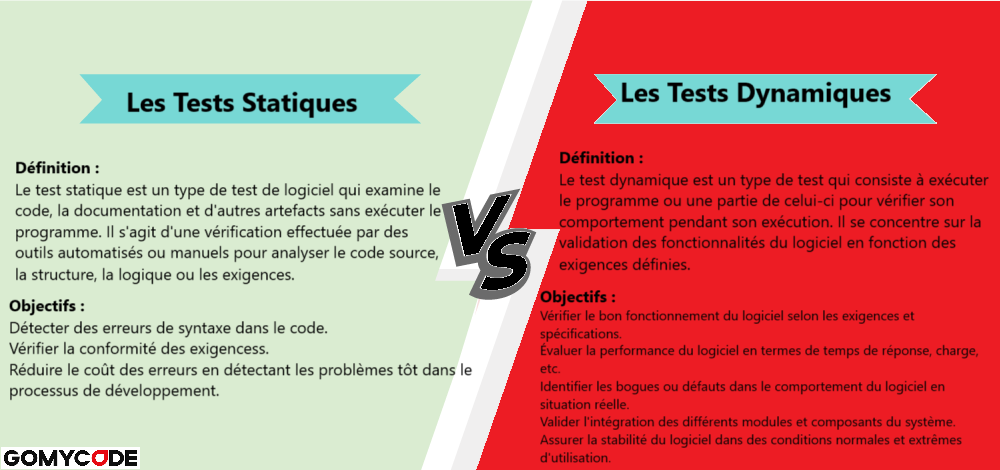
**Différences entre les tests statiques et les tests dynamiques**

Certains types de défauts ne peuvent être trouvés que par le test statique ou le test dynamique.

* Les tests statiques sont effectués manuellement sans exécuter l'application (par exemple, en lisant le code pour détecter les erreurs de syntaxe).
* Les tests dynamiques impliquent l'exécution des cas de test sur le produit logiciel réel ou le code de programmation.
* Pour en savoir plus sur les tests statiques et dynamiques, consultez ce [**Lien**](https://emnamagri.github.io/teststatique-dynamique/).



Les pratiques de test statique et de test dynamique se complètent mutuellement. Voici quelques différences:

* Le test statique détecte directement les défauts, tandis que le test dynamique provoque des défaillances à partir desquelles les défauts associés sont déterminés par une analyse ultérieure
* Le test statique peut détecter plus facilement les défauts qui se trouvent sur des chemins dans le code qui sont rarement exécutés ou difficiles à atteindre en utilisant le test dynamique.
* Le test statique peut être appliqué à des produits d'activité non exécutables, alors que le test dynamique ne peut être appliqué qu'à des produits d'activité exécutables
* Le test statique peut être utilisé pour mesurer des caractéristiques de qualité qui ne dépendent pas de l'exécution du code (par ex, la maintenabilité), tandis que les tests dynamiques peuvent être utilisés pour mesurer les caractéristiques de qualité qui dépendent de l'exécution du code (par exemple, l'efficacité des performances)

Les défauts typiques qui sont plus faciles et/ou moins coûteux à trouver grâce aux tests statiques comprennent :

* Les défauts dans les exigences (par exemple, les incohérences, les ambiguïtés, les contradictions, les erreurs, etc, incohérences, ambiguïtés, contradictions, omissions, inexactitudes, duplications).
* Défauts de conception (par exemple, structures de base de données inefficaces, mauvaise modularisation).
* Certains types de défauts de codage (par exemple, variables avec des valeurs non définies, variables non déclarées.
* Code inaccessible ou dupliqué, complexité excessive du code).
* Déviances par rapport aux normes (par exemple, manque d'adhésion aux conventions de dénomination).
* Défauts de codage (par exemple, code non déclaré, etc, manque d'adhésion aux conventions de dénomination dans les normes de codage).
* Spécifications d'interface incorrectes (par ex, nombre, type ou ordre des paramètres inadéquats).
* Types spécifiques de vulnérabilités en matière de sécurité.
* Lacunes ou imprécisions dans la couverture de la base de test.