

Les tests logiciel pour aller vers plus de pragmatisme (T6)

Cette thématique se propose de présenter les tests logiciels et ce en quoi ils constituent concrètement un très précieux outil pour faciliter et améliorer le travail de développement.

Les tests logiciels peuvent être utilisés à différentes étapes de le cycle de vie d'un logiciel. Ils permettent aux acteurs, ingénieurs comme chercheurs, de construire les besoins, façonner l'architecture, formaliser le comportement, favoriser la maintenabilité.

Pourtant, dans notre communauté, ils sont parfois perçus comme une charge supplémentaire, voire un luxe pour les équipes de développement. Nous voulons vous convaincre du contraire. Incontournables dans l'industrie du logiciel, ils sont nécessaires à une production logicielle de qualité, et sont de puissants alliés pour le développeur au quotidien. Accessibles, ils libèrent sa créativité, limitent les effets de bords et permettent de travailler dans la sérénité.

Dans une optique pragmatique, nous vous présenterons un état de l'art en regard du cycle de développement, notamment dans l'approche Agile. Vous serez sensibilisé aux approches usuelles, à la notion de dette technique, à la pratique d'outils d'intégration continue, de traçabilité et à la couverture de tests des exigences fonctionnelles. .

Un cas d'école sur la validation d'un logiciel scientifique sera présenté.

Présentations

Jeudi 14h00-17h30, Amphi Gay-Lussac

Les présentations seront assurées par

- (leader) Fabrice Bouquet - PU - INRIA Projet CASSIS - [fabrice.bouquet\(at\)inria.fr](mailto:fabrice.bouquet@inria.fr)
- Fabien Peureux - MCF - FEMTO-ST (UMR CNRS 6174) - [fabien.peureux\(at\)femto-st.fr](mailto:fabien.peureux@femto-st.fr)
- Bruno Legeard - PU - FEMTO-ST (UMR CNRS 6174) - [bruno.legeard\(at\)femto-st.fr](mailto:bruno.legeard@femto-st.fr)

T6.P1 Etat de l'art du test en regard avec le cycle de développement logiciel et les méthodes agiles.

Intervenant : Fabrice Bouquet, INRIA, Besançon

Les différents types de tests, leurs caractéristiques et leurs usages dans le cycle de développement. Coût de la non-qualité et de la dette technique.

T6.P2 Améliorer la qualité et la maintenance de mon code par les tests de non-régression et l'intégration continue.

Etat de l'art. Illustration de l'automatisation des tests avec la plateforme jenkins.

T6.P3 Passage des exigences fonctionnelles aux tests : traçabilité, couverture de tests des exigences

Intervenant : Bruno Legeard (Université de Franche-Comté/FEMTO-ST & Smartesting).

T6.P3 Cas d'école sur la validation d'un logiciel scientifique (spécificités de la recherche, spike)

Tests fonctionnels. Conditions limites des algo, choix des données de tests (couverture de tests optimale).

Cheminement dans les algorithmes, tests unitaires, robustesse et condition d'usage des algorithmes.

Illustration par le développement d'un logiciel fictif dans un laboratoire.

Ateliers

T6.A1 Définition et écriture des tests de couverture des exigences logiciels.

Intervenant : Fabien Peureux (FEMTO-ST, Besançon)

Mise en oeuvre de l'application Squash TM (application Full Web).

Utilisation d'une instance de Squash TM hébergée dans les locaux de la société HENIX disposant d'une Base de Données dédiée.

T6.A2 Industrialisation des tests - Intégration continue et automatisation des tests.

Intervenants : Cédric Joffroy (FEMTO-ST, Besançon), Fabrice Ambert (FEMTO-ST, Besançon)

Découverte de l'environnement Jenkins + Nexus.

Utilisation de Maven (gestion des dépendances).

Mise en place de l'intégration continue (Build continu).

Publication au sein de Nexus des packages créés.

T6.A3/T4.A5 Qualité du logiciel - règle de codage – SONAR Voir T4-A5

Intervenant : Thomas Lallart (INRA Avignon)

T6.A4 Bien écrire les tests de son composants logiciel PHP.

Intervenants: Ivan Enderlin (Inria) , Jean-Marie Gauthier (FEMTO-ST)

- Partie 1 (1 heure 30) : présentation des tests unitaires manuels et de PHP, découverte de l'outil et exercices. A partir d'un petit cours et de « code à trous », nous écrivons une demi-douzaine de tests concrets de qualité.
- Partie 2 (1 heure 30) : poursuite de l'écriture de tests puis introduction de tests unitaires automatiques et exercices. Pour ce dernier point, nous allons reprendre le code précédent et le spécifier, afin de générer automatiquement des tests et les comparer avec les tests manuels.

T6.A4 Bien écrire les tests de son composants logiciel Java.

Intervenants : Fabrice Ambert (FEMTO-ST), Alexandre Vernotte (FEMTO-ST)

- Partie 1 : Sur un code java simple fourni, écriture guidée des tests permettant la découverte des différents éléments du test unitaire avec Junit.
- Partie 2 : Introduction des concepts de mock et stubbing au travers de l'utilisation de mockito. Impact du test sur le design du code.

Groupes de travail

T6.GT1 Cas d'étude et retour d'expérience sur la validation des logiciels scientifiques

Animateur(s) : Marc Dexet (CNRS, DSI)

T6.GT2 Test pour expliciter les besoins logiciels

Animateur(s) : Olivier Inizan (INRA Versailles)

T6.GT3 Mettez du test dans votre existant

Animateur(s) : Marc Dexet (CNRS, DSI)

T6.GT4 Spécifications et émergence de la conception par les tests.

Animateur(s) : Fabien Peureux (FEMTO-ST, Besançon), Thomas Lallart (INRA Avignon, DSI)