

Tests

Comprendre la mise en oeuvre des
tests



Compétence demandée :
**Comprendre la mise en oeuvre de
tests**

A votre avis, à quoi servent les tests ?



Pourrait-on automatiser des tests ?



Les tests

Les **tests** sont classés en 3 catégories : les **tests unitaires**, les **tests d'intégration** et les **tests e2e**. Les tests sont implémentés en code visant à tester une autre partie du code.

Les **tests unitaires** sont donc des programmes et testent de manière unitaire **chaque fonction ou méthode** du programme.

Les **tests d'intégration** sont donc des programmes et testent **l'intégration de plusieurs fonction ou méthode** du programme.

Les **tests e2e** sont aussi appelé tests d'application et sont donc des programmes et testent **l'ensemble d'un programme, de bout à bout.**

Mise en oeuvre

```
<?php
// tests/Service/PlanningServiceTest.php

namespace App\Tests\Service;

use App\Entity\Cursus;
use App\Service\PlanningService;
use Symfony\Bundle\FrameworkBundle\Test\KernelTestCase;

class PlanningServiceTest extends KernelTestCase
{
    public function testGet()
    {
        self::bootKernel();
        $container = static::getContainer();

        $fakeCursus = new Cursus();
        $fakeCursus->setName('DWM14');

        $planningService = $container->get(PlanningService::class);
        $planning = $planningService->get($fakeCursus);

        $this->assertEquals('DWM14', $planning['title']);
    }
}
```

```
<?php
// tests/Service/PlanningServiceTest.php

namespace App\Tests\Service;

use App\Entity\Cursus;
use App\Service\PlanningService;
use Symfony\Bundle\FrameworkBundle\Test\KernelTestCase;

class PlanningServiceTest extends KernelTestCase
{
    public function testGet()
    {
        self::bootKernel();
        $container = static::getContainer();

        $fakeCursus = new Cursus();
        $fakeCursus->setName('DWM14');

        $planningService = $container->get(PlanningService::class);
        $planning = $planningService->get($fakeCursus);

        $this->assertEquals('DWM14', $planning['title']);
    }
}
```

Qu'est-ce que génie logiciel ?



Le génie logiciel, l'ingénierie logicielle ou l'ingénierie du logiciel (en anglais : regroupant **computer science** et **software engineering**) est une science de génie industriel qui étudie **les méthodes de travail et les bonnes pratiques des ingénieurs** qui développent des logiciels.

QA

L'assurance qualité (QA) est un moyen de prévenir les erreurs et les défauts dans les produits fabriqués et d'éviter les problèmes lors de la livraison des produits ou des services aux clients ; que l'ISO 9000 définit comme « une partie du management de la qualité visant à garantir que les exigences de qualité seront satisfaites ».

Cette **prévention des défauts** dans l'assurance qualité diffère subtilement de la **détection des défauts** et du rejet dans le contrôle qualité et a été appelée un décalage vers la gauche car elle se concentre sur la qualité **plus tôt dans le processus**

Autres types de tests

Les tests unitaires sont donc des programmes et testent de manière unitaire chaque fonction ou méthode du programme.

Les tests d'intégration sont donc des programmes et testent l'intégration de plusieurs fonction ou méthode du programme.

- Tests unitaires
- Tests d'intégration
- Tests d'acceptation ou de recette
- Tests de performance et de charge

En informatique, le **test d'acceptation** (ou recette) est une phase de développement des projets, visant à **assurer formellement que le produit est conforme** aux spécifications (réponse donnée à un instant « t » aux attentes formulées). Elle s'inscrit dans les activités plus générales de qualification.

Cette étape implique, en la présence effective des différents acteurs du projet, **MOE et MOA**, le déroulement rigoureux de procédures de tests préalablement décrits (par un **plan de tests**), et l'identification de tout écart fonctionnel ou technique (par des **defects**).

La recette est l'étape de tests fait part la MOA qui intervient **avant une mise en production** (au moins majeure).

Les tests de charge

L'acceptation la plus courante de ce terme est celle dans laquelle ces tests logiciels vont avoir pour objectif de mesurer les temps de réponse d'un système applicatif en fonction de sa sollicitation.

Le comportement d'un système en fonction de la charge d'utilisateurs simultanés.

Seuls les tests de charge permettent de valider correctement une application ou un système avant déploiement, tant en qualité de service qu'en consommation de ressources.

La couverture de tests

La **couverture des tests** est la proportion des exigences (techniques ou fonctionnelles) couvertes par les tests.

L'objectif est d'avoir une couverture à 100%.

Sur le plan technique, la couverture de tests est mesurée sur tous les chemins possibles (aussi appelé branches) dans l'exécution d'un programme.

