



LES TECHNIQUES DE TEST

By Dr. Imèn Khanchouch

CATÉGORIES DE TECHNIQUES DE TEST

Le choix des techniques de test à utiliser dépend d'un certain nombre de facteurs :

- Type de composant ou de système
- Complexité du composant ou des systèmes
- Exigences client
- Types de risques
- Objectifs du test

CHOIX DES TECHNIQUES DE TEST

- Documentation disponible
- Connaissances et compétences des testeurs
- Outils disponibles
- Temps et budget
- Modèle de cycle de vie du développement logiciel
- Utilisation prévue du logiciel
- Expérience antérieure de l'utilisation des techniques de test sur le composant ou le système à tester.

CATÉGORIES DE TECHNIQUES DE TEST ET LEURS CARACTÉRISTIQUES

Les techniques de test sont classées en :

- Techniques **boîte-noire**
- Techniques **boîte blanche**
- Techniques **basées sur l'expérience.**

→ Les testeurs combinent les techniques pour un meilleur résultat

LES TECHNIQUES DE TEST BOÎTE-NOIRE (BASÉES SUR LE COMPORTEMENT OU TECHNIQUES DE CONCEPTION)

- Les techniques de test de boîte-noire (aussi appelées techniques comportementales ou techniques basées sur le comportement) sont basées sur une analyse de la base de test appropriée (par exemple, documents d'exigences formelles, spécifications, cas d'utilisation, User Stories ou processus métier).
- Ces techniques sont applicables aux tests fonctionnels et non-fonctionnels.
- Les techniques de test de boîte noire se concentrent sur les entrées et sorties de l'objet de test sans référence à sa structure.

LES TECHNIQUES DE TEST BOÎTE-NOIRE (BASÉES SUR LE COMPORTEMENT OU TECHNIQUES DE CONCEPTION)

Les caractéristiques communes des techniques de test boîte-noire incluent :

- Les conditions de test, les cas de test et les données de test sont dérivés d'une base de test qui peut inclure des exigences logicielles, des spécifications, des cas d'utilisation, et des User Stories.
- Les cas de test peuvent être utilisés pour détecter les écarts entre les exigences et l'implémentation des exigences, ainsi que les défauts au niveau des exigences.
- La couverture est mesurée en fonction des éléments de la base de test évalués et de la technique appliquée à cette base de test.

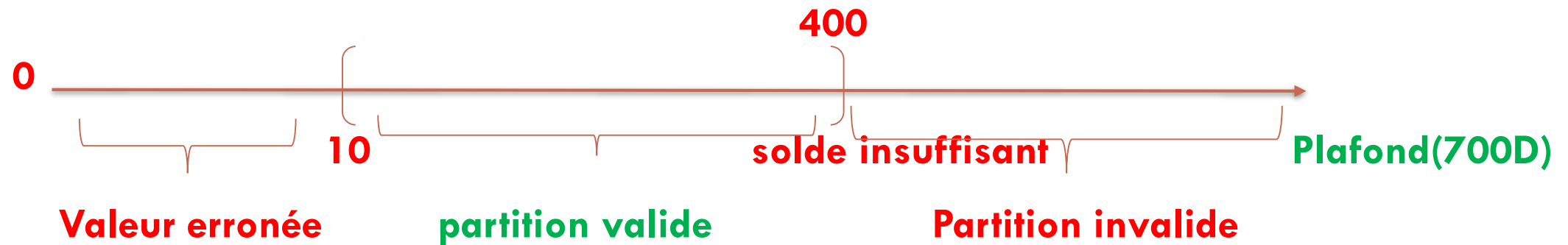
PARTITIONS D'ÉQUIVALENCE

- Les partitions d'équivalence divisent les données en partitions (également connues sous le nom de classes d'équivalence) de telle sorte que tous les éléments d'une partition donnée sont supposés être traités de la même manière.
- Il y a des partitions d'équivalence pour les valeurs valides et invalides.
- Les valeurs valides sont des valeurs qui doivent être **acceptées** par le composant ou le système.
- Les valeurs invalides sont des valeurs qui doivent être **rejetées** par le composant ou le système -> le système refuse la donnée (message d'erreur)

PARTITIONS D'ÉQUIVALENCE

- Toute partition peut être divisée en sous-partitions si nécessaire.
- Chaque valeur doit appartenir à une et une seule partition d'équivalence.
- Pour obtenir une couverture de 100% avec cette technique, les cas de test doivent couvrir toutes les partitions identifiées (y compris les partitions invalides) en utilisant au moins une valeur de chaque partition.
- La couverture est mesurée comme étant le **nombre de partitions d'équivalence testées** par au moins une valeur, divisé par **le nombre total de partitions d'équivalence identifiées**, généralement exprimé en pourcentage.
- Les partitions d'équivalence sont applicables à tous les niveaux de test.

EXEMPLE



1. Retrait Argent – Valeur erronée (5d)
2. Retrait Argent – Montant servi (200d)
3. Retrait Argent – Solde insuffisant

EXAMPLE

Invalide		Valide		Invalide
0	9	10	400	401
Partition 1		Partition 2		Partition 3

ANALYSE DES VALEURS LIMITES

- L'analyse des valeurs limites est une extension des partitions d'équivalence, mais ne peut être utilisée que lorsque la partition est ordonnée, composée de données numériques ou séquentielles.
- Les valeurs minimale et maximale (ou première et dernière valeur) d'une partition sont ses valeurs limites.
- Le comportement aux bornes de chaque partition d'équivalence.

TEST DE TABLES DE DÉCISION

- Les tables de décision sont un bon moyen pour répertorier les règles métier complexes qu'un système doit mettre en œuvre.
- Lors de la création de tables de décision, le testeur identifie les conditions (souvent des entrées) et les actions résultantes (souvent des sorties) du système. Celles-ci forment les lignes du tableau, généralement avec les conditions en haut et les actions en bas.
- Chaque colonne correspond à une règle de décision qui définit une combinaison unique de conditions qui aboutit à l'exécution des actions associées à cette règle. → Fait partie des techniques de **test combinatoires**.
- Les valeurs des conditions et des actions sont généralement affichées sous forme de valeurs booléennes (vraies ou fausses) ou de valeurs discrètes (par exemple, rouge, vert, bleu), mais peuvent également être des nombres ou des plages de nombres.

EXEMPLE

Avantages accordés à un client :

- Nouveau client : 15%
- Carte fidélité : 10%
- Coupon : 20%

Un seul avantage peut être appliqué à la fois. Quelle remise sera appliquée par le système ?

SOLUTION

$$2^3 = 8$$

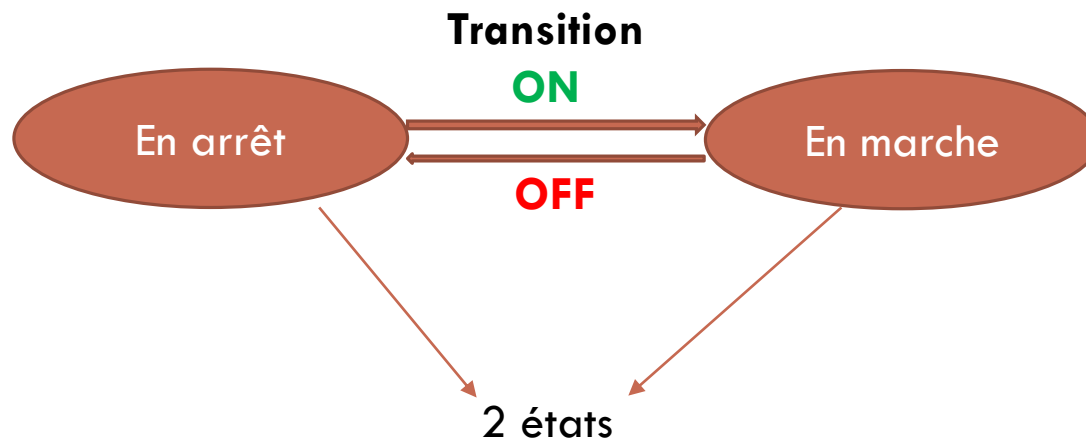
	1	2	3	4	5	6	7	8
Nouveau client 15%	V	V	V	V	F	F	F	F
Carte fidélité 10%	V	V	F	F	V	V	F	F
Coupon 20%	V	F	V	F	V	F	V	F
Remise ??	10%	10%	15%	15%	10%	10%	20%	0%

TEST DES TRANSITIONS D'ÉTAT

- Un diagramme de transitions d'état montre les états possibles du logiciel, ainsi que la façon dont le logiciel entre, sort et évolue entre ces états.
- Une transition est déclenchée par un événement (par exemple, l'entrée d'une valeur par l'utilisateur dans un champ). L'événement entraîne une transition. Si le même événement peut entraîner deux transitions différentes ou plus à partir d'un même état, cet événement peut être qualifié par une condition de garde.
- Le changement d'état peut entraîner une action du logiciel (par exemple, l'affichage du résultat d'un calcul ou d'un message d'erreur).

TEST DES TRANSITIONS D'ÉTAT

- Les tests peuvent être conçus pour couvrir une séquence de plusieurs états, pour exercer tous les états, pour exercer toutes les transitions, pour exercer des séquences spécifiques de transitions ou pour tester des transitions invalides.

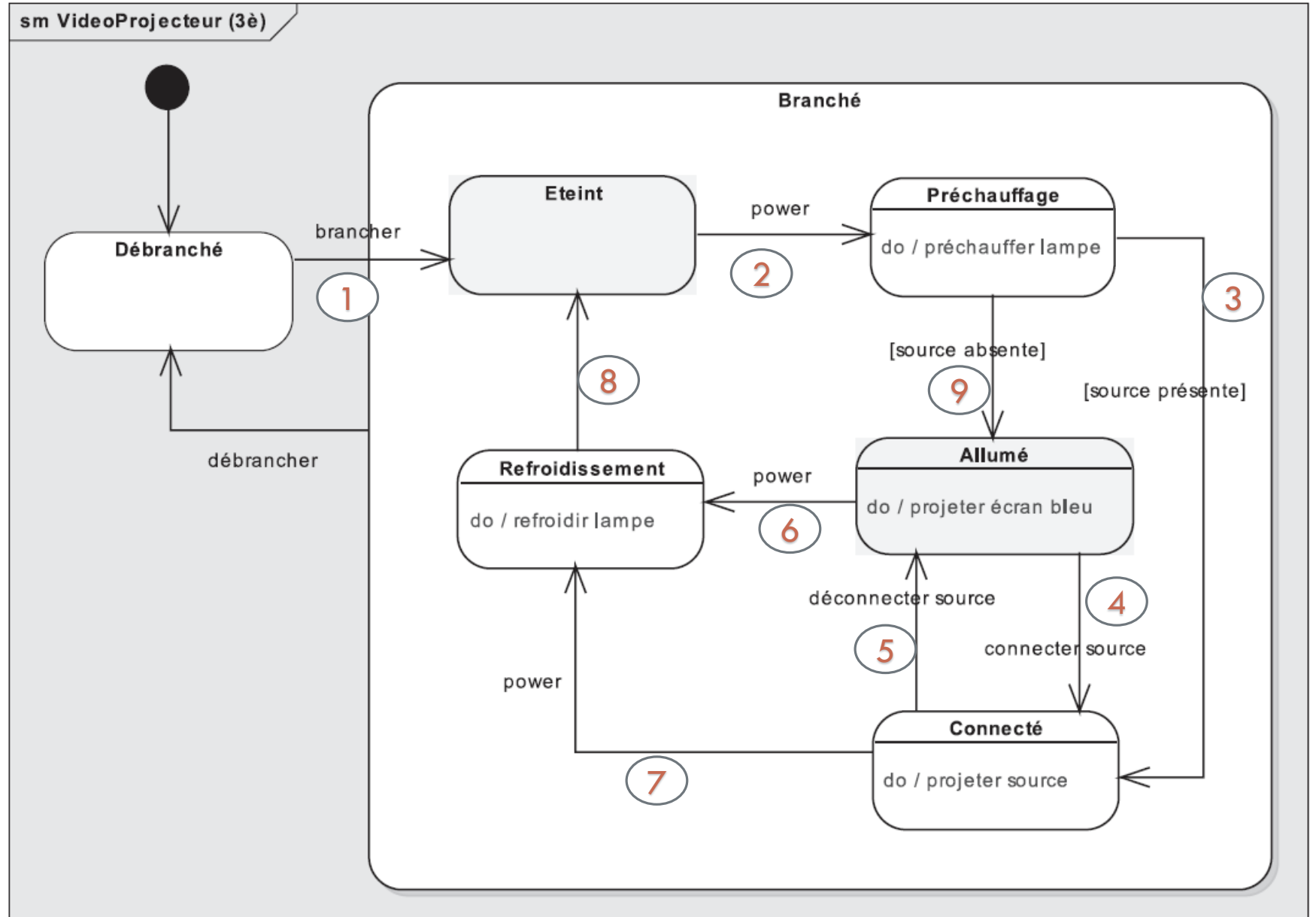


TEST DES TRANSITIONS D'ÉTAT

Ces transitions, états et conditions peuvent être représentés dans :

- Un **tableau de transition d'état** : montre les transitions valides et invalides
- Un **diagramme état-transition (UML)** : montre les transitions disponibles
- Les diagrammes de transition d'états ne montrent normalement que les transitions valides et excluent les transitions invalides.
- La couverture est généralement mesurée comme étant **le nombre d'états ou de transitions identifiés testés**, divisé par **le nombre total d'états ou de transitions identifiés** dans l'objet de test, normalement exprimé en pourcentage.

EXEMPLE



TEST DES TRANSITIONS D'ÉTAT

Nombre de cas de test total = nombre d'états * nombre de transition (incluant les transitions valides et invalides)

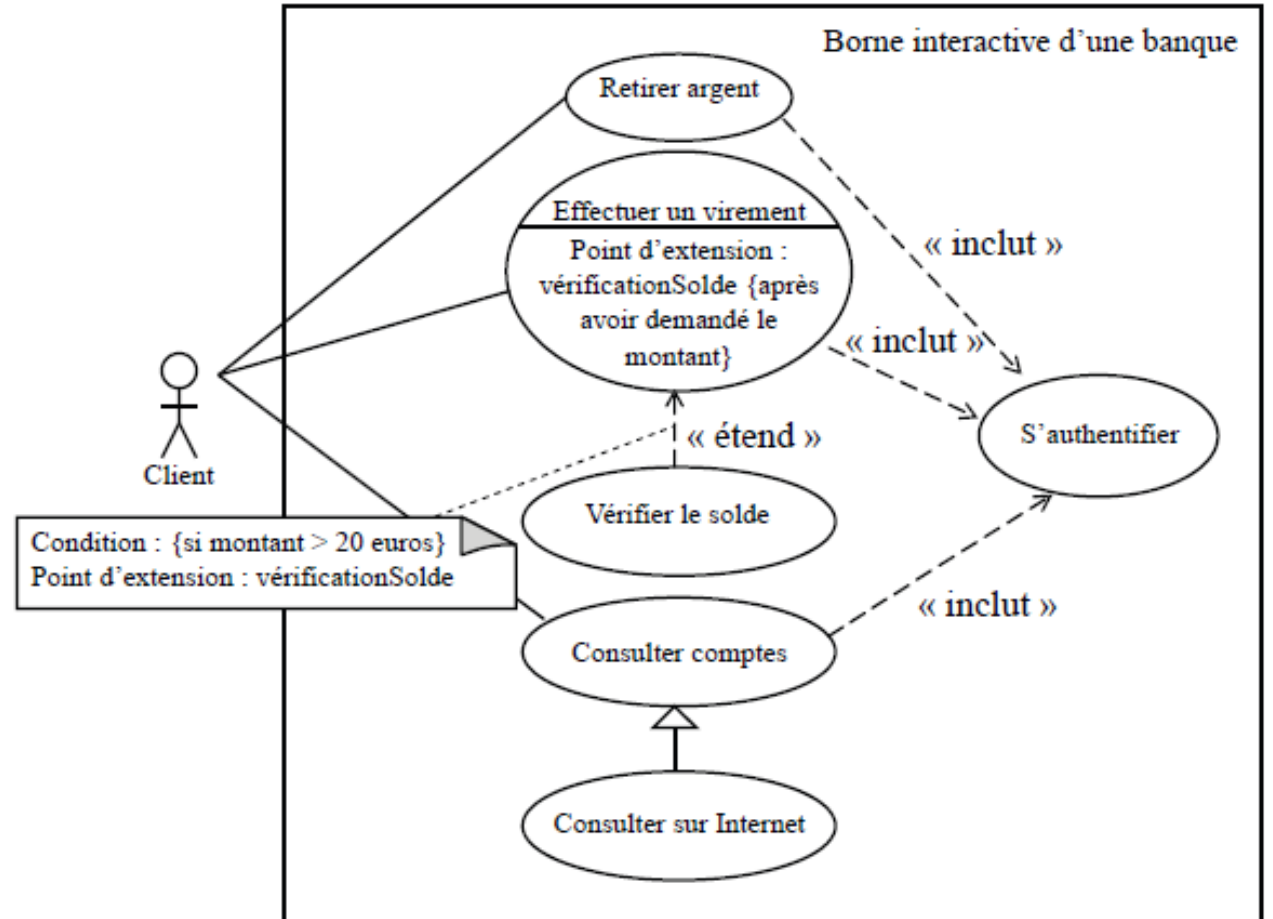
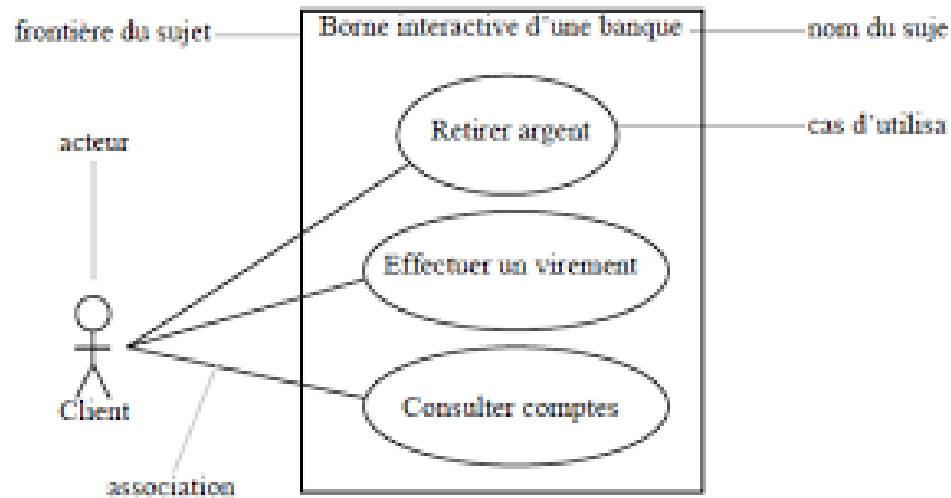
Pour l'exemple précédant :

- Nombre de transitions valides (Ce que le système doit faire) : 9
- Nombre des états : 6
- Nombre de transitions invalides : 2
- Nombre de cas de test total : $(9+2) * 6 = 66$

TEST DES CAS D'UTILISATION

- Les tests peuvent être dérivés de cas d'utilisation. Les cas d'utilisation sont une façon spécifique de concevoir les interactions avec le logiciel pour représenter des exigences.
- Les cas d'utilisation sont associés à des acteurs (utilisateurs humains, matériel externe ou autres composants ou systèmes) et à des sujets (le composant ou système auquel le cas d'utilisation est appliqué).
- La couverture peut être mesurée par le pourcentage **des comportements de cas d'utilisation testés** divisé par **le nombre total des comportements du cas d'utilisation**, généralement exprimé en pourcentage.

EXEMPLE



LES TECHNIQUES DE TEST DE BOITE BLANCHE

Les techniques de test en boîte blanche (aussi appelées techniques structurelles ou techniques basées sur la structure) sont basées sur une analyse de l'architecture, de la conception détaillée, de la structure interne ou du code de l'objet de test. Contrairement aux techniques de test de boîte-noire, les techniques de test de boîte-blanche se concentrent sur la structure et le traitement à l'intérieur de l'objet de test.

Les caractéristiques communes des techniques de test des boîte-blanche incluent :

- Les conditions de test, les cas de test et les données de test sont dérivés d'une base de test qui peut inclure le code, l'architecture logicielle, la conception détaillée ou toute autre source d'information concernant la structure du logiciel.
- La couverture est mesurée en fonction des éléments testés au sein d'une structure donnée (p. ex. le code ou les interfaces).
- Les spécifications sont souvent utilisées comme source d'information supplémentaire pour déterminer le résultat attendu des cas de test.

LES TECHNIQUES DE TEST DE BOITE BLANCHE

- **Niveau de composant** : la structure d'un composant logiciel ; instructions, décisions..
- **Niveau d'intégration** : la structure peut être un arbre ou graphe d'appel ou un diagramme (où des modules font appel à d'autres modules)
- **Niveau système** : la structure peut être une structure de menus, ou la structure d'une page web.

LES TECHNIQUES DE TEST DE BOITE BLANCHE

■ Test et couverture des instructions

Le test des instructions exerce les instructions exécutables dans le code.

La couverture est mesurée comme le *nombre d'instructions exécutées par les tests*, divisé par *le nombre total d'instructions exécutables dans l'objet de test*, généralement exprimé en pourcentage.

■ Test et couverture des décisions

La couverture est mesurée comme le *nombre de résultats de décision exécutés par les tests*, divisé par *le nombre total de résultats de décision dans l'objet de test*, généralement exprimé en pourcentage.

➤ L'obtention d'une couverture à 100 % des décisions garantit une couverture à 100 % des instructions (mais pas l'inverse).

LES TECHNIQUES DE TEST BASÉES SUR L'EXPÉRIENCE

- Lors de l'application de techniques de test basées sur l'expérience, les cas de test sont basés sur les compétences et l'intuition du testeur, ainsi que sur son expérience avec des applications et des technologies similaires. Ces techniques peuvent être utiles pour identifier les tests qui n'ont pas été facilement identifiés par d'autres techniques plus systématiques. Selon l'approche et l'expérience du testeur, ces techniques peuvent atteindre des degrés de couverture et d'efficacité très variables.

Les techniques basées sur l'expérience les plus couramment utilisées sont :

- L'estimation d'erreur
- Les tests exploratoires
- Les tests basés sur des checklists.

LES TECHNIQUES DE TEST BASÉES SUR L'EXPÉRIENCE

L'estimation d'erreur est une technique utilisée pour anticiper les erreurs, les défauts et les défaillances, sur la base des connaissances du testeur, y compris :

- Comment l'application a fonctionné antérieurement
- Quels types d'erreurs les développeurs ont tendance à faire
- Les défaillances qui se sont produites dans d'autres applications.

Une approche méthodique de la technique par estimation d'erreur consiste à créer une liste d'erreurs, de défauts et de défaillances possibles, et à concevoir des tests qui exposeront ces défaillances et les défauts qui les ont causées.

Ces listes d'erreurs, de défauts et de défaillances peuvent être construites sur la base de l'expérience, de données sur les défauts et les défaillances, ou à partir de connaissances générales sur les causes des défaillances logicielles.

LES TECHNIQUES DE TEST BASÉES SUR L'EXPÉRIENCE

- Dans **les tests exploratoires**, des tests informels (non prédéfinis) sont conçus, exécutés, enregistrés et évalués **dynamiquement** pendant l'exécution des tests (après la livraison)
- Les tests exploratoires sont généralement utilisés pour enrichir le cahier de test.
- Les tests exploratoires sont les plus utiles lorsqu'il y a peu de spécifications ou des spécifications inadéquates ou des contraintes de temps importantes sur les tests.
- Les tests exploratoires sont également utiles pour compléter d'autres techniques de test plus formelles.

LES TECHNIQUES DE TEST BASÉES SUR L'EXPÉRIENCE

- Dans **les tests basés sur des checklists**, les testeurs conçoivent, implémentent et exécutent des tests pour couvrir les conditions de test figurant dans une checklist. Au cours de l'analyse, les testeurs créent une nouvelle checklist ou complètent une checklist existante.
- De telles checklists peuvent être construites sur la base de l'expérience, de la connaissance de ce qui est important pour l'utilisateur, ou de la compréhension du pourquoi et du comment des défaillances logicielles.
- Des checklists peuvent être créées pour prendre en charge différents types de tests, y compris les tests fonctionnels et non-fonctionnels. En l'absence de cas de tests détaillés, les tests basés sur des checklists peuvent fournir des lignes directrices et un certain degré de cohérence. Comme il s'agit de listes de haut niveau, il est probable qu'il y ait une certaine variabilité dans les tests réels, ce qui pourrait entraîner une plus grande couverture des tests.

Merci 😊