

# Test et validation des logiciels

Responsable du cours: Wahid Bannour

ISET de Mahdia

Septembre 2018

M.A.J. V2 2022

# Description du Test logiciel

- ▶ Quel est le rôle de test dans le processus de développement logiciel ?
- ▶ Comment peut-on mesurer la qualité d'un logiciel?

# Description du Test logiciel

## Panoramic du Test

- Le test est le processus d'examen d'une application pour être sûre qu'elle répond aux exigences de la conception et assure la qualité attendu
- Le test mesure la qualité d'une application ou d'un projet
- Le développeur doit être capable de voir les bugs ou les dysfonctionnements d'un projet qui n'ont pas pu être révélés encore.
- Le test permet de fixer et corriger ces erreurs.
- Un bug est une erreur de codage ou de logique qui cause un mauvais fonctionnement du programme ou produit un résultat erroné.

# Description du Test logiciel

## L'importance du test

- Réduit le coût de développement d'un programme
  - » Les estimations courante indiquent qu'un problème qui ne sera détecté et corrigé qu'après son déploiement, peut être 40-100 fois plus chère à résoudre que le détecter au cours du cycle de développement.
- Assurer que votre application se comporte exactement comme attendu.
- Réduit le coût total de l'application pour les utilisateurs finaux.
- Développement de la notoriété de l'entreprise sur le marché.

# Description du Test logiciel

## Les avantages du test pour l'utilisateur final

- Les tests réalisés au cours du cycle de développement nous garantissent un logiciel de meilleure utilisabilité et fiabilité et avec un moindre coût de revient.
  - » Les bugs détectés au cours du test ne requièrent pas que l'utilisateur les identifie au cours de l'exécution
  - » Les bugs détectés avant la livraison nous épargnent un temps considérable de correction, d'installation et de mise à jour.
  - » Les logiciels qui se comportent comme attendu, requièrent moins de temps d'apprentissage et de support technique.
  - » Les logiciels bien testés nous donnent une meilleure satisfaction Utilisateur.

# Description du Test logiciel

## Mesure de la qualité de logiciel

« Tu ne peux pas gérer ce que tu ne peux pas mesurer »  
– Lord Kelvin (1824-1907) –

- Il existe différents types de métrique (standard de mesure) pour la qualité de logiciel.

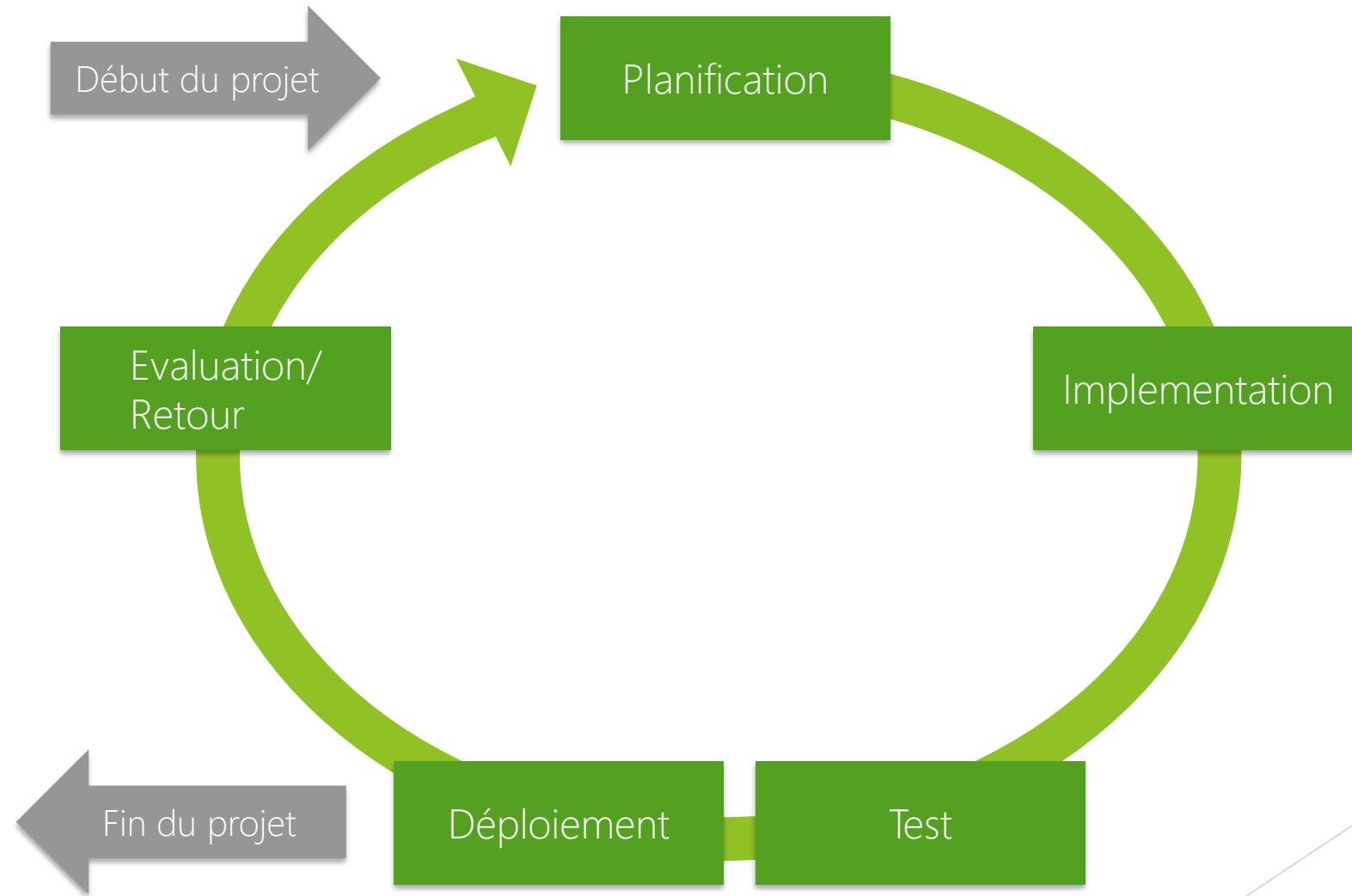
Les plus connu sont:

- Métrique de performance, comme le temps nécessaire pour accomplir un processus ou le volume d'espace requis pour une application.
- Métrique de fiabilité tel que le nombre de bug total, ou la densité de défaillance (defect density) qui est le nombre de bug divisé par le nombre de ligne de code.

# Cycle de vie logiciel

- ▶ Chute d'eau
- ▶ Cycle en V
- ▶ Cycle spirale
- ▶ Méthodologie AGILE
  - ▶ Petit cycle (Sprint)
  - ▶ Incrémental
  - ▶ Test de chaque release (version déployé)

# LE MODELE AGILE





# Les types de test

- ▶ Test manuel
- ▶ Test automatisé
- ▶ Test boîte noir et test boîte blanche

# Les types de test

## Test Manuel

- ▶ La première expérience de test on l'a fait avec le premier programme qu'on a écrit! Exécute le programme et regarde s'il fonctionne correctement!
- ▶ Dans ce type de test, le testeur joue le rôle de l'Utilisateur et s'assure qu'il n'existe aucun comportement indésirable
- ▶ Généralement, les testeurs manuel, utilise un plan de test avec des cas spécifiques pour assurer une couverture maximale des scénarii dans le projet.
- ▶ Le testeur peut faire partie, ou non, de l'équipe de développement.

# Les types de test

## Test Automatisé

- ▶ Les tests automatisés utilisent un software de test (comme Microsoft® Test Manager dans Microsoft Visual Studio®) pour contrôler et traquer les tests exécutés automatiquement.
- ▶ Les tests automatisés peuvent être créés et configurés pour s'exécuter à chaque création d'une nouvelle version du projet (ou un bout de code).
- ▶ L'exécuteur de test (test runner) fournit un rapport détaillé du résultat de test pour chaque test automatisé.

# Les types de test

## Test manuel vs. test automatisé

- ▶ Les deux types sont complémentaires et sont importants pour assurer un logiciel de haute qualité
- ▶ Les tests automatisés sont rapides et peuvent tester plusieurs variantes de données. Ils peuvent facilement répéter la liste exhaustive des tests au fur et à mesure que le logiciel évolue. Et puisque, ils sont exécutés par un ordinateur, ils deviennent insensible aux répétitions.
- ▶ Malgré que le test manuel généralement prend plus de temps pour s'exécuter (puisque'il requiert un humain), il demande moins de temps de configuration. Il serait un bon choix pour les tests qui doivent s'exécuter occasionnellement, ou dans le cas où la configuration demande un coût exorbitant.

# Les types de test

## Test boîte noire

- ▶ Le test boîte noire est un test conduit sans connaissance du fonctionnement interne du système à tester.
- ▶ Ce type de test simule l'utilisation de l'utilisateur final
- ▶ En général, le testeur ne sait pas comment le code fonctionne. Il fournit les entrées et examine les sorties.
- ▶ Le testeur n'est pas censé savoir programmer (en cas de test de l'IHM)
- ▶ Exemple de scénarios
  - » Tester les interfaces utilisateurs (vues) pour voir s'ils répondent à toutes les exigences et sont fonctionnelles
  - » Test de différents types d'entrées
  - » Test le système sous grande charge et en stress
  - » Tester la sécurité d'un système.
  - » Test des fonctionnalités du programme par un département spécialisé en test

# Les types de test

## Test boîte blanche

- ▶ Aussi connu sous le nom de boîte de verre, boîte claire et test boîte ouverte.
- ▶ Le test boîte blanche est conduit par l'examen du code de programmation selon des scénarii d'échec potentiel.
- ▶ Les cas de test boîte blanche sont créés par une personne qui analyse le code de l'application et les prépare dans le but de s'assurer que le code se comporte en accordance avec la spécification.
- ▶ Les scénarii de test boîte blanche inclus:
  - » Test des méthodes et sous routines utilisées en interne.
  - » Test des boucles et des structures conditionnelles pour s'assurer de la justesse
  - » Test de performance pour le chemin de code ou l'algorithme.

# Hiérarchie des tests

