

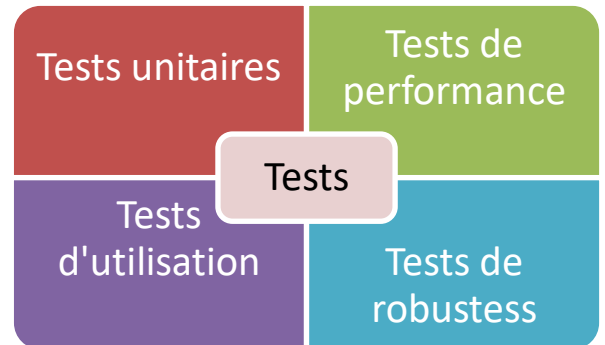
Génie Logiciel et Projet – TD 5 – Tests du logiciel

1. Les tests à faire pour votre projet

Pour votre projet, vous réfléchissez à une question primordiale : Comment prouver le bon fonctionnement de votre logiciel pour :

- ✓ Les utilisateurs finaux
- ✓ Les développeurs d'un point de vue de la qualité du système développé (extensibilité, modularité...)

Ensuite, vous intégrez une partie « tests » dans votre travail du projet. Les tests doivent être catégorisés. Le schéma à droite illustre différents types de test possible pour votre projet.



Note : Tous les tests ne peuvent pas être automatisés.

Les tests devront être réalisés au fur et à mesure de l'avancement du projet.

3. Révision des programmes vus en CM

Les programmes expliqués par l'enseignant en CM sont très importants pour comprendre les théories et les techniques du génie logiciel. Ils vous sont aussi très utiles pour réaliser votre projet en termes de mise en œuvre.

Pour en profiter au maximum et pour aussi réviser le contrôle écrit individuel, vous pouvez :

- 1) Réviser ces programmes en équipe afin d'avoir une compréhension plus complète sur leur fonctionnement.
- 2) Revoir à nouveau ce qui pourrait vous être utile dans ces programmes, pour la réalisation de votre projet.

Rappel des 6 programmes (support de cours) vus en CM :

- 1) Programme « chronometer » : Conception de logiciel générale. De l'algorithme (mathématique), via la conception à la programmation. Java Graphics, Java Threading. Réalisation des animations logicielles.
- 2) Programme « Tree » version 1 : Conception des classes de données, design patterns Singleton, Simple Factory, Builder et Template Method.
- 3) Programme « Aircraft » : Architecture logicielle, une séparation pure et interaction entre classes IHM graphique, classes de traitement et classes de données.
- 4) Programme « Tree » version 2 : Conception des classes de traitement, séparation entre l'IHM graphique et le noyau (moteur). Conception de l'IHM graphique avec la gestion de l'échelle (algorithmique). Design patterns Visitor et Strategy.
- 5) Programme « Tree » version 3 : Testabilité du logiciel et tests automatisés avec le framework Junit. Instrumentation du logiciel avec le framework JFreechart. Système de journal avec le framework Log4j.
- 6) Programme « Train » : Réalisation du logiciel en multi-threading avec la concurrence d'accès. Conception et réalisation de l'IHM graphique avancée.