

The background of the slide is a photograph of a building's exterior. It features a series of vertical stripes in a light beige and a muted terracotta color. Below these stripes are several windows with white frames and frosted glass panes. The overall aesthetic is modern and architectural.

# Méthodologie de tests et tests unitaires

Mastère 2 - Ynov

# Objectif du cours



Comprendre les usages des  
différents types de test



Comprendre comment mettre  
en place des tests efficaces



Écrire des tests en utilisant des  
frameworks adaptés



Apprendre à écrire du code  
testable et maintenable



Savoir mettre en place les outils  
nécessaires à l'exécution des  
tests

# Critères d'évaluation

Test (30% de la note finale)

Vérification des acquis théoriques de la formation

TP sur un nouveau cas d'utilisation (70% de la note finale)

Mise en pratique du cours en autonomie

Capacité à reproduire les différents types de test

# Plan du cours

1. Introduction aux tests logiciels
2. Tests unitaires et bonnes pratiques
3. Mise en place des tests dans un projet
4. Tests d'intégration et de composants
5. Tests non-fonctionnels
6. Tests End-To-End

# Introduction aux tests logiciels

# Qu'est-ce qu'un test ?

## ***Objectifs des tests logiciels***

- Garantir la qualité et la fiabilité des logiciels.
- Détecter les bugs et les problèmes avant qu'ils n'atteignent les utilisateurs.
- Assurer la conformité aux spécifications et aux exigences.

## ***Rôle essentiel des tests dans le développement***

- Les tests ne sont pas une étape optionnelle, mais un processus intégré.
- Éviter les perturbations majeures et coûteuses en cours de développement.
- Contribuer à une itération plus rapide et plus efficace du cycle de développement.

## ***Garantie de la qualité et de la fiabilité***

- Les tests renforcent la confiance des utilisateurs dans le logiciel.
- Réduisent les risques d'erreurs critiques et de dysfonctionnements.
- Permettent des mises à jour plus fluides et des corrections plus rapides.

# Conséquences des bugs non détectés

## ***Cas historiques de bugs catastrophiques***

- Bug de l'an 2000 (Y2K) - Perturbations dues aux années à deux chiffres.
- Fusée Ariane 5 (1996) - Surcharge d'entiers, explosion en vol.

## ***Pertes financières et de réputation***

- Exemple : Faille de sécurité Heartbleed (2014) - Vol de données, impact sur la réputation.
- Impact sur la confiance des utilisateurs et les parts de marché.

## ***Importance de la détection précoce***

- Détecter et corriger les bugs tôt réduit les coûts de correction.
- Éviter les coûts associés aux rappels, aux remplacements et aux litiges.

# Avantages des tests

## ***Fiabilité***

- Identifier les faiblesses du logiciel
- Réduire les risques de dysfonctionnements en production

## ***Maintenance***

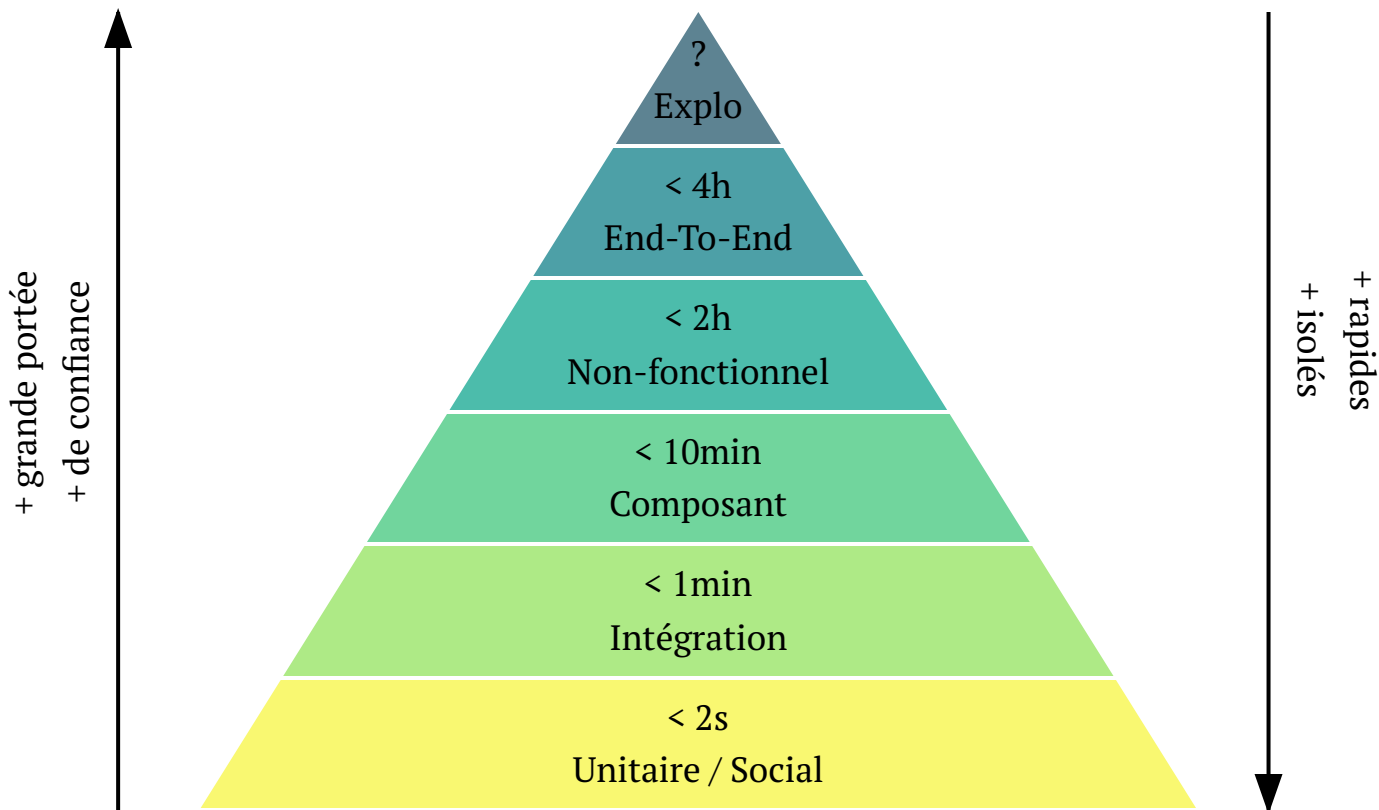
- Détection précoce = réduction des coûts
- Nouvelles évolutions plus simples à mettre en place

## ***Stabilité***

- Logiciel de bonne qualité = clients satisfaits
- Gain en image de marque



# Pyramide des tests



# Tests unitaires et bonnes pratiques

# Mise en place des tests dans un projet

# Tests d'intégration et de composants

# Tests non-fonctionnels

Tests End-To-End

Merci de votre attention