



DEPARTEMENT INFORMATIQUE
DE LA FACULTE DES SCIENCES

Ahmed Kaci, Guilhèm Blanchard et Yanis Allouch

TER : Feuille de route

HMIN201 — Travail d'Etude et de Recherche

Référent: Nicolas Hlad

2021

1 Le TER

1.1 Le Contexte

Mr Hlad, représentant l'équipe [MAREL](#) du [LIRMM](#) (UMR 5506 (CNRS)), développe l'outil **IsiSPL** qui génère un *FeatureModel* à partir d'un corpus de code annoté incrémental. Avec sa collaboration, nous allons développer un outil qui évaluera la qualité des annotations générées par **IsiSPL** pour la comparer avec une annotation développée à la main. Nous avons introduit la méthode **VITAL** publiée en 2014.

1.2 Le nécessaire

- Nom du groupe : TçaPöt
- Noms, prénoms, mails des membres :
 1. Allouch, Yanis, yanis.allouch@etu.umontpellier.fr
 2. Blanchard, Guilhèm, guilhem.blanchard@etu.umontpellier.fr
 3. Kaci, Ahmed, ahmed.kaci@etu.umontpellier.fr
- Listes des documents à lire :
 1. [\[Mei+17\]](#)
 2. [\[Zha15\]](#)
 3. [\[ZB14\]](#)
 4. [\[Ale+12\]](#)
 5. [\[ZB12\]](#)
 6. [\[KA08\]](#)
 7. [\[WB09\]](#)
- **Gantt**, voir la pièce jointe [Planning_TER] à ouvrir avec **ProjectLibre**.

1.3 Les objectifs

- Phase **Analyse et conception** du projet.
 - Identification des bonnes pratiques utilisées pour la réalisation de code annoté.
 - Élaboration de métriques dont le but est d'identifier des annotations de qualité et qui nous aideront pour classer des projets annotés.
- Phase **Développement** du projet.
 - Implémentation des critères d'évaluation et de comparaison de l'approche **VITAL** afin de lui ajouter de nouvelles métriques, plus spécifique à l'équipe **MAREL**.
 - Implémentation d'un logiciel basé sur la réalisation précédente en Java ou en Python (à étudier). Dans le but d'avoir un outil portable et exécutable sur toutes plates-formes.
 - Nous testerons notre outil sur des lignes de produits réalisés à la main et générés par **isySPL**. Nous réaliserons ces tests progressivement, en premier lieu sur la première métrique implémentée, puis à chaque métrique ajoutée pour mesurer notamment leur complexité temporelle.
- Étudier d'éventuelles utilisations des designs patterns dont la mise en œuvre serait intéressante lors de la création d'un code avec des annotations dans le cadre de la construction d'une ligne de produits logiciels. (Perspective)

Références

- [Ale+12] ALEIXO, F. A. et al. « A Comparative Study of Compositional and Annotative Modelling Approaches for Software Process Lines ». In : *2012 26th Brazilian Symposium on Software Engineering*. IEEE, sept. 2012.
- [KA08] KÄSTNER, C. et APEL, S. « Integrating Compositional and Annotative Approaches for Product Line Engineering ». In : (2008), p. 91-98.
- [Mei+17] MEINICKE, J. et al. *Mastering Software Variability with FeatureIDE*. Springer International Publishing, 2017.
- [WB09] WILSON, J. et BALL, T. *Preprocessing .java with Munge*. 2009. URL : <https://publicobject.com/2009/02/preprocessing-java-with-munge.html>.
- [ZB12] ZHANG, B. et BECKER, M. « Code-based variability model extraction for software product line improvement ». In : *Proceedings of the 16th International Software Product Line Conference on - SPLC '12 -volume 1*. ACM Press, 2012.
- [ZB14] ZHANG, B. et BECKER, M. « Variability code analysis using the VITAL tool ». In : *Proceedings of the 6th International Workshop on Feature-Oriented Software Development - FOSD '14*. ACM Press, 2014.
- [Zha15] ZHANG, B. *VITAL - Reengineering Variability Specifications and Realizations in Software Product Lines*. T. 53. Fraunhofer IESE, Kaiserslautern : Fraunhofer Verlag, 2015.