

## Chapitre 4: Techniques de test

Catégories de techniques de test	Les bases de test
Boîte-noire	Exigences, cas d'utilisation, Histoire de l'utilisateur
Boîte-blanche	analyse d'architecture, conception détaillée, structure interne, code
Basées sur l'expérience	Connaissance de (testeurs, développeurs, utilisateurs)

Techniques de test boîte-noire	Description
<b>Partitions d'équivalence</b>	<ul style="list-style-type: none"> <li>• Les partitions sont appelées classes d'équivalence</li> <li>• Une partition d'équivalence contenant des valeurs valides est appelée "partition d'équivalence valide".</li> <li>• Une partition d'équivalence contenant des valeurs invalides est appelée "partition d'équivalence invalide".</li> <li>• Chaque valeur doit appartenir à une et une seule partition d'équivalence</li> <li>• Toute partition peut être divisée en sous-partitions si nécessaire</li> <li>• 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.</li> <li>• Lorsque des partitions d'équivalence invalides sont utilisées dans des cas de test, elles doivent être testées individuellement</li> </ul>
<b>Analyse des valeurs limites</b>	<ul style="list-style-type: none"> <li>• 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 d'une partition sont ses valeurs limites.</li> <li>• Certaines variantes de cette technique identifient trois valeurs limites par limite: les valeurs en-dessous, sur et juste au-dessus de la limite.</li> <li>• L'analyse des valeurs limites peut être appliquée à tous les niveaux de test. Cette technique est généralement utilisée pour tester les exigences qui nécessitent une série de valeurs numériques</li> <li>• La couverture des limites d'une partition est mesurée comme le nombre de valeurs limites testées, divisé par le nombre total de valeurs limites identifiées, généralement exprimé en pourcentage.</li> </ul>
<b>Test de tables de décision</b>	<ul style="list-style-type: none"> <li>• 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.</li> <li>• 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</li> <li>• Ces différents types de conditions et d'actions peuvent être regroupés dans la même table.</li> <li>• La table peut être réduite en supprimant les colonnes contenant des combinaisons impossibles de conditions, les colonnes contenant des combinaisons de conditions possibles mais irréalisables et les colonnes qui testent des combinaisons de conditions qui n'affectent pas le résultat.</li> <li>• La couverture minimale courante pour le test de tables de décision est d'avoir au moins un cas type par règle de décision dans la table.</li> <li>• La couverture est mesurée comme le nombre de règles de décision testées par au moins un cas de test, divisé par le nombre total de règles de décision, généralement exprimé en pourcentage.</li> </ul>
<b>Test des transitions d'état</b>	<ul style="list-style-type: none"> <li>• 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.</li> <li>• Une transition est déclenchée par un événement</li> <li>• Le changement d'état peut entraîner une action du logiciel</li> <li>• Un tableau de transition d'état montre toutes les transitions valides et les transitions potentiellement invalides entre les états</li> <li>• Les diagrammes de transition d'états ne montrent normalement que les transitions valides et excluent les transitions invalides.</li> <li>• 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.</li> <li>• 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.</li> </ul>
<b>Test des cas d'utilisation</b>	<ul style="list-style-type: none"> <li>• Les cas d'utilisation sont associés à des acteurs et à des sujets</li> <li>• Chaque cas d'utilisation spécifie un comportement qu'un sujet peut accomplir en collaboration avec un ou plusieurs acteurs</li> <li>• Un cas d'utilisation peut être décrit par des interactions et des activités, ainsi que par des préconditions, des post conditions et du langage naturel</li> <li>• Un cas d'utilisation peut inclure des variations possibles de son comportement de base, y compris un comportement exceptionnel et la gestion des erreurs</li> <li>• Les tests sont conçus pour exercer les comportements définis</li> <li>• 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.</li> </ul>

Techniques de test boîte-blanche	Description
<b>Test et couverture des instructions</b>	<ul style="list-style-type: none"> <li>• Les tests exercent les instructions exécutables potentielles dans le code.</li> <li>• le pourcentage des instructions exécutables qui ont été exécutées par une suite de tests .</li> </ul>
<b>Decision Testing and Coverage</b>	<ul style="list-style-type: none"> <li>• Le test de décision exerce les décisions dans le code et teste le code qui est exécuté en fonction des résultats de la décision.</li> <li>• La couverture des résultats des décisions.</li> </ul>

- PS : =>** • Lorsque la couverture à 100% des instructions est atteinte, elle garantit que toutes les instructions exécutables du code ont été testées au moins une fois, mais elle ne garantit pas que toutes les décisions ont été testées.
- L'obtention d'une couverture à 100 % des décisions garantit une couverture à 100 % des instructions (pas l'inverse).

Techniques de test basées sur l'expérience	Description
<b>Estimation d'erreur</b>	<p>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 :</p> <ul style="list-style-type: none"> <li>• Comment l'application a fonctionné antérieurement</li> <li>• Quels types d'erreurs les développeurs ont tendance à faire</li> <li>• Les défaillances qui se sont produites dans d'autres applications</li> <li>• Une <b>approche méthodique</b> 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.</li> </ul>
<b>Tests exploratoires</b>	<ul style="list-style-type: none"> <li>• Dans les tests exploratoires, des tests informels (non prédéfinis) sont conçus, exécutés, enregistrés et évalués dynamiquement pendant l'exécution des tests.</li> <li>• Les résultats des tests sont utilisés pour en apprendre davantage sur le composant ou le système, et pour créer des tests pour les parties qui pourraient avoir besoin de plus de tests.</li> <li>• Les tests exploratoires sont parfois réalisés en utilisant <b>des sessions de test</b> pour structurer l'activité.</li> <li>• Les tests exploratoires sont le plus utile lorsqu'il y a <b>peu de spécifications</b> ou des spécifications inadéquates ou <b>des contraintes de temps</b> importantes sur les tests.</li> </ul>
<b>Tests basés sur des checklists</b>	<ul style="list-style-type: none"> <li>• 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.</li> <li>• Au cours de l'analyse, les testeurs créent une nouvelle checklist ou complètent une checklist existante, mais les testeurs peuvent également utiliser une checklist existante sans modification.</li> <li>• Des checklists peuvent être créées pour prendre en charge différents types de tests, y compris les tests fonctionnels et non-fonctionnels.</li> <li>• 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.</li> <li>• 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, mais une reproductibilité plus faible.</li> </ul>