



POLYTECH
NICE SOPHIA



UNIVERSITÉ
CÔTE D'AZUR

RAPPORT INTERMEDIAIRE DE STAGE

EPU 4

Groupe de projet :

Florian AINADOU, Paul-Marie DJEKINNOU, Djotiham NABAGOU

Période de stage : **22 Juin 2020** au **31 Juillet 2020**

Année universitaire : 2019-2020

Entreprise : Polytech Nice Sophia

Encadrants :

M. Philippe Collet, M. Johann Mortara, Mme Anne-Marie Dery Pinna

Titre : **Symfinder API**

La variabilité d'un système informatique se définit par la complexité de son implémentation et les différents éléments (patrons de conception, mécanismes d'héritage et de surcharges de méthodes...) mis en œuvre. Pour déterminer cette variabilité dans un code source orienté objet, il faut être en mesure d'analyser le-dit code source et d'en ressortir un résultat. [Symfinder](#) est une base de code écrite en Java permettant l'analyse de projets (open source jusqu'ici, tel que JUnit4) et permettant de ressortir une analyse modélisée sous forme de graphe et donnant des précisions sur divers patrons de conception utilisés, des relations entre les différentes classes, interfaces, méthodes et bien d'autres éléments d'un code source. Toutefois, dans une release de code source mise à la disposition des utilisateurs, toutes les méthodes, constructeurs et voire même classes et interfaces, ne sont pas toutes utilisées. Elles peuvent l'être implicitement à travers des appels internes, mais ne sont pas directement exposées à l'utilisateur final. On peut donc considérer à partir de là, que les méthodes "*public*" d'une classe Java, constituent l'API de cette dernière, car elles sont exposées. Le but de *Symfinder API* (dans un premier temps) sera donc de détecter l'API d'un code source et vient donc s'inscrire dans le processus de détermination de la variabilité, qui elle-même, consiste en fait en de la rétro-ingénierie.

Durant cette première semaine de stage, nous avons commencé à appréhender le sujet et prendre en main la chaîne d'outils *Symfinder* déjà existante et sur laquelle nous nous baserons pour développer *Symfinder API*. Un "Sprint 0" a été défini avec pour but la détection des différents packages de JUnit 4 et leur visualisation de façon interactive. Ce sprint va s'étendre sur la semaine prochaine en raison des divers soucis techniques rencontrés lors de la prise en main de la chaîne d'outils.

Les prochains sprints auront pour objectif d'analyser toute une base de code source et en ressortir l'API. Une piste intéressante sera également d'explorer le code source de JUnit 5 qui a changé en termes de modularité par rapport à JUnit 4.