

1. Les différents types de tests

- Tests unitaires

Les tests unitaires sont très bas niveau et proches de la source d'une application. Ils consistent à tester les méthodes et fonctions individuelles des classes, composants ou modules utilisés par votre logiciel. Les tests unitaires sont généralement assez simples à automatiser et peuvent s'exécuter très rapidement grâce à un serveur d'intégration continue.

- Tests d'intégration

Les tests d'intégration vérifient que différents modules ou services utilisés par votre application fonctionnent bien ensemble. Par exemple, cela peut être tester l'interaction avec la base de données ou s'assurer que les microservices fonctionnent ensemble comme prévu.

- Tests fonctionnels

Les tests fonctionnels se concentrent sur les exigences métier d'une application. Ils vérifient uniquement la sortie d'une action et ne vérifient pas les états intermédiaires du système lors de l'exécution de cette action.

Il y a parfois une confusion entre les tests d'intégration et les tests fonctionnels car ils exigent tous deux que plusieurs composants interagissent les uns avec les autres. La différence est qu'un test d'intégration peut simplement vérifier que vous pouvez interroger la base de données, tandis qu'un test fonctionnel s'attendait à obtenir une valeur spécifique de la base de données telle que définie par les exigences du produit.

- Tests de bout en bout

Les tests de bout en bout reproduisent le comportement d'un utilisateur avec le logiciel dans un environnement d'application complet. Ils vérifient que divers flux d'utilisateurs fonctionnent comme prévu et peuvent être aussi simples que le chargement d'une page web ou la connexion, ou des scénarios beaucoup plus complexes vérifiant des notifications par e-mail, des paiements en ligne, etc.

- Tests d'acceptation

Les tests d'acceptation sont des tests formels qui vérifient si un système satisfait les exigences métier. Ils nécessitent que l'ensemble de l'application soit en cours d'exécution pendant les tests et se concentrent sur la reproduction des comportements des utilisateurs. Cependant, ils peuvent également aller plus loin et mesurer les performances du système, refuser des changements si certains objectifs ne sont pas atteints.

- Tests de performance

Les tests de performance évaluent comment un système se comporte sous une charge de travail particulière. Ces tests aident à mesurer la fiabilité, la vitesse, la scalabilité et la réactivité d'une application. Par exemple, un test de performance peut observer les temps de réponse lors de l'exécution d'un grand nombre de requêtes, ou déterminer comment un système se comporte avec une quantité importante de données. Ils peuvent déterminer si une application respecte les exigences de performance, localiser des goulots d'étranglement, mesurer la stabilité pendant les périodes de trafic intense, et plus encore.

2. Teste doubles

Les "test doubles" sont des objets ou composants utilisés dans les tests logiciels pour remplacer un composant ou une dépendance réelle par une version simplifiée dans le but de réaliser des tests. Le terme "test double" est un concept général qui englobe divers types d'objets utilisés dans les tests, notamment :

- Objets Dummies :

Ce sont des objets qui sont passés en paramètre mais qui ne sont jamais réellement utilisés. Ils sont généralement utilisés pour satisfaire des exigences de paramètres sans influencer le comportement du test.

- Objets Fakes :

Les objets Fakes sont des implémentations simplifiées d'un composant, souvent créées spécifiquement pour les tests. Ils possèdent une version simplifiée du comportement du composant réel.

- Stubs :

Les Stubs fournissent une réponse prédéfinie aux appels effectués pendant le test. Ils sont utilisés pour simuler des conditions ou des scénarios spécifiques afin de tester différentes trajectoires de code.

- Spies :

Les Spies sont des objets qui enregistrent des informations sur les appels de méthodes effectués pendant un test. Ils permettent au test de vérifier que certaines méthodes ont été appelées et avec quels paramètres.

- Mocks :

Les Mocks sont des objets qui simulent le comportement d'objets réels, mais qui permettent également au test de définir des attentes sur la manière dont ils doivent être appelés. Les Mocks sont utilisés pour vérifier que des méthodes spécifiques sont appelées avec des paramètres spécifiques.

L'utilisation de test doubles est une pratique courante en tests unitaires, en particulier lorsqu'il s'agit de dépendances externes telles que des bases de données, des services web ou des API externes. En remplaçant les dépendances réelles par des test doubles, les tests peuvent être rendus plus isolés, déterministes et axés sur des comportements spécifiques sans dépendre des systèmes externes réels