

- **Qu'est ce qu'un test statique ?**

Un test statique est une méthode d'analyse du code source ou de la documentation logicielle sans exécution, visant à détecter des erreurs, des incohérences ou des problèmes potentiels avant que le logiciel ne soit testé ou déployé.

- **Qu'est ce qu'un test dynamique ?**

Un test dynamique est une méthode d'évaluation de logiciels qui implique l'exécution du programme pour identifier des erreurs ou des anomalies en observant son comportement.

- **Quelle sont les différences entre un test statique et un test dynamique ?**

1- Test Statique :

- Le test statique examine le code source ou la documentation du logiciel sans exécution.
- Il vise à détecter des erreurs, des incohérences ou des problèmes potentiels en se basant sur l'analyse statique du code.
- Les exemples incluent l'examen de code, l'inspection de code et la revue de documentation.

2- Test Dynamique :

- Le test dynamique consiste à exécuter le logiciel avec des entrées spécifiques pour observer son comportement.
- Il vise à détecter des erreurs, des bogues ou des problèmes lors de l'exécution réelle du logiciel.
- Les exemples comprennent les tests unitaires, les tests d'intégration, les tests de système, les tests de performance et les tests de régression.

⇒ le test statique se concentre sur l'analyse du code et de la documentation sans exécution, tandis que le test dynamique consiste à exécuter le logiciel pour identifier des problèmes en cours d'exécution.

- **Est ce qu'on peut faire un test dynamique et un test statique en même temps ?**

Oui, il est possible de réaliser des tests statiques et dynamiques en même temps pour une évaluation plus complète de la qualité logicielle.

(réponse plus détaillée pour mieux comprendre)

Oui, il est tout à fait possible de réaliser des tests statiques et des tests dynamiques en même temps, et il est souvent recommandé de le faire pour garantir la qualité d'un logiciel. Les tests statiques et dynamiques sont des approches complémentaires qui permettent de détecter différents types de problèmes logiciels.

Lors du développement logiciel, vous pouvez effectuer des tests statiques tels que l'examen de code, l'inspection de code et la revue de documentation pour identifier des erreurs potentielles dans le code source et la conception avant même d'exécuter le programme. Cela permet de repérer des problèmes tels que des erreurs de logique, des violations de normes de codage, des incohérences ou des lacunes dans la documentation.

Une fois que le code a été examiné et corrigé en fonction des résultats des tests statiques, vous pouvez ensuite effectuer des tests dynamiques en exécutant le logiciel avec différentes entrées et conditions pour vérifier son fonctionnement réel. Cela permet de détecter des erreurs qui ne peuvent être identifiées que lors de l'exécution, telles que des erreurs de comportement, des bugs, des problèmes de performance, etc.

En combinant des tests statiques et dynamiques, vous augmentez vos chances de détecter un large éventail de problèmes, ce qui contribue à améliorer la qualité et la fiabilité de votre logiciel.

- **Qu'est ce qu'un test en boîte blanche ?**

Le test en boîte blanche (white box) est une technique de test qui évalue le logiciel en examinant sa structure interne, y compris le code source. Il se concentre sur la compréhension des détails de l'implémentation pour concevoir des tests basés sur la logique du code.

- **Qu'est ce qu'un test en boîte noire ?**

Le test en boîte noire évalue le logiciel en se basant sur ses entrées, sorties, et comportements, sans considération de sa structure interne ni de son code source.

- **Quelle est la différence entre le white box et le black box ?**

- Le test en boîte blanche examine la structure interne du logiciel, y compris le code source, pour concevoir des tests basés sur la logique du code.
- Le test en boîte noire se concentre sur les entrées, sorties et comportements observables du logiciel sans considérer la structure interne, en se basant sur les spécifications et les exigences.

- **Différences entre tester et déboguer :**

- Tester consiste à évaluer un logiciel pour identifier des erreurs, des bugs et des problèmes potentiels. Cela peut inclure l'exécution de scénarios de test pour vérifier que le logiciel fonctionne conformément aux spécifications.
- Déboguer, en revanche, consiste à trouver, analyser et corriger spécifiquement les erreurs et les bugs dans le code source d'un logiciel. Cela implique généralement l'utilisation d'outils de développement pour localiser et résoudre des problèmes spécifiques.
- Test se fait généralement par les testeurs et le débogage par les développeurs .

- **Qu'est ce qu'un test de régression ?**

Un test de régression consiste à vérifier qu'une modification logicielle n'a pas introduit de nouveaux bugs en réexécutant des tests déjà validés après une mise à jour ou une modification du logiciel.

- **Un test de performance?**

Un test de performance évalue la capacité et les performances d'un logiciel, notamment sa vitesse, sa robustesse et sa réactivité sous différentes conditions de charge ou de stress.

- **Qu'est ce qu'un test d'intégration?**

Un test d'intégration vérifie comment différents modules ou composants logiciels interagissent ensemble pour s'assurer qu'ils fonctionnent correctement en tant qu'ensemble intégré.

- **Un test de composant ?**

Un test de composant évalue individuellement un module ou une partie isolée d'un logiciel pour vérifier son bon fonctionnement.

- **Un test système?**

Un test système évalue l'ensemble du logiciel ou d'un système pour vérifier s'il fonctionne conformément aux exigences et aux spécifications globales.

- **Un test d'acceptation?**

Un test d'acceptation vérifie si le logiciel satisfait aux critères et aux exigences convenus avec le client ou les utilisateurs finaux, avant sa mise en production.

- **Modèle en V :**

C'est un modèle de développement linéaire où chaque phase de développement (comme la conception, le codage, les tests) est suivie d'une phase de test correspondante. Il garantit une bonne couverture des tests, mais les retours d'information sont limités.

- **Modèle itératif :**

Il divise le développement en itérations où chaque itération comprend des phases de conception, de développement et de test. Les itérations permettent d'ajuster le logiciel en fonction des retours, améliorant la flexibilité et la réactivité aux changements.