de les connaître et d'y répondre.

- **Réglementation et sécurité :** Les organisations qui développent des logiciels critiques pour la sûreté ou la mission, ou qui sont soumises à la conformité réglementaire, peuvent préférer les outils commerciaux car ils répondent plus souvent aux normes requises et possèdent souvent la certification appropriée.
- **Aspects financiers :** Les outils Open-Source ont généralement un coût initial moins élevé en raison du support et du développement de la communauté. Les outils commerciaux peuvent avoir un prix d'achat unique ainsi que des coûts de licence récurrents. Le coût initial d'un outil personnalisé est difficile à déterminer car il dépend des exigences et du stade de développement de l'outil. Outre les coûts initiaux, les coûts de formation et de maintenance pendant la durée de vie d'un outil doivent être calculés et pris en compte. Tous les outils peuvent avoir des coûts de maintenance et de support élevés.
- **Exigences des parties prenantes :** Il est important de recueillir les exigences de toutes les parties prenantes pour évaluer et identifier l'outil le plus approprié. Les outils commerciaux et les outils Open-Source ne répondent pas nécessairement à toutes les exigences en détail. Les outils personnalisés peuvent être le meilleur choix pour répondre à toutes les exigences individuelles et dans les cas où aucun autre outil ne fournit la fonctionnalité requise.
- **L'environnement logiciel existant et la stratégie en matière d'outils :** La composition existante des outils (paysage logiciel) et la stratégie d'outils associée doivent être évaluées, car il peut y avoir des fournisseurs privilégiés ou verrouillés, des systèmes intégrés qui ont des dépendances avec d'autres produits, ou un modèle spécial de support étendu à l'ensemble du paysage logiciel avec des réglementations spécifiques.

1.6.3 Considérations relatives au processus de sélection et évaluation du retour sur investissement

Les outils de test peuvent constituer un investissement à long terme, s'étendant parfois sur plusieurs itérations d'un même projet, et/ou s'appliquant à de nombreux projets. Un outil prospectif doit donc être envisagé sous différents angles.

- Pour la direction, un retour sur investissement positif est une exigence.

- Pour l'équipe de support et d'exploitation, un nombre limité mais nécessaire d'outils utilisés dans l'ensemble de l'organisation est préférable. La maintenabilité d'un plus grand nombre d'outils, le suivi de leurs licences et la gestion de la pile d'outils ne doivent pas être coûteux ou chronophages.
- Pour les chefs de projet, l'outil doit apporter une valeur ajoutée mesurable au projet ou à l'organisation.
- Pour les personnes qui utilisent l'outil, l'utilisabilité est très importante. L'utilisabilité comprend, par exemple, le support pour des tâches données, la facilité d'apprentissage et l'opérabilité.
- Pour les membres du personnel d'exploitation, la maintenabilité est importante.

Les fonctionnalités doivent être analysées pour chaque type d'outil métier et technique. Différents points de vue et intérêts ont une influence sur cette analyse : le management des tests, l'analyse (technique) des tests, l'automatisation des tests ou le développement. La personne de l'organisation qui est responsable de l'outil doit s'assurer que l'analyse est effectuée et que les points mentionnés ci-dessus sont pris en compte.

Tous les outils introduits dans le processus de test doivent également garantir un retour sur investissement positif pour l'organisation. Il est de la responsabilité du Test Manager de s'occuper du calcul et de l'évaluation ultérieure du ROI (NDT : Retour sur investissement). Dans le cadre du développement logiciel en mode Agile, cette responsabilité peut incomber à l'ensemble de l'équipe de développement.

Une analyse coût-bénéfice doit être réalisée avant l'acquisition ou le développement d'un outil afin de s'assurer qu'il est bénéfique pour l'organisation. Cette analyse doit prendre en compte les coûts récurrents et non récurrents.

Les activités et les coûts non récurrents comprennent les éléments suivants :

- Définition et détermination des exigences en matière d'outils pour atteindre les objectifs.
- Évaluation et sélection de l'outil et du fournisseur adéquats, validation du concept (P.O.C.)
- Achat, adaptation ou développement de l'outil pour une première utilisation.
- Définition des lignes directrices pour l'utilisation de l'outil.
- Formation initiale à l'outil.
- Intégration de l'outil dans l'environnement existant.
- Acquisition du matériel et des logiciels nécessaires au soutien de l'outil.

Les activités et les coûts récurrents sont les suivants :

- Frais récurrents de licence et de support.
- Frais de maintenance.
- Coûts de formation continue.
- Portage de l'outil dans différents environnements.

Les coûts d'opportunité doivent également être pris en compte. Cela signifie que le temps passé à évaluer, administrer, former et utiliser l'outil aurait pu être consacré aux tâches de test proprement dites.

Par conséquent, des ressources de test supplémentaires peuvent être nécessaires avant que l'outil puisse être utilisé pour les activités prévues.

Les risques suivants concernant le retour sur investissement doivent être pris en compte lors de la sélection des outils :

- L'immaturité de l'organisation peut conduire à une utilisation inefficace de l'outil.
- Des changements dans la politique de maintenance du fournisseur peuvent augmenter la charge de travail.
- Coûts plus élevés que prévu.
- Bénéfices moins importants que prévu.

Les avantages suivants peuvent s'appliquer aux outils de test :

- Réduction du travail manuel répétitif (par exemple, tests de régression).
- Accélération du cycle des tests grâce à l'automatisation.
- Réduction des coûts d'exécution du test par une diminution des activités manuelles.
- Augmentation de la couverture pour certains types de tests soutenus par l'outil.
- Réduction des erreurs humaines grâce à la diminution des activités manuelles.
- Accès plus rapide aux informations sur les tests.

Des avantages et des risques supplémentaires, en particulier pour les outils d'automatisation des tests, sont décrits dans le syllabus niveau Fondation version 4 et dans le syllabus Ingénieur en Automatisation des Tests de l'ISTQB®.

En général, une organisation de test utilise rarement un seul outil. Le retour sur investissement total pour une organisation est généralement un mélange du retour sur investissement de tous les outils utilisés pour tester. Les outils doivent partager des informations et travailler en coopération. Il est conseillé d'adopter une stratégie globale à long terme pour les outils de test, qui tienne compte des risques, des coûts et des avantages.

1.6.4 Cycle de vie d'un outil

Le cycle de vie d'un outil comporte quatre stades différents. Un administrateur d'outil doit être désigné pour veiller à ce que les activités de ces étapes soient définies, exécutées et gérées.

- **Acquisition :** Dans un premier temps, la décision a été prise de sélectionner un outil. La deuxième étape consiste à désigner un responsable (parfois appelé propriétaire) de l'outil. Cette personne prend les décisions relatives à l'utilisation de l'outil (par exemple, les conventions de nommage des produits d'activités et l'endroit où ces produits seront stockés). Prendre ces décisions dès le départ peut faire une différence significative dans le retour sur investissement de l'outil.
- **Support et maintenance** : Le responsable de l'outil est responsable de la maintenabilité de l'outil. La responsabilité des activités de maintenance devrait incomber à l'administrateur de l'outil ou à un groupe dédié aux outils. Dans le cas de l'interopérabilité, l'échange de données et les processus de coopération et de communication doivent être pris en compte. Des décisions sur la sauvegarde et la restauration des artefacts liés à l'outil sont également requises.

- **Evolution :** Au fil du temps, l'environnement, les exigences métier ou les décisions des fournisseurs peuvent nécessiter des modifications de l'outil. Plus l'environnement d'exploitation d'un outil devient complexe, plus un changement peut facilement perturber son utilisation.
- **Décommissionnement :** À la fin de sa durée de vie, l'outil doit être décommissionné. Dans la plupart des cas, les fonctionnalités fournies par l'outil seront remplacées et les données devront être préservées et/ou archivées. Cette décision peut être prise par le fournisseur ou parce qu'il est arrivé à un point où les avantages et les possibilités de passer à un nouvel outil dépassent ses coûts et ses risques.

1.6.5 Métriques des outils

Les métriques objectives des outils sont conçues et collectées en fonction des besoins de l'équipe de test et des autres parties prenantes. Les outils de test capturent la plupart du temps des données précieuses en temps réel et réduisent les efforts de collecte de données. Ces données sont utilisées pour gérer l'ensemble des tests et identifier les domaines à optimiser.

Différents outils se concentrent sur la collecte de différents types de données. En voici quelques exemples :

- Les outils de gestion des tests peuvent fournir une variété de métriques différentes liées aux éléments de test disponibles, aux tests, aux tests planifiés ainsi qu'à l'état actuel et passé de l'exécution du test (par exemple, réussi, échoué, ignoré, bloqué ou planifié).
- Les outils de gestion des exigences assurent la traçabilité de la couverture des exigences par les cas de test passés et échoués.
- Les outils de gestion des défauts peuvent fournir des informations sur les défauts telles que l'état, la sévérité, la priorité et la densité de défauts des éléments du test. D'autres données précieuses, telles que le pourcentage de détection des défauts, les niveaux de test auxquels les défauts sont introduits et le délai de détection des défauts contribuent à l'amélioration du processus, mais elles ne sont pas nécessairement toutes fournies uniquement par l'outil de gestion des défauts.
- Les outils d'analyse statique, entre autres, fournissent des métriques liées à la complexité du code.
- Les outils de test de performance peuvent fournir des informations précieuses telles que les temps de réponse et les taux de défaillance en période de pointe (de charge).
- Les outils de couverture du code aident à comprendre quelles parties de l'objet du test ont été exercées par les tests.
- Bien que les outils de test puissent être utilisés pour collecter des métriques, ils doivent également se suivre eux-mêmes. Dans ce contexte, la qualité du processus de test peut être mesurée (par exemple, le nombre de défauts constatés avec et sans outils et la couverture des exigences).
- L'efficience des tests (par exemple, la durée de l'exécution du test et le nombre de tests exécutés).

Plus de détails sur la collecte et l'utilisation des métriques peuvent être obtenus dans la section 2.1 de ce syllabus, Métriques de test.

2 Management du produit – 390 minutes

Mots clés

anomalie, défaut, rapport de défaut, cycle de vie des défauts, défaillance, métrique, estimation de test, objectif du test, avancement du test

Mots clés spécifiques au domaine

planning poker, estimation en trois points, Delphi à large bande

Objectifs d'apprentissage pour le chapitre 2 :

2.1 Métriques de test

TM-2.1.1 (K2) Donner des exemples de métriques permettant d'atteindre les objectifs du test.

TM-2.1.2 (K2) Expliquer comment contrôler l'avancement des tests à l'aide de métriques de test.

TM-2.1.3 (K4) Analyser les résultats des tests pour créer des rapports de test qui permettent aux parties prenantes de prendre des décisions.

2.2 Estimation des tests

TM-2.2.1 (K2) Expliquer les facteurs à prendre en compte dans l'estimation de test.

TM-2.2.2 (K2) Donnez des exemples de facteurs susceptibles d'influencer les estimations de test.

TM-2.2.3 (K4) Choisir une technique ou une approche appropriée pour l'estimation de test dans un contexte donné.

2.3 Gestion des défauts

TM-2.3.1 (K3) Implémenter un processus de gestion des défauts, y compris le cycle de vie des défauts, qui peut être utilisé pour suivre et contrôler les défauts.

TM-2.3.2 (K2) Expliquer le processus et les participants nécessaires à une gestion des défauts efficace.

TM-2.3.3 (K2) Expliquer les spécificités de la gestion des défauts dans le cadre du développement logiciel en mode Agile.

TM-2.3.4 (K2) Expliquer les défis de la gestion des défauts dans le développement de logiciels hybrides.

TM-2.3.5 (K3) Utiliser les données et les informations de classification qui devraient être recueillies lors de la gestion des défauts.

TM-2.3.6 (K2) Expliquer comment les statistiques des rapports de défaut peuvent être utilisées pour concevoir l'amélioration des processus.

2.1 Métriques de test

Introduction – Pourquoi des métriques de test ?

Dans le domaine du management, on dit que "ce qui est mesuré est fait". De même, ce qui n'est pas mesuré n'est pas susceptible d'être fait car il est facile de l'ignorer. Par conséquent, il est important d'établir un ensemble approprié de métriques pour toute tâche, y compris les tests.

Les objectifs du test sont la réponse à la question de savoir pourquoi nous testons (voir Section 1.4, La stratégie de test du projet). Pour déterminer si les objectifs du test ont été atteints, il faut définir un moyen de les mesurer. Les métriques de test sont les indicateurs qui nous aident à répondre à cette question.

Les métriques de test peuvent être classées comme suit :

- **Les métriques de projet** mesurent les progrès réalisés par rapport aux critères de sortie du projet, tels que le pourcentage de tests exécutés, réussis et échoués.
- **Les métriques de produit** mesurent les attributs du produit, tels que le degré auquel le produit répond aux attentes de qualité des utilisateurs prévus.
- **Les métriques de processus** mesurent la capacité du processus de test et l'efficacité des tests. Les métriques de processus sont donc utilisées pour rendre compte de l'efficacité et de l'efficience liées au processus.

Le lecteur trouvera plus d'informations sur la gestion des métriques de produit et de processus dans le syllabus ISTQB® de niveau Expert « Test Management » et sur les métriques de processus dans le syllabus ISTQB® de niveau Expert « Improving the Test Process ».

Les sections suivantes traitent des métriques pour la planification des tests, le suivi des tests, le contrôle des tests et la clôture des tests. Ce sont les quatre principales activités de management liées aux métriques.

2.1.1 Métriques pour les activités de gestion des tests

Le syllabus de niveau Avancé "Test Management" se concentre sur les activités génériques de gestion des tests suivantes :

- Planification des tests.
- Pilotage et contrôle des tests.
- Clôture des tests (voir section 1.1, Le processus de test).

Le management des tests doit être en mesure de définir un ensemble de métriques de test pour le suivi des tests, le contrôle des tests et la clôture des tests dans le cadre des activités de planification des tests. Chaque métrique doit être définie, mesurée, suivie et rapportée.

Lors de la planification des tests, les métriques de test appropriées sont définies en fonction des objectifs de test de la stratégie de test du projet.

Les métriques utilisées lors du pilotage des tests et du contrôle des tests peuvent être différentes de celles utilisées lors de la clôture des tests. Pendant le pilotage des tests et le contrôle des tests, les métriques concernent l'avancement des activités de test. Lors de la clôture des tests, les objectifs du test doivent être atteints. Une ou plusieurs métriques peuvent être combinées pour mesurer les critères de sortie du test assignés à des objectifs de test donnés.

Le tableau suivant donne des exemples de métriques (il y en a beaucoup d'autres) utilisées dans les activités de management des tests:

Métrique (planifiée/suivie pour des jalons définis)	**Pilotage des tests et contrôle des tests**	**Clôture des tests**
Couverture des exigences	X	X
Couverture des risques produit	X	X
Couverture du code	X	
Estimation réelle par rapport à la planification (en heures) pour les activités de test	X	
Pourcentage de cas de test exécutés par statut (par exemple, échec, blocage) par rapport aux cas de test planifiés	X	X
Nombre cumulé de défauts résolus par rapport au nombre cumulé de défauts	X	
Cas de tests automatisés réels par rapport aux cas de tests automatisés planifiés		X
Coût réel des tests par rapport au coût planifié	X	

Tableau 2: Exemples de mesures utilisées dans les activités de Management des Tests

La métrique utilisée dans une activité de test spécifique est indiquée dans le tableau. Les métriques avec un X dans le suivi et le contrôle des tests sont principalement utilisées pour mesurer l'avancement et sont rapportées dans les rapports d'avancement des tests. Les métriques avec un X dans la clôture des tests sont principalement utilisées pour mesurer l'atteinte des objectifs du test et sont rapportées dans le rapport de clôture des tests. Les métriques avec un X dans les deux colonnes peuvent être utilisées pour l'un ou l'autre.

Il existe également des métriques pour le suivi de l'efficacité des tests (par exemple, le pourcentage de détection des défauts (DDP)).

Le DDP est couvert dans le syllabus de niveau Expert de l'ISTQB®, Improving the Test Process, spécifiquement dans la section Implémentation de l'amélioration du processus de test.

2.1.2 Pilotage, contrôle et cloture

Les métriques de test sont des indicateurs qui montrent à quel point le test a progressé et si les critères de sortie ou les tâches de test correspondantes ont été atteints.

Le pilotage des tests est l'activité qui consiste à collecter des données concernant le test et l'évaluation associés. Il sert à évaluer l'avancement du test et à vérifier le respect des critères de sortie ou des activités de test associées (voir section 1.1.2, Suivi et contrôle des tests). Les critères de sortie sont dérivés des objectifs du test.

Le contrôle des tests utilise les informations issues du suivi des tests pour fournir des directives et des actions correctives afin de tester de manière efficace et efficiente. Parmi les exemples de directives de

contrôle des tests, on peut citer la redéfinition des priorités des tests lorsqu'un risque identifié devient un problème, en réévaluant la conformité d'un élément de test aux critères d'entrée ou aux critères de sortie en raison d'un remaniement ; en ajustant le calendrier des tests pour tenir compte d'un retard dans la livraison de l'environnement de test ; et en ajoutant de nouvelles ressources lorsque cela s'avère nécessaire.

La clôture des tests recueille les données des activités de test achevées afin de consolider les leçons apprises, les testware et d'autres informations pertinentes. La clôture des tests intervient à des jalons du projet tels que la clôture d'un niveau de test, la clôture d'une itération, la clôture (ou l'annulation) d'un projet de test, la sortie d'un produit ou la clôture d'une version de maintenance.

Les métriques de test couramment utilisées dans les activités de management des tests comprennent les métriques d'avancement du projet et les métriques qui montrent l'avancement par rapport au calendrier et au budget prévus, la qualité actuelle des éléments du test, et l'efficacité des tests par rapport aux objectifs du test ou aux objectifs de l'itération.

2.1.3 Rapports de test

Le Management des Tests devrait comprendre comment interpréter et utiliser les métriques pour comprendre et rapporter l'état des tests. Pour les niveaux de test supérieurs tels que les tests système, les tests d'intégration de système, les tests d'acceptation et les tests de sécurité ; la base de test principale est généralement constituée par les produits d'activités tels que les spécifications des exigences, les cas d'utilisation, les User Stories et les risques produits. Les métriques de couverture structurelle sont plus applicables aux niveaux de test inférieurs tels que les tests de composants (par exemple, la couverture des instructions) et les tests d'intégration des composants (par exemple, la couverture des interfaces). Alors que le Management des Tests peut utiliser des métriques de couverture de code pour mesurer combien leurs tests exercent la structure du système sous test, le reporting des résultats des tests de plus haut niveau devrait être adapté au contexte et aux besoins spécifiques du projet. Par exemple, dans les environnements qui changent fréquemment, les métriques de couverture de code peuvent être utiles pour suivre l'impact des changements de code sur la suite de tests et identifier les lacunes ou les risques potentiels. En outre, le Management des Tests doit comprendre que même si les tests de composants et les tests d'intégration des composants atteignent 100 % de leur couverture structurelle, des défauts et des risques de qualité restent à traiter à des niveaux de test plus élevés.

L'objectif du reporting des métriques est de fournir une facilité de compréhension immédiate de l'information à des fins de gestion. Les métriques peuvent être présentées sous la forme d'un instantané d'une métrique à un moment donné ou sous la forme de l'évolution d'une métrique dans le temps afin d'évaluer les tendances.

Les risques produits, les défauts, l'avancement des tests, la couverture et les coûts associés ainsi que l'effort de test sont mesurés et rapportés de manière spécifique à la fin du projet.

Voici des exemples de métriques qui peuvent être utilisées à différentes fins :

Les métriques relatives aux risques produits comprennent :

- Pourcentage de risques dont tous les tests ont été réussis.
- Pourcentage de risques dont une partie ou la totalité des tests ont échoué.
- Pourcentage de risques qui n'ont pas encore été complètement testé.

Ces métriques peuvent être utilisées pour évaluer la qualité de la base de test et l'efficacité des cas de test pour couvrir les risques liés au produit.

Les mesures relatives aux défauts comprennent :

- Nombre cumulé de défauts résolus par rapport au nombre cumulé de défauts
- Ventilation du nombre ou du pourcentage de défauts catégorisés par :
    - Éléments du test ou composants
    - Source du défaut (par exemple, spécification d'une exigence, nouvelle fonctionnalité ou régression)
    - Version testée
    - Niveau de test ou itération introduite, détectée et supprimée
    - Priorité/sévérité
    - Cause racine
    - Statut (par exemple, rejeté, dupliqué, ouvert, fermé)

Ces métriques peuvent être utilisées pour suivre le processus de détection et de résolution des défauts, identifier les zones de forte densité de défauts ou de sévérité des défauts, et évaluer l'efficience et l'efficacité des tests.

Les métriques relatives à l'avancement des tests comprennent :

- État de l'exécution des tests : Nombre total de tests planifiés, implémentés, exécutés, réussis, échoués, bloqués et ignorés.
- Effort de test : Nombre d'heures de ressources réelles par rapport au nombre d'heures planifiées consacrées aux tests.

Les métriques relatives à la couverture comprennent :

- Couverture des exigences : Pourcentage d'exigences couvertes par des cas de test.
- Couverture des risques produits : Pourcentage de risques produits identifiés qui sont atténués par les cas de test.
- Couverture du code : Pourcentage d'instructions, de branches, de chemins ou de conditions du code qui sont exécutés par les cas de test.

Les métriques liées aux coûts et à l'effort de test comprennent :

- Risques résiduels pour les composants non testés : L'impact potentiel et la probabilité des défauts dans les composants qui ne sont pas testés.
- Coût des tests : Le coût réel des tests par rapport au coût planifié.

En outre, il est utile de combiner des métriques de différentes catégories (par exemple, une métrique qui montre la corrélation entre les tendances des défauts ouverts et les tendances des tests exécutés, ou une métrique qui montre la qualité de la base de test sur la base du nombre de défauts constatés dans les exigences). Lorsque l'exécution des tests se poursuit et que de moins en moins de défauts sont identifiés, il est possible de prendre la décision de mettre fin aux tests. Cette décision doit être basée sur les rapports des métriques et les critères de sortie convenus.

2.2 Estimation de test

Introduction

Il existe des meilleures pratiques de gestion de projet pour l'estimation de l'ingénierie des systèmes et des logiciels, concernant tous les types de ressources (par exemple, le coût, les personnes ou le temps). L'estimation de test est l'application de ces meilleures pratiques aux tests associés à un projet ou à une exploitation.

2.2.1 Estimation des activités qu'impliqueront les tests

L'estimation des tests est une activité de management des tests qui permet d'estimer le temps, les efforts et les coûts nécessaires à la clôture des tests. L'estimation de test est l'une des tâches majeures et importantes du management des tests.

Les principales caractéristiques de l'estimation dans le management des tests sont :

- L'**effort** est généralement calculé en personnes-heures ou en story points nécessaires pour terminer les tâches de test du projet. Souvent, l'effort de test et la durée du test (temps écoulé) peuvent être différents, et le management des tests peut avoir besoin d'estimer la durée totale de l'activité. Combien de personnes-heures seront nécessaires ?
- Le **temps** nécessaire pour terminer le projet. Le temps est une ressource critique dans un projet. La planification des tests doit estimer l'effort de test en jours calendaires et en jours ouvrables. Chaque projet comporte des jalons et une date limite de livraison. Combien de temps faudra-t-il pour terminer le projet de test ?
- Le **coût** est le budget du projet. Il comprend les dépenses pour les ressources, les outils et l'infrastructure de test. Quel sera le coût du projet de test?

Le test est souvent un sous-projet au sein d'un (grand) projet, parfois réparti sur plusieurs sites de test (par exemple, des centres de test). Pour réaliser l'estimation de test, la première étape consiste à identifier les niveaux de test, les activités de test et les tâches de test. Ensuite, il faut diviser le projet de test en ses principales activités de test (par exemple, la planification des tests et l'exécution des tests) au sein du processus de test (voir le syllabus de niveau Fondation de l'ISTQB® V.4). Dans les projets en mode Agile, les activités de test sont souvent estimées au sein du travail de développement, et non comme des valeurs distinctes. L'étape suivante consiste à estimer l'effort de test nécessaire pour terminer les tâches ou les produits d'activités et les coûts qui en résultent.

Parce que tester est une sous-activité d'un projet, il y a toujours des contraintes naturelles du projet qui l'influencent et nécessitent des compromis, et nous ne pouvons pas manipuler ces valeurs de manière arbitraire. Cela peut être constaté dans la gestion de la qualité sous la forme du triangle temps-coût-qualité. Dans la gestion de projet, le triangle temps-coût-qualité comprend trois valeurs qui sont interdépendantes, ce qui signifie qu'elles sont étroitement liées et s'influencent mutuellement. Cette relation est couramment observée dans les scénarios de projet.

2.2.2 Facteurs susceptibles d'influer sur l'effort de test

L'estimation de l'effort de test consiste à prédire la quantité de travail liée aux activités de test qui sera nécessaire pour atteindre les objectifs du test dans le cadre d'un projet, d'une release ou d'une itération donnés. Les facteurs qui influencent l'effort de test peuvent inclure les caractéristiques du projet, de la version ou de l'itération :

Produit :

- La qualité de la base de test.
- La taille du produit à tester (c'est-à-dire l'objet du test).
- La complexité du domaine du produit (par exemple, l'environnement, l'infrastructure et l'historique).
- Les exigences relatives au test des caractéristiques de qualité (par exemple, la sécurité et la fiabilité).

Ces facteurs liés au produit peuvent influencer les estimations des tests car ils créent un contexte spécifique au système sous test.

Processus de développement :

- La stabilité et la maturité des processus de développement de l'organisation.
- Le modèle de développement (par exemple, modèle de développement logiciel Agile/itératif, ou modèle de développement logiciel hybride) en vigueur.
- Les facteurs matériels (par exemple, la disponibilité de l'automatisation des tests, des outils et des environnements de test).

Ces facteurs liés au processus de développement peuvent influencer les estimations de test car les tests sont directement liés au développement.

Personnes :

- La satisfaction des personnes (par exemple, les jours fériés, les vacances, d'autres avantages attendus)
- Les compétences et l'expérience des personnes impliquées, notamment en ce qui concerne des projets et des produits similaires (par exemple, la connaissance du domaine).

Les personnes sont les ressources les plus nécessaires, et toute instabilité doit donc être prise en compte. Par conséquent, les personnes sont un facteur important dans l'estimation de l'effort de test. Voir également la section 3.1 de ce syllabus, L'équipe de test.

Résultats de test :

- Le nombre et la sévérité des défauts constatés lors de l'exécution du test
- La quantité de travail de reprise exigée.

Les statistiques historiques soutiennent l'estimation des tests. Ainsi, la connaissance de ces facteurs permettra une estimation avec des valeurs plus précises.

Contexte du test :

- La répartition des tests sur plusieurs filiales, la composition et la localisation des équipes, la complexité du projet (par exemple, plusieurs sous-systèmes).
- Le type de travail (par exemple, virtuel ou sur site).

Les facteurs liés au contexte sont ceux qui affectent l'ensemble de l'estimation de test. Voir aussi la section 1.2. de ce syllabus, Le contexte des tests.

2.2.3 Sélection des techniques d'estimation de test

L'estimation de test devrait couvrir toutes les activités impliquées dans le processus de test. L'estimation du coût, de l'effort et, en particulier, de la durée de l'exécution du test est souvent la plus importante pour le Management des Tests car ces valeurs auront un impact sur le projet. Cependant, les estimations concernant l'exécution des tests ont tendance à être difficiles à générer lorsque la qualité globale du logiciel est faible ou inconnue. De plus, la familiarité et l'expérience avec le produit affecteront probablement la qualité des estimations. Une pratique courante consiste à estimer le nombre de cas de test dérivés de la base de test (par exemple, les exigences ou les User Stories). Les hypothèses formulées lors de l'estimation de test devraient toujours être documentées dans le cadre de l'estimation.

Les techniques ou approches d'estimation de test peuvent être classées en deux catégories : celles basées sur des métriques et celles basées sur l'expertise.

De plus amples détails sur les techniques d'estimation de test sont expliqués dans le Syllabus V.4 du niveau Fondation de l'ISTQB®.

Dans la plupart des cas, l'estimation, une fois préparée, doit être remise au management du projet, accompagnée d'une justification. Il arrive fréquemment que certains paramètres d'entrée soient modifiés (par exemple, le périmètre du test), ce qui entraîne souvent un ajustement de l'estimation. Idéalement, l'estimation de test finale représente le meilleur équilibre possible entre les objectifs de l'organisation et ceux du projet dans les domaines de la qualité, du calendrier, du budget et des fonctionnalités.

Il est important de garder à l'esprit que toute estimation est basée sur les informations disponibles au moment où elle est préparée. Au début d'un projet, les informations peuvent être assez limitées. En outre, ces informations peuvent changer au fil du temps. Pour rester précises, les estimations doivent être mises à jour afin de refléter les nouvelles informations et les changements.

Le choix de la technique d'estimation dépend de plusieurs facteurs, tels que :

- **Erreur d'estimation :** Certaines techniques permettent de calculer l'écart type, qui est une mesure de l'incertitude ou de la variabilité de l'estimation. Par exemple, la technique d'estimation en trois points utilise les estimations optimiste, pessimiste et la plus probable pour calculer la valeur attendue et l'écart-type de l'estimation (voir le syllabus ISTQB® V.4, de niveau Fondation section 5.1.4 pour plus de détails).
- **Disponibilité des données :** Certaines techniques exigent des données historiques provenant de projets antérieurs ou similaires, qui peuvent ne pas être disponibles ou fiables. Par exemple, l'estimation basée sur des ratios et l'extrapolation s'appuient sur des données historiques pour dériver les ratios ou les tendances pour le projet en cours.
- **Disponibilité des experts :** Certaines techniques nécessitent la participation d'experts qui possèdent les connaissances et l'expérience nécessaires pour fournir des estimations précises et réalistes. Par exemple, la méthode Delphi et le planning poker s'appuient sur les opinions et les jugements d'experts ou de membres de l'équipe.

- **Connaissances en matière de modélisation :** Certaines techniques exigent l'utilisation de modèles ou de formules mathématiques pour calculer les estimations, ce qui peut nécessiter certaines compétences et connaissances en modélisation. Par exemple, l'extrapolation et l'estimation en trois points utilisent des formules pour calculer la valeur attendue et l'écart-type de l'estimation.
- **Contraintes de calendrier : La mise en œuvre de** certaines techniques nécessite plus de temps et d'efforts que d'autres, ce qui peut affecter la faisabilité et l'adéquation de la technique. Par exemple, le planning poker est facile à réaliser, alors que l'extrapolation peut être plus difficile.

Ceci montre que les critères de sélection des techniques d'estimation des tests sont fortement dépendants du contexte des tests (par exemple, le SDLC, les parties prenantes, les niveaux de test et les types de test utilisés dans le projet) (voir Section 1.2 de ce syllabus, Le contexte des tests). Le Test Manager doit être capable de coordonner et d'appliquer les techniques d'estimation de test (par exemple, avec différents modèles de SDLC dans un projet dans différentes filiales).

Par exemple, pour choisir la bonne technique d'estimation, il faut d'abord déterminer la complexité du sujet. Si la complexité est faible, des techniques basées sur des métriques peuvent être utilisées. Si la complexité est élevée, des techniques basées sur l'expertise peuvent être utilisées. Si un modèle de développement séquentiel est utilisé, la technique d'estimation Delphi à large bande peut être utilisée. Si un modèle de développement logiciel en mode Agile est utilisé, alors le planning poker pourrait être utilisé.

2.3 Gestion des défauts

Introduction

Le syllabus V.4 du niveau Fondation de l'ISTQB® décrit les activités qui commencent après l'observation de résultats réels qui diffèrent des résultats attendus. Le syllabus désigne ces activités par le terme de "gestion des défauts". D'autres normes utilisent le terme "gestion des incidents" (norme ISO/IEC/IEEE 29119-3) ou "gestion des anomalies" (TMAP) pour souligner le fait qu'au début du processus, nous pouvons ne pas savoir si l'écart est dû à un défaut dans un produit d'activités, ou à quelque chose d'autre (par exemple, une défaillance de l'automatisation des tests ou une mauvaise compréhension des exigences par le testeur). La gestion des défauts et l'outil utilisé pour gérer les défauts sont d'une importance cruciale pour les testeurs et les autres membres de l'équipe impliqués dans le développement du logiciel. Les informations issues d'un processus de gestion des défauts efficace permettent à l'équipe de test et aux autres parties prenantes du projet de se faire une idée de l'état d'un projet tout au long de son cycle de développement. La gestion des défauts est également cruciale pour décider quels défauts seront corrigés. Cela permet de s'assurer que les efforts sont consacrés à travailler sur les bons défauts. La collecte et l'analyse des données relatives aux défauts au fil du temps peuvent aider à localiser les domaines d'amélioration potentielle à la fois pour les tests et pour d'autres processus du cycle de développement durable (par exemple, une meilleure prévention des défauts grâce à une architecture et une conception technique améliorées).

En plus de comprendre le cycle de vie du défaut et comment il est utilisé pour surveiller et contrôler les processus de développement logiciel et de test, le Test Manager et les testeurs (ou l'ensemble de l'équipe Agile dans le cadre du développement logiciel en mode Agile) doivent également être familiers avec les données critiques à capturer. Le Test Manager doit être un défenseur de la bonne utilisation du processus de gestion des défauts et de l'outil de gestion des défauts sélectionné.

2.3.1 Cycle de vie des défauts

Chaque phase du cycle de développement logiciel devrait comporter des activités visant à détecter et à éliminer les défauts potentiels. Par exemple, les techniques de test statique (c'est-à-dire les revues et l'analyse statique) peuvent être utilisées sur les spécifications de la conception, les spécifications des exigences et le code avant de livrer ces produits d'activités dans les activités ultérieures. Plus chaque défaut est détecté et éliminé rapidement, plus le coût de qualité global du produit est faible. Le coût de la qualité est minimisé lorsque chaque défaut est éliminé au cours de la même phase que celle où il a été introduit (c'est-à-dire lorsque le processus logiciel atteint un confinement de phase parfait).

Lors des tests statiques, on recherche les défauts. Lors des tests dynamiques, la présence d'un défaut est révélée lorsqu'il provoque une défaillance, ce qui se traduit par une divergence entre les résultats réels et les résultats attendus d'un test (c'est-à-dire une anomalie). Dans certains cas, un résultat faux négatif se produit lorsque le testeur n'observe pas l'anomalie. Lorsqu'une anomalie est observée, il convient de mener une enquête plus approfondie. Cette investigation commence généralement par la création d'un rapport de défaut conformément au processus de gestion des tests et des défauts définis. Un test échoué ne donne pas toujours lieu à la création d'un rapport de défaut. (Par exemple, dans le développement piloté par les tests, où les tests des composants, généralement automatisés, sont utilisés comme une forme de spécification de conception exécutable). Tant que le développement du composant n'est pas achevé, une partie ou la totalité des tests doit initialement échouer. Par conséquent, le résultat d'un tel test n'est pas nécessairement causé par un défaut et n'est généralement pas testé par le biais d'un rapport de défaut.

Un rapport de défaut progresse le long d'un cycle de vie (NDT : workflow) (par souci de simplicité et de cohérence avec la plupart des outils de gestion des défauts, nous continuerons à utiliser le terme "cycle de vie des défauts") et passe par une séquence d'états de défaut. Dans la plupart de ces états, une personne est "propriétaire" du rapport de défaut et est responsable de l'exécution d'une tâche (par exemple, l'analyse, la suppression du défaut ou le test de confirmation). Le diagramme suivant représente cycle de vie des défauts simple :

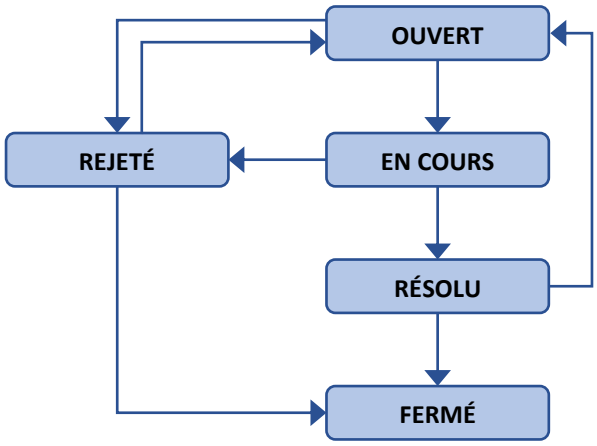

Figure 2 : Un cycle de vie simple pour les défauts

Un cycle de vie simple peut couvrir les états de défaut suivants :

- OUVERT (peut être appelé NOUVEAU) : L'état initial lorsque le rapport de défaut est créé.
- EN COURS : L'équipe travaille sur l'analyse et/ou la correction du rapport de défaut.
- REJETÉ : Un rapport de défaut est rejeté par la personne qui l'a traité (généralement un développeur ou un analyste). Les raisons du rejet peuvent être nombreuses (par exemple, informations non valides, test incorrect, rapport de défaut en double) et ces informations sont ajoutées au rapport de défaut.
- RÉSOLU (peut être appelé CORRIGÉ, PRÊT POUR UN RE-TEST) : Un testeur exécute un test de confirmation en suivant souvent les étapes de reproduction de la défaillance à partir du rapport de défaut lui-même afin de déterminer si la correction a bien permis de tester le défaut.
- FERMÉ : Le rapport de défaut a atteint son état terminal et aucun travail supplémentaire n'est prévu. Le testeur fait passer le rapport de défaut dans cet état, soit après un test de confirmation réussi, soit pour confirmer le rejet du rapport de défaut.

Un cycle de vie des défauts simple est utilisé dans de nombreuses organisations et est étendu par l'utilisation d'autres états de défauts pertinents pour un contexte donné (par exemple, RÉOUVERT, ACCEPTÉ, CLARIFICATION ou REPORTÉ).

Le cycle de vie des défauts peut varier d'une organisation à l'autre en ce qui concerne les noms des états de défaut, les règles de transition entre les états de défaut et les rôles responsables des tâches dans des états de défaut donnés. Souvent, le cycle de vie des défauts est plus simple dans le développement logiciel en mode Agile que dans les modèles de développement séquentiels. Le cycle de vie des défauts

doit être adapté à un contexte donné. Lors de la conception du cycle de vie des défauts, il est conseillé de respecter plusieurs bonnes pratiques :

- Si possible, le cycle de vie des défauts doit être défini à l'échelle de l'organisation afin d'assurer une gestion des défauts unifiée dans tous les projets.
- Les défauts dupliqués et les faux positifs doivent être représentés par un état distinct ou une combinaison de l'état REJETÉ avec le choix de la raison du rejet. Ils peuvent être utiles lors d'analyses de défauts plus poussées dans le but d'améliorer le processus de test.
- Il est recommandé de n'utiliser qu'un seul état terminal (par exemple, FERMÉ). Le passage à cet état exige souvent de choisir une raison de fermeture, utile pour l'évaluation du processus et les activités d'amélioration du processus.
- Les noms des états dans le cycle de vie des défauts doivent être les mêmes que ceux des états analogues utilisés pour d'autres entités (par exemple, les User Stories et les tâches de test) afin de simplifier leur utilisation.
- Les états de défauts consécutifs doivent appartenir à des rôles de responsabilité différents. Si deux états consécutifs ou plus appartiennent au même rôle responsable, il doit y avoir une bonne raison (par exemple, pour mesurer le temps passé dans un état de défaut).
- Chaque état de défaut, à l'exception de l'état final, doit avoir plus d'une transition sortante pour permettre au rôle responsable de prendre une décision concernant l'étape suivante. Les exceptions à cette règle doivent être justifiées (par exemple, pour contrôler le temps consacré à une activité donnée).
- L'ensemble des exigences à saisir lors de l'exécution d'une transition d'état doit être limité à celles qui confèrent une valeur substantielle à la gestion des défauts.

2.3.2 Gestion cross-fonctionnelle des défauts

Bien que l'organisation de test et le Test Manager soient souvent propriétaires du processus global de gestion des défauts et de l'outil de gestion des défauts, une équipe cross-fonctionnelle est généralement responsable de la gestion des défauts pour un projet donné. Cette équipe, parfois appelée comité de gestion des défauts, peut comprendre le Test Manager, des représentants du développement, des fournisseurs, des représentants de la gestion de projet, de la gestion de produit ou du Product Owner et d'autres parties prenantes qui ont un intérêt dans le logiciel testé.

Au fur et à mesure que des anomalies sont découvertes et saisies dans l'outil de gestion des défauts, le comité de gestion des défauts doit déterminer si chaque rapport de défaut représente un défaut valable et s'il doit être corrigé (et par quelle partie dans le cas où plusieurs équipes de développement participent à la validation), rejeté ou reporté. Pour prendre cette décision, le comité de gestion des défauts doit tenir compte des avantages, des risques et des coûts associés à la correction du défaut. Il est utile de discuter de ces considérations lors d'une réunion (souvent appelée réunion de triage). Si le défaut doit être corrigé, l'équipe doit établir la priorité de la correction du défaut par rapport aux autres tâches. Le Test Manager et l'équipe de test peuvent être consultés sur l'importance relative d'un défaut et doivent fournir les informations objectives disponibles.

Dans le cas de très grands projets, la nomination d'un responsable des défauts à temps plein peut se justifier par les efforts nécessaires à la préparation et au suivi des décisions prises lors des réunions du comité de gestion des défauts, au moins pendant les phases du SDLC où les tests sont les plus intensifs. Dans d'autres situations, plusieurs grands projets peuvent se partager un responsable des défauts.

Un outil de gestion des défauts ne doit pas se substituer à une bonne communication, pas plus qu'un comité de gestion des défauts ne doit se substituer à l'utilisation efficace d'un bon outil de gestion des défauts. La communication, un soutien adéquat de l'outil, un cycle de vie des défauts bien défini (y compris les propriétés des rapports de défauts) et une équipe de gestion des défauts engagée sont tous nécessaires pour une gestion des défauts efficace et efficiente.

2.3.3 Particularités de la gestion des défauts dans les équipes en mode Agile

La gestion des défauts dans les organisations utilisant le développement logiciel en mode Agile est souvent légère et/ou moins formelle que dans les modèles de développement séquentiel. Si les équipes en mode Agile sont colocalisées ou disposent de moyens de communication bien établis, les informations sur un défaut ou une défaillance sont souvent échangées entre les testeurs, les représentants des clients et les développeurs sans qu'un rapport de défaut formel ne soit établi. Des rapports de défaut devraient cependant être créés pour :

- Les défauts qui bloquent d'autres activités du sprint en cours (c'est-à-dire le développement, les tests ou autres) et qui ne peuvent pas être corrigés immédiatement au sein de l'équipe Agile.
- Les défauts qui ne peuvent pas être résolus au cours de la même itération. Certaines équipes en mode Agile ont pour règle de créer un rapport de défaut si le défaut ne peut être résolu au cours de la journée où la défaillance a été constatée.
- Les défauts qui doivent être résolus par ou en coopération avec d'autres équipes dans les organisations multi-équipes.
- Les défauts qui doivent être résolus par un fournisseur.
- Les défauts pour lesquels un rapport de défaut est explicitement demandé (par exemple, lorsqu'un développeur ne peut pas travailler immédiatement sur un correctif).

La pratique courante consiste à ajouter les défauts qui ne peuvent pas être résolus dans la même itération au backlog du produit afin qu'ils puissent être classés par ordre de priorité parmi d'autres défauts et User Stories pour une itération ultérieure.

Bien que les fondements de la gestion des défauts doivent être établis dans la stratégie de test d'une organisation, de nombreux aspects, y compris le niveau de formalité, les déclencheurs de la création d'un rapport de défaut et les attributs de défaut à capturer, peuvent être laissés à l'appréciation des membres de l'équipe en mode Agile. En général, le niveau de formalité de la gestion des défauts et l'approche de la création de rapports de défauts devraient refléter ce qui suit :

- Colocalisation des membres de l'équipe.
- Répartition des membres de l'équipe entre les différents fuseaux horaires.
- Le nombre d'équipes qui coopèrent au développement du produit.
- Maturité de l'équipe ou des équipes.
- Taille de l'équipe ou des équipes.
- Risques associés au produit.
- Les exigences réglementaires, contractuelles ou autres (le cas échéant).

La décision finale de l'équipe en mode Agile concernant les détails de la gestion des défauts doit toujours être documentée (par exemple, avec des lignes directrices dans un outil de gestion des connaissances).

2.3.4 Défis liés à la gestion des défauts dans le développement de logiciels hybride

Dans la pratique, plusieurs équipes collaborent souvent à la livraison du système ou du système de systèmes. Il s'agit par exemple du développement logiciel hybride, lorsqu'un client utilise le développement logiciel en mode Agile et qu'un de ses fournisseurs utilise un modèle de développement séquentiel, ou lorsqu'une organisation utilisant un modèle de développement séquentiel exige la livraison d'un sous-système de la part d'une équipe utilisant le développement logiciel en mode Agile. Un tel environnement multi-équipes pose plusieurs défis :

- **Alignement sur les attributs des défauts et les outils à utiliser pour la gestion des défauts** : Dans un scénario idéal, toutes les équipes utilisent un seul outil de gestion des défauts. Dans la pratique, il est courant que chaque équipe utilise un outil de gestion des défauts différent, en particulier lorsque plusieurs équipes de fournisseurs contribuent à la réalisation du projet. Dans ce cas, il est bon d'établir une synchronisation entre les outils de gestion des défauts (de préférence automatiquement).
- **Priorisation des défauts** : Le(s) Product Owner(s) devrait(ent) être inclus dans les réunions de gestion des défauts et rechercher activement des informations sur les conséquences et les risques associés aux défauts. Les réunions de gestion des défauts devraient avoir lieu plus souvent dans le cadre du développement logiciel en mode Agile que dans celui des modèles de développement séquentiel, afin de suivre le rythme plus rapide de livraison d'incréments de produit de l'équipe en mode Agile. Ces réunions peuvent toutefois être plus courtes avec les équipes en mode Agile. Il est parfois bénéfique qu'un groupe plus restreint de parties prenantes à la gestion des défauts ait le dernier mot sur la hiérarchisation des défauts.
- **Alignement et transparence du plan de test pour les nouveaux développements et les corrections de défauts** : Le travail de toutes les équipes doit s'aligner sur le même plan de projet, qu'elles utilisent un modèle de développement logiciel en mode Agile ou un modèle de développement séquentiel. Tous les livrables, y compris les corrections de défauts, devraient être alignés sur ce plan de projet. Un meilleur alignement peut être obtenu par la participation active des membres de toutes les équipes au processus de planification (par exemple, la participation des équipes du modèle de développement séquentiel aux réunions de développement logiciel en mode Agile au cours desquelles les défauts sont discutés et classés par ordre de priorité). La transparence des plans de développement peut être améliorée en les partageant entre les équipes (par exemple, via des tableaux de bord, ou via le Backlog du produit).

2.3.5 Informations des rapports de défaut

Les informations contenues dans un rapport de défaut doivent être suffisantes pour répondre aux objectifs suivants :

- Gestion du rapport des défauts tout au long du cycle de vie du défaut.
- Evaluation de l'état général du projet, notamment en termes de qualité du produit et de progression des tests
- Evaluation de l'état d'un incrément de produit en termes de qualité du produit
- Evaluation du processus.

Les informations nécessaires à la gestion des défauts et à l'état d'avancement du projet peuvent varier en fonction du moment où le défaut est détecté au cours du cycle de développement. En outre, les rapports de défauts liés à des caractéristiques de qualité non fonctionnelles peuvent nécessiter davantage

d'informations (par exemple, les conditions de charge pour les problèmes de performance). Toutefois, les principales informations recueillies doivent être cohérentes tout au long du cycle de développement et, idéalement, dans tous les projets d'une organisation, afin de permettre une comparaison pertinente des données relatives aux défauts tout au long du projet et dans l'ensemble des projets.

De nombreuses données peuvent être collectées dans un rapport de défaut. Le Test Manager doit décider quelles informations sont appropriées pour une gestion des défauts efficace dans un contexte de projet donné. Étant donné que chaque attribut supplémentaire augmente le temps consacré au rapport de défauts et peut accroître la confusion de la personne qui saisit le rapport de défaut, il est conseillé de ne collecter que les données qui sont nécessaires à la gestion des défauts dans le contexte donné et/ou qui seront utilisées pour l'amélioration du processus.

Pour gérer un rapport de défaut dans la plupart des environnements, les éléments suivants sont obligatoires :

- Un titre de défaut avec un bref résumé de l'anomalie.
- Une description détaillée de l'anomalie comprenant de préférence les étapes pour reproduire la défaillance.
- La sévérité de l'impact sur le système sous test et/ou les parties prenantes du produit.
- La priorité de correction de l'anomalie.

D'autres données importantes sont souvent créées par l'outil de gestion des défauts :

- Identifiant unique du rapport de défaut.
- Date/heure de création du rapport de défaut.
- Nom de la personne qui a découvert et/ou signalé l'anomalie.
- Projet et phase du SDLC au cours de laquelle l'anomalie a été découverte.
- État actuel du rapport de défaut.
- "Propriétaire" actuel (c'est-à-dire la personne actuellement chargée de travailler sur le défaut).
- Historique des modifications, c'est-à-dire la séquence des actions, y compris les informations relatives à la date et à l'heure, entreprises par les membres de l'équipe de projet pour isoler, réparer et confirmer que le défaut a été corrigé.
- Références (par exemple, au cas de test, aux défauts connectés).

Selon le contexte, d'autres informations (par exemple, la traçabilité) peuvent également être recueillies dans un rapport de défaut (voir la norme ISO/IEC/IEEE 29119-3 pour plus d'informations). Les points suivants regroupent les informations en fonction de l'objectif visé :

- **Aider à la résolution des défauts** : Le sous-système ou le composant dans lequel se situe le défaut, l'élément du test spécifique et son numéro de version dans lequel l'anomalie a été observée ou l'environnement de test dans lequel le défaut a été observé.
- **Evaluer l'état général du projet** : Informations permettant de suivre les progrès réalisés (par exemple, risques, coûts, opportunités et avantages liés à la correction ou à la non-réparation du défaut, description de toute solution de contournement disponible ou exigences affectées par les défauts).

- **Evaluer l'état d'un incrément de produit en termes de qualité** : Le type de défaut (correspondant généralement à une taxonomie des défauts), le produit d'activités dans lequel le défaut a été introduit, ou la caractéristique/sous-caractéristique de qualité affectée par le défaut.
- **Evaluer le processus** : Les informations permettant de contrôler l'efficacité et l'efficience des processus de développement (par exemple, la phase SDLC d'introduction, de détection et de suppression du défaut ou de la cause racine du défaut).

2.3.6 Définition des actions d'amélioration du processus à l'aide des rapports de défaut

Comme discuté dans la section 2.3.5 de ce syllabus, "Information sur les rapports de défauts", les rapports de défauts peuvent être utiles pour le suivi de l'état du projet et le reporting. Alors que les implications des métriques sur le processus de test sont principalement abordées dans le syllabus de niveau Expert "Management des Tests", au niveau Avancé " Management des Tests", les Test Managers devraient être conscients de ce que les rapports de défauts signifient pour l'évaluation de la capacité des processus de développement et de test de logiciels.

En plus des informations de suivi de l'avancement des tests mentionnées dans ce syllabus, dans la section 2.1.2, Pilotage, contrôle et clôture, et dans la section 2.1.3, Rapports de tests, les informations sur les défauts devraient soutenir les initiatives d'amélioration du processus comme discuté pendant les rétrospectives. Les exemples incluent :

- L'utilisation des informations relatives aux phases d'introduction, de détection et d'élimination des défauts afin d'évaluer le confinement de phase et/ou d'effectuer une analyse du coût de la qualité dans le but de suggérer des moyens d'améliorer l'efficacité de la détection des défauts dans chaque phase et de minimiser le coût associé aux défauts.
- D'utiliser les informations relatives à la phase d'introduction pour analyser les phases au cours desquelles le plus grand nombre de défauts sont introduits, afin de permettre des améliorations ciblées en matière de prévention des défauts.
- L'utilisation des informations sur la cause racine des défauts pour déterminer les raisons sous-jacentes de l'introduction des défauts, afin de permettre des améliorations du processus qui réduisent le nombre total de défauts.
- D'utiliser les informations relatives à la localisation des défauts pour effectuer une analyse des risques, afin de mieux comprendre les risques techniques (pour les tests basés sur les risques) et de permettre le remaniement des composants qui posent problème.
- L'utilisation d'informations sur les défauts ré-ouverts afin d'auditer la qualité des implémentations de débogage.
- D'utiliser les informations sur les défauts dupliqués et rejetés pour évaluer la qualité de la création des rapports de défauts.
- De permettre des améliorations de processus qui réduisent le nombre total de défauts en introduisant des mesures proactives pour éviter les erreurs en amont.

L'utilisation de métriques pour évaluer l'efficacité et l'efficience du processus de test est abordée dans le syllabus de niveau Expert "Test Management".

Dans certains cas, les équipes décident de ne pas suivre les défauts constatés pendant certaines ou toutes les phases du SDLC. Bien que cette décision soit souvent prise au nom de l'efficience et dans le but de réduire les frais généraux du processus, elle réduit considérablement la visibilité sur les capacités

du processus de développement et de test de logiciels. Cela rend les améliorations suggérées ci-dessus difficiles à mettre en œuvre en raison d'un manque de données de soutien fiables.

3 Management de l'équipe – 225 minutes

Mots clés

évaluation, coût de la qualité, défaut, prévention des défauts, défaillance externe, défaillance interne

Objectifs d'apprentissage pour le chapitre 3:

3.1 L'équipe de test

TM-3.1.1 (K2) Donner des exemples de compétences typiques requises pour les membres de l'équipe de test dans quatre domaines de compétence.

TM-3.1.2 (K4) Analyser le contexte d'un projet donné afin de déterminer les compétences requises pour les membres de l'équipe de test.

TM-3.1.3 (K2) Expliquer les techniques typiques pour évaluer les compétences des membres de l'équipe de test.

TM-3.1.4 (K2) Différencier les approches typiques pour développer les compétences des membres de l'équipe de test.

TM-3.1.5 (K2) Expliquer les compétences requises pour gérer une équipe de test.

TM-3.1.6 (K2) Donner des exemples de facteurs de motivation et d'hygiène pour les membres de l'équipe de test.

3.2 Relations avec les parties prenantes

TM-3.2.1 (K2) Donnez des exemples pour chacune des quatre catégories déterminant le coût de la qualité.

TM-3.2.2 (K3) Appliquer un calcul coûts-bénéfices pour estimer la valeur ajoutée des tests pour les parties prenantes.

3.1 L'équipe de test

Introduction

Toute équipe qui exécute des tâches de test est composée d'individus ayant des compétences différentes. Alors que dans certaines organisations, les équipes sont auto-organisées, dans d'autres, les Test Managers recrutent et développent ces équipes. La bonne combinaison de compétences est un facteur essentiel pour que toutes les équipes clôturent avec succès les tests.

Les compétences requises pour un membre de l'équipe de test peuvent changer au fil du temps. Il est important de sélectionner les bonnes personnes pour l'équipe de test et de fournir une formation adéquate et des opportunités de croissance. En outre, des personnes extérieures à l'équipe de test peuvent apporter des compétences spécifiques supplémentaires.

Cette section examine le processus fondamental d'analyse et de développement des compétences requises pour les membres de l'équipe de test, ainsi que les compétences requises pour diriger ou encadrer une équipe de test. Cela comprend également la connaissance des facteurs qui motivent ou démotivent les membres de l'équipe de test et d'autres facteurs permettant de garantir un travail d'équipe réussi.

Chaque individu possède déjà des compétences et peut les développer davantage par divers moyens tels que l'expérience professionnelle, l'éducation et la formation. L'équipe de test idéale possède toutes les compétences nécessaires pour des tâches de test données ou n'est responsable que des tâches pour lesquelles elle possède les compétences requises. Pour réussir, une équipe de test a besoin de diverses compétences à différents niveaux. En fonction du contexte du projet, certaines compétences seront plus importantes ou nécessaires que d'autres. Il peut être judicieux de faire appel à des experts externes pour des tâches de test spécifiques qui dépassent les capacités de l'équipe de test.

3.1.1 Compétences typiques dans les quatre domaines de compétence

Les aptitudes d'une personne peuvent être classées en quatre domaines de compétence (Sonntag & Schmidt-Rathjens, 2005) (Erpenbeck & von Rosenstiel, 2017) :

- **Compétence professionnelle** : Consiste en des compétences permettant d'effectuer des tâches spécialisées. Il s'agit par exemple de compétences en techniques de test, d'expertise technologique et métier dans le domaine d'application, ainsi que de compétences en gestion des projets.
- **Compétence méthodologique** : Comprend les compétences générales qu'une personne peut utiliser de manière indépendante dans un domaine et qui permettent de réaliser de manière indépendante des tâches complexes ou nouvelles. Les exemples incluent les compétences analytiques, conceptuelles et de jugement.
- **Compétence sociale** : Comprend les aptitudes liées à la communication, à la coopération et à la gestion des conflits dans des contextes intra et interculturels. Elles permettent d'établir des relations avec les autres afin d'agir de manière appropriée dans une situation donnée et d'atteindre des objectifs individuels et communs. Il s'agit par exemple des compétences en matière de communication, de résolution des conflits, de la capacité à travailler en équipe, de l'adaptabilité et de l'affirmation de soi.
- **Compétence personnelle** : Comprend la capacité et la volonté de se développer soi-même et de développer ses propres talents, sa motivation et sa volonté de performance ainsi que le

développement d'attitudes spécifiques et d'une personnalité individuelle. Les exemples incluent l'autogestion, la responsabilité personnelle, la capacité à recevoir des critiques, la fiabilité, la résilience, la capacité à agir avec confiance, la discipline, l'ouverture aux changements, la volonté d'aider et d'apprendre, et la capacité à déléguer.

Tous les domaines de compétence sont importants pour la réussite d'une équipe de test. Les compétences méthodologiques, sociales et personnelles n'étant pas spécifiques aux tests, l'ISTQB® se concentre sur le développement des compétences professionnelles. Cela inclut des compétences en matière de gestion des tâches de test, d'analyse de la base de test, de conception des tests, d'identification et d'analyse des risques et de développement, de configuration et de maintenance des données de test, des environnements de test et des scripts de test.

3.1.2 Analyse des compétences requises pour les membres de l'équipe de test

L'assignation des personnes est une activité qui fait partie de la planification des tests. Elle comprend la tâche d'identifier les rôles et les compétences des personnes requises pour implémenter les tests dans la stratégie de test. Une analyse détaillée du contexte est nécessaire pour déterminer les compétences requises pour un projet.

Compétences professionnelles et méthodologiques

Pour tester, l'accent est mis sur les compétences requises pour les tâches du test. Voici quelques exemples :

- La planification des tests nécessite des connaissances conceptuelles pour développer une stratégie de test.
- Le pilotage des tests et le contrôle des tests requièrent des compétences en gestion de projet pour gérer toutes les tâches de test.
- L'analyse de test requiert des compétences analytiques pour analyser la base de test et les risques produit.
- La conception des tests requiert des compétences en techniques de test pour concevoir les cas de test et des connaissances conceptuelles pour concevoir les environnements de test.
- L'implémentation des tests requiert des capacités de jugement pour la sélection des tests, ainsi qu'une expertise technique pour la programmation des scripts de test et la mise en place des environnements de test.
- L'exécution des tests requiert des compétences techniques pour exécuter des tests automatisés, effectuer des tests exploratoires et évaluer les résultats des tests.
- La clôture des tests exige la capacité de communiquer les résultats du projet et la responsabilité personnelle des décisions prises.

Les différents types et niveaux de test requièrent des compétences différentes (par exemple, une expertise métier dans le domaine d'application pour évaluer l'aptitude fonctionnelle d'un système, ou une expertise technique pour évaluer la maintenabilité du code).

En outre, le contexte du projet fournit des informations précieuses sur les compétences professionnelles requises :

- Le domaine du système requiert une expertise métier dans le domaine d'application tel que les technologies de l'information, l'industrie automobile ou l'industrie du jeu.
- L'architecture du logiciel et du système et les technologies utilisées dans le projet requièrent, par exemple, une expertise technique en matière de langages de programmation, de technologie d'interface ou de vulnérabilités en matière de sécurité.
- Le cycle de développement nécessite, par exemple, des connaissances sur les niveaux de test, les rôles des testeurs et les techniques de test spécifiques.

Compétence sociale

Dans le contexte du test, la compétence sociale permet aux membres de l'équipe de test de se comporter de manière appropriée dans leurs relations avec les autres membres de l'équipe et d'atteindre les objectifs du test. En particulier, elle comprend les compétences en matière de communication, de coopération et de résolution des conflits (par exemple, traiter de manière constructive des conditions de test non optimales ou rapporter les défauts aux développeurs).

Le développement et le test de logiciels sont généralement effectués par différents membres de groupes (différents) qui coordonnent leurs tâches par le biais de la communication. Les compétences en matière de communication, la capacité à travailler en équipe et la capacité à résoudre les conflits sont des exigences pour la réussite d'un projet. Toutefois, le niveau de compétences sociales requis peut varier en fonction du contexte du projet. Par exemple, le développement logiciel en mode Agile peut nécessiter des exigences plus élevées en matière de compétences sociales que les modèles de développement séquentiels centrés sur les documents ainsi que les tests hors site.

Compétence personnelle

L'efficacité et l'efficience des membres de l'équipe de test dépendent également de leur capacité et de leur volonté de se développer, de développer leurs compétences et leurs attitudes. Par exemple, travailler dans une équipe Agile auto-organisée peut exiger un niveau plus élevé d'autogestion et de discipline de la part de tous les membres de l'équipe, tandis qu'un Test Manager d'une équipe de test hiérarchique, par exemple, doit être capable de déléguer du travail. Un haut degré de fiabilité et de résilience est souvent exigé, en particulier dans les projets où le temps est compté. En outre, la volonté d'aider, d'apprendre et l'ouverture au changement sont importantes dans un processus de changement dans tous les modèles SDLC.

3.1.3 Évaluation des compétences des membres de l'équipe de test

Dans de nombreux cas, les équipes de test sont formées avec le personnel existant. Pour comprendre les capacités des membres de l'équipe et les besoins de développement personnel, le Management des Tests doit évaluer les compétences existantes de l'équipe de test et les comparer aux exigences, qui peuvent être documentées dans une matrice de compétences.

Il existe des modèles pour aider les équipes et leurs membres à travailler plus efficacement (par exemple, les "rôles d'équipe" de Meredith Belbin, DISG® ou PCM®). Selon Belbin (Belbin, 2010), les équipes fonctionnent efficacement lorsqu'elles sont composées de différents types de personnalités et de rôles. Ces modèles aident les équipes à identifier les compétences dont elles disposent et celles qui leur font défaut.

Les compétences professionnelles et méthodologiques des membres de l'équipe de test peuvent être évaluées en testant des tâches typiques de test :

Définir une stratégie de test et tester le retour d'information avec les collègues.

- Réviser la base de test et communiquer les constatations, ce qui peut également révéler des compétences en matière de communication.
- Déterminer les techniques de test permettant d'atteindre des objectifs de test spécifiques dans le cadre d'un projet donné.
- Appliquer les différentes techniques de test de manière appropriée
- Rédiger un rapport de clôture des tests comprenant une évaluation des résultats du test.

En outre, les compétences peuvent être évaluées par le biais de titres externes, de certifications, d'expériences professionnelles et de diplômes.

Dans le cadre du développement logiciel en mode Agile en particulier, les équipes identifient les compétences dont elles ont besoin en participant régulièrement à des rétrospectives et en recevant un retour d'information. Des coaches ou des mentors expérimentés les aident à développer leurs compétences et à identifier et résoudre les lacunes en matière de connaissances.

3.1.4 Développer les compétences des membres de l'équipe de test

Une équipe de test peut ne pas disposer de toutes les exigences requises au début d'un projet. Bien qu'un ensemble parfait d'individus puisse ne pas être disponible, une équipe solide peut équilibrer les forces et les faiblesses des individus.

Le Management des Tests ou l'équipe de test peuvent identifier les besoins de développeurs en comparant les compétences requises et les compétences disponibles dans une matrice de compétences. Sur cette base, ils peuvent déterminer les approches de développement des compétences :

- La formation et l'éducation enseignent des connaissances et des pratiques prédéfinies, généralement dans une salle de classe (virtuelle) (par exemple, en envoyant des personnes suivre une formation, en organisant des sessions de formation en interne, en développant des formations personnalisées ou en utilisant des cours d'apprentissage en ligne en direct).
- L'auto-apprentissage est une manière d'apprendre un sujet qui implique d'étudier seul, plutôt que dans une classe (virtuelle), (par exemple, lire des livres, regarder des vidéos enregistrées ou faire des recherches sur Internet).
- L'apprentissage par les pairs, dans lequel les collègues partagent leurs connaissances, leurs idées et leurs expériences et apprennent les uns avec les autres et les uns des autres.
- Le mentorat ou le coaching sont des approches dans lesquelles un membre de l'équipe qui est nouveau dans un rôle reçoit des conseils individuels de la part d'un coach, ou des connaissances, des compétences et/ou de l'expérience de la part d'un mentor expérimenté. La personne expérimentée agit en tant que ressource permanente pour fournir des conseils et de l'aide.
- La formation "sur le terrain" est également bien connue et constitue un mélange d'auto-apprentissage, d'apprentissage par les pairs et de mentorat ou de coaching.

Toutes les approches de développement des compétences ne sont pas aussi efficaces et efficientes les unes que les autres. L'auto-apprentissage et la formation, par exemple, sont bien adaptés au développement des compétences professionnelles et méthodologiques. Pour cette raison, les connaissances de base sur les tests peuvent être développées en participant aux sessions de formation

de l'ISTQB® ou par auto-apprentissage sur la base des syllabus de l'ISTQB®. Cependant, pour développer les compétences sociales et personnelles, il est recommandé d'utiliser des approches telles que la formation et le coaching, qui sont souvent plus prometteuses que l'auto-apprentissage. L'échange social, le retour d'information et la réflexion font partie des facteurs clés de succès pour développer les compétences sociales et personnelles.

3.1.5 Les exigences en matière de management pour gérer une équipe de test

Toute personne souhaitant diriger avec succès une équipe de test doit posséder des compétences en management. Celles-ci comprennent des compétences professionnelles et méthodologiques dans les activités fondamentales de management (par exemple, la planification, le pilotage des progrès, le contrôle et l'établissement de rapports). Des connaissances et des compétences spécifiques en matière de Management des Tests sont exigées pour les tests (par exemple, la connaissance des différentes approches de test, le développement de stratégies de test, ou l'utilisation de techniques de test ou du SDLC appliqué).

Diriger ou encadrer une équipe de test signifie agir de manière appropriée dans les relations avec les autres membres de l'équipe de test et avoir la capacité et la volonté d'évoluer dans des circonstances changeantes. C'est pourquoi les compétences sociales et personnelles sont des facteurs de réussite essentiels pour diriger une équipe de test. Cela inclut la résilience, la capacité à déléguer et la capacité à être accepté par l'équipe de test. De plus, cela inclut la capacité à faire valoir les intérêts du test dans le projet, à promouvoir les avantages du test, et à communiquer et résoudre les conflits avec toutes les parties prenantes.

Pour recruter des personnes pour l'équipe de test, des compétences en matière d'analyse des conditions sociales, des conditions d'équipe et des conditions de travail sont nécessaires. Ces compétences permettent de s'assurer que l'équipe de test correspond aux conditions de travail ou, si possible, que les conditions de travail sont adaptées à l'équipe de test. En outre, les équipes de test sont soumises à des processus de développement dynamiques et nécessitent donc des compétences situationnelles (par exemple, selon les phases du modèle de développement des petits groupes de Tuckman) (Tuckman, 1965) (Bonebright, 2010) :

- La volonté d'aider les membres de l'équipe de test à s'intégrer dans l'équipe de test (Forming).
- La capacité à résoudre les conflits au sein de l'équipe de test (Storming).
- Discipline et leadership orienté vers les objectifs pour garantir des valeurs et des règles convenues (Norming)
- La capacité à déléguer afin de donner à l'équipe de test une responsabilité personnelle (Performing)
- La capacité à agir avec appréciation et confiance avec les membres de l'équipe de test qui quittent l'équipe (Adjourning)

3.1.6 Facteurs de motivation ou de démotivation d'une équipe de test dans certaines situations

Des membres de l'équipe de test satisfaits et motivés augmentent la productivité et la performance et ont donc un impact significatif sur la réussite des projets. Lorsque ces conditions sont remplies, la formation croisée se fait de manière informelle, les membres de l'équipe de test peuvent gérer leur propre charge de travail et le Test Manager a plus de temps pour s'occuper des questions externes. La théorie de la

motivation et de l'hygiène (Herzberg, et al., 1993) fait la distinction entre les facteurs de motivation et les facteurs d'hygiène :

Les facteurs de motivation sont perçus consciemment et peuvent conduire à la croissance et à la satisfaction. Il peut s'agir de :

- La reconnaissance et l'appréciation du travail accompli (par exemple, des incitants et toute autre approche individuelle que les membres de l'équipe de test trouvent appréciable).
- Une responsabilité et une autonomie accrues (par exemple, définir les processus de test au sein d'une équipe de test).
- Des tâches intéressantes, significatives et stimulantes que les membres de l'équipe de test considèrent comme réalisables et qui valent la peine d'être testées (par exemple, la sélection et l'introduction d'un nouvel outil pour l'automatisation des tests).
- L'avancement et le développement professionnels (par exemple, le passage d'un testeur expérimenté à Test Manager ou à responsable du processus de test).

Les facteurs d'hygiène sont généralement considérés comme acquis. Le fait de les remplir n'entraîne pas automatiquement une plus grande satisfaction. S'ils font défaut, ils peuvent avoir un effet démotivant sur les membres de l'équipe de test :

- Une rémunération appropriée (par exemple, un salaire conforme au marché, des heures supplémentaires payées, de bons avantages sociaux)
- Une politique du personnel et un style de management basés sur l'appréciation (par exemple, gestion allégée, objectifs réalistes, protection contre l'accessibilité externe et la surcharge).
- Des conditions de travail agréables (par exemple, des spécifications sans ambiguïté, des objets de test mûrs, des défauts corrigés de manière adéquate, un lieu de travail approprié, un environnement de test stable)
- La sûreté en tant que besoin existentiel (par exemple, un lieu de travail sûr et des accords respectés)
- De bonnes relations interpersonnelles (par exemple, avec les collègues, les superviseurs).

Par conséquent, le Management des Tests doit continuellement éliminer les facteurs de démotivation et en même temps créer et renforcer les facteurs de motivation.

De plus amples informations peuvent être trouvées dans les documents suivants (Belbin, 2010) (Marston, 1999) (Kahler, 2008).

3.2 Relations avec les parties prenantes

Introduction

Dans le cadre du Management des Tests, il est important d'optimiser les tests pour obtenir une bonne valeur métier. Des tests excessifs peuvent entraîner des retards déraisonnables et des coûts supérieurs aux bénéfices, tandis que des tests insuffisants peuvent conduire à la livraison d'un produit de faible qualité aux utilisateurs. L'approche optimale se situe entre ces deux extrêmes. Il est de la responsabilité du Test Manager d'aider les parties prenantes à comprendre cet équilibre et la valeur ajoutée des tests pour atteindre cet équilibre, tout en gardant à l'esprit, par exemple, les contraintes de temps typiques d'un projet.

3.2.1 Le coût de la qualité

Les avantages des tests sont compensés par les coûts de la qualité. Le coût de la qualité est un moyen de quantifier le coût total des efforts et des défauts liés à la qualité. Le coût de la qualité consiste à classer les coûts de projet et d'exploitation en quatre catégories liées aux coûts des défauts du produit :

- **Coûts de prévention des défauts** : Le coût de toutes les activités planifiées et proactives pour prévenir la mauvaise qualité (par exemple, la qualification des développeurs pour leurs tâches, comme la formation à la création d'un code maintenable ou sécurisé, la revue de la base de test le plus tôt possible, et une communication appropriée au sein de l'équipe).
- **Coûts d'évaluation** : Le coût de toutes les activités visant à détecter les défauts (par exemple, effectuer des tests statiques et des tests dynamiques et revoir les produits d'activités).
- **Coûts de défaillance interne** : Le coût de toutes les activités réactives (par exemple, la correction des défauts constatés pendant les tests, la fourniture de solutions de contournement).
- **Coûts de défaillance externe** : Coût de toutes les activités réactives et sans valeur ajoutée (par exemple, perte de revenus, d'actifs, de santé humaine, de vies humaines ou d'environnement, coûts juridiques liés à la correction des défauts, aux tests, au déploiement et au soutien en raison de la livraison d'un produit défectueux au client ("post release"), correction des défauts sur le terrain (signalés par les clients)).

Les coûts totaux d'évaluation et les coûts de défaillance interne sont généralement nettement inférieurs aux coûts de défaillance externe. Cela rend donc les tests extrêmement utiles. En déterminant les coûts dans ces quatre catégories, les Test Managers peuvent créer une analyse de rentabilité (business case) convaincante pour les tests.

D'autres approches peuvent être envisagées pour définir le coût de la qualité. Les syllabus de l'ISTQB® décrivent deux d'entre elles. Ce syllabus est basé sur l'approche de Feigenbaum, et le syllabus ISTQB® de niveau Fondation V.4 présente l'approche de Boehm (voir le syllabus ISTQB® niveau Fondation V.4, Section 1.3, Principes de test). Ces deux approches ont été sélectionnées pour parvenir à une facilité de compréhension du coût de la qualité. L'approche de Feigenbaum (Feigenbaum, Nov/Dec 1956) considère la qualité comme un processus orienté vers le client et à l'échelle de l'entreprise, tandis que l'approche de Boehm (Boehm, 1979) se concentre sur le compromis entre le coût de la prévention et le coût de la défaillance dans le développement de logiciels (Hadjicostas, 2004).

3.2.2 Rapport coûts-bénéfices du test

Alors que la plupart des organisations considèrent que les tests ont une certaine valeur, peu de managers, y compris les Test Managers, peuvent quantifier, décrire ou articuler cette valeur. De plus, de nombreux Test Managers, responsables de tests et testeurs se concentrent sur les détails opérationnels des tests (c'est-à-dire les aspects spécifiques à la tâche de test ou au niveau du test), tout en ignorant les questions tactiques et stratégiques plus larges (niveau supérieur) liées aux tests, auxquelles d'autres parties prenantes, en particulier les managers, s'intéressent.

Les tests apportent des bénéfices à l'organisation, au projet et/ou à l'exploitation de manière quantitative et qualitative :

- **Les avantages qualitatifs** comprennent une meilleure réputation de qualité, des releases plus fluides et plus prévisibles, une confiance accrue, une protection contre la responsabilité juridique et une réduction du risque de perte de missions entières, voire de vies humaines.
- **Les avantages quantitatifs** comprennent les défauts constatés, évités ou corrigés avant la release, les défauts connus avant la release (c'est-à-dire non corrigés mais documentés, éventuellement avec des solutions de contournement), les avantages en termes de coûts (Bohm 1981, Böhler 2008), la réduction du niveau de risque par l'exécution de tests et la fourniture d'informations sur l'état du projet, du processus et du produit.

Un autre avantage des tests est que toutes les parties prenantes obtiennent des informations adéquates pour prendre des décisions en connaissance de cause et déterminer si la qualité du produit est suffisante pour le mettre en service, avec ou sans défauts. Parfois, il est préférable de mettre le produit en service avec des défauts connus plutôt que d'attendre que les défauts soient résolus. Dans ces cas où un défaut peut être toléré, cela dépend fortement de la probabilité d'occurrence et de la sévérité du défaut.

Le coût de la qualité par défaut de test est calculé comme suit :

Économie moyenne par défaut = Moyenne des coûts de défaillance externes - (Moyenne des coûts évalués + Moyenne des coûts de défaillance internes).

Coût total de la qualité = (Coûts de prévention des défauts + (Coût évalué * Nombre de défauts constatés avant la release)) + ((Coût des défaillances internes * Nombre de défauts constatés avant la release) + (Coût des défaillances externes * Nombre de défauts constatés après la release))

À titre d'exemple, supposons que vous ayez calculé le coût de la qualité par défaut suivant pour un produit :

Coûts de prévention des défauts : 180 €

Moyenne des coûts évalués par défaut : 500 €

Moyenne des coûts de défaillance interne par défaut : 200 €

Moyenne des coûts de défaillance externe par défaut : 4 000 €

Les coûts moyens de prévention des défauts, les coûts évalués et les coûts de défaillance interne sont calculés sur la base du nombre de défauts constatés avant la release, tandis que les coûts moyens de défaillance externe sont calculés sur la base du nombre de défauts constatés après la release. Avec ces valeurs, nous pouvons calculer l'économie moyenne par défaut comme suit :

Économie moyenne par défaut = 4 000 - (500 + 200) = 3 300 €.

La courbe de Boehm est une représentation graphique des coûts de correction des défauts au fil du temps dans le cadre du cycle de développement. Elle conclut que les tests devraient être effectués plus tôt dans le cycle de développement afin de réduire le coût de la correction des défauts. La courbe de Boehm montre que plus un défaut est découvert tard dans le SDLC, plus les coûts de défaillance interne, ou le coût de correction d'un défaut, augmentent. À l'aide de ces informations, le Test Manager doit s'efforcer de trouver la relation optimale entre les coûts de prévention des défauts et les coûts internes et externes.

L'effort de test doit être basé sur les risques spécifiques du projet et du produit et sur le risque que le Métier est prêt à prendre. Trop tester peut entraîner des coûts plus élevés que le bénéfice issu de la réduction du niveau de risque. Si l'on teste trop peu, les défauts manqués peuvent présenter un risque élevé de générer des coûts supérieurs à ceux qu'auraient engendrés les tests omis. Le test basé sur les risques (voir Section 1.3 de ce syllabus, Test basé sur les risques) soutient la relation coûts-bénéfices des tests en investissant des niveaux d'effort de test proportionnels aux niveaux de risque, et en priorisant les tests en fonction de leurs niveaux de risque.

Les Test Managers devraient comprendre lesquels de ces bénéfices et coûts s'appliquent à leur organisation, projet, et/ou exploitation, et être capables de communiquer la valeur ajoutée des tests en termes de ces bénéfices et du coût de la qualité par défaut.

4 Références

Normes

- **IEC 61508** (2010) Functional safety of electrical/electronic/programmable electronic safety-related systems - Parts 1 to 7
- **ISO/IEC/IEEE 29119-2** (2021) ISO/IEC/IEEE 29119-2 Software and systems Engineering-Software testing-Part 2 Test processes
- **ISO/IEC/IEEE 29119-3** (2021) ISO/IEC/IEEE 29119-3 Software and systems Engineering-Software testing-Part 3 Test documentation

Documents ISTQB®

- ISTQB® Certified Tester Agile Test Leadership at Scale Syllabus v2.0 (2023)
- ISTQB® Certified Tester Foundation Level Syllabus V.4 (2023)
- ISTQB® Certified Tester Expert Level Test Management Syllabus v1.0 (2011)
- ISTQB® Certified Tester Expert Level Improving the Test Process Syllabus v1.0.1 (2011)

Livres

- Basili, V., Trendowicz, A., Kowalczyk, M., Heidrich, J., Seaman, C., Münch, J., & Rombach, D. (2014). *Aligning Organizations Through Measurement – The GQM+ Strategies Approach.* Springer International.
- Bath, G., & van Veenendaal, E. (2014). *Improving the Test Process - chapter 6: Process for Improvement.* Rocky Nook.
- Belbin, R. M. (2010). *Management Teams: Why They Succeed or Fail.* London: Routledge.
- Black, R. (2009). *Managing the Testing Process, 3rd Edition.* John Wiley & Sons.
- Boehm, B. (1979). *Software Engineering Economics.* Prentice-Hall.
- Bonebright, D. A. (2010). *40 years of storming: a historical review of Tuckman's model of small group development* (1 Ausg., Bd. 13). Human Resource Development International, 1, 2010, Vol. 13.
- Craig, R., & Jaskiel, S. P. (2002). *Systematic Software Testing.* Artech House.
- Derby, E., & Larsen, D. (2006). *Agile Retrospectives – Making Good Teams Great,.* The Pragmatic Bookshelf.
- Erpenbeck, J., & von Rosenstiel, L. (2017). *Handbuch Kompetenzmessung.* Stuttgart: Schäffer-Poeschel.
- Fowler, M. (2010). *Hybrid development processes.* IEEE Software, 27(2), 57-63.

- Herzberg, F., Mausner, B., & Bloch Snyderman, B. (1993). *Motivation to Work.* London: Routledge.
- Kahler, T. (2008). *The Process Therapy Model: The Six Personality Types with Adaptations.* Taibi Kahler Associates, Inc.
- Marston, W. M. (1999). *Emotions Of Normal People.* London: Routledge.
- Sonntag, K., & Schmidt-Rathjens, C. (2005). *Anforderungsanalyse und Kompetenzmodelle.* Wiesbaden: VS Verlag für Sozialwissenschaften.
- Tuckman, B. W. (1965). *Developmental sequence in small groups* (Bd. 63(6)). Psychological Bulletin.
- van Ewijk, A. (2013). *TPI NEXT – Business Driven Test Process Improvement,.* Sogeti Nederland B.V.
- van Solingen, R., & Berghout, E. (1999). *The Goal Question Metric Method – A Practical Guide for Quality Improvement of Software Development.* McGraw-Hill.
- van Veenendaal, E. (2012). *The PRISMA Approach: Practical Risk-Based Testing.* UTN Publishers.
- van Veenendaal, E. (2020). *TMMi in the Agile world, version 1.4.* TMMi Foundation.
- van Veenendaal, E., & Cannegieter, J. J. (2011). *The Little TMMi – Objective-Driven Test Process Improvement.* UTN Publishers.

Articles

- Feigenbaum, Armand V. (Nov/Dec 1956) Harvard Business Review, Vol. 34 Issue 6, p93-101
- Hadjicostas, Evsevios (2004) Total Quality Management and Cost of Quality, Springer https://link.springer.com/chapter/10.1007/978-3-662-09621-5_7

Pages Web

- www.tmmi.org Test Maturity Model integration (TMMi®); last visit January 31st, 2024
- www.tmap.net Test Process Improvement (TPI); last visit January 31st, 2024
- www.wikipedia.org/wiki/PDCA Plan-Do-Check-Act; last visit January 31st, 2024

Les références précédentes renvoient à des informations disponibles sur Internet et ailleurs. Bien que ces références aient été vérifiées au moment de la publication de ce syllabus, l'ISTQB® ne peut être tenue responsable si les références ne sont plus disponibles.

5 Appendice A – Objectifs d'apprentissage/Niveau cognitif de connaissance

Les objectifs d'apprentissage spécifiques qui s'appliquent à ce syllabus sont indiqués au début de chaque chapitre. Chaque sujet du syllabus sera examiné en fonction de son objectif d'apprentissage.

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

Niveau 1: Se souvenir (K1)

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

Verbes d'action : Rappeler, reconnaître.

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

Niveau 2: Comprendre (K2)

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

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

Exemples	**Notes**
Classer les outils de test en fonction de leur objectif et des activités de test qu'ils soutiennent.	
Comparer les différents niveaux de test.	Peut être utilisé pour rechercher des similitudes, des différences ou les deux.
Différencier les tests du débogage.	Recherche de différences entre les concepts.
Faire la distinction entre les risques projet et les risques produit.	Permet de classer séparément deux concepts (ou plus).
Expliquer l'impact du contexte sur le processus de test.	
Donner des exemples de raisons pour lesquelles les tests sont nécessaires.	
Déduire la cause racine des défauts à partir d'un profil donné de défaillances.	

Exemples	Notes
Résumer les activités du processus de révision du produit d'activités.	

Niveau 3: Appliquer (K3)

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

Verbes d'action : Appliquer, implémenter, préparer, utiliser.

Exemples	Notes
Appliquer l'analyse des valeurs limites pour dériver des cas de test à partir d'exigences données.	Doit faire référence à une procédure/technique/processus, etc.
Implémenter des méthodes de collecte de métriques pour soutenir les exigences techniques et de management.	
Préparer des tests de facilité d'installation pour les applications mobiles.	
Utiliser la traçabilité pour suivre la progression des tests afin d'en vérifier la complétude et la cohérence avec les objectifs du test, la stratégie de test et le plan de test.	Peut être utilisé dans un LO qui demande au candidat d'être capable d'utiliser une technique ou une procédure. Semblable à "appliquer".

Niveau 4: Analyser (K4)

Le candidat peut séparer les informations relatives à une procédure ou à une technique en ses éléments constitutifs pour une meilleure compréhension et peut faire la distinction entre les faits et les déductions. Une application typique consiste à analyser un document, un logiciel ou la situation d'un projet et à proposer des actions appropriées pour résoudre un problème ou achever une tâche.

Verbes d'action : Analyser, déconstruire, tracer les grandes lignes, établir des priorités, sélectionner.

Exemples	Notes
Analyser la situation d'un projet donné afin de déterminer quelles techniques de test boîte noire ou basés sur l'expérience doivent être appliquées pour atteindre des objectifs spécifiques.	Examinable uniquement en combinaison avec un objectif mesurable de l'analyse.
Classer par ordre de priorité les cas de test d'une suite de tests donnée en vue de leur exécution, en se basant sur les risques liés au produit.	Doit être de la forme "Analyser xxxx à xxxx" (ou similaire).

Exemples	Notes
Sélectionner les niveaux de test et les types de test appropriés pour vérifier un ensemble donné d'exigences.	

Référence

(Pour les niveaux cognitifs des objectifs d'apprentissage)

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

Les objectifs d'apprentissage spécifiques qui s'appliquent à ce syllabus sont indiqués au début de chaque chapitre.

6 Appendice B – Matrice de traçabilité des objectifs métier avec les objectifs d'apprentissage

Cette section énumère la traçabilité entre les objectifs métier et les objectifs d'apprentissage du syllabus de Management des Tests au niveau avancé..

Objectifs métier : Management des Tests - niveau Avancé			BO1	BO2	BO3	BO4	BO5	BO6	BO7	BO8	BO9	BO10	BO11
TM_01	Gérer les tests dans le cadre de divers projets de développement de logiciels en appliquant les processus de gestion des tests établis pour l'équipe de projet ou l'organisation chargée des tests.		12										
TM_02	Identifier les parties prenantes des tests et les modèles de cycle de vie du développement logiciel qui sont pertinents dans un contexte donné			4									
TM_03	Organiser des sessions d'identification des risques et d'évaluation des risques dans le cadre d'un cycle de vie du développement logiciel et utiliser ces résultats pour guider les tests afin d'atteindre les objectifs du test				6								
TM_04	Définir une stratégie de test de projet cohérente avec la stratégie de test de l'organisation et le contexte du projet					11							
TM_05	Suivre et contrôler en continu les tests pour atteindre les objectifs du test						4						

Objectifs métier : Management des Tests - niveau Avancé			BO1	BO2	BO3	BO4	BO5	BO6	BO7	BO8	BO9	BO10	BO11
TM_06	Évaluer et rapporter l'avancement des tests aux parties prenantes du projet.							3					
TM_07	Identifier les compétences nécessaires et développer ces compétences au sein de votre équipe.								6				
TM_08	Préparer et présenter un business case pour les tests dans différents contextes, en soulignant les coûts et les avantages attendus.									5			
TM_09	Diriger les activités d'amélioration du processus des tests dans les projets ou les flux de produits de développement de logiciels et contribuer aux initiatives organisationnelles d'amélioration du processus de test.										5		
TM_10	Planifier les activités de test et estimer l'effort de test.											9	
TM_11	Créer des rapports de défauts et un cycle de vie des défauts adapté à un cycle de vie du développement logiciel.												6
LO Unique	Objectif d'apprentissage.	Niveau K											
1	Gérer les activités de test.												
1.1	Le processus de test.												
TM-1.1.1	Résumer la planification des tests.	K2	X			X							

Objectifs métier : Management des Tests - niveau Avancé			BO1	BO2	BO3	BO4	BO5	BO6	BO7	BO8	BO9	BO10	BO11
TM-1.1.2	Résumer le suivi des tests et le contrôle des tests.	K2	X				X						
TM-1.1.3	Résumer la clôture des tests.	K2	X					X					
1.2	Le contexte du test.												
TM-1.2.1	Comparer les raisons pour lesquelles les différentes parties prenantes s'intéressent aux tests.	K2		X		X							
TM-1.2.2	Expliquer pourquoi la connaissance des parties prenantes est importante dans le Management des Tests.	K2		X		X							
TM-1.2.3	Expliquer comment tester dans un modèle de développeur de logiciels hybrides.	K2		X		X							
TM-1.2.4	Résumer les activités de Management des Tests pour différents cycles de vie du développement logiciel.	K2	X	X		X							
TM-1.2.5	Comparer les activités de Management des Tests à différents niveaux de test.	K2	X			X							
TM-1.2.6	Comparer les activités de Management des Tests pour différents types de test.	K2	X			X							
TM-1.2.7	Analyser un projet donné et déterminer les activités de Management des Tests, en mettant l'accent sur la planification des tests, le pilotage des tests et le contrôle des tests.	K4	X			X							

Objectifs métier : Management des Tests - niveau Avancé			BO1	BO2	BO3	BO4	BO5	BO6	BO7	BO8	BO9	BO10	BO11
1.3	Test basé sur les risques.												
TM-1.3.1	Expliquer les différentes mesures prises par le test basé sur les risques pour répondre aux risques.	K2			X								
TM-1.3.2	Donner des exemples de différentes techniques qu'un Test Manager peut utiliser pour identifier les risques liés à la qualité d'un produit.	K2			X								
TM-1.3.3	Résumer les facteurs qui déterminent les niveaux de risque liés à la qualité du produit.	K2			X								
TM-1.3.4	Sélectionner les activités de test appropriées pour atténuer les risques en fonction de leur niveau de risque dans un contexte donné.	K4			X								
TM-1.3.5	Faire la différence entre les exemples lourds et légers de techniques de tests basés sur le risque.	K2			X								
TM-1.3.6	Donner des exemples de métriques de succès et de difficultés associées au test basé sur les risques.	K2			X								
1.4	La stratégie de test du projet.												
TM-1.4.1	Expliquer les choix typiques d'une approche de test.	K2				X							

Objectifs métier : Management des Tests - niveau Avancé			BO1	BO2	BO3	BO4	BO5	BO6	BO7	BO8	BO9	BO10	BO11
TM-1.4.2	Analyser une stratégie de test organisationnelle et le contexte du projet pour sélectionner l'approche de test appropriée.	K4				X							
TM-1.4.3	Utiliser la méthodologie S.M.A.R.T. (méthodologie des objectifs) pour définir des objectifs de test mesurables et des critères de sortie.	K3				X							
1.5	Amélioration du processus de test.												
TM-1.5.1	Expliquer comment utiliser le modèle IDEAL pour l'amélioration du processus de test sur un projet donné.	K2									X		
TM-1.5.2	Résumer l'approche de l'amélioration du processus de test basée sur les modèles et comprendre comment l'appliquer dans le cadre d'un projet donné.	K2									X		
TM-1.5.3	Résumer l'approche d'amélioration du processus de test basée sur l'analyse et comprendre comment l'appliquer dans le contexte d'un projet.	K2									X		
TM-1.5.4	Implémenter une rétrospective de projet ou d'itération pour évaluer les processus de test et découvrir les domaines de test à améliorer.	K3									X		
1.6	Outils de test.												

Objectifs métier : Management des Tests - niveau Avancé			BO1	BO2	BO3	BO4	BO5	BO6	BO7	BO8	BO9	BO10	BO11
TM-1.6.1	Résumer les meilleures pratiques pour l'introduction d'un outil.	K2										X	
TM-1.6.2	Expliquer l'impact des différents aspects techniques et métiers lors du choix d'un type d'outil.	K2										X	
TM-1.6.3	Analyser une situation donnée afin d'élaborer un plan de sélection d'un outil, en tenant compte des risques, des coûts et des avantages.	K4										X	
TM-1.6.4	Distinguer les différents stades du cycle de vie d'un outil.	K2										X	
TM-1.6.5	Donner des exemples de collecte et d'évaluation de métriques à l'aide d'outils.	K2									X	X	
2	Management du produit.												
2.1	Métriques de test.												
TM-2.1.1	Donner des exemples de métriques pour atteindre les objectifs du test.	K2					X						
TM-2.1.2	Expliquer comment contrôler l'avancement des tests à l'aide des métriques de test.	K2					X						
TM-2.1.3	Analyser les résultats des tests pour créer des rapports de tests qui permettent aux parties prenantes de prendre des décisions.	K4					X	X					
2.2	Estimation des tests.												

Objectifs métier : Management des Tests - niveau Avancé			BO1	BO2	BO3	BO4	BO5	BO6	BO7	BO8	BO9	BO10	BO11
TM-2.2.1	Expliquer les facteurs qui doivent être pris en compte dans l'estimation des tests.	K2	X							X		X	
TM-2.2.2	Donner des exemples de facteurs qui peuvent influencer l'estimation des tests.	K2	X							X		X	
TM-2.2.3	Choisir une technique ou une approche appropriée pour l'estimation de test dans un contexte donné.	K4	X							X		X	
2.3	Gestion des défauts.												
TM-2.3.1	Implémenter un processus de gestion des défauts, y compris le cycle de vie des défauts, qui peut être utilisé pour surveiller et contrôler les défauts.	K3											X
TM-2.3.2	Expliquer le processus et les participants nécessaires à une gestion des défauts efficace.	K2											X
TM-2.3.3	Expliquer les spécificités de la gestion des défauts dans le cadre du développement logiciel en mode Agile.	K2	X										X
TM-2.3.4	Expliquer les défis de la gestion des défauts dans le cadre du développement de logiciels hybrides.	K2	X										X
TM-2.3.5	Utiliser les données et les informations de classification qui devraient être recueillies au cours de la gestion des défauts.	K3											X

Objectifs métier : Management des Tests - niveau Avancé			BO1	BO2	BO3	BO4	BO5	BO6	BO7	BO8	BO9	BO10	BO11
TM-2.3.6	Expliquer comment les statistiques des rapports de défauts peuvent être utilisées pour concevoir l'amélioration des processus.	K2										X	X
3	Management de l'équipe.												
3.1	L'équipe de test.												
TM-3.1.1	Donner des exemples de compétences typiques requises par les membres de l'équipe de test dans quatre domaines de compétence.	K2							X				
TM-3.1.2	Analyser le contexte d'un projet donné pour déterminer les compétences requises pour les membres de l'équipe de test.	K4							X				
TM-3.1.3	Expliquer les techniques typiques pour évaluer les compétences des membres de l'équipe de test.	K2							X				
TM-3.1.4	Différencier les approches typiques pour développer les compétences des membres de l'équipe de test.	K2							X				
TM-3.1.5	Expliquer les compétences requises pour gérer une équipe de test.	K2							X				
TM-3.1.6	Donner des exemples de facteurs de motivation et d'hygiène pour les membres de l'équipe de test.	K2							X				

Objectifs métier : Management des Tests - niveau Avancé			BO1	BO2	BO3	BO4	BO5	BO6	BO7	BO8	BO9	BO10	BO11
3.2	Relations avec les parties prenantes.												
TM-3.2.1	Donner des exemples de chacune des quatre catégories déterminant le coût de la qualité.	K2								X			
TM-3.2.2	Appliquer un calcul coûts-bénéfices pour estimer la valeur ajoutée des tests pour les parties prenantes.	K3						X		X			

7 Appendice C – Notes de release

Le syllabus de niveau avancé Management des Tests de l'ISTQB® v3.0 est une mise à jour majeure basée sur le syllabus de niveau avancé Test Manager 2012. Pour cette raison, il n'y a pas de notes de release détaillées par chapitre et section. Cependant, un résumé des principaux changements est fourni ci-dessous.

Dans cette version, tous les objectifs d'apprentissage (LO) ont été édités pour les rendre atomiques et pour créer une traçabilité biunivoque entre les LO et les sections du syllabus, ce qui permet de ne pas avoir de contenu sans avoir également un LO. L'objectif est de rendre cette version plus facile à lire, à comprendre, à apprendre et à traduire, en se concentrant sur l'augmentation de l'utilité pratique et de l'équilibre entre les connaissances et les compétences.

Cette release majeure a apporté les changements suivants :

- Réduction de la taille du syllabus global. Un syllabus n'est pas un manuel, mais un document qui sert à présenter les éléments de base d'un cours avancé sur les tests de logiciels, y compris les sujets qui doivent être couverts et à quel niveau. Par conséquent, en particulier :
    - Dans la plupart des cas, les exemples sont exclus du texte. Il incombe à l'organisme de formation de fournir les exemples, ainsi que les exercices, pendant la formation.
    - La "la checklist pour la rédaction du syllabus" a été suivie, qui suggère la taille maximale du texte pour les objectifs d'apprentissage à chaque niveau K (K2 = maximum de 1 500 caractères sans espace, K3 = maximum de 2 500 caractères sans espace, K4 = maximum de 3 000 caractères sans espace, +/- 20 %).
- Réduction du nombre d'objectifs métier par rapport au syllabus CTAL TM 2012
    - 36 LO K2 contre 39 LO dans CTAL TM 2012.
    - 5 LO K3 contre 12 LO dans la CTAL TM 2012.
    - 7 objectifs d'apprentissage K4 contre 10 objectifs d'apprentissage dans la CTAL TM 2012.
- La structure complète du syllabus a été revue.
- L'alignement sur le syllabus ISTQB® niveau Fondation V.4 est complète.
- Changements majeurs dans l'ancien chapitre 1 (Processus de test) de 2012
    - Limité au Management des activités de test (Planification des tests, pilotage des tests, contrôle des tests et clôture des tests).
    - Intégré en tant que section dans le nouveau chapitre "Management des activités de test".
- Nouveau chapitre : **"Manager les activités de test"**.
    - Section 1.1 - Le processus de test : voir ci-dessus.
    - Section 1.2 - Le contexte des tests : Élargi pour couvrir les modèles de développement non séquentiels.
    - Section 1.3 - Test basé sur les risques : Entièrement réécrite pour la rendre plus applicable au niveau d'un projet.

- Section 1.4 - La stratégie de test du projet : Le plan de test étant déjà défini dans le syllabus V.4 du niveau Fondation de l'ISTQB®, l'accent est mis sur la sélection de l'approche de test adéquate et sur la définition d'objectifs de test mesurables.
- Section 1.5 - Amélioration du processus de test : L'intégration dans le Management des Activités de Test, en montrant comment l'appliquer dans un contexte de projet, et l'implémentation à l'aide de rétrospectives au sein d'une itération ou d'un projet.
- Section 1.6 - Outils de test : L'introduction des outils a été déplacée du syllabus de niveau Fondation de l'ISTQB® V.3.1 (n'est pas présente dans le syllabus de niveau Fondation de l'ISTQB® V.4).

- Nouveau chapitre : **Management du produit**
  - Section 2.1 - Métriques de test : Anciennes sections définissant les métriques et l'utilisation des métriques, Métriques de test
  - Section 2.2 - Estimation de test : Le syllabus du niveau Fondation de l'ISTQB® V.4 couvre déjà le calcul de l'estimation de test. Développé pour sélectionner à un niveau K4 les techniques d'estimation de test adéquates et l'utilisation des estimations de test à travers les modèles SDLC.
  - Section 2.3 - Gestion des défauts : alignée sur les dernières éditions des normes et étendue à l'utilisation dans le cadre du développement logiciel en modes Agile et hybride.
- Nouveau chapitre **Management de l'équipe**
  - Section 3.1 - L'équipe de test : Les principaux sujets sont les mêmes que dans le syllabus TM 2012. Identifier les compétences individuelles et composer les équipes de test.
  - Section 3.2 - Les relations avec les parties prenantes : Il s'agit de l'ancienne section du Syllabus TM 2012 intitulée "Valeur métier des tests".
- Principaux changements et sections/chapitres supprimés dans le CTAL TM Syllabus 2012
  - La section sur les tests distribués, externalisés et internalisés a été supprimée.
  - Section sur le Management de l'application des normes industrielles supprimée.
  - Chapitre sur les Revues supprimé.
  - Sous-sections sur l'amélioration du processus de test CTP et STEP supprimées.
  - Sous-sections sur l'analyse des tests, la conception des tests, l'implémentation des tests et l'exécution des tests supprimées.

8 Appendice D – Mots clés spécifiques au domaine

Nom du terme	Definition
Métrique-questions-objectifs (GQM)	Approche de la mesure des logiciels utilisant un modèle à trois niveaux comprenant un niveau conceptuel (objectif), un niveau opérationnel (question) et un niveau quantitatif (métrique).
IDEAL	Un modèle d'amélioration organisationnelle qui sert de feuille de route pour initier, planifier et implémenter des actions d'amélioration.
indicateur	Une mesure qui fournit une estimation ou une évaluation d'attributs spécifiés dérivés d'un modèle par rapport à des besoins d'information définis.
mesure	Le nombre ou la catégorie attribué(e) à un attribut d'une entité en effectuant une mesure.
métrique	Une échelle de mesure et la méthode utilisée pour la mesure.
planning poker	Une technique d'estimation basée sur le consensus, principalement utilisée pour estimer l'effort ou la taille relative des histoires d'utilisateurs dans le développement logiciel en mode Agile. Il s'agit d'une variante de la méthode Delphi à large bande qui utilise un jeu de cartes dont les valeurs représentent les unités dans lesquelles l'équipe fait ses estimations.
Estimation à trois points	Il s'agit d'une technique basée sur l'expertise. Trois estimations sont faites par les experts : l'estimation la plus optimiste (a), l'estimation la plus probable (m) et l'estimation la plus pessimiste (b). L'estimation finale (E) est leur moyenne arithmétique pondérée.
Delphi à large bande	Une technique d'estimation de test basée sur l'expertise qui vise à faire une estimation précise en utilisant la sagesse collective des membres de l'équipe.

9 Appendice E – Marques Déposées

CMMI® est une marque déposée aux États-Unis. U.S. Patent and Trademark Office by Carnegie Mellon University.

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

TMMi® est une marque déposée de TMMi Foundation.

TPI-Next® est une marque déposée de Sogeti, The Netherlands.

10 Index

Tous les termes sont définis dans le glossaire de l'ISTQB®. (http://glossary.istqb.org/).