

Examen Blanc – Réponses

Examen Blanc - Ensemble C
Version 1.5

ISTQB® Testeur certifié Niveau Fondation

Compatible avec la version 4.0 du Syllabus

International Software Testing Qualifications Board
Comité canadien en tests logiciels



Avis de droit d'auteur

Avis de droit d'auteur © International Software Testing Qualifications Board (ci-après appelé ISTQB®).

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

Tous droits réservés.

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

Des extraits de ce document peuvent être copiés à des fins non commerciales, à condition que la source soit mentionnée.

Tout organisme de formation accrédité peut utiliser cet examen blanc dans son cours de formation si les auteurs et l'ISTQB® sont reconnus comme la source et les propriétaires des droits d'auteur de l'examen blanc et à condition que toute publicité d'une telle formation ne soit faite qu'après que l'accréditation officielle du matériel de formation ait été reçue d'un comité reconnu de l'ISTQB®.

Tout individu ou groupe d'individus peut utiliser cet examen blanc dans des articles et des livres, à condition que les auteurs et l'ISTQB® soient reconnus comme la source et les détenteurs des droits d'auteur de l'examen blanc.

Toute autre utilisation de cet examen blanc est interdite sans l'accord écrit de l'ISTQB® .

Tout comité membre reconnu par l'ISTQB® peut traduire cet examen blanc à condition de reproduire l'avis de droit d'auteurs susmentionné dans la version traduite de l'examen blanc.

La traduction française est la propriété du CCTL (Comité canadien en tests logiciels).

Responsabilité des documents

Le groupe de travail sur les examens de l'ISTQB® est responsable de ce document.

Ce document est mis à jour par un groupe de travail de l'ISTQB® composé du groupe de travail sur le syllabus et du groupe de travail sur les examens.

Remerciements

Ce document a été réalisé par une équipe de travail de l'ISTQB® Stuart Reid et Adam Roman

L'équipe de travail remercie l'équipe de révision du groupe de travail sur les examens, le groupe de travail sur le programme et les conseils membres pour leurs suggestions et leurs contributions.

Notes sur la version française

Pour le compte du comité canadien en tests logiciels, le partenaire Zentelia, zentelia.com, a réalisé la traduction de cet examen blanc (modèle C) pour le syllabus de testeur certifié ISTQB® niveau fondation, version 4.0.

Historique des révisions

Exemple d'examen – Modèle de présentation des réponses utilisé : Version 2.11 Date : 16 octobre

Version	Date	Remarques
1.5	04 juillet 2024	Modification à la réponse : #20
1.4	29 mai 2024	Correction de la réponse : #17
1.3	14 mars 2024	Correction des réponses #20 et #25
1.2	20 décembre 2023	Alignement sur la version du document ' Questions '
1.1	06 novembre 2023	Correction de "Compatible avec"
1.0	16 octobre 2023	Première version

Table des matières

Avis de droit d'auteur	2
Responsabilité des documents	2
Remerciements	2
Notes sur la version française	3
Historique des révisions	4
Table des matières	5
Introduction	7
Objet de ce document	7
Instructions	7
Clé de réponses	8
Réponses	9
1	9
2	10
3	12
4	14
5	16
6	17
7	19
8	20
9	22
10	23
11	24
12	25
13	26
14	27
15	28
16	30
17	31
18	33
19	34
20	35
21	37
22	38

23.....	39
24.....	40
25.....	41
26.....	42
27.....	43
28.....	44
29.....	45
30.....	46
31.....	47
32.....	48
33.....	49
34.....	50
35.....	51
36.....	52
37.....	53
38.....	54
39.....	55
40.....	56

Introduction

Objet de ce document

Les exemples de questions et de réponses ainsi que les justifications associées dans cet exemple d'examen ont été créés par une équipe d'experts en la matière et de rédacteurs de questions expérimentés dans le but de :

- Assistance à l'ISTQB® Comités membres et comités d'examen dans leurs activités de rédaction de questions
- Fournir aux prestataires de formation et aux candidats aux examens des exemples de questions d'examen

Ces questions ne peuvent pas être utilisées telles quelles dans un examen officiel.

Note, que les examens réels peuvent inclure une grande variété de questions, et cet exemple d'examen **n'est pas** destiné à inclure des exemples de tous les types, styles ou longueurs de questions possibles, cet exemple d'examen peut également être à la fois plus difficile ou moins difficile que n'importe quel examen officiel.

Instructions

Vous trouverez dans ce document :

- Tableau des corrigés, comprenant pour chaque bonne réponse :
 - Niveau K, objectif d'apprentissage et valeur en points
- Ensembles de réponses, y compris pour toutes les questions :
 - Bonne réponse
 - Justification de chaque option de réponse
 - Niveau K, objectif d'apprentissage et valeur en points
- Ensembles de réponses supplémentaires, y compris pour toutes les questions [ne s'applique pas à tous les exemples d'examens] :
 - Bonne réponse
 - Justification de chaque option de réponse (réponse)
 - Niveau K, objectif d'apprentissage et valeur en points

Les questions sont contenues dans un document séparé.

Clé de réponses

Numéro de la Question (#)	Bonne réponse	LO	Niveau K	Points
1	b	FL-1.1.1	K1	1
2	c	FL-1.1.2	K2	1
3	b	FL-1.3.1	K2	1
4	b, e	FL-1.4.1	K2	1
5	a	FL-1.4.3	K2	1
6	c	FL-1.4.5	K2	1
7	b	FL-1.5.2	K1	1
8	a	FL-1.5.3	K2	1
9	d	FL-2.1.2	K1	1
10	d	FL-2.1.3	K1	1
11	b	FL-2.1.5	K2	1
12	c	FL-2.1.6	K2	1
13	d	FL-2.2.1	K2	1
14	b	FL-2.2.3	K2	1
15	d	FL-3.1.3	K2	1
16	a	FL-3.2.1	K1	1
17	b	FL-3.2.4	K2	1
18	b	FL-3.2.5	K1	1
19	c	FL-4.1.1	K2	1
20	a	FL-4.2.1	K3	1

Numéro de la Question (#)	Bonne réponse	LO	Niveau K	Points
21	d	FL-4.2.2	K3	1
22	d	FL-4.2.3	K3	1
23	a	FL-4.2.4	K3	1
24	c	FL-4.3.2	K2	1
25	a	FL-4.3.3	K2	1
26	b	FL-4.4.1	K2	1
27	d	FL-4.4.3	K2	1
28	b	FL-4.5.2	K2	1
29	d	FL-4.5.3	K3	1
30	a	FL-5.1.1	K2	1
31	c	FL-5.1.4	K3	1
32	a	FL-5.1.5	K3	1
33	b	FL-5.1.6	K1	1
34	d	FL-5.1.7	K2	1
35	c	FL-5.2.3	K2	1
36	b	FL-5.3.2	K2	1
37	d	FL-5.4.1	K2	1
38	b	FL-5.5.1	K3	1
39	d	FL-6.1.1	K2	1
40	d	FL-6.2.1	K1	1

Réponses

Numéro de la Question (#)	Bonne réponse	Explications / Justifications	Objectifs d'apprentissage (LO)	Points	Nombre de points
1	b	<p>a) Ce n'est pas correct. Valider que les exigences documentées sont satisfaites est incorrect car la validation concerne la satisfaction des exigences et des attentes des utilisateurs, tandis que la vérification concerne la satisfaction des exigences spécifiées. Cela serait donc correct si nous remplacions « valider » par « vérifier ».</p> <p>b) Est exact. Provoquer des pannes et identifier des défauts est probablement l'objectif le plus courant des tests dynamiques.</p> <p>c) Ce n'est pas correct. Initier des erreurs et identifier les causes profondes est incorrect car les testeurs ne déclenchent pas d'erreurs, ils essaient de provoquer des échecs. Les erreurs sont généralement commises par les développeurs (et ne peuvent pas vraiment être initiées) et entraînent des défauts, que les testeurs tentent d'identifier soit directement via des tests statiques, soit indirectement via des échecs lors de tests dynamiques. L'identification des causes profondes est utile mais fait partie du débogage, qui est une activité distincte des tests.</p> <p>d) Ce n'est pas correct. Vérifier que l'objet de test répond aux attentes de l'utilisateur est incorrect, car la vérification concerne la vérification du respect des exigences spécifiées (documentées), tandis que la validation concerne la satisfaction des exigences et des attentes de l'utilisateur. Cela serait donc correct si nous remplacions « vérifier » par « valider ».</p>	FL-1.1.1	K1	1

Numéro de la Question (#)	Bonne réponse	Explications / Justifications	Objectifs d'apprentissage (LO)	Points	Nombre de points
2	c	<p>a) N'est pas correct. Le test dynamique peut effectivement provoquer des défaillances (à partir desquelles les défauts peuvent ensuite être localisés et corrigés). Cependant, le débogage est axé sur la localisation des défauts et la correction de ces défauts. Par conséquent, le débogage ne répare pas les défaillances.</p> <p>b) Le test et le débogage contribuent tous deux à améliorer la qualité de l'objet testé, donc ils devraient vraiment tous deux être considérés de manière positive. Le débogage est généralement considéré comme une activité positive car il corrige quelque chose. Le test dynamique implique intentionnellement de provoquer la défaillance de l'objet testé, c'est pourquoi certaines personnes le considèrent comme une activité négative, mais c'est une vision très limitée (et pas typiquement partagée par les testeurs). Des cas de test positifs et négatifs sont possibles. Les cas de test positifs vérifient que l'objet testé effectue correctement ce qu'il est censé faire, tandis que le test négatif vérifie que l'objet testé ne fait pas ce qu'il n'est pas supposé faire.</p> <p>c) Est correct. Le test détermine l'existence de défauts soit directement par l'observation du défaut lors de revues (ou par un outil dans l'analyse statique), soit indirectement en provoquant une défaillance par le test dynamique. Le débogage est une activité séparée du test (normalement effectuée par les développeurs) et concerne la localisation des défauts (uniquement pour le test dynamique) et la correction des défauts.</p> <p>d) N'est pas correct. Les causes des défauts sont typiquement des</p>	FL-1.1.2	K2	1

		<p>erreurs humaines. Le test trouve des défauts soit directement par le test statique, soit indirectement en provoquant des défaillances dans le test dynamique, et le débogage corrige les défauts. Ainsi, le test ne trouve pas la cause des défauts et le débogage ne corrige pas les causes des défauts.</p>			
--	--	--	--	--	--

Numéro de la Question (#)	Bonne réponse	Explications / Justifications	Objectifs d'apprentissage (LO)	Points	Nombre de points
3	b	<p>L'illusion de l'absence de défauts concerne l'idée que garantir la conformité aux exigences (c'est-à-dire, vérifier l'absence de défauts d'implémentation) ne garantit pas la satisfaction des utilisateurs avec le système. Pour y remédier, il est également nécessaire de valider que le système répond aux besoins et attentes des utilisateurs, atteint les objectifs commerciaux et surpasse les systèmes concurrents.</p> <p>a) N'est pas correct. Le principe « le test montre la présence, et non l'absence, de défauts » explique que, bien que le test puisse détecter l'existence de défauts dans l'objet testé, il n'est pas possible de démontrer qu'il n'y a aucun défaut et, par conséquent, de garantir sa justesse. Par conséquent, expliquer qu'il est impossible pour les tests de montrer l'absence de défauts aborderait partiellement ce principe, et non l'illusion de « l'absence de défauts ».</p> <p>b) Est correct. En soutenant l'utilisateur final pour réaliser le test d'acceptation, il devrait être possible de valider que le système répond aux besoins et attentes des utilisateurs.</p> <p>c) N'est pas correct. Il n'est pas possible de garantir qu'aucun défaut d'implémentation ne subsiste dans le système livré, car le principe « le test montre la présence, et non l'absence, de défauts » explique que, bien que le test puisse détecter l'existence de défauts dans l'objet testé, il n'est pas possible de démontrer qu'il n'y a aucun défaut et, par conséquent, de garantir sa justesse.</p> <p>d) Ce n'est pas correct. Modifier les tests qui ne provoquent aucune défaillance pour garantir que peu de défauts subsistent est une</p>	FL-1.3.1	K2	1

		<p>façon de répondre au principe de « l'usure des tests ». Ce principe repose sur l'idée selon laquelle il est peu probable que la répétition de tests identiques sur du code non modifié révèle de nouveaux défauts et que, par conséquent, la modification des tests peut être essentielle. Cela ne validera pas que le système répond aux besoins et aux attentes des utilisateurs.</p>			
--	--	--	--	--	--

Numéro de la Question (#)	Bonne réponse	Explications / Justifications	Objectifs d'apprentissage (LO)	Points	Nombre de points
4	b, e	<p>Compte tenu de la description suivante de l'analyse des tests : Pour identifier les fonctionnalités qui nécessitent des tests, la base de test est analysée et définie sous forme de conditions de test, qui sont ensuite hiérarchisées avec les risques associés. L'identification systématique des conditions de test en tant qu'éléments de couverture implique souvent l'utilisation de techniques de test à la fois pendant l'analyse des tests et dans le cadre de l'activité de conception des tests.</p> <p>D'après la description ci-dessus, il apparaît que les techniques de test sont souvent utilisées dans les activités d'analyse de test et de conception de test. L'analyse des valeurs limites et le partitionnement d'équivalence sont des techniques de test</p> <ul style="list-style-type: none"> a) N'est pas correct. L'implémentation du test n'est probablement pas susceptible d'impliquer l'utilisation de techniques de test car elle concerne principalement l'assemblage des cas de test en procédures de test, tandis que les techniques de test créent des cas de test. b) Est correct. La conception de test est susceptible d'impliquer l'utilisation de techniques de test pour créer des cas de test à partir des conditions de test et des éléments de couverture. c) N'est pas correct. L'exécution du test n'est probablement pas susceptible d'impliquer l'utilisation de techniques de test car elle concerne principalement l'exécution des procédures de test (et donc des cas de test), tandis que les techniques de test créent des cas de test. d) N'est pas correct. La surveillance du test n'est probablement pas susceptible d'impliquer l'utilisation de techniques de test. La surveillance du test concerne principalement les vérifications en 	FL-1.4.1	K2	1

		<p>cours pour s'assurer que le plan est suivi, tandis que les techniques de test créent des cas de test.</p> <p>e) Est correct. L'analyse de test est susceptible d'impliquer l'utilisation de techniques de test pour identifier les conditions de test</p>			
--	--	--	--	--	--

Numéro de la Question (#)	Bonne réponse	Explications / Justifications	Objectifs d'apprentissage (LO)	Points	Nombre de points
5	a	<p>Compte tenu de chacune des activités de test répertoriées et de leur logiciel de test de sortie :</p> <ul style="list-style-type: none"> a) Analyse des tests - conditions de test priorisées (4) (par exemple, critères d'acceptation), et rapports de défauts pour les défauts identifiés dans la base de test b) c) Conception des tests - cas de test priorisés, chartes de test, éléments de couverture (1), exigences relatives aux données de test et exigences relatives à l'environnement de test d) Implémentation des tests - procédures de test, scripts de test automatisés, suites de test, données de test, calendrier d'exécution des tests (3), et éléments de l'environnement de test tels que les bouchons, les pilotes, les simulateurs et les virtualisations de service e) Clôture des tests - rapport de clôture des tests, leçons apprises documentées, actions d'amélioration, et demandes de changement (2) (comme éléments du carnet de produit) <p>Ainsi:</p> <ul style="list-style-type: none"> a) Est correct. La correspondance correcte est : 1B, 2D, 3C, 4A b) N'est pas correct c) N'est pas correct d) N'est pas correct 	FL-1.4.3	K2	1

Numéro de la Question (#)	Bonne réponse	Explications / Justifications	Objectifs d'apprentissage (LO)	Points	Nombre de points
6	c	<p>a) N'est pas correct. Bien qu'il soit exact de dire que, dans le développement logiciel Agile, certaines des tâches de gestion des tests peuvent être prises en charge par l'équipe Agile elle-même, le rôle de testeur n'est pas principalement la responsabilité d'un seul individu extérieur à l'équipe. Au contraire, les tests sont plus susceptibles d'être effectués par divers membres de l'équipe, suivant l'Approche équipe intégrée.</p> <p>b) N'est pas correct. Le rôle de test manager implique principalement des activités liées à la planification des tests, à la surveillance et au contrôle des tests, et à la clôture des tests. Ainsi, bien que cette affirmation soit partiellement correcte, il est erroné de dire que le rôle de testeur est principalement responsable de la surveillance et du contrôle des tests.</p> <p>c) Est correct. Dans le développement logiciel Agile, certaines des tâches de gestion des tests peuvent être prises en charge par l'équipe Agile elle-même. Cependant, pour les activités de test qui concernent plusieurs équipes au sein d'une organisation, les test managers extérieurs à l'équipe de développement peuvent effectuer ces tâches.</p> <p>d) N'est pas correct. Le rôle de test manager implique principalement des activités liées à la planification des tests, à la surveillance et au contrôle des tests, et à la clôture des tests, tandis que le rôle de testeur est principalement responsable des aspects techniques et d'ingénierie des tests, tels que l'analyse des tests, la conception des tests, l'implémentation des tests et l'exécution des tests. Ainsi, le rôle de test manager n'est normalement pas responsable de l'analyse et</p>	FL-1.4.5	K2	1

		de la conception des tests, bien qu'il soit exact de dire que le rôle de testeur est principalement responsable de l'implémentation et de l'exécution des tests.			
--	--	--	--	--	--

Numéro de la Question (#)	Bonne réponse	Explications / Justifications	Objectifs d'apprentissage (LO)	Points	Nombre de points
7	b	<p>a) N'est pas correct. Dans l'approche équipe intégrée, les testeurs jouent un rôle vital en partageant leur expertise en matière de tests avec l'équipe et en guidant le développement du produit. Ils collaborent avec les autres membres de l'équipe pour atteindre les niveaux de qualité souhaités et travaillent avec les représentants des affaires(métiers) pour créer des tests d'acceptation. Les testeurs s'associent également aux développeurs pour déterminer la stratégie de test optimale et les approches d'automatisation.</p> <p>b) Est correct. En tirant parti des compétences diversifiées de chaque membre de l'équipe de manière la plus efficace, l'approche équipe intégrée favorise une dynamique d'équipe supérieure, promeut une communication et une collaboration robustes, et génère un effet synergique qui profite à l'ensemble du projet.</p> <p>c) N'est pas correct. L'approche équipe intégrée permet à tout membre de l'équipe possédant les compétences et connaissances requises d'entreprendre toute tâche, ainsi les membres spécialisés de l'équipe ne constituent pas un avantage de cette approche.</p> <p>d) N'est pas correct. Il n'y a pas de directives spécifiques concernant la taille optimale des équipes utilisant l'approche équipe intégrée, et il n'est pas suggéré que des équipes plus grandes soient meilleures.</p>	FL-1.5.2	K1	1

Numéro de la Question (#)	Bonne réponse	Explications / Justifications	Objectifs d'apprentissage (LO)	Points	Nombre de points
8	a	<p>a) Est correct. L'avantage principal de l'indépendance dans les tests est que les testeurs sont plus susceptibles d'identifier différents types de défaillances et de défauts par rapport aux développeurs, en raison de leurs antécédents variés, de leurs points de vue techniques et de leurs biais potentiels, y compris le biais cognitif. Cependant, l'inconvénient principal de l'indépendance dans les tests est que les testeurs peuvent devenir isolés de l'équipe de développement, menant à des problèmes de communication, un manque de collaboration et potentiellement une relation conflictuelle, avec les testeurs étant blâmés pour les retards et les goulets d'étranglement dans le processus de sortie.</p> <p>b) N'est pas correct. La familiarité d'un développeur avec le code ne signifie pas qu'ils trouvent rarement des défauts dans celui-ci, au contraire cette familiarité signifie qu'ils peuvent trouver efficacement de nombreux défauts dans leur propre code. Et, plutôt que les développeurs et les testeurs ayant un background partagé, le fait que les développeurs aient un background différent des testeurs est normalement cité comme la raison pour laquelle les testeurs et les développeurs trouvent différents types de défauts.</p> <p>c) N'est pas correct. Les tests peuvent être conduits à différents niveaux d'indépendance, allant d'une absence d'indépendance pour l'auteur à une très haute indépendance pour les testeurs extérieurs à l'organisation. Dans la plupart des projets, plusieurs niveaux d'indépendance sont utilisés, avec les développeurs effectuant les tests de composant et d'intégration de composant, l'équipe de test effectuant les tests de système et d'intégration de système, et les représentants des affaires effectuant les tests d'acceptation. Ainsi, les</p>	FL-1.5.3	K2	1

		<p>testeurs peuvent être dans l'équipe de développeurs et n'ont pas besoin de venir de l'extérieur de l'organisation. La connaissance du domaine d'application changera d'un cas à l'autre et n'est pas dépendante du niveau d'indépendance.</p> <p>d) N'est pas correct. Les tests peuvent être effectués à différents niveaux d'indépendance, allant d'une absence d'indépendance pour l'auteur à une très haute indépendance pour les testeurs extérieurs à l'organisation, avec les testeurs extérieurs à l'équipe de développeurs généralement plus indépendants que les testeurs au sein de l'équipe. Cependant, il y a plus de raisons de croire que les testeurs extérieurs à l'équipe sont susceptibles d'être plus isolés des développeurs et donc plus susceptibles d'être blâmés pour les retards dans la sortie du produit.</p>			
--	--	--	--	--	--

Numéro de la Question (#)	Bonne réponse	Explications / Justifications	Objectifs d'apprentissage (LO)	Points	Nombre de points
---------------------------	---------------	-------------------------------	--------------------------------	--------	------------------

9	d	<p>a)Ce n'est pas correct. Le contrôle qualité s'applique à toutes les activités de développement, ce qui signifie que chaque activité de développement logiciel correspond à une activité de test. Cependant, nous essayons ici d'assimiler les niveaux de test aux niveaux de développement et, bien que nous sachions ce que l'on entend par « niveaux de test », il n'existe pas de compréhension commune du terme « niveau de développement ».</p> <p>b)N'est pas correct. Chaque activité de développement logiciel a une activité de test correspondante ; cependant, les objectifs de test sont assez différents. Par exemple, il pourrait y avoir un objectif de test assurant qu'un objet de test adhère à une exigence contractuelle qu'un certain type de test doit être effectué avant la livraison. Dans ce cas, il n'y a aucune raison pour qu'il y ait un objectif de développement correspondant.</p> <p>c)N'est pas correct. Le contrôle de qualité s'applique à toutes les activités de développement, signifiant que chaque activité de développement logiciel a une activité de test correspondante. Cependant, la même symétrie ne s'applique pas aux activités de test et aux activités des utilisateurs. Par exemple, pour certains systèmes, il est difficile d'identifier même les utilisateurs finaux. De plus, certaines activités de test sont focalisées sur les développeurs (par exemple, le test pour la facilité de maintenance), ce qui n'a aucun aspect utilisateur.</p> <p>d)Est correct. Le contrôle de qualité s'applique à toutes les activités de développement, signifiant que chaque activité de développement logiciel a une activité de test correspondante.</p>	FL-2.1.2	K1	1
---	---	---	----------	----	---

Numéro de la Question (#)	Bonne réponse	Explications / Justifications	Objectifs d'apprentissage (LO)	Points	Nombre de points
10	d	<p>a) N'est pas correct. Le Développement piloté par les tests de composants n'est pas un exemple correct d'une approche de développement pilotée par les tests.</p> <p>b) N'est pas correct. Le Développement piloté par les tests d'intégration n'est pas un exemple correct d'une approche de développement pilotée par les tests.</p> <p>c) N'est pas correct. Le Développement piloté par les tests système n'est pas un exemple correct d'une approche de développement pilotée par les tests.</p> <p>d) Est correct. Le Développement piloté par les tests d'acceptation (ATDD) est un exemple bien connu d'une approche de développement pilotée par les tests.</p>	FL-2.1.3	K1	1

Numéro de la Question (#)	Bonne réponse	Explications / Justifications	Objectifs d'apprentissage (LO)	Points	Nombre de points
11	b	<p>a) N'est pas correct. Les pratiques impliquées dans le test shift left visent à mettre en œuvre davantage d'activités de test dans les phases précoces du cycle de vie du développement, représentant le SDLC comme se déplaçant de gauche à droite. Il n'existe pas de côté gauche du processus de test.</p> <p>b) Est correct. Le shift left souligne l'importance de commencer les tests plus tôt dans le cycle de vie du développement logiciel (SDLC). La mise en œuvre du test shift left nécessite une formation supplémentaire, ainsi qu'un effort et des coûts accrus durant les premières étapes du SDLC, néanmoins, les économies globales devraient être plus élevées.</p> <p>c) N'est pas correct. Bien que les tests automatisés de composants et les tests d'intégration de composants pour les tests de régression soient généralement précieux, la création de ces tests est normalement la responsabilité des développeurs, et si une approche d'intégration continue/livraison continue (CI/CD) est suivie, alors ces tests auront été soumis avec le code. Dans certaines situations, le testeur peut automatiser des tests pour les tests de régression, et parfois même pour les tests de composants et les tests d'intégration de composants, cependant cela ne fait pas partie d'une approche «shift left» qui déplace les tests plus tôt dans le SDLC.</p> <p>d) N'est pas correct. Former les testeurs à effectuer des tâches tôt dans le SDLC soutiendrait une approche shift left en soulignant l'importance de commencer les tests plus tôt dans le SDLC. Cependant, automatiser davantage d'activités de test à effectuer plus tard dans le SDLC ne fait pas partie d'une approche «shift left».</p>	FL-2.1.5	K2	1

Numéro de la Question (#)	Bonne réponse	Explications / Justifications	Objectifs d'apprentissage (LO)	Points	Nombre de points
12	c	<p>a) N'est pas correct. L'un des objectifs des rétrospectives est d'identifier les améliorations de processus potentielles qui, si mises en pratique, devraient aboutir à une qualité supérieure des futurs résultats du processus de développement (objets de test). Ainsi, cela est susceptible de se produire à la suite d'une rétrospective.</p> <p>b) N'est pas correct. L'un des avantages des rétrospectives pour les tests comprend l'augmentation de l'efficacité des tests grâce aux améliorations de processus. Ainsi, cela est susceptible de se produire à la suite d'une rétrospective.</p> <p>c) Est correct. Les participants aux rétrospectives incluent typiquement les testeurs, les développeurs, les architectes, les propriétaires de produit et les analystes d'affaires, mais les utilisateurs finaux sont rarement invités ou assistent à ces réunions – et il est également peu probable qu'ils reçoivent des rapports de ces réunions. Ainsi, il est très peu probable qu'ils apprennent et comprennent davantage les processus de développement et de test à travers les rétrospectives.</p> <p>d) N'est pas correct. L'un des avantages des rétrospectives pour les tests comprend l'amélioration de la qualité du testware (y compris les scripts de test automatisés) grâce aux revues conjointes avec les développeurs. Ainsi, cela est susceptible de se produire à la suite d'une rétrospective.</p>	FL-2.1.6	K2	1

Numéro de la Question (#)	Bonne réponse	Explications / Justifications	Objectifs d'apprentissage (LO)	Points	Nombre de points
13	d	<p>a) N'est pas correct. Les tests de composants (également appelés tests unitaires) impliquent de tester des composants individuels de manière isolée et consistent principalement en une vérification par rapport à une spécification, plutôt qu'une validation par rapport aux besoins des utilisateurs. Cependant, ces tests ne sont normalement pas effectués par les testeurs, car les développeurs effectuent généralement ces tests dans leur environnement de développement.</p> <p>b) N'est pas correct. Les tests d'intégration de composants impliquent de tester les interfaces et les interactions entre les composants et consistent principalement en une vérification par rapport à une spécification, plutôt qu'une validation par rapport aux besoins des utilisateurs. Cependant, ces tests ne sont normalement pas effectués par les testeurs, car les développeurs effectuent généralement ces tests.</p> <p>c) N'est pas correct. Les tests d'intégration système examinent les interfaces avec d'autres systèmes et services externes et consistent principalement en une vérification par rapport à une spécification, plutôt qu'une validation par rapport aux besoins des utilisateurs. Ce type de tests est également le plus souvent effectué par des testeurs.</p> <p>d) Est correct. Les tests d'acceptation visent à valider que le système répond aux besoins commerciaux de l'utilisateur et qu'il est prêt à être déployé. Idéalement, ces tests sont effectués par les utilisateurs finaux.</p>	FL-2.2.1	K2	1

Numéro de la Question (#)	Bonne réponse	Explications / Justifications	Objectifs d'apprentissage (LO)	Points	Nombre de points
14	b	<p>a) N'est pas correct. Il est nécessaire d'effectuer des tests de confirmation pour vérifier que les mises à jour ont abouti à une implémentation correcte, cependant, il serait ensuite judicieux de réaliser des tests de régression pour s'assurer qu'aucun défaut n'a été introduit ou découvert dans des zones inchangées du système.</p> <p>b) Est correct. Les tests de confirmation vérifieront que les mises à jour ont abouti à une implémentation correcte, puis les tests de régression seront utilisés pour s'assurer qu'aucun défaut n'a été introduit ou découvert dans des zones inchangées du système.</p> <p>c) N'est pas correct. Les tests de régression devraient être utilisés pour s'assurer qu'aucun défaut n'a été introduit ou découvert dans des zones inchangées du système lors de la mise à jour, cependant, il est également nécessaire d'effectuer des tests de confirmation qui vérifieront que les mises à jour ont abouti à une implémentation correcte.</p> <p>d) N'est pas correct. Les tests de confirmation vérifieront que les mises à jour ont abouti à une implémentation correcte, et les tests de régression seront utilisés pour s'assurer qu'aucun défaut n'a été introduit ou découvert dans des zones inchangées du système. Cependant, lorsqu'ils sont effectués (c'est-à-dire lorsqu'une mise à jour doit être testée), les tests de confirmation précèdent les tests de régression.</p>	FL-2.2.3	K2	1

Numéro de la Question (#)	Bonne réponse	Explications / Justifications	Objectifs d'apprentissage (LO)	Points	Nombre de points
15	d	<p>En considérant chacun des défauts exemples listés :</p> <ul style="list-style-type: none"> i. Deux parties différentes de la spécification de conception sont en désaccord en raison de la complexité de la conception : il s'agit d'un exemple de défaut de spécification, qui comprend des incohérences, des ambiguïtés, contradictions, omissions, inexactitudes et duplications, qui peuvent être plus facilement détectées par des tests statiques. ii. Un temps de réponse est trop long et fait perdre patience aux utilisateurs. Il s'agit d'un exemple de défaut de temps de réponse, qui ne peut être détecté en pratique qu'en exécutant le programme et en mesurant le temps de réponse, ce qui peut être trouvé plus facilement par des tests dynamiques. iii. Un chemin dans le code ne peut pas être atteint lors de l'exécution - ceci est un exemple de défaut de codage, qui inclut des variables avec des valeurs non définies, des variables non déclarées, dupliquées ou code inaccessible et complexité excessive du code, qui peut être trouvée plus facilement par des tests statiques iv. Une variable est déclarée mais n'est jamais utilisée par la suite dans le programme. Il s'agit d'un exemple de défaut de codage, qui inclut des variables avec des valeurs non définies, des variables non déclarées, du code dupliqué ou inaccessible et une complexité de code excessive, qui peuvent être détectées plus facilement par des tests statiques. v. La quantité de mémoire nécessaire au programme pour générer un rapport est trop élevée – ceci est un exemple de défaut de performance, qui ne peut être détecté en pratique qu'en exécutant le programme et en mesurant la mémoire utilisée, qui peut être trouvée le plus facilement par tests dynamiques. 	FL-3.1.3	K2	1

		<p>Ainsi:</p> <ul style="list-style-type: none">a) N'est pas correctb) N'est pas correctc) N'est pas correctd) Est correct. La correspondance correcte pour les tests statiques est i, iii et iv			
--	--	---	--	--	--

Numéro de la Question (#)	Bonne réponse	Explications / Justifications	Objectifs d'apprentissage (LO)	Points	Nombre de points
16	a	<p>a) Est correct. Obtenir des retours d'information des parties prenantes tôt et souvent dans le processus de développement logiciel peut être très bénéfique. Cela facilite la communication précoce des problèmes de qualité potentiels, peut prévenir les malentendus concernant les exigences, et assure que tout changement dans les exigences des parties prenantes est compris et mis en œuvre plus tôt.</p> <p>b) N'est pas correct. Le retour d'information vient des parties prenantes, et le fait qu'elles fournissent des retours est peu susceptible d'améliorer leur compréhension de leurs propres exigences utilisateur.</p> <p>c) N'est pas correct. Obtenir des retours d'information des parties prenantes tôt et souvent dans le processus de développement logiciel peut être très bénéfique. Cela facilite la communication précoce des problèmes de qualité potentiels, peut prévenir les malentendus concernant les exigences, et assure que tout changement dans les exigences des parties prenantes est compris et mis en œuvre plus tôt. Cependant, le fait que les changements d'exigences puissent être compris et mis en œuvre plus tôt ne signifie pas que des changements illimités aux exigences sont encouragés.</p> <p>d) N'est pas correct. Le retour d'information vient des parties prenantes et ne couvre pas la communication vers elles. Les communications aux utilisateurs finaux pourraient inclure le fait de les informer des exigences qui ne seront pas mises en œuvre avant la sortie, mais idéalement cela ne devrait pas se produire du tout.</p>	FL-3.2.1	K1	1

Numéro de la Question (#)	Bonne réponse	Explications / Justifications	Objectifs d'apprentissage (LO)	Points	Nombre de points
17	b	<p>En considérant chacun des types de revue listés :</p> <ol style="list-style-type: none"> 1. Revue technique - Ce type de revue est réalisé par des réviseurs techniquement qualifiés et dirigé par un modérateur. Les objectifs sont de parvenir à un consensus et de prendre des décisions sur des problèmes techniques tout en évaluant également la qualité et en renforçant la confiance dans le produit de travail, en générant de nouvelles idées, en motivant et en permettant aux auteurs de s'améliorer, et en détectant des anomalies. 2. Revue informelle - L'objectif principal est de détecter des anomalies. Le processus n'est pas défini et ne nécessite pas de sortie documentée formelle. 3. Inspection - Il s'agit du type de revue le plus formel, et elle suit le processus de revue générique complet. L'objectif principal est de trouver le plus d'anomalies possible, et d'autres objectifs incluent l'évaluation de la qualité et le renforcement de la confiance dans le produit de travail, la motivation et la capacité des auteurs à s'améliorer, et la collecte de métriques qui peuvent être utilisées pour améliorer le cycle de vie du développement logiciel (SDLC), y compris le processus d'inspection. L'auteur ne peut pas agir en tant que leader de la revue ou scribe. 4. Relecture technique (Walkthrough) - Dirigée par l'auteur, ce type de revue sert divers objectifs tels que l'évaluation de la qualité et le renforcement de la confiance dans le produit de travail, l'éducation des réviseurs, l'obtention d'un consensus, la génération de nouvelles idées, la motivation et la capacité des auteurs à s'améliorer, et la détection d'anomalies. Les réviseurs peuvent effectuer une revue individuelle avant la relecture technique, mais 	FL-3.2.4	K2	1

		<p>cela n'est pas obligatoire.</p> <ul style="list-style-type: none">A. Comprend des objectifs tels que parvenir à un consensus, générer de nouvelles idées et motiver les auteurs à s'améliorerB. Comprend des objectifs tels que former les réviseurs, parvenir à un consensus, générer de nouvelles idées et détecter des défauts potentiels.C. L'objectif principal est de détecter des défauts potentiels et cela nécessite la collecte de métriques pour soutenir l'amélioration du processusD. L'objectif principal est de détecter des défauts potentiels et cela ne génère aucun résultat formellement documenté.			
--	--	---	--	--	--

Numéro de la Question (#)	Bonne réponse	Explications / Justifications	Objectifs d'apprentissage (LO)	Points	Nombre de points
		Ainsi: a) N'est pas correct b) Est correct. c) N'est pas correct d) N'est pas correct			
18	b	a) N'est pas correct. Pour garantir la réussite des revues, il est important de sécuriser le soutien de la direction pour le processus de revue, cependant cela ne signifie pas qu'ils doivent participer en tant que réviseurs. b) Est correct. Pour garantir la réussite des revues, il est important de diviser le produit de travail en parties suffisamment petites pour être examinées dans un délai raisonnable afin d'éviter que les réviseurs ne perdent leur concentration pendant les revues individuelles ou les réunions de revue. c) N'est pas correct. Pour garantir la réussite des revues, il est important de définir clairement les objectifs et les critères de sortie mesurables, sans évaluer les participants. d) N'est pas correct. Pour garantir la réussite des revues, il est important de décomposer la revue en morceaux plus petits pour éviter que les réviseurs ne perdent leur concentration pendant les revues individuelles ou les réunions de revue. Donc, vous ne devriez pas prévoir de couvrir un document par revue.	FL-3.2.5	K1	1

Numéro de la Question (#)	Bonne réponse	Explications / Justifications	Objectifs d'apprentissage (LO)	Points	Nombre de points
19	c	<p>a) N'est pas correct. Dans la plupart des cas, les techniques de test boîte noire et les techniques de test basées sur l'expérience peuvent être utilisées pour les mêmes objets de test.</p> <p>b) N'est pas correct. Les techniques de test boîte noire et les techniques de test basées sur l'expérience peuvent être utilisées à tous les niveaux de test.</p> <p>c) Est correct. Les techniques de test boîte noire (également connues sous le nom de techniques basées sur la spécification) sont basées sur une analyse du comportement spécifié de l'objet de test sans référence à sa structure interne. Ainsi, la base de test est généralement une spécification. Les techniques de test basées sur l'expérience utilisent efficacement les connaissances et l'expérience des testeurs pour la conception et la mise en œuvre des cas de test. Cela signifie que le testeur, lors de la conception des tests, peut ne pas utiliser du tout la spécification.</p> <p>d) N'est pas correct. Les techniques de test basées sur l'expérience peuvent détecter des défauts qui peuvent être manqués en utilisant des techniques de test boîte noire (et boîte blanche). Par conséquent, les techniques de test basées sur l'expérience sont complémentaires aux techniques de test boîte noire et boîte blanche, et les deux, techniques de test boîte noire et basées sur l'expérience, peuvent être utilisées dans tous les SDLCs.</p>	FL-4.1.1	K2	1

Numéro de la Question (#)	Bonne réponse	Explications / Justifications	Objectifs d'apprentissage (LO)	Points	Nombre de points
20	a	<p>Il existe cinq partitions d'équivalence :</p> <p>(Longueur d'entrée) :</p> <p style="padding-left: 40px;">L1 : Entre 1 et 3 L2 : 4 L3 : Plus de 4</p> <p>(Chiffres d'entrée) :</p> <p style="padding-left: 40px;">D1 : NIP composé uniquement des mêmes chiffres D2 : NIP avec au moins deux chiffres différents</p> <p>(Sortie) Deux partitions :</p> <p style="padding-left: 40px;">G1 : NIP valide (satisfait à la fois L2 et D2) G2 : NIP invalide</p> <p>a) Est correct. 112 couvre D2, L1, G2 ; 1111 couvre D1, L2, G2 ; 1234 couvre D2, L2, G1 ; 123456 couvre D2, L3, G2</p> <p>b) N'est pas correct. Il ne couvre pas L3. 1 couvre D1, L1, G2 ; 123 couvre D2, L1, G2 ; 1111 couvre D1, L2, G2 ; 12345 couvre D2, L2, G1</p> <p>c) N'est pas correct. Il ne couvre pas D1. 12 couvre D2, L1, G2 ; 112 couvre D2, L1, G2 ; 1112 couvre D2, L2, G1 ; 11112 couvre D2, L3, G2</p> <p>d) N'est pas correct. Il ne couvre pas D2 (et G1). 1 couvre D1, L1, G2 ; 111 couvre D1, L1, G2 ; 1111 couvre D1, L2,</p>	FL-4.2.1	K3	1

		G2 ; 11111 couvre D1, L3, G2			
--	--	------------------------------	--	--	--

Numéro de la Question (#)	Bonne réponse	Explications / Justifications	Objectifs d'apprentissage (LO)	Points	Nombre de points
21	d	<p>Les partitions d'équivalence sont : {..., 99, 100}, {101, 102, ..., 198, 199}, {200, 201, ...}.</p> <p>Ainsi, il existe 4 valeurs limites, qui sont : 100, 101, 199 et 200. Dans l'analyse des valeurs limites, la technique des deux valeurs stipule que, pour chaque valeur limite, il existe deux éléments de couverture (la valeur limite et son voisin le plus proche appartenant à la partition adjacente). Comme les voisins les plus proches sont également des valeurs limites dans la partition adjacente, il n'y a alors que quatre éléments de couverture.</p> <p>Ainsi:</p> <ul style="list-style-type: none"> a) N'est pas correct. Seuls 100 et 200 sont des articles de couverture valides pour l'analyse des valeurs limites à 2 valeurs, nous obtenons donc une couverture de 50 % b) N'est pas correct. Seuls 100 et 200 sont des articles de couverture valides pour l'analyse des valeurs limites à 2 valeurs, nous obtenons donc une couverture de 50 % c) N'est pas correct. Seuls 100 et 101 sont des articles de couverture valides pour l'analyse des valeurs limites à 2 valeurs, nous obtenons donc une couverture de 50 % d) Est correct. 101, 199 et 200 sont des articles de couverture valides pour l'analyse des valeurs limites à 2 valeurs, nous obtenons donc une couverture de 75 % 	FL-4.2.2	K3	1

22	d	<p>a) N'est pas correct. La combinaison (V, V, F) ne correspond à aucune règle. Ceci est un exemple d'omission, pas une contradiction</p> <p>b) N'est pas correct. La combinaison (V, F, V) ne correspond qu'à une seule colonne, R2, il n'y a donc pas de contradiction</p> <p>c) N'est pas correct. Les deux combinaisons (V, V, V) et (F, V, V) ne correspondent qu'à une seule colonne, R1, il n'y a donc pas de contradiction</p> <p>d) Est correct. La combinaison (F, F, F) correspond à la fois à R2 et R3, mais R2 et R3 ont des actions différentes, cela montre donc une contradiction entre R2 et R3.</p>	FL-4.2.3	K3	1
----	---	---	----------	----	---

Numéro de la Question (#)	Bonne réponse	Explications / Justifications	Objectifs d'apprentissage (LO)	Points	Nombre de points
23	a	<p>Les trois transitions suivantes :</p> <p>« REQUESTING -> CONFIRMED » « WAITING LIST -> CONFIRMED » « WAITING LIST -> END »</p> <p>ne peuvent pas apparaître dans le même cas de test, ce qui suggère qu'au moins trois cas de test sont nécessaires. Toutes les autres transitions peuvent apparaître en combinaison avec une ou plusieurs de ces trois transitions, donc nous avons besoin d'un minimum de trois cas de test. En fait, seulement trois séquences sont possibles :</p> <p>TC1 : START (Room request) -> REQUESTING (Available) -> CONFIRMED (Pay) -> END TC2 : START (Room request) -> REQUESTING (Not available) -> WAITING LIST (Available) -> CONFIRMED (Pay) -> END TC3 : START (Room request) -> REQUESTING (Not available) -> WAITING LIST (Cancel) -> END</p> <p>Ainsi :</p> <p>a) Est correct b) N'est pas correct c) N'est pas correct d) N'est pas correct</p>	FL-4.2.4	K3	1

24	c	<p>Dans les tests de branche, les éléments de couverture sont les branches, qui sont représentées par les arêtes d'un graphique de flux de contrôle. Il y a 8 arêtes dans le graphique de flux de contrôle.</p> <p>Ainsi:</p> <ul style="list-style-type: none">a) N'est pas correctb) N'est pas correctc) Est correctd) N'est pas correct	FL-4.3.2	K2	1
----	---	---	----------	----	---

Numéro de la Question (#)	Bonne réponse	Explications / Justifications	Objectifs d'apprentissage (LO)	Points	Nombre de points
25	a	<p>a) Est correct. Réaliser uniquement des tests en boîte noire ne fournit pas une mesure de la couverture réelle du code. Les mesures de couverture en boîte blanche fournissent une mesure objective de la couverture et fournissent les informations nécessaires pour permettre la génération de tests supplémentaires afin d'augmenter cette couverture, et par conséquent d'augmenter la confiance dans le code.</p> <p>b) N'est pas correct. Cette affirmation est correcte, mais elle n'a rien à voir avec les tests boîte noire.</p> <p>c) N'est pas correct. En général, il n'existe pas de relations d'inclusion entre les techniques de boîte blanche et de boîte noire.</p> <p>d) N'est pas correct. Les techniques de boîte blanche sont utilisées pour concevoir des tests basés sur l'objet de test lui-même, tandis que les techniques boîte noire sont utilisées pour concevoir des tests basés sur la spécification. Par conséquent, il n'y a pas de relation entre les éléments de couverture dérivés de ces deux types de techniques.</p>	FL-4.3.3	K2	1

Numéro de la Question (#)	Bonne réponse	Explications / Justifications	Objectifs d'apprentissage (LO)	Points	Nombre de points
26	b	<p>a) N'est pas correct. Le test exploratoire utilise des chartes de test, pas une liste de défauts/défaillances possibles. Bien que le test exploratoire puisse intégrer l'utilisation d'autres techniques de test, dans ce cas, l'injection de défauts est l'option la plus probable.</p> <p>b) Est correct. Il s'agit d'une liste de défaillances potentielles. L'attaque de fautes est une approche méthodique de l'implémentation de l'estimation d'erreurs et nécessite que le testeur crée ou acquiert une liste d'erreurs, de défauts et de défaillances potentielles, et de concevoir des tests qui vont identifier les défauts liés aux erreurs, révéler les défauts, ou provoquer les défaillances.</p> <p>c) N'est pas correct. Le testeur utilise une checklist pour appuyer son processus de test. L'estimation d'erreurs et le test basé sur des checklists utilisent de telles listes, mais la liste en question concerne des défaillances potentielles, et non des conditions de test. Ainsi, la technique de test LA PLUS PROBABLE est l'injection de défauts, qui se concentre sur les erreurs, les défauts et les défaillances.</p> <p>d) N'est pas correct. L'analyse des valeurs limites se base sur l'analyse des valeurs aux frontières des partitions d'équivalence. La liste mentionnée ne fait pas référence aux partitions d'équivalence ou à leurs frontières.</p>	FL-4.4.1	K2	1

Numéro de la Question (#)	Bonne réponse	Explications / Justifications	Objectifs d'apprentissage (LO)	Points	Nombre de points
27	d	<p>a) N'est pas correct. Bien qu'il soit vrai que le testeur peut mettre en œuvre et exécuter des cas de test détaillés basés sur la checklist, cela n'explique pas comment cela pourrait entraîner une couverture accrue.</p> <p>b) N'est pas correct. Les éléments de la checklist ne devraient pas être automatisés. Mais même s'ils le sont, les scripts de test automatisés exécutent toujours les tests de la même manière, ce qui n'entraîne généralement pas une couverture accrue.</p> <p>c) N'est pas correct. Il est vrai que chaque élément de la checklist devrait être testé séparément et indépendamment. Mais cela impacte l'ordre d'exécution des tests et n'affecte pas la couverture atteinte, et donc cela n'entraîne pas une couverture accrue.</p> <p>d) Est correct. Si les checklists sont de haut niveau, une certaine variabilité dans les tests réels est susceptible de se produire, entraînant <u>potentiellement une plus grande couverture</u> mais moins de répétabilité. Si deux testeurs suivent une checklist d'éléments de haut niveau, chacun d'eux peut utiliser des données de test, des étapes de test, etc., différentes. De cette manière, un testeur couvrira probablement certaines zones non couvertes par l'autre testeur et cela entraînera une couverture accrue.</p>	FL-4.4.3	K2	1

Numéro de la Question (#)	Bonne réponse	Explications / Justifications	Objectifs d'apprentissage (LO)	Points	Nombre de points
28	b	<p>a) N'est pas correct. Ce critère d'acceptation décrit quelles règles ou réglementations le système doit respecter (dans ce cas, le droit à l'oubli). C'est un exemple de critère d'acceptation orienté règle.</p> <p>b) Est correct. Ce critère d'acceptation décrit un scénario exemple qui doit être réalisable par le système. C'est un exemple de critère d'acceptation orienté scénario.</p> <p>c) N'est pas correct. Cette phrase ressemble davantage à une ligne de code qui implémente une règle commerciale. Les critères d'acceptation devraient être rédigés en collaboration avec les représentants du métier, et donc devraient être écrits dans un langage qu'ils comprennent. Cette phrase sera très probablement inintelligible pour ces parties prenantes.</p> <p>d) N'est pas correct. Ce critère d'acceptation décrit quelles règles ou réglementations le système doit respecter et comment la conformité sera assurée. Par conséquent, c'est un exemple de critère d'acceptation orienté règle, et non un critère d'acceptation basé sur un scénario.</p>	FL-4.5.2	K2	1

29	d	<p>a)Utilisateurs réguliers, donc nous devons tester les droits d'accès pour un utilisateur Spécial, et non pour un utilisateur Régulier.</p> <p>b)N'est pas correct. Nous voulons vérifier que les utilisateurs Spéciaux ont les droits des utilisateurs Réguliers, donc nous devons tester les droits d'accès pour un utilisateur Spécial, et non pour un utilisateur Régulier.</p> <p>c)N'est pas correct. Aucun étage 5 n'est décrit dans les critères d'acceptation. Les cas de test ne devraient pas étendre le champ de la User Story. Mais même si nous aimerions effectuer un test négatif, ce test n'est pas directement lié à AC3.</p> <p>d)Est correct. De cette façon, nous pouvons vérifier si un utilisateur Spécial peut accéder aux étages accessibles à un utilisateur Régulier.</p>	FL-4.5.3	K3	1
----	---	--	----------	----	---

Numéro de la Question (#)	Bonne réponse	Explications / Justifications	Objectifs d'apprentissage (LO)	Points	Nombre de points
30	a	<p>a) Est correct. Le plan de test peut inclure les exigences en matière de données de test (comme partie de l'approche de test), mais pas les données de test détaillées pour les cas de test. Les données de test font partie des cas de test, pas du plan de test. De plus, il est généralement impossible de définir de telles données lors de la création du plan de test, car on ne sait pas exactement à quoi ressembleront les composants.</p> <p>b) N'est pas correct. L'un des objectifs d'un plan de test est d'aider à garantir que les activités de test effectuées répondront aux critères établis, en incluant des critères d'entrée et de sortie. Les critères de couverture de code sont un exemple de tels critères pour le niveau de test de composant.</p> <p>c) N'est pas correct. Les modèles de documentation sont un contenu typique d'un plan de test. Cela aide à faciliter la communication entre les parties prenantes en définissant une manière standard de communiquer ou de rapporter.</p> <p>d) N'est pas correct. L'un des objectifs d'un plan de test est de démontrer que les tests adhéreront à la politique de test et à la stratégie de test existantes, ou d'expliquer pourquoi les tests dévieront de celles-ci. C'est un exemple d'explication de l'écart, concernant les niveaux de test qui seront (ou ne seront pas) suivis.</p>	FL-5.1.1	K2	1

31	c	<p>Du graphique nous avons : A(4)=6 et A(3)=8 (les deux dernières cases grises). De la formule on obtient : $E(5) = (3 \cdot A(4) + A(3)) / 4 = (3 \cdot 6 + 8) / 4 = 26 / 4 = 6,5$ jours-personnes.</p> <p>Ainsi:</p> <ul style="list-style-type: none">a) N'est pas correctb) N'est pas correctc) Est correctd) N'est pas correct	FL-5.1.4	K3	1
----	---	---	----------	----	---

Numéro de la Question (#)	Bonne réponse	Explications / Justifications	Objectifs d'apprentissage (LO)	Points	Nombre de points
32	a	<p>Nous souhaitons exécuter des cas de test en fonction de leurs priorités, mais nous devons également prendre en compte les dépendances. Si nous ne considérons que les priorités, nous voulons d'abord exécuter TC 5 et TC 7 (priorité la plus élevée), puis TC 1, TC 3 et TC 4, et enfin TC 2 et TC 6 (priorité la plus basse). Cependant, pour exécuter TC 7, nous devons d'abord exécuter TC 4. Pour exécuter TC 5, nous devons exécuter TC 4 et TC 2, mais TC 2 est bloqué par TC1, qui doit être exécuté avant TC 2. Ainsi, afin d'exécuter les cas de test de priorité 1 le plus tôt possible, les cinq premiers cas de test doivent être : TC 4 - TC 7 - TC 1 - TC 2 - TC 5. Ensuite, nous devons exécuter TC 3, car il a une priorité plus élevée que TC 6. Ainsi, le planning complet sera TC 4 – TC 7 – TC 1 – TC 2 – TC 5 – TC 3 – TC 6.</p> <p>Ainsi, le sixième cas de test sera TC 3.</p> <p>Ainsi: a) Est-ce exact b) N'est pas correct c) N'est pas correct d) N'est pas correct</p>	FL-5.1.5	K3	1

33	b	<p>a) N'est pas correct. Le modèle de pyramide de tests ne fournit pas d'informations sur les priorités de test</p> <p>b) Est exact. Le modèle de pyramide de tests montre effectivement que différents tests ont différents niveaux de granularité. Les tests unitaires, situés à la base de la pyramide, sont plus granulaires et se concentrent sur des composants spécifiques du système. Les tests d'intégration et les tests de bout en bout, situés au sommet de la pyramide, sont moins granulaires et couvrent de plus larges parties du système.</p> <p>c) N'est pas correct. Le modèle de pyramide de test est indépendant des critères de couverture</p> <p>d) N'est pas correct. Le modèle de pyramide de tests ne montre aucune relation entre les différents tests</p>	FL-5.1.6	K1	1
----	---	---	----------	----	---

Numéro de la Question (#)	Bonne réponse	Explications / Justifications	Objectifs d'apprentissage (LO)	Points	Nombre de points
34	d	<p>a) Ce n'est pas correct. Les quadrants de test regroupent les niveaux de test et les types de test séparément selon plusieurs critères. Ils ne représentent aucune combinaison de niveaux de test et de types de test et ils ne sont pas liés à une position dans un cycle de vie de développement logiciel. Dans le modèle de quadrants de test, les niveaux de test et les types de test sont traités séparément.</p> <p>b) N'est pas correct. Les quadrants de test regroupent les niveaux de test et les types de test selon plusieurs critères. Ils ne décrivent pas le degré de granularité des différents types de tests effectués à chaque niveau de test. Un tel modèle, concernant les niveaux de test, est appelé la pyramide de test.</p> <p>c) N'est pas correct. L'affirmation est fausse, car en général, tout type de test peut être effectué à n'importe quel niveau de test.</p> <p>d) Est correct. Les quadrants de test regroupent les niveaux de test, les types de test, les activités, les techniques de test et les produits de travail dans le développement logiciel Agile. Dans ce modèle, les tests peuvent être orientés vers les activités d'affaires ou vers la technologie. Les tests peuvent soutenir l'équipe (c'est-à-dire guider le développement) ou critiquer le produit (c'est-à-dire mesurer son comportement par rapport aux attentes). La combinaison de ces deux points de vue détermine les quatre quadrants.</p>	FL-5.1.7	K2	1

35	c	<p>a)Ce n'est pas correct. La surveillance des risques fait partie du contrôle des risques, pas de l'analyse des risques.</p> <p>b)Ce n'est pas correct. L'identification des risques en elle-même ne permet pas de mettre en œuvre des activités d'atténuation des risques. Les actions d'atténuation sont définies pendant la phase de contrôle des risques.</p> <p>c)C'est correct. Il s'agit d'un exemple de la façon dont l'analyse des risques influence l'exhaustivité et la portée des tests.</p> <p>d)Ce n'est pas correct. Les éléments de couverture sont dérivés à l'aide de techniques de test, pas par l'analyse des risques.</p>	FL-5.2.3	K2	1
----	---	---	----------	----	---

Numéro de la Question (#)	Bonne réponse	Explications / Justifications	Objectifs d'apprentissage (LO)	Points	Nombre de points
36	b	<p>a) Ce n'est pas correct. Les rapports d'avancement des tests sont principalement utilisés pendant le suivi et le contrôle des tests, ainsi que pendant la clôture des tests, et non pendant la conception des tests.</p> <p>b) C'est correct. Un rapport de clôture des tests est préparé pendant la clôture des tests, lorsqu'un projet, un niveau de test ou un type de test est terminé et que, idéalement, ses critères de sortie ont été respectés. Ce rapport utilise des informations provenant des rapports d'avancement des tests et d'autres données.</p> <p>c) Ce n'est pas correct. Les rapports d'avancement des tests sont principalement utilisés pendant le suivi et le contrôle des tests, ainsi que pendant la clôture des tests, et non pendant l'analyse des tests.</p> <p>d) Ce n'est pas correct. Les rapports d'avancement des tests sont principalement utilisés pendant le suivi et le contrôle des tests, ainsi que pendant la clôture des tests, et non pendant la planification des tests.</p>	FL-5.3.2	K2	1

37	d	<p>a) N'est pas correct. Lorsqu'un utilisateur signale une panne logicielle, grâce à l'identification unique des commits, il est possible de réassembler les fichiers de la version du logiciel qui a été utilisée par l'utilisateur (ainsi que les versions correspondantes des scripts de test) et ainsi reproduire la panne. et localisez le défaut plus rapidement</p> <p>b) N'est pas correct. Si une modification apportée à l'environnement de test entraîne des problèmes inattendus lors des tests, la gestion de la configuration permet aux testeurs de revenir à une version précédente de l'environnement. Cela garantit que les tests peuvent continuer sans être affectés par le changement.</p> <p>c) N'est pas correct. La gestion de la configuration garantit que toute la documentation identifiée (par exemple, les spécifications des exigences) et les éléments logiciels sont référencés sans ambiguïté dans la documentation de test (par exemple, les plans de test)</p> <p>d) Est correct. Ceci est assuré par la gestion des défauts et non par le processus de gestion de la configuration.</p>	FL-5.4.1	K2	1
----	---	--	----------	----	---

Numéro de la Question (#)	Bonne réponse	Explications / Justifications	Objectifs d'apprentissage (LO)	Points	Nombre de points
38	b	<p>a) N'est pas correct. C'est important, mais pas aussi important que les éléments de l'environnement de test.</p> <p>b) Correct. Ce qui manque le plus, c'est l'identification du navigateur et de l'appareil utilisé pour le test. Les informations sur le navigateur et l'appareil sont importantes car un tel défaut peut être spécifique au navigateur ou à l'appareil. Par exemple, un bouton de connexion peut fonctionner correctement sur un navigateur (ou une version d'un navigateur spécifique) mais pas sur un autre. Par conséquent, les informations sur le navigateur et l'appareil peuvent aider les développeurs à reproduire le problème et à trouver plus rapidement la cause première du problème.</p> <p>c) N'est pas correct. L'objet de test est identifié (Boutique en ligne v0.99).</p> <p>d) N'est pas correct. L'impact est inclus – c'est la gravité (élevée)</p>	FL-5.5.1	K3	1

39	d	<p>a) N'est pas correct. Les outils d'exécution et de couverture des tests facilitent l'exécution automatisée des scénarios de test et la mesure de la couverture obtenue lors de l'exécution de ces scénarios de test. Cependant, ces outils n'aident pas à l'organisation des défauts et à la gestion de la configuration.</p> <p>b) N'est pas correct. Les outils de conception et de mise en œuvre des tests facilitent la génération de cas de test, de données de test et de procédures de test, mais ils n'aident pas à l'organisation des défauts et à la gestion de la configuration.</p> <p>c) N'est pas correct. Les outils de gestion des défauts sont utilisés pour gérer les défauts mais ne sont pas des outils de test et ne sont pas utilisés pour organiser les cas de test ou la gestion de la configuration.</p> <p>d) Est correct. Les outils de gestion des tests augmentent l'efficacité du processus de test en facilitant la gestion du cycle de vie du développement logiciel (SDLC), des exigences, des tests, des défauts et de la gestion de la configuration.</p>	FL-6.1.1	K2	1
----	---	---	----------	----	---

Numéro de la Question (#)	Bonne réponse	Explications / Justifications	Objectifs d'apprentissage (LO)	Points	Nombre de points
40	d	<p>a) N'est pas correct. « La capacité de générer des cas de test sans accès à la base de test » n'est pas possible. La génération de cas de tests par des testeurs ou des outils nécessite un accès à la base de test</p> <p>b) N'est pas correct. « L'obtention d'une couverture accrue grâce à une évaluation plus objective » n'est pas un avantage direct de l'automatisation des tests. L'automatisation des tests fournira une évaluation plus objective de la couverture, mais cette évaluation objective n'augmentera pas la couverture. Ce n'est qu'en utilisant les résultats de la couverture pour rédiger d'autres cas de test que la couverture peut éventuellement être augmentée.</p> <p>c) N'est pas correct. « L'augmentation des temps d'exécution des tests disponibles avec une puissance de traitement plus élevée » est une affirmation contradictoire, car une puissance de traitement plus élevée réduirait normalement les temps d'exécution, et des temps d'exécution plus longs ne constituent pas un avantage car les tests prendraient plus de temps.</p> <p>d) Est correct. La prévention des erreurs humaines grâce à une plus grande cohérence et répétabilité est un avantage de l'automatisation des tests, car l'automatisation des tests ne peut pas souffrir d'erreurs humaines. Par exemple, cela signifie que les tests sont systématiquement dérivés des exigences, que les données de test sont créées de manière systématique et que les tests sont exécutés par un outil dans le même ordre et à la même fréquence.</p>	FL-6.2.1	K1	1