

Question #1 (un point)

Dans quel cas l'utilisation d'une approche de test spécifique justifie-t-elle la nécessité des tests logiciels ?

- a) L'exécution de tests dynamiques permet d'identifier des défaillances dans un logiciel en provoquant des comportements anormaux impossibles à reproduire par un utilisateur final.
- b) Les techniques de test statique offrent aux développeurs la possibilité de détecter des défauts dans le code source avant même l'exécution du logiciel, réduisant ainsi les coûts de correction.
- c) L'analyse statique d'un programme assure aux parties prenantes que les composants du système ne produisant aucune sortie sont conformes aux exigences et peuvent être déployés en production.
- d) Les revues techniques permettent d'améliorer la qualité des spécifications et réduisent considérablement la nécessité de tests ultérieurs sur les produits dérivés.

Sélectionnez UNE réponse.

Question #2 (un point)

Quelle est la relation correcte entre les tests et l'assurance qualité ?

- a) Les tests sont une partie de l'assurance qualité
- b) Les tests remplacent complètement l'assurance qualité
- c) L'assurance qualité se limite uniquement aux activités de test
- d) Les tests et l'assurance qualité sont des concepts totalement distincts et indépendants

Sélectionnez UNE réponse.

Question #5 (un point)

Lesquels des éléments suivants (i–v) reflètent **correctement** l'impact que le contexte peut avoir sur le processus de test ?

- i. Le niveau de risque du système influence le niveau de rigueur attendu dans les activités de test.
- ii. Tous les projets doivent suivre le même processus de test pour garantir l'uniformité.
- iii. Un projet Agile privilégiera des cycles de test courts et itératifs.
- iv. Dans un projet soumis à des réglementations strictes, les tests devront être formellement documentés.
- v. Le contexte n'affecte pas les objectifs des tests, seulement leur exécution.

- a) i, iii, iv reflètent correctement l'impact du contexte ; ii, v non
- b) ii, iv, v reflètent correctement l'impact du contexte ; i, iii non
- c) i, ii, v reflètent correctement l'impact du contexte ; iii, iv non
- d) iii, iv, v reflètent correctement l'impact du contexte ; i, ii non

Sélectionnez UNE réponse.

Question #8 (un point)

Quel est l'un des **avantages principaux** d'une approche équipe intégrée pour les tests ?

- a) Elle permet aux testeurs de travailler de manière totalement indépendante des développeurs.
- b) Elle améliore la communication et la collaboration entre les différents membres de l'équipe projet.
- c) Elle supprime le besoin de planifier des tests en amont du développement.
- d) Elle réduit la nécessité d'exécuter des tests de régression, car les défauts sont corrigés immédiatement.

Sélectionnez UNE réponse.

Question #9 (un point)

Comment le choix du cycle de vie du développement logiciel impacte-t-il les activités de test ?

- a) Dans un cycle en V, les tests sont planifiés tôt et exécutés après le développement
- b) En mode Agile, les tests ne sont réalisés qu'en fin de projet pour valider le produit final
- c) Dans un modèle en cascade, les tests peuvent être réalisés à tout moment du projet
- d) Tous les modèles de cycle de vie du développement ont le même impact sur les activités de test

Sélectionnez UNE réponse.

Question #12 (un point)

Lors d'une rétrospective d'équipe après une livraison de sprint, les testeurs remontent un problème récurrent : les exigences arrivent trop tard et ne sont pas suffisamment détaillées pour écrire les cas de test à temps.

Quel usage **approprié** de la rétrospective pourrait aider à améliorer cette situation ?

- a) Supprimer l'écriture des cas de test pour gagner du temps
- b) Augmenter la taille des sprints pour inclure toutes les activités de test
- c) Identifier les causes racines de ces retards et mettre en place une action d'amélioration convenue par l'équipe
- d) Passer les exigences directement aux développeurs sans les testeurs

Sélectionnez UNE seule option.

Question #13 (un point)

Associez les types de test (1–4) aux objectifs correspondants (A–D) :

- 1. Test de régression
- 2. Test de confirmation
- 3. Test d'acceptation
- 4. Test d'intégration

- A. Vérifie qu'une modification n'a pas causé d'effets secondaires
- B. Permet de valider que le système répond aux besoins métier
- C. Vérifie qu'un défaut a bien été corrigé
- D. Évalue si les modules interagissent correctement entre eux

- a) 1A, 2C, 3B, 4D
- b) 1C, 2D, 3A, 4B
- c) 1B, 2A, 3C, 4D
- d) 1D, 2C, 3B, 4A

Sélectionnez UNE réponse.

Question #16 (un point)

Quel est l'un des **AVANTAGES** d'un retour d'information précoce et régulier des parties prenantes pendant les activités de test ?

- a) Il garantit que tous les cas de test sont automatisés
- b) Il permet de détecter et corriger les défauts plus tôt dans le cycle de développement
- c) Il réduit le besoin de tests exploratoires en fin de projet
- d) Il évite complètement les modifications d'exigences pendant le projet

Sélectionnez UNE réponse.

Question #17 (un point)

Quelles activités du processus de revue (1–4) correspondent aux descriptions suivantes (A–D) ?

- 1. Planification de la revue
- 2. Examen du document
- 3. Réunion de revue
- 4. Re-travail (rework)

- A. Identifier les anomalies de manière individuelle
- B. Corriger les défauts ou incohérences détectés
- C. Définir le type de revue, les rôles et le calendrier
- D. Discuter collectivement des défauts pour valider ou rejeter leur pertinence

- a) 1C, 2A, 3D, 4B
- b) 1A, 2D, 3C, 4B
- c) 1B, 2C, 3A, 4D
- d) 1C, 2B, 3A, 4D

Sélectionnez UNE réponse.

Question #20 (un point)

Vous testez un site de réservation hôtelière qui comporte seulement deux critères de recherche :

- **Catégorie d'hôtel** (avec trois options possibles : économique; standard; luxe)
- **Type de vue** (avec trois options possibles : pas de vue; vue partielle; vue panoramique)

Seuls les hôtels de catégorie luxe proposent une vue panoramique. Le formulaire dispose d'un mécanisme de validation intégré qui ne vous permettra pas d'utiliser les critères de recherche qui violent cette règle.

Chaque test comprend deux valeurs d'entrée : catégorie d'hôtel et type de vue. Vous souhaitez appliquer le partitionnement en classes d'équivalence (EP) pour couvrir chaque catégorie d'hôtel et chaque type de vue dans vos tests.

Quel est le nombre **minimal** de cas de test pour atteindre une couverture EP de 100% ?

- a) 3
- b) 4
- c) 5
- d) 6

Sélectionnez UNE réponse.

Dans le tableau suivant : **V = VRAI** et **F = FAUX**.

Conditions	R1	R2	R3	R4	R5	R6	R7	R8
Abonné Premium	V	V	V	V	F	F	F	F
Abonnement expiré	V	F	V	F	V	F	V	V
12 mois d'abonnement actif	F	V	F	V	F	V	V	V
Actions								
Accès anticipé aux films	X			X				
Un mois gratuit		X		X				X

Basé **UNIQUEMENT** sur la description des fonctionnalités du programme de fidélité, laquelle des règles ci-dessus décrit une situation impossible ?

- a) R4
- b) R2
- c) R6
- d) R8

Sélectionnez **UNE** réponse.

Question #25 (un point)

Une organisation développe un système de contrôle d'accès critique pour une infrastructure gouvernementale. Le chef de projet hésite à investir des ressources dans les tests boîte blanche.

Quelle est la **MEILLEURE** explication de la valeur ajoutée des tests boîte blanche pour ce type de projet ?

- a) Les tests boîte blanche remplacent efficacement les revues de code et les inspections formelles, réduisant ainsi le besoin d'autres activités de vérification
- b) Les tests boîte blanche sont plus importants que les tests boîte noire car ils garantissent que 100% des défauts seront détectés avant la mise en production
- c) Les tests boîte blanche permettent d'identifier les défauts dans le code qui sont difficiles à détecter avec des tests boîte noire, comme les défauts liés aux chemins d'exécution rarement empruntés.
- d) Les tests boîte blanche sont principalement utiles pour réduire la durée du cycle de développement en diminuant le nombre de tests fonctionnels nécessaires

Sélectionnez **UNE** réponse.

Question #26 (un point)

Un important projet gouvernemental de système d'information sur la santé est entré en phase de test final. Le directeur des tests souhaite appliquer une technique d'estimation d'erreurs pour déterminer quand arrêter les tests. L'équipe a identifié 350 défauts pendant les tests. En utilisant différentes méthodes d'estimation, ils ont calculé que le nombre total probable de défauts dans le système se situe entre 410 et 480. Après deux semaines supplémentaires de test, 40 nouveaux défauts ont été découverts.

Quelle est l'affirmation **la plus précise** concernant l'utilisation de l'estimation d'erreurs dans ce contexte ?

Question #28 (un point)

Laquelle des propositions suivantes concernant la rédaction des critères d'acceptation lors de l'acceptation utilisateur (UAT) est la **plus exacte** ?

- a) Les critères d'acceptation doivent obligatoirement être rédigés sous la forme de scénarios Given/When/Then (GWT) pour garantir une couverture complète.
- b) Les critères d'acceptation peuvent être définis sous différents formats, par exemple : scénarios Gherkin, checklists structurées ou gabarits de tests fonctionnels, en fonction des besoins du projet.
- c) Les critères d'acceptation sont généralement rédigés par l'équipe de développement seule, car ce sont eux qui connaissent les contraintes techniques.
- d) Les critères d'acceptation doivent inclure uniquement des métriques de performance afin de mesurer la rapidité d'exécution des scénarios critiques.

Sélectionnez **UNE** réponse.

Question #29 (un point)

Une équipe agile développe un système de gestion d'inventaire pharmaceutique soumis à des réglementations strictes. L'équipe a décidé d'utiliser le Développement Piloté par les Tests d'Acceptation (ATDD). Pendant un atelier d'ATDD avec les parties prenantes, l'équipe élabore les scénarios suivants pour la fonctionnalité "Alerte de péremption des médicaments" :

Fonctionnalité: Alerte de péremption des médicaments

En tant que pharmacien

Je veux être alerté des médicaments approchant leur date de péremption

Afin de garantir la sécurité des patients et la conformité réglementaire

Scénario 1: Alerte pour médicaments approchant la péremption

Étant donné que j'ai des médicaments en stock

Question #30 (un point)

Quel est l'un des **rôles principaux** du testeur lors de la planification des itérations dans un environnement Agile ?

- a) Réécrire les exigences pour les rendre plus techniques
- b) Estimer le nombre de sprints nécessaires pour l'équipe de développement
- c) Définir la méthode d'implémentation des tâches techniques
- d) Apporter une vision qualité sur les fonctionnalités prioritaires et leurs risques

Sélectionnez UNE réponse.

Question #31 (un point)

Laquelle des affirmations suivantes met **correctement** en opposition un critère d'entrée et un critère de sortie ?

- a) Avoir une exigence approuvée est un critère de sortie ; une couverture de test de 80 % est un critère d'entrée
- b) L'environnement de test prêt est un critère de sortie ; la livraison du produit est un critère d'entrée
- c) La disponibilité des données de test est un critère d'entrée ; le taux d'échec des tests est un critère de sortie
- d) Le début du développement est un critère d'entrée ; la fin des revues est un critère de sortie

Sélectionnez UNE réponse.

Question #33 (un point)

Vous travaillez sur une application de gestion de bibliothèque. Le tableau suivant montre la matrice de traçabilité des cas de test (TC) par rapport aux fonctionnalités dans une application de gestion de bibliothèque. "X" signifie qu'un cas de test donné couvre la fonctionnalité correspondante.

- Fonc1 : Inscription des lecteurs
- Fonc2 : Recherche de livres
- Fonc3 : Réservation d'ouvrage
- Fonc4 : Gestion des emprunts
- Fonc5 : Système de recommandation
- Fonc6 : Gestion des amendes
- Fonc7 : Rapport d'activité

	Fonc1	Fonc2	Fonc3	Fonc4	Fonc5	Fonc6	Fonc7
TC1	X	X	X				
TC2				X			
TC3			X	X			
TC4				X	X	X	X

Vous voulez prioriser les cas de test en utilisant la technique de priorisation par couverture supplémentaire. Vous exécutez tous les quatre cas de test.

Quel cas de test devrait être exécuté **EN DERNIER** ?

- a) TC1
- b) TC2
- c) TC3
- d) TC4

Sélectionnez **UNE** réponse.

Question #35 (un point)

Voici les risques identifiés dans le cadre du développement d'une application de gestion de bibliothèque universitaire :

1. Le système plante lorsque plusieurs utilisateurs réservent un livre en même temps
2. Un nouveau règlement interne impose une modification soudaine des règles de prêt
3. Une panne d'électricité empêche l'accès aux serveurs
4. Les étudiants en fin de cycle voient s'afficher des livres restreints qui ne leur sont pas destinés

Et les mesures suivantes ont été proposées :

- A. Acceptation du risque
- B. Test de performance
- C. Utilisation des tests de partitionnement d'équivalence
- D. Transfert du risque (externalisation ou assurance)

Quelle est la **MEILLEURE** correspondance entre les risques et les mesures proposées ?

- a) 1B, 2D, 3A, 4C
- b) 1C, 2A, 3D, 4B
- c) 1B, 2A, 3D, 4C
- d) 1A, 2B, 3D, 4C

Sélectionnez UNE réponse.

Question #38 (un point)

Voici un rapport d'anomalie rédigé par un testeur :

L'application plante.

Utilisateur : client_test02.

Reproduit en environnement pré-prod.

Voir capture écran.

Quelle information **ESSENTIELLE** manque pour que ce rapport soit exploitable par les développeurs ?

- a) Le niveau de sévérité du défaut
- b) Le résultat attendu
- c) Le nom du testeur
- d) Le navigateur utilisé

Sélectionnez UNE réponse.

Question #39 (un point)

Étant donné les activités de test suivantes :

1. Conception des cas de test
2. Exécution automatisée des tests
3. Gestion des environnements de test
4. Analyse des résultats de test

Réponse question #1

Cette question correspond à l'objectif d'apprentissage **FL-1.2.1 (K2)**- Donner des exemples montrant la nécessité des tests.

- **a) Faux** → Les tests dynamiques détectent des défauts, mais ils ne garantissent pas qu'ils provoquent uniquement des comportements anormaux impossibles à reproduire par les utilisateurs.
- **b) Correct** → Les tests statiques permettent d'identifier des défauts dans le code source avant même l'exécution, ce qui réduit les coûts de correction.
- **c) Faux** → L'analyse statique ne garantit pas qu'un composant est prêt pour la production, elle identifie uniquement des erreurs structurelles.
- **d) Faux** → Les revues améliorent la qualité des spécifications, mais ne remplacent pas les tests.

Réponse question #2

Cette question correspond à l'objectif d'apprentissage **FL-1.2.2 (K1)** – Rappeler la relation entre les tests et l'assurance qualité.

- **a) Correct** → Les tests font partie des activités d'assurance qualité, qui incluent également l'amélioration des processus, la documentation et le respect des normes.
- **b) Faux** → Les tests ne remplacent pas l'ensemble de l'assurance qualité, qui couvre un périmètre bien plus large (gestion des risques, validation des exigences, etc.).
- **c) Faux** → L'assurance qualité ne se limite pas aux tests, elle englobe des pratiques préventives et correctives.
- **d) Faux** → Les tests et l'assurance qualité sont liés : l'assurance qualité vise à améliorer la qualité globale du produit, et les tests en sont un des moyens.

Réponse question #5

Cette question correspond à l'objectif d'apprentissage **FL-1.4.2 (K2)** – Expliquer l'impact du contexte sur le processus de test.

La bonne réponse est a) i, iii, iv reflètent correctement l'impact du contexte ; ii, v non

- **i) Vrai** – Le **niveau de risque** conditionne la **profondeur des tests** requis.
- **ii) Faux** – Il n'existe **pas de processus unique** applicable à tous les projets.
- **iii) Vrai** – En **Agile**, les tests sont **intégrés** dans des cycles courts.
- **iv) Vrai** – Les **contraintes réglementaires** (ex. médical, aéronautique) exigent une **traçabilité et une documentation renforcées**.
- **v) Faux** – Le contexte **impacte aussi les objectifs** (par ex. : conformité, performance critique...).

Réponse question #6

Cette question correspond à l'objectif d'apprentissage **FL-1.4.5 (K2)** – Comparer les rôles impliqués dans les tests.

La bonne réponse est d) 1A, 2B, 3C, 4D

- **1A) : Correct** : Le testeur **exécute, observe, consigne**
- **2B) : Correct** : L'analyste métier définit les **attentes et exigences**
- **3C) : Correct** : Le développeur **corrige les défauts**
- **4D) : Correct** : Le responsable **planifie, suit, et coordonne**

Réponse question #7

Cette question correspond à **FL-1.5.1 (K2)** – Donner des exemples de compétences génériques requises pour le test.

- **a) Faux** → Trop spécifique, concerne plutôt les outils ou automatisation
- **b) Correct** → La **communication claire** est une **compétence générique clé** pour un testeur

Réponse question #10

Cette question correspond à l'objectif d'apprentissage **FL-2.1.2 (K1)** – Rappeler les bonnes pratiques de test applicables à tous les cycles de vie.

- **a) Faux** → Cela correspond à une approche tardive (type Waterfall rigide), et va à l'encontre des bonnes pratiques modernes.
- **b) Faux** → Les testeurs doivent être impliqués dès les phases d'analyse et de conception.
- **c) Correct** → Tester dès que possible (shift-left) est une **bonne pratique universelle** qui réduit les coûts de détection de défauts.
- **d) Faux** → Collaboration entre test et développement est essentielle, peu importe le cycle.

Réponse question #11

Cette question correspond à l'objectif d'apprentissage **FL-2.1.4 (K2)** - Résumer la façon dont DevOps pourrait avoir un impact sur le test.

- **a) Faux** → Dans DevOps, les tests deviennent plus critiques, pas moins importants, car les déploiements fréquents augmentent le besoin de détection rapide des problèmes.
- **b) Faux** → DevOps encourage la collaboration et le partage des responsabilités, mais n'élimine pas le besoin d'expertise en test. Les compétences spécialisées en test restent valorisées.
- **c) Faux** → DevOps ne préconise pas de tester principalement en production. Au contraire, il encourage le test approfondi avant le déploiement, mais dans des environnements qui reflètent fidèlement la production.
- **d) Correct** → Cette réponse décrit correctement l'impact principal de DevOps : l'automatisation des tests et l'intégration continue permettent un feedback plus rapide et facilitent les déploiements fréquents et fiables.

- **1A) Correct** → Le test de confirmation (ou "retest") vérifie qu'un défaut corrigé est effectivement résolu.
- **2B) Correct** → Le test de régression s'assure que la correction ou une nouvelle fonctionnalité n'a pas causé d'effets secondaires indésirables.
- **3A) Correct** → Les tests de confirmation ciblent spécifiquement la correction appliquée et ne testent pas d'autres parties du système.
- **4B) Correct** → Les tests de régression sont exécutés à une échelle plus large pour vérifier que le reste du système fonctionne toujours correctement.

Réponse question #15

Cette question correspond à l'objectif d'apprentissage **FL-3.1.2 (K2)** - Expliquer la valeur du test statique.

- **a) Correct** → Les tests statiques permettent d'identifier des défauts **sans exécuter le logiciel**, via des revues ou des analyses statiques.
- **b) Faux** → Tester le logiciel en l'exécutant relève des **tests dynamiques**.
- **c) Faux** → Tester la performance du logiciel sous charge relève des **tests de performance**, qui sont des tests dynamiques.
- **d) Faux** → Les tests statiques ne concernent pas l'implémentation des tests automatisés.

Réponse question #18

Cette question correspond à l'objectif d'apprentissage **FL-3.2.3 (K1)** – Rappeler quelles sont les responsabilités attribuées aux rôles principaux lors des revues.

- **a) Faux** → Le modérateur organise la revue, mais ne corrige pas les défauts.
- **b) Correct** → **L'auteur est la personne qui a produit l'artefact analysé et qui doit corriger les défauts détectés.**
- **c) Faux** → Le lecteur examine les documents, mais n'effectue pas de corrections.
- **d) Faux** → Le responsable qualité veille au respect du processus, mais n'intervient pas directement dans la correction des défauts.

Réponse question #19

Cette question correspond à l'objectif d'apprentissage **FL-4.1.1 (K2)** - Distinguer les techniques de test boîte noire, boîte blanche et basées sur l'expérience.

- **a) Faux** → Cette description correspond aux techniques de test boîte blanche, qui examinent la structure interne du code pour concevoir des tests couvrant différents chemins d'exécution. Les tests boîte noire ne s'intéressent pas au code source.
- **b) Faux** → Cette description correspond aux techniques de test basées sur l'expérience, qui s'appuient sur les connaissances et l'expérience des testeurs pour identifier les défauts potentiels. Les tests boîte noire sont basés sur des spécifications documentées plutôt que sur l'expérience seule.
- **c) Faux** → Cette description correspond également aux techniques de test boîte blanche, qui analysent la structure interne et mesurent la couverture (instructions, branches, chemins). Les tests boîte noire ne s'intéressent pas à la structure interne.
- **d) Correct** → Cette description caractérise précisément les techniques de test boîte noire. Ces techniques considèrent le système comme une "boîte noire" dont le fonctionnement interne est inconnu ou ignoré. Les cas de test sont dérivés des spécifications, des exigences et des comportements attendus, sans connaissance de l'implémentation.

Réponse question #22

Cette question correspond à l'objectif d'apprentissage **FL-4.2.3 (K3)** - Utiliser les tests par tables de décisions pour dériver les cas de test.

- **a) Faux** → **R4** : Possible – Un abonné Premium peut avoir accès anticipé aux films et recevoir un mois gratuit après 12 mois d'abonnement.
- **b) Faux** → **R2** : Possible – Un abonné Premium ayant 12 mois d'abonnement actif reçoit un mois gratuit, sans contradiction.
- **c) Faux** → **R6** : Possible – Cette règle concerne un non-abonné, il n'y a pas d'erreur.
- **d) Correct** → **R8** : **Impossible** – Un **non-abonné** ne devrait **pas recevoir un mois gratuit** après 12 mois, car seuls les abonnés Premium y ont droit selon les règles définies.

Réponse question #23

Cette question correspond à l'objectif d'apprentissage **FL-4.2.4 (K3)** – Utiliser les tests de transition d'état pour dériver les cas de test.

Transitions identifiées :

- 1 Accueil → Saisie du code PIN
- 2 Saisie du code PIN → Sélection du montant
- 3 Sélection du montant → Éjection du billet
- 4 Éjection du billet → Fin de transaction
- 5 Fin de transaction → Retour à l'accueil
- 6 Annulation à tout moment → Retour à l'accueil

6 transitions minimum doivent être testées pour couvrir tous les cas.

- **a) Faux** → 5 transitions ne suffisent pas.
- **b) Correct** → **6 transitions couvrent toutes les possibilités.**
- **c) Faux** → 7 transitions incluraient un test redondant.
- **d) Faux** → 8 transitions sont **excessives**.

- **c) Correct** → Cette réponse capture la principale valeur ajoutée des tests boîte blanche : ils permettent d'identifier les défauts dans des parties du code qui pourraient ne pas être facilement atteignables par les tests boîte noire.
- **d) Faux** → Les tests boîte blanche ne visent pas principalement à réduire la durée du cycle de développement ou à diminuer le nombre de tests fonctionnels. Au contraire, ils représentent souvent un effort supplémentaire.

Réponse question #26

Cette question correspond à l'objectif d'apprentissage **FL-4.4.1 (K2)** - Expliquer l'estimation d'erreurs.

- **a) Faux** → Cette réponse simplifie excessivement l'utilisation de l'estimation d'erreurs en appliquant un "seuil standard" arbitraire de 80%. L'estimation d'erreurs ne se limite pas à un pourcentage fixe et ne peut pas, à elle seule, déterminer si les tests doivent être arrêtés. De plus, le calcul ignore les 40 défauts supplémentaires découverts, ce qui porterait le total à 390 défauts identifiés.
- **b) Faux** → Ce n'est pas une conclusion fiable. Le fait de trouver 40 défauts en deux semaines n'indique pas nécessairement une inefficacité des tests. La détection des défauts n'est généralement pas linéaire, et le taux peut varier selon les phases de test.
- **c) Faux** → L'estimation d'erreurs donne une fourchette (ici 410–480), et les 40 nouveaux défauts sont encore dans cette fourchette. Le modèle reste valide, il faut juste le réévaluer progressivement, pas le rejeter.
- **d) Correct** → Cette réponse reflète avec précision la valeur et les limites de l'estimation d'erreurs. L'estimation fournit une base quantitative pour évaluer la progression des tests, mais ne doit pas être utilisée comme seul critère de décision pour arrêter les tests. D'autres facteurs qualitatifs comme la sévérité des défauts restants, la couverture des risques métier, et les contraintes de projet doivent également être pris en compte. Cette approche équilibrée est particulièrement importante pour un système critique comme un système d'information sur la santé.

- **d) Faux** → Les critères d'acceptation peuvent effectivement comporter des exigences de performance, mais ils ne se limitent pas à cet aspect. Ils doivent également couvrir la fonctionnalité, la facilité d'utilisation, la sécurité, la fiabilité, etc.

Réponse question #29

Cette question correspond à l'objectif d'apprentissage **FL-4.5.3 (K3)** - Utiliser le développement piloté par les tests d'acceptation (ATDD) pour dériver les cas de tests.

- **a) Faux** → Cette réponse reflète une confusion entre les tests d'acceptation et les tests non-fonctionnels. L'ATDD se concentre principalement sur les comportements fonctionnels et les critères d'acceptation métier.
- **b) Faux** → Cette réponse contredit fondamentalement le principe de l'ATDD. Dans l'ATDD, les scénarios sont développés collaborativement avec toutes les parties prenantes métier et développeurs compris qui sont les mieux placées pour définir les critères d'acceptation.
- **c) Correct** → Cette réponse capture l'essence de l'ATDD. Les scénarios présentés suivent un bon format qui définit clairement les comportements attendus, mais ils restent un peu abstraits.
- **d) Faux** → Cette réponse va à l'encontre des bonnes pratiques d'ATDD. Les différents scénarios présentés capturent des variations importantes du comportement métier (médicaments standard vs contrôlés, suppression d'alertes) qui doivent être explicitement testées au niveau des tests d'acceptation, pas seulement au niveau unitaire..

Réponse question #32

Cette question correspond à l'objectif d'apprentissage **FL-5.1.4 (K3)** - Utiliser des techniques d'estimation pour calculer l'effort de test requis.

L'estimation à 3 points utilise la formule : (Optimiste + 4 × Plus probable + Pessimiste) ÷ 6

Donc : $(6 + 4 \times 15 + 24) \div 6 = (6 + 60 + 24) \div 6 = 90 \div 6 = 15$

- **a) Correct** → 15 jours-personnes est le résultat correct de la formule d'estimation à trois points.
- **b) Faux** → 16 jours-personnes n'est pas le résultat correct.
- **c) Faux** → 17 jours-personnes n'est pas le résultat correct.
- **d) Faux** → 18 jours-personnes n'est pas le résultat correct.

Réponse question #33

Cette question correspond à l'objectif d'apprentissage **FL-5.1.5 (K3)** - Appliquer la priorisation des cas de test.

- **TC1** couvre Fonc1, Fonc2, Fonc3 → 3 nouvelles fonctionnalités
- **TC4** couvre Fonc5, Fonc6, Fonc7 → 3 autres nouvelles fonctionnalités
 - Après exécution de **TC1** et **TC4**, 6 fonctionnalités sont déjà couvertes
- **TC3** couvre Fonc3 (déjà couverte par TC1) + **Fonc4 (nouvelle)** → utile pour **compléter la couverture à 100 %**
 - Donc **TC3** doit être exécuté **avant la fin**.
- **TC2** ne couvre que **Fonc4**, qui est **déjà couverte par TC3**
 - **TC2 est redondant, il n'apporte aucune nouvelle couverture**

TC2 (Réponse b) a la couverture supplémentaire la plus faible, il devrait donc être exécuté en dernier.

- **4C) Correct** → Mauvaise attribution de droits → peut être testé avec **partitionnement d'équivalence** (groupes autorisés vs interdits).

Réponse question #36

Cette question correspond à l'objectif **FL-5.3.3 (K2)** – Donner des exemples de communication de l'état d'avancement des tests.

- **a) Faux** → Trop technique, inutile pour un suivi métier.
- **b) Faux** → La revue de code concerne la qualité du code, pas l'avancement des tests.
- **c) Correct** → Le **burndown chart** est un excellent moyen visuel de **suivre la progression dans le temps**.
- **d) Faux** → Les commits ne donnent pas de vue globale sur l'exécution des tests.

Réponse question #37

Cette question correspond à l'objectif **FL-5.4.1 (K2)** – Résumer la manière dont la gestion de configuration soutient les tests.

- **a) Faux** → Ce n'est pas un rôle de la configuration mais de la stratégie de test.
- **b) Correct** → Le **cœur de la gestion de configuration**, c'est le **suivi des versions, des liens entre livrables et des artefacts**.
- **c) Faux** → Ce n'est ni toujours souhaitable, ni une responsabilité de la configuration.
- **d) Faux** → Les spécifications sont sous contrôle de gestion produit.

Réponse question #38

Cette question correspond à l'objectif **FL-5.5.1 (K3)** – Préparer un rapport de défaut.

- **a) Faux** → Utile, mais **pas bloquant** pour comprendre l'erreur.
- **b) Correct** → **Sans résultat attendu**, il est **impossible de savoir ce qui aurait dû se produire** → c'est l'élément clé manquant

Réponse question #40

Cette question correspond à l'objectif d'apprentissage **FL-6.2.1 (K1)** - Rappeler les avantages et les risques de l'automatisation des tests.

- **a) Faux** → L'automatisation n'améliore pas nécessairement la qualité de conception des cas de test, qui dépend davantage des compétences des testeurs.
- **b) Correct** → Le principal avantage de l'automatisation des tests de régression est qu'elle permet d'exécuter rapidement un grand nombre de tests qui doivent être répétés fréquemment.
- **c) Faux** → L'automatisation des tests ne garantit pas l'absence de défauts, elle permet seulement de détecter ceux pour lesquels des tests ont été écrits.
- **d) Faux** → L'automatisation ne remplace pas complètement les tests manuels, qui restent nécessaires pour certains aspects comme les tests exploratoires.

iv. Les tests sont principalement axés sur la performance du système, et non sur la détection des défauts.

v. Une modification du logiciel peut introduire de nouveaux défauts imprévus.

a) i, ii, iii, v ont une influence significative ; iv non.

b) i, iii, iv ont une influence significative ; ii, v non.

c) ii, iv, v ont une influence significative ; i, iii non.

d) iii, v ont une influence significative ; i, ii, iv non.

Sélectionnez UNE réponse.

Question #4 (un point)

Quelle correspondance est la **PLUS** appropriée entre les activités de test (1–4) et leurs objectifs (A–D) ?

1. Planification du test
2. Conception du test
3. Exécution du test
4. Clôture du test

A. Rassembler les métriques, évaluer les critères de sortie atteints

B. Choisir les cas à exécuter et les données de test

C. Identifier les objectifs, ressources, et risques

D. Lancer les tests et comparer les résultats attendus et obtenus

a) 1A, 2D, 3B, 4C

b) 1B, 2A, 3C, 4D

c) 1D, 2C, 3A, 4B

d) 1C, 2B, 3D, 4A

Sélectionnez UNE réponse.

Question #6 (un point)

Quelle est la principale valeur du maintien de la traçabilité dans le processus de test ?"

- a) Il permet d'optimiser la performance du logiciel en production
- b) Il garantit que chaque exigence a été testée et qu'aucun élément critique n'a été oublié
- c) Il évite complètement la nécessité de documenter les tests manuels
- d) Il remplace les rapports de test en fournissant directement des métriques de qualité

Sélectionnez UNE réponse.

Question #7 (un point)

Quel est un **AVANTAGE** de l'approche "équipe intégrée" pour le test ?

- a) Elle permet aux testeurs de rester indépendants en dehors de l'équipe
- b) Elle favorise la collaboration entre développeurs, testeurs et analystes métier
- c) Elle rend le rôle de testeur facultatif dans l'équipe
- d) Elle permet d'éviter toute forme de documentation

Sélectionnez UNE réponse.

Question #8 (un point)

Quel est un risque **potentiel** d'une équipe de test totalement indépendante ?

- A) Un manque de compréhension du produit et des objectifs métier.
- B) Une meilleure objectivité dans l'identification des défauts.
- C) Une réduction des conflits d'intérêts avec les développeurs.
- D) Une amélioration de la rapidité d'exécution des tests.

Sélectionnez UNE réponse.

Question #11 (un point)

Quels défis les tests doivent-ils relever dans un environnement DevOps ?

- i. Maintenir un équilibre entre tests manuels et automatisés
- ii. Assurer une exécution rapide des tests pour s'adapter aux livraisons fréquentes
- iii. Se concentrer uniquement sur les tests de non-régression
- iv. Gérer la stabilité et la disponibilité de l'environnement de test

- a) i, ii ont une influence significative ; iii, iv non.
- b) i, iii ont une influence significative ; ii, iv non.
- c) i, ii, iv ont une influence significative ; iii non.
- d) iii, iv ont une influence significative ; i, ii non.

Sélectionnez UNE réponse.

Question #12 (un point)

Lors d'une rétrospective, une équipe identifie que les testeurs reçoivent les user stories trop tard et n'ont pas le temps de les analyser. Quelle action représente le **mieux** l'objectif d'une rétrospective dans ce cas ?

- a) Documenter le problème dans un fichier Excel partagé
- b) Répartir la charge de travail des testeurs entre plusieurs équipes
- c) Décider collectivement d'inclure les testeurs dans les ateliers de raffinement
- d) Ajouter un nouveau testeur à l'équipe dès le prochain sprint

Sélectionnez UNE seule option.

Les tests sont exécutés après chaque nouvelle version livrée, qui inclut des corrections pour les défauts précédemment identifiés.

Quels tests ont été exécutés en tant que tests de **régression** ?

- a) Seulement 4, 7, 8
- b) Seulement 4, 5, 7, 8
- c) Seulement 6, 9
- d) Seulement 4, 5, 6, 7, 8, 9

Sélectionnez UNE réponse.

Question #15 (un point)

Lequel des éléments suivants **n'est pas** une technique de test statique ?

- a) La revue des exigences.
- b) L'analyse statique du code source.
- c) L'exécution de tests unitaires automatisés.
- d) L'inspection formelle d'un document de conception.

Sélectionnez UNE réponse.

Question #16 (un point)

Quels bénéfices des tests statiques (1–4) correspondent aux descriptions suivantes (A–D) ?

1. Réduction du coût de correction des défauts
2. Détection précoce d'erreurs dans les documents
3. Amélioration de la compréhension partagée des exigences
4. Identification d'anomalies sans exécution du code

Question #18 (un point)

Quel facteur contribue le **plus** à la réussite d'une revue ?

- a) La participation active des parties prenantes
- b) La suppression des discussions pour éviter les désaccords
- c) L'absence de modérateur pour une meilleure autonomie des participants
- d) La réalisation exclusive de revues informelles

Sélectionnez UNE réponse.

Question #19 (un point)

Une organisation développe un nouveau système de contrôle pour des équipements médicaux. L'équipe de test souhaite utiliser une combinaison de différentes approches de test. Laquelle des combinaisons suivantes utilise **UNIQUEMENT** des techniques basées sur l'expérience ?

- a) Tests exploratoires, analyses de valeurs limites et tests par paires
- b) Tests basés sur l'intuition, tests exploratoires et tests basés sur les check-lists
- c) Tests de chemins, tests de couverture d'instructions et tests basés sur les défauts
- d) Partitionnement d'équivalence, tests basés sur les risques et tests de structure de contrôle

Sélectionnez UNE réponse.

- a) 23, 27
- b) 24, 26
- c) 23, 24, 26, 27
- d) 22, 24, 26, 28

Sélectionnez **UNE** réponse.

Question #22 (un point)

Vous concevez des cas de test à partir de la table de décision suivante pour un système de validation de l'accès à une salle de sport :

Les règles sont les suivantes :

- L'accès est autorisé si la personne est âgée de 18 ans ou plus, a un abonnement actif et a fourni un certificat médical de moins de 6 mois.
- Si une de ces conditions n'est pas remplie, l'accès est refusé.

Vous avez déjà conçu les cas de test suivants :

- CT1 : 20 ans, abonnement actif, certificat valide → accès autorisé
- CT2 : 17 ans, abonnement actif, certificat valide → accès refusé
- CT3 : 25 ans, abonnement inactif, certificat valide → accès refusé
- CT4 : 30 ans, abonnement actif, certificat expiré → accès refusé
- CT5 : 28 ans, abonnement inactif, certificat expiré → accès refusé

Lequel des cas de test suivants, s'il est ajouté à la suite existante, augmentera la **couverture de la table de décision** ?

- a) 22 ans, abonnement inactif, certificat expiré → accès refusé
- b) 19 ans, abonnement actif, certificat valide → accès autorisé
- c) 17 ans, abonnement inactif, certificat expiré → accès refusé
- d) Aucun des cas supplémentaires n'augmentera la couverture de la table de décision

Sélectionnez **UNE** réponse.

Question #24 (un point)

Considérez le fragment de pseudo-code suivant, écrit pour vérifier les limites d'un score d'évaluation.

```
1:  fonction évaluer(score):  
2:      if score > 100:  
3:          afficher("Erreur : score trop élevé")  
4:      else:  
5:          if score >= 50:  
6:              afficher("Réussi")  
7:          else:  
8:              afficher("Échec")  
9:      afficher("Évaluation terminée")
```

Vous devez construire un ensemble de cas de test pour atteindre une **couverture des instructions de 100 %**.

Parmi les ensembles de valeurs suivantes, lequel est le plus adapté pour atteindre cet objectif ?

- a) score = 45, score = 110
- b) score = 50, score = 101
- c) score = 30, score = 75, score = 120
- d) score = 55, score = 45

Sélectionnez UNE réponse.

Question #27 (un point)

Dans une entreprise développant des applications mobiles, les testeurs suivent une **checklist** de vérifications pour s'assurer que l'application respecte les standards iOS et Android.

Quel est l'**objectif principal** de cette approche ?

- a) Garantir une couverture exhaustive de toutes les fonctionnalités
- b) Fournir une liste de points clés à vérifier pour éviter d'oublier des aspects critiques
- c) Remplacer complètement l'exécution de tests exploratoires
- d) Automatiser entièrement le processus de validation

Sélectionnez **UNE** réponse.

Question #28 (un point)

Vous participez à la rédaction collaborative d'une User Story concernant une nouvelle fonctionnalité de tri par note moyenne des clients dans une application e-commerce.

*En tant que client d'une boutique en ligne,
Je veux pouvoir trier les résultats de recherche par note moyenne des clients,
afin de pouvoir trouver les produits les mieux évalués plus facilement.*

Parmi les cas de test suivants, lequel illustre le **MIEUX** un scénario réaliste basé sur cette User Story et ses critères d'acceptation ?

- a) Accéder à la page produit. Trier par note moyenne en ordre décroissant. Résultat attendu : les produits les mieux notés apparaissent en haut de la liste.
- b) Modifier la couleur du bouton de tri. Résultat attendu : le bouton passe du bleu au vert.
- c) Désactiver JavaScript dans le navigateur. Résultat attendu : les résultats ne se mettent plus à jour automatiquement.
- d) Cliquer sur un produit dans la liste. Résultat attendu : la fiche produit s'ouvre dans un nouvel onglet.

Sélectionnez **UNE** réponse.

Question #30 (un point)

Une équipe teste un logiciel de gestion de contrats d'assurance. Après plusieurs campagnes de test, elle souhaite vérifier si les conditions pour clôturer les tests sont remplies.

Parmi les éléments suivants, lesquels peuvent être considérés comme des critères de sortie typiques ?

- a) Tous les cas de test de non-régression ont été exécutés
- b) L'environnement de recette a été validé par l'équipe technique
- c) La couverture des exigences critiques a atteint 95 %
- d) Les plans de test ont été approuvés par le chef de projet
- e) Les outils de gestion des tests sont opérationnels

Sélectionnez DEUX réponses.

Question #31 (un point) (à adapter)

Vous souhaitez estimer le budget testing pour un nouveau produit pharmaceutique en utilisant l'estimation basée sur des ratios. Vous calculez le ratio budget développement/coût de test en utilisant des données moyennes provenant de quatre projets historiques similaires. Le tableau montre ces données historiques.

Projet	Budget Développement (k€)	Budget Testing (k€)
Projet 1	2,500	300
Projet 2	1,800	270
Projet 3	3,200	560
Projet 4	2,700	430

Question #33 (un point)

Quelles sont les principales caractéristiques des tests situés dans le quadrant Q4 ?

- a) Tests unitaires automatisés et tests de performance
- b). Tests d'acceptance et tests exploratoires
- c). Tests de programmation et tests de configuration
- d) Tests de charge et tests de sécurité

Sélectionnez UNE réponse.

Question #34 (un point)

Lors de la planification des tests pour un système de réservation de salles de réunion, l'équipe QA identifie le risque suivant :

- **Risque** : En cas de surcharge, les notifications ne sont pas envoyées à temps.
- **Probabilité** : élevée
- **Impact** : critique
- **Réponse au risque** :
 - Exécution de tests de performance ciblés pendant les tests système
 - Ajustement de l'architecture pour répartir les charges sur plusieurs serveurs avant la mise en production

Quelle est la mesure appliquée dans ce contexte ?

- a) Acceptation du risque
- b) Plan de secours
- c) Transfert du risque
- d) Réduction du risque (mitigation)

Sélectionnez UNE réponse.

Question #37 (un point)

Pourquoi la gestion de configuration est-elle essentielle dans un projet où plusieurs versions logicielles sont testées en parallèle ?

- a) Pour centraliser tous les plans de test dans un seul fichier
- b) Pour s'assurer que les tests manuels sont plus rapides
- c) Pour éviter que les développeurs modifient le code en même temps
- d) Pour garantir que chaque version du logiciel est testée avec les bonnes données, scripts et environnements

Sélectionnez UNE réponse.

Question #38 (un point)

Tu es en phase de test système pour une nouvelle application bancaire. Tu effectues une opération de virement entre deux comptes en utilisant un navigateur Firefox. Tu constates un comportement inattendu.

Voici le rapport d'anomalie que tu prépares :

Erreur affichée lors de la tentative de virement de compte A vers compte B.

Étapes :

1. Connexion avec l'utilisateur test_user_01
2. Accès à la section "Virements"
3. Saisie des données suivantes : montant = 0, destinataire = Compte B
4. Clic sur "Valider"

Environnement : QA-01

Résultat obtenu :

Message d'erreur : "Opération non autorisée" alors qu'il est attendu "Opération autorisée"

Question #40 (un point)

Pourquoi l'automatisation des tests ne peut-elle **pas** remplacer complètement les tests manuels ?

- a) Parce que l'automatisation est toujours plus coûteuse que les tests manuels
- b) Parce que certains tests nécessitent un jugement humain et une exploration
- c) Parce que l'automatisation ne peut pas être utilisée pour les tests de régression
- d) Parce que l'automatisation ne permet pas de générer des rapports de test

Sélectionnez UNE réponse.

Réponse question #3

Cette question correspond à l'objectif d'apprentissage **FL-1.3.1 (K2)** – Expliquer les sept principes du test.

Réponse correcte : a) i, ii, iii, v ont une influence significative ; iv non.

- **i) Correct** → Il est impossible de tester toutes les combinaisons d'entrées, ce qui limite la couverture des tests.
- **ii) Correct** → Certains défauts peuvent être masqués par d'autres erreurs et ne se manifester que plus tard.
- **iii) Correct** → Les tests sont conçus selon les spécifications, mais ne couvrent pas tous les scénarios réels.
- **iv) Faux** → Les tests ne sont pas uniquement axés sur la performance, ils incluent aussi la recherche de défauts.
- **v) Correct** → Modifier le logiciel peut introduire de nouveaux défauts, nécessitant des tests de régression.

Réponse question #4

Cette question correspond à l'objectif d'apprentissage **FL-1.4.1 (K2)** – Résumer les différentes activités et tâches de test.

La bonne réponse est : d) 1C, 2B, 3D, 4A

- **1C) : Correct** : La planification sert à **préparer le cadre du test**
- **2B) : Correct** : La conception permet de définir les **données, cas et conditions de test**
- **3D) : Correct** : L'exécution vérifie le **résultat réel par rapport à l'attendu**
- **4A) : Correct** : Clôturer = **évaluer, nettoyer, archiver, analyser les résultats**

Réponse question #7

Cette question correspond à l'objectif d'apprentissage **FL-1.5.2 (K1)** – Rappeler les avantages de l'approche intégrée.

- **a) Faux** → L'intégration **réduit la séparation**
- **b) Correct** → Une **équipe intégrée = collaboration renforcée**
- **c) Faux** → Le testeur reste **essentiel** même dans une équipe Agile
- **d) Faux** → La documentation existe, même si elle est **plus légère**

Réponse question #8

Cette question correspond à l'objectif d'apprentissage **FL-1.5.3 (K2)** – Distinguer les avantages et les inconvénients de l'indépendance du test.

- **a) Correct** → Une équipe trop indépendante peut manquer d'informations essentielles sur le produit et son contexte d'utilisation.
- **b) Faux** → Une plus grande objectivité est un avantage, et non un inconvénient.
- **c) Faux** → Une séparation trop marquée peut au contraire **augmenter les conflits** entre testeurs et développeurs.
- **d) Faux** → Une équipe totalement indépendante n'accélère pas forcément les tests, cela dépend de l'organisation en place.

Réponse question #9

Cette question correspond à l'objectif d'apprentissage **FL-2.1.2 (K1)** – Rappeler les bonnes pratiques de test qui s'appliquent à tous les cycles de vie du développement logiciel.

- **a) Faux** → Les tests doivent couvrir plusieurs niveaux (unitaires, intégration, système, acceptation), et pas uniquement l'interface utilisateur.
- **b) Faux** → Attendre la fin du développement pour tester expose le projet à un risque accru de corrections coûteuses en fin de cycle.

Réponse question #12

Cette question correspond à l'objectif d'apprentissage **FL-2.1.6 (K2)** – Utiliser les rétrospectives pour améliorer le processus.

- **a) Faux** → Identifier le problème sans agir n'apporte aucune amélioration.
- **b) Faux** → Déplacer le problème ailleurs ne résout rien.
- **c) Correct** → Impliquer les testeurs **plus tôt dans le processus** est une **amélioration concrète** issue d'une réflexion collective.
- **d) Faux** → Ajouter des ressources sans changer le processus ne garantit pas de résolution.

Réponse question #13

Cette question correspond à l'objectif d'apprentissage **FL-2.2.1 (K2)** - Distinguer les différents niveaux de test.

La bonne réponse est a) 1A, 2C, 3B, 4A

- **1A) Correct** → **Les tests d'intégration** vérifient comment les systèmes logiciels interagissent entre eux.
- **2C) Correct** → **Les tests unitaires** permettent de détecter des défauts techniques dans des unités de code isolées.
- **3B) Correct** → **Les tests système** valident le comportement global du système dans un environnement réaliste.
- **4A) Correct** → **Les tests d'acceptation** vérifient que le produit répond aux attentes métiers et fonctionnelles.

Réponse question #15

Cette question correspond à l'objectif d'apprentissage **FL-3.1.1 (K1)** - Reconnaître les types de produits qui peuvent être examinés par les différentes techniques de test statique.

- **a) Faux** → La **revue des exigences** est une activité de test statique permettant d'identifier des incohérences dans les spécifications.
- **b) Faux** → L'**analyse statique du code** est une technique statique qui permet de détecter des erreurs sans exécuter le programme.
- **c) Correct** → L'**exécution de tests unitaires** est une activité **dynamique**, car elle nécessite de faire fonctionner le logiciel.
- **d) Faux** → L'**inspection formelle** est une technique statique qui permet de valider la conception avant l'implémentation.

Réponse question #16

Cette question correspond à l'objectif d'apprentissage **FL-3.1.2 (K2)** – Expliquer la valeur du test statique.

La réponse correcte est a) 1A, 2B, 3C, 4D

- **1A) Correct** : Moins de défauts en production = **moins de coûts en fin de cycle**.
- **2B) Correct** : Le test statique permet d'identifier les erreurs **avant que le code soit écrit**.
- **3C) Correct** : Les revues statiques aident l'équipe à **mieux comprendre les exigences**.
- **4 D) Correct** : Le test statique **n'exécute pas le code**, il l'analyse.

Réponse question #19

Cette question correspond à l'objectif d'apprentissage **FL-4.1.1 (K2)** - Distinguer les techniques de test boîte noire, boîte blanche et basées sur l'expérience.

- a) **Faux** - Cette combinaison mélange des techniques basées sur l'expérience (tests exploratoires) avec des techniques de test boîte noire (analyses de valeurs limites et tests par paires). Les tests par paires et les analyses de valeurs limites sont des techniques systématiques basées sur les spécifications, pas sur l'expérience.
- b) **Correct** - Cette combinaison contient uniquement des techniques basées sur l'expérience :
 - Tests basés sur l'intuition : utilisation de l'expérience et de l'intuition du testeur pour identifier les zones problématiques
 - Tests exploratoires : apprentissage, conception et exécution des tests simultanément
 - Tests basés sur les check-lists : utilisation de listes de contrôle développées à partir de l'expérience pour guider les tests
- c) **Faux** - Cette combinaison contient principalement des techniques de test boîte blanche (tests de chemins et tests de couverture d'instructions), avec une technique basée sur l'expérience (tests basés sur les défauts). Les tests de structure interne sont clairement des techniques boîte blanche.
- d) **Faux** - Cette combinaison mélange des techniques de test boîte noire (partitionnement d'équivalence), des techniques basées sur l'expérience (tests basés sur les risques), et des techniques de test boîte blanche (tests de structure de contrôle). Ce n'est donc pas une liste de techniques uniquement basées sur l'expérience.

Réponse question #22

Cette question correspond à l'objectif d'apprentissage **FL-4.2.3 (K3)** – Utiliser les tests par tables de décisions pour dériver les cas de test.

Analyse:

Le contexte définit 3 conditions :

1. Âge ≥ 18 ans ? (Oui/Non)
2. Abonnement actif ? (Oui/Non)
3. Certificat médical valide ? (Oui/Non)

La table de décision complète comporterait 8 combinaisons possibles ($2^3 = 8$) :

Règle	Âge ≥ 18	Abonnement actif	Certificat valide	Décision
R1	Oui	Oui	Oui	Autorisé
R2	Oui	Oui	Non	Refusé
R3	Oui	Non	Oui	Refusé
R4	Oui	Non	Non	Refusé
R5	Non	Oui	Oui	Refusé
R6	Non	Oui	Non	Refusé
R7	Non	Non	Oui	Refusé
R8	Non	Non	Non	Refusé

Les cas de tests proposés couvrent :

- CT1 (20 ans, abonnement actif, certificat valide) → couvre R1
- CT2 (17 ans, abonnement actif, certificat valide) → couvre R5
- CT3 (25 ans, abonnement inactif, certificat valide) → couvre R3
- CT4 (30 ans, abonnement actif, certificat expiré) → couvre R2
- CT5 (28 ans, abonnement inactif, certificat expiré) → couvre R4

b) Disponible, Changer Véhicule, Non Disponible, Disponible, Payer : En attente → Réserve → Réserve → En attente → Réserve → Confirmé. **Couvre 5 transitions incluant l'état Confirmé**

c) Disponible, Changer Véhicule, Disponible, Changer Véhicule, Non Disponible : En attente → Réserve → Réserve → Réserve → Réserve → En attente **Couvre seulement 3 transitions uniques**

d) Non Disponible, Annuler, Changer Véhicule, Disponible, Payer : En attente → En attente → Terminé → (impossible de continuer car Terminé est un état final) **Séquence invalide après la deuxième transition**

L'option b) offre la meilleure couverture car elle couvre 5 transitions et atteint l'état "Confirmé", qui n'est pas couvert par les autres options valides.

Réponse question #24

Cette question correspond à l'objectif d'apprentissage **FL-4.3.1 (K2)** – Expliquer le test des instructions et le principe de couverture des instructions.

L'objectif est ici de **faire exécuter chaque ligne (instruction)** au moins une fois.

- **Instruction 3** → exécutée si **score > 100**
- **Instruction 6** → exécutée si **score >= 50** ET ≤ 100
- **Instruction 8** → exécutée si **score < 50**
- **Instruction 9** → exécutée dans tous les cas

Pour atteindre une couverture des instructions de 100%, nous devons nous assurer que chaque ligne de code est exécutée au moins une fois. Analysons les différents chemins d'exécution:

1. **Chemin 1** (score > 100): lignes 1, 2, 3, 9
2. **Chemin 2** (score >= 50 et score ≤ 100): lignes 1, 2, 4, 5, 6, 9
3. **Chemin 3** (score < 50): lignes 1, 2, 4, 5, 7, 8, 9

Réponse question #25

Cette question correspond à l'objectif d'apprentissage **FL-4.3.2 (K2)** – Expliquer le test des branches.

Étapes pour résoudre la question :

On identifie les **conditions logiques** et leurs **branches** :

1. **if disponible == FAUX**
 - **Vrai** → afficher "Produit indisponible"
 - **Faux** → entrer dans le **else**
2. **if solde >= seuil** (dans le **else**)
 - **Vrai** → afficher "Commande validée"
 - **Faux** → afficher "Fonds insuffisants"

Chaque condition a **2 résultats possibles** → cela donne :

- **Condition 1** : 2 branches
- **Condition 2** : 2 branches

Total = 4 branches

La bonne réponse est la réponse c)

Réponse question #28

Cette question correspond à l'objectif d'apprentissage **FL-4.5.1** - Expliquer comment rédiger des User Stories en collaboration avec des développeurs et des représentants du métier.

- **a) Correct** → Ce cas de test est **directement aligné avec le besoin métier** (voir les produits les mieux notés) et **répond à un critère d'acceptation de la User Story** (tri décroissant).
- **b) Faux** → Ce changement de couleur **n'est pas mentionné dans la User Story**.
- **c) Faux** → Le comportement sans JavaScript **n'est pas un critère couvert**, et relève de tests techniques spécifiques.
- **d) Faux** → L'ouverture de la fiche produit **n'est pas liée à la fonctionnalité de tri** décrite dans la User Story.

Réponse question #29

Cette question correspond à l'objectif d'apprentissage **FL-4.5.3 (K3)** - Utiliser le développement piloté par les tests d'acceptation (ATDD) pour dériver les cas de tests.

- **a) Faux** → Cette approche confond les tests unitaires avec les tests d'acceptation. Dans l'ATDD, l'objectif est de créer des tests d'acceptation automatisés qui valident les comportements attendus du point de vue de l'utilisateur, pas de décomposer le scénario en tests unitaires.
- **b) Faux** → Ajouter des détails techniques d'implémentation va à l'encontre des principes de l'ATDD. Les tests d'acceptation doivent se concentrer sur le comportement observable et les critères d'acceptation métier, pas sur les détails d'implémentation technique.
- **c) Correct** → Cette approche est parfaitement alignée avec les principes de l'ATDD. Le scénario actuel est trop vague et ne fournit pas d'exemples concrets permettant de déterminer si l'implémentation est correcte. En ajoutant plusieurs exemples avec des valeurs d'entrée spécifiques (différents niveaux de revenus, scores de crédit, valeurs de propriété) et les résultats attendus correspondants (éligible/non éligible), le scénario devient beaucoup plus clair.

Réponse question #31

Cette question correspond à l'objectif d'apprentissage **FL-5.1.4 (K2)** - Utiliser des techniques d'estimation pour calculer l'effort de test requis.

- **a) Correct** → Pour calculer l'estimation basée sur des ratios, il faut d'abord calculer le ratio moyen entre le budget développement et le coût du testing pour tous les projets historiques:
 - Projet 1 : $300/2,500 = 0.12$
 - Projet 2 : $270/1,800 = 0.15$
 - Projet 3 : $560/3,200 = 0.175$
 - Projet 4 : $430/2,700 = 0.159$
 - Moyenne = $(0.12+0.15+0.175+0.159)/4 = 0.16$ Ensuite, on applique ce ratio au coût de recherche estimé: $2,400 \text{ k€} \times 0.16 = 384 \text{ k€}$
- **b) Faux** → Cette valeur semble être une simple approximation sans appliquer la technique d'estimation par ratios.
- **c) Faux** → Ce résultat pourrait provenir d'un calcul erroné utilisant un ratio approximatif de 0.156 plutôt que le ratio moyen exact.
- **d) Faux** → Cette réponse semble résulter d'un calcul avec un ratio de 0.17, qui n'est pas la moyenne exacte des ratios individuels.

Réponse question #32

Cette question correspond à l'objectif d'apprentissage **FL-5.1.5 (K3)** - Appliquer la priorisation des cas de test.

Pour déterminer l'ordre d'exécution, nous devons considérer à la fois les dépendances logiques et les priorités.

- TC 101 n'a pas de dépendance et doit être exécuté en premier
- TC 102 dépend de TC 101 et doit être exécuté en deuxième
- TC 103 dépend de TC 102 et a la priorité la plus élevée (1), donc il doit être exécuté en troisième
- TC 104 et TC 105 dépendent tous deux de TC 103

Réponse question #35

Cette question correspond à l'objectif d'apprentissage **FL-5.3.1 (K1)** – Rappeler des métriques utilisées dans les tests.

- **a) Faux** → C'est une métrique de gestion de projet, pas de test.
- **b) Faux** → Utile à l'analyse des exigences, mais pas une métrique standard de test.
- **c) Correct** → Le **taux de réussite des cas de test** est une **métrique clé** pour suivre la progression et la qualité.
- **d) Faux** → Ce n'est ni pertinent ni représentatif de la qualité du produit.

Réponse question #36

Cette question correspond à l'objectif d'apprentissage **FL-5.3.2 (K2)** – Résumer les objectifs, le contenu et les destinataires des rapports de test.

- **a) Faux** → Le rapport ne vise pas à convaincre, mais à **informer objectivement**.
- **b) Correct** → C'est exactement ce que doit contenir un **rapport de test clair et utile** pour les parties prenantes.
- **c) Faux** → Ce rôle revient aux développeurs, pas aux testeurs.
- **d) Faux** → Le fonctionnement de l'outil n'est pas pertinent pour la majorité des destinataires.

Réponse question #37

Cette question correspond à l'objectif d'apprentissage **FL-5.4.1 (K2)** – Résumer la manière dont la gestion de configuration soutient les tests.

- **a) Faux** → Ce serait insuffisant, et trop simpliste.
- **b) Faux** → Ce n'est ni lié à la vitesse ni au type de test.
- **d) Correct** → C'est l'essence même de la gestion de configuration dans un contexte de test multi-version.
- **c) Faux** → Cela relève du contrôle de version de code, pas du test.

Réponse question #40

Cette question correspond à l'objectif d'apprentissage **FL-6.2.1 (K1)** – Rappeler les avantages et les risques de l'automatisation des tests.

- **a) Faux** → L'automatisation peut être coûteuse à mettre en place, mais cela dépend du contexte et du type de tests automatisés.
- **b) Correct** → Les tests exploratoires, UX et certains tests de validation métier nécessitent une prise de décision humaine et ne peuvent pas être automatisés.
- **c) Faux** → L'automatisation est justement idéale pour les tests de régression, car elle permet de les exécuter régulièrement.
- **d) Faux** → Les outils d'automatisation peuvent générer des rapports de test, mais cela ne signifie pas que l'automatisation remplace tous les tests manuels.