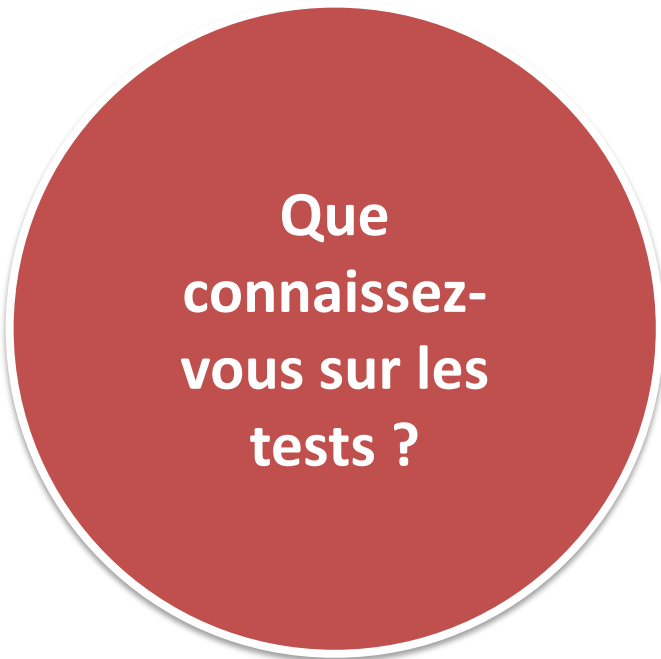


Les tests automatisés


Yannick Mazières

Département informatique

Cégep de Sainte-Foy

A solid red circle with a thin white border, containing white text.

**Que
connaissez-
vous sur les
tests ?**

A solid green circle with a thin white border, containing white text.

**Quelle est
votre opinion
sur les tests ?**

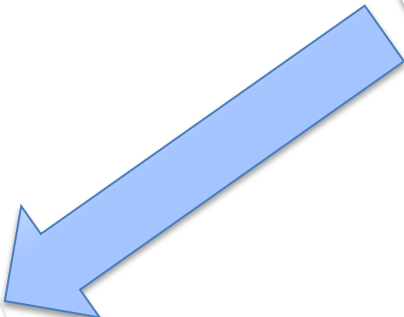
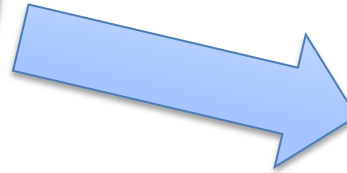
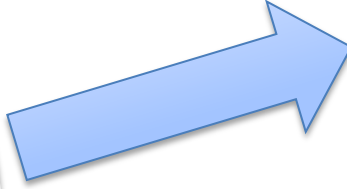
Que faire pour
savoir si le
code est
fonctionnel ?

Revue
de
code

Tests

Que faire pour
savoir si un
test est
fonctionnel ?

En l'ayant
déjà vu
échouer



Pourquoi faire des tests ?

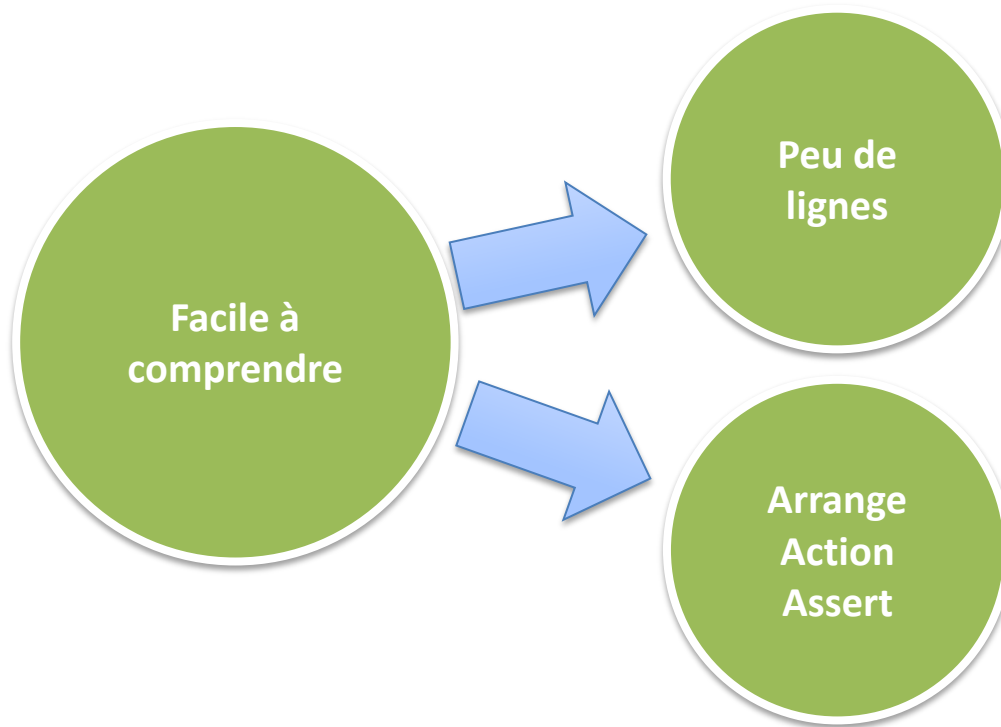
**Avoir
confiance au
code**

Ne pas avoir peur de faire des
modifications ou du réusinage.

**Permet
d'identifier
plus
rapidement
les problèmes**

**Valeur
documentaire**

Qu'est-ce qu'un bon test ?



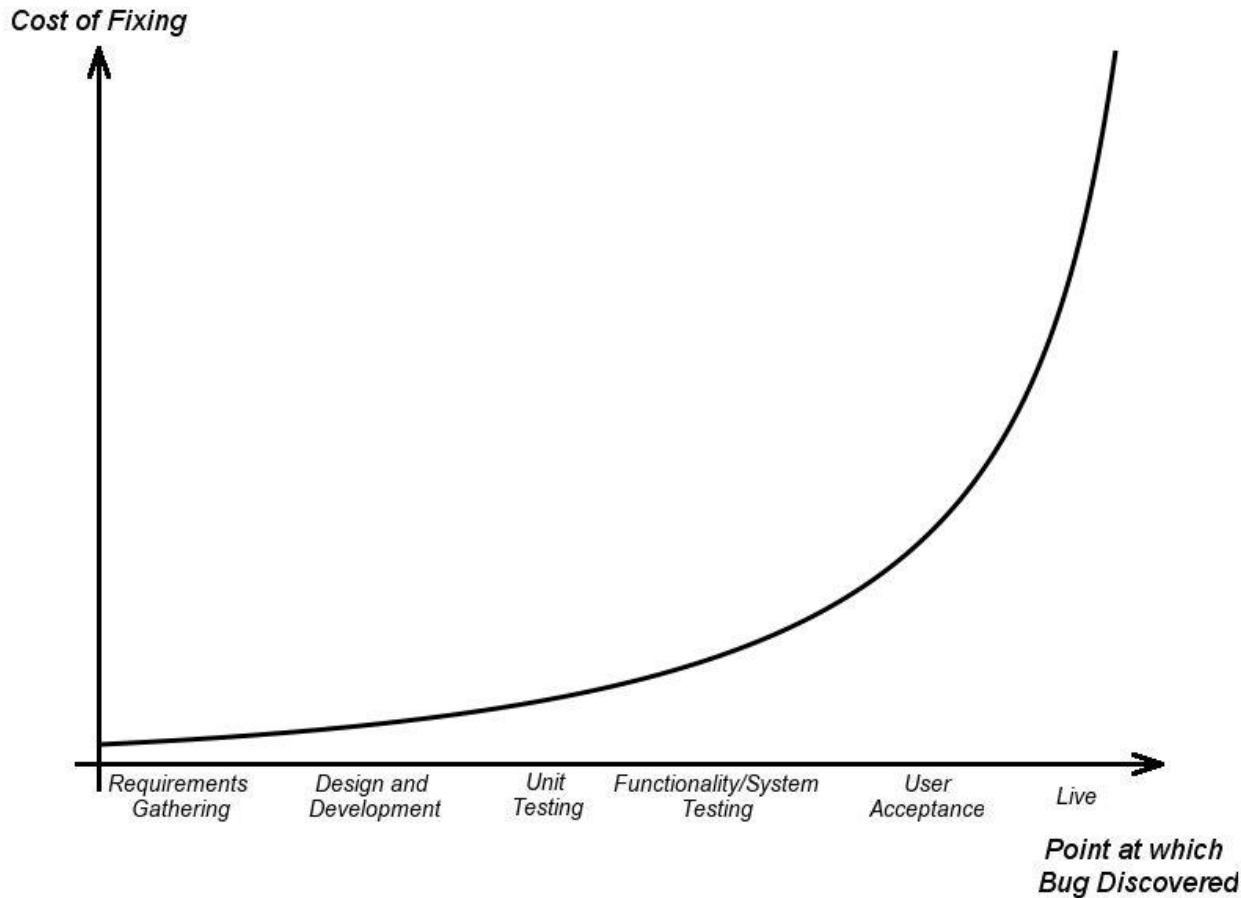
Les tests doivent être indépendants les uns des autres

Un test
=
Un
comportement
testé

Le nom du test
exprime le
comportement
désiré du test

Pourquoi automatiser les tests ?

Plus on tarde à découvrir un bug, plus le coût pour le corriger est élevé.



3 grandes catégories de tests

Unitaire

Tester les "unités" en isolation
Tester le comportement logique

Intégration

Tester les "unités" ensemble
Tester la communication entre les parties du système
Tester l'interaction avec la BD
Tester l'interaction avec des services

Bout en bout

Système

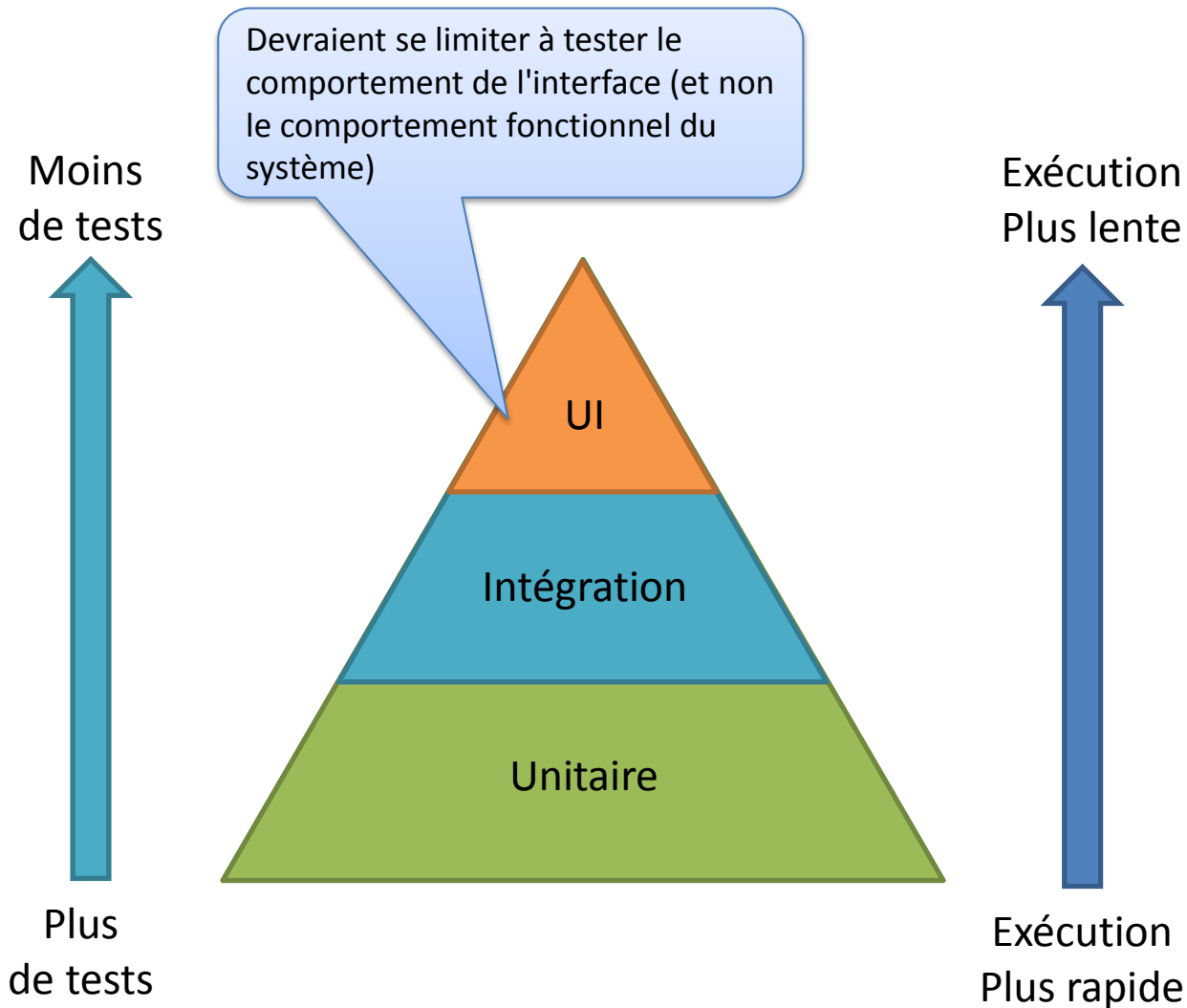
Fonctionnel

UI

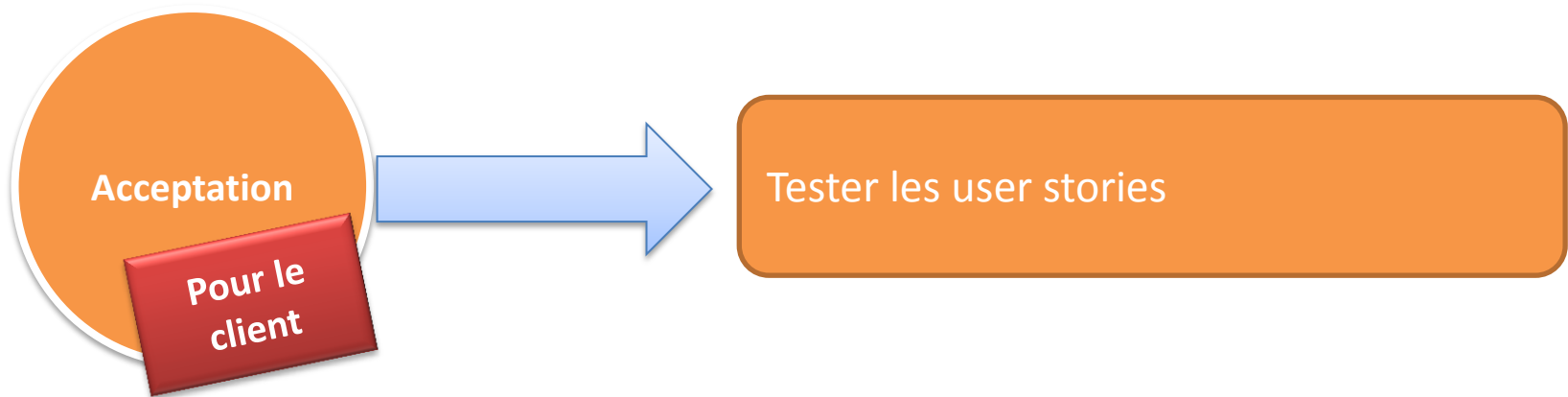
Tester l'interaction avec l'interface

Les types de tests automatisés pour l'équipe

Pour l'équipe



Les tests d'acceptation automatisés pour le client



Arrange / Action / Assert

Mise en place
pour le test

Exécution du
comportement
à tester

Validation des
résultats

```
public void home_controller_index_should_return_view_with_restaurants()
{
    // Arrange
    const int RESTAURANT_COUNT = 5;
    var restaurants = _fixture.CreateMany<Restaurant>(RESTAURANT_COUNT);
    _repo.GetAll<Restaurant>().Returns(restaurants.AsQueryable());

    // Action
    var result = _controller.Index() as ViewResult;
    var model = result.Model as IEnumerable<RestaurantListViewModel>;

    // Assert
    Assert.AreEqual(RESTAURANT_COUNT, model.Count());
}
```

Peu de lignes

Le nom du test exprime
le comportement désiré
du test

Un test
=
Un comportement testé

Le test est indépendant
des autres tests