Guillaume André - Alexandre Hiltcher - David Lang - Jean-Adam Puskaric

Sujet GIT - Make merging great again

Question générale

Existe-t-il une corrélation entre le nombre de Git merge et la qualité logicielle dans un projet Git et peut on la mesurer ?

Nous avons choisi ce sujet car utiliser Git pour gérer nos projets semble aujourd'hui être une évidence, et il nous paraît intéressant d'analyser et quantifier l'impact de cet outil sur le code produit et sa qualité. Plus particulièrement nous nous intéresserons aux *merges* et *merge conflicts* (ce qui représente pour nous les *merges* qui entraînent l'apparition de marqueurs de conflits et demande donc une intervention humaine), qui sont pour nous la fonctionnalité basique de Git pouvant être la plus disruptive pour le code (et pour les relations humaines parfois!)

Sous-questions

A quels endroits ont tendance à apparaître les *merge conflicts* ? Le code de ces zones est-il de bonne qualité logicielle ?
Hypothèse :
Les zones avec plus de *merges* existent et on est capable de les identifier, ce qui présuppose qu'on peut identifier les *merges* ainsi que les *merge conflicts* dans la *timeline* d'un projet Git.
Mise en place :
Nous pouvons récupérer sur github un grand ensemble de dépôts libres d'accès grâce à l'api publique disponible. Ensuite nous pouvons regarder si ceux-ci comportent un nombre minimum de lignes du langages voulu, ces données étant contenues dans les résultats de l'api Github. Une fois le dataset de dépôts établi, nous pourrons parcourir l'historique Git de chaque dépôt afin de récupérer l'ensemble des informations des commits de merges (les commits de merges étant les commits ayant 2 parents). Ensuite, nous pouvons rejouer tous les merges avec les informations précédemment extraites. Cela nous permettra d'extraire les merges conflicts et leurs informations, comme les fichiers concernés. Finalement, nous pourrons analyser avec des outils comme sonarqube la qualité de code des fichiers concernés par les conflits pour chaque tuple *base commit / left commit / right commit / result commit*.

Peut-on trouver une corrélation entre le nombre/emplacement de merge et la qualité logicielle?

Hypothèse :

On est capable de trouver des métriques isolables qui, en fonction du langage et de la taille du projet, identifient une corrélation entre la quantité de merge et la qualité logicielle.

Mise en place :

Avec plusieurs langages on peut **comparer le taux de git merge/merge conflict dans les langages** pour identifier quel langage va être intéressant (on dispose déjà d'une base de projets Git en Java)

Trouver une manière de mesurer la qualité va probablement dépendre du langage (exemple : SonarQube pour Java)

S'il existe une corrélation, peut-on isoler des métriques ou sous-métriques en résultant et les appliquer à d'autres langages?

Hypothèses :

- Nous avons trouvé une corrélation pour un langage
- Le travail que nous avons produit pour un langage est applicable à d'autres
- Nous pouvons utiliser le même outils d'analyse du code ou des outils similaires sur plusieurs langages afin d'obtenir de métriques comparables.

Mise en place :

Le travail fait pour un langage devra être pensé pour pouvoir être applicable à d'autres afin de pouvoir comparer les résultats.

Démarche

Dans un premier temps il va falloir obtenir des bases de projet git pour quelques langages qui nous semblent pertinent ainsi que d'un outil pour extraire automatiquement un modèle git interrogeable contenant les merges, leur type, et le code associé.

Une fois qu'on aura déterminé le langage le plus pertinent à analyser on pourra, à l'aide d'outil d'analyse de la qualité du code et/ou en se basant sur l'historique résultant du *merge* (exemple : si le merge a entraîné un bug), tenter de déterminer des liens de causalité entre les *merges* et la qualité grâce à des métriques choisis.

Dans une premier temps nous allons donc travailler sur un seul langage puis, si le temps nous le permet, nous allons pouvoir analyser d'autres langages et comparer les résultats obtenus dans la mesure du possible (on ne va pas comparer la qualité générale du code puisque cela dépend des outils et de nombreux facteurs mais plutôt si nos analyses tendent à être généralisables sur les différents langages et à quel point).

Articles scientifiques en lien

Ces deux articles présentent des outils que nous pourrions utiliser afin de récupérer des métriques concernant la qualité logicielle.

Code Scene:
https://empear.com/docs/CodeSceneBook.pdf

Titan:
https://www.cs.drexel.edu/~lx52/LuXiao/papers/FSE-TD-14.pdf