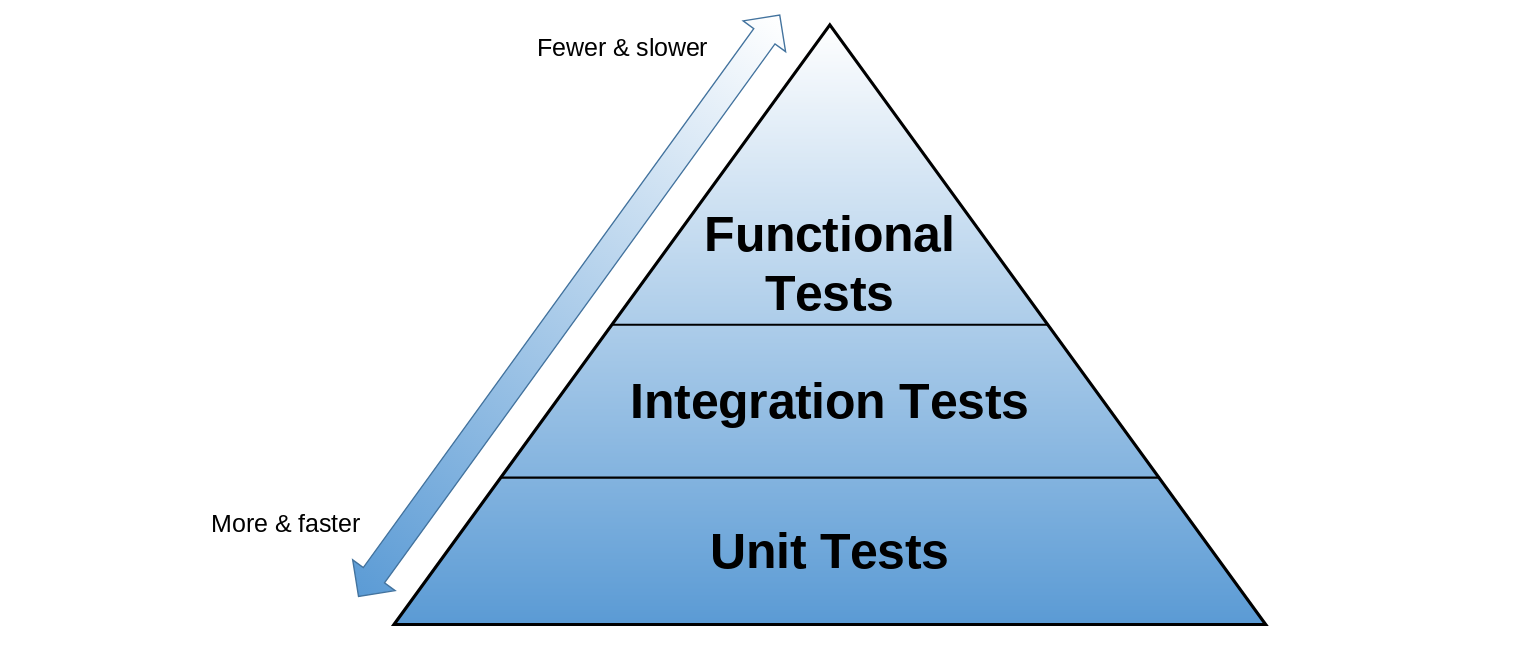
LES TESTS UNITAIRES ET LE TDD (Test Driving Développement)



Les différents types de tests :

* [**Tests unitaires**](https://en.wikipedia.org/wiki/Unit_testing) : ce sont les [**tests de développement**](https://en.wikipedia.org/wiki/Unit_testing) les plus nombreux, de bas niveau. Les tests unitaires ne garantissent pas seulement que le système fait ce qu'il est censé faire, mais aussi qu'il est correctement pris en compte là où les composants individuels sont découplés. Les tests unitaires se concentrent sur l'exercice de classes et de méthodes spécifiques s'exécutant dans un isolement complet, ce qui devient plus difficile si la classe que vous souhaitez tester est étroitement associée à ses dépendances / collaborateurs. Ces tests valident le comportement des constructions de programmation de base comme les classes et les algorithmes qu'elles contiennent. Ils doivent être petits et rapides, car leur intention est d'être fréquemment exécutés par les développeurs lorsqu'ils implémentent des fonctionnalités et modifient celles existantes.
* [**Tests d'intégration**](https://martinfowler.com/bliki/IntegrationTest.html) : ces tests vont d'un niveau d'abstraction plus élevé que les tests unitaires. Ils s'assurent que les composants développés indépendamment fonctionnent correctement ensemble.
* [**Tests fonctionnels**](https://en.wikipedia.org/wiki/Functional_testing) : ce sont les tests au plus haut niveau d'abstraction. Ceux-ci essaient d'imiter étroitement l'expérience de l'utilisateur avec le système en interagissant avec l'application comme le ferait un utilisateur. En termes d'application Web, cela signifie faire des requêtes HTTP, cliquer sur des boutons, remplir et soumettre des formulaires, inspecter les résultats HTML, etc. Comme ces tests exercent le système complet de bout en bout, ils sont autorisés à être un peu plus lents. Ce n'est pas un problème car ils ne sont pas exécutés aussi fréquemment que les tests unitaires ou d'intégration.

Pratiquement tous les écosystèmes de développement logiciel sérieux ont au moins un framework ou une bibliothèque de tests automatisés que nous pouvons exploiter pour écrire nos tests. Pour nos besoins dans cet article, nous allons utiliser le framework PHP Symfony qui s'intègre parfaitement à **[PHPUnit](https://phpunit.de/)** pour fournir aux développeurs un moyen efficace et même amusant d'écrire des tests

**Le TDD : Test-Driven Development**, ou Développements Pilotés par les **Tests** en français, est une méthode de développement de logiciel, qui consiste à concevoir un logiciel par petits pas, de façon itérative et incrémentale, en écrivant chaque **test** avant d'écrire le code source et en remaniant le code continuellement.

**Notre but est de commençer à intégrer des tests unitaires pour éviter les bugs de non régréssions et d’initier une solution de tests automatisés pour amener plus de qualité et avoir une meilleure vision globale de notre développement.**

TP : Nous allons aborder les tests unitaires via la construction d’une simple application qui permettra d’afficher la météo sur une page web.

https://github.com/jprivet-dev/symfony-tdd#pr%C3%A9sentation