**NJOYA YOUCHAOU**

Je présente ici mon apport dans un travail collectif qui consistait à analyser la qualité du code source d’un projet au choix à l’aide des outils d’analyse métrique du logiciel au choix.

Le projet est un petit site web fait à base du PHP, du HTML et du CSS.

J’ai utilisé principalement deux outils : **CLOC** et **Sonar**

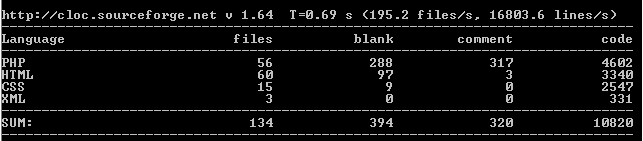
1. **Analyse de code avec l’outil CLOC**
   1. **Présentation de l’outil**

CLOC est un simple outil en ligne de commande qui compte les fichiers d’un projet par langage et pour chaque fichier, il compte les nombres de ligne de code, de ligne de commentaires, et de lignes vides.

* 1. **Résultat d’analyse**

Le résultat d’analyse de notre projet (Figure 1) nous montre que :

* Les langages utilisés sont PHP, HTML, CSS et XML ;
* Le projet compte 134 fichiers au total ;
* Le nombre de ligne de code source (SLOC) est de **10 000**



***Figure 1****: Résultat d’analyse du projet avec CLOC*

1. **Analyse de code avec l’outil Sonar**
   1. **Présentation de l’outil**

[**Sonar**](http://sonar.codehaus.org/) est un outil open source initialement développé par la société suisse [**Hortis**](http://www.hortis.ch/). Depuis novembre 2008, c'est la société suisse [**SonarSource**](http://www.sonarsource.com/) qui se charge du développement et du support de Sonar. Le but principal de cet outil est de fournir une analyse complète de la qualité d'une application en fournissant de nombreuses statistiques (ou **métriques**) sur ses projets. Ces données permettent ainsi d'évaluer la qualité du code, et d'en connaître l'évolution au cours du développement. Nous listons ici les principales fonctionnalités de l'outil Sonar.

* Tableau de bord complet des différents projets suivis.
* Détection rapide du code à risque.
* Mesures quantitatives : nombre de classes, duplication de code, etc.
* Mesures qualitatives : couverture et taux de réussite des tests, complexité du code, respect des règles de codage...
* Historiques des statistiques, pour en voir l'évolution au cours du temps.
* Support de plus de 600 règles de qualité.
* Gestion de profils pour les règles de codage.
* Visualisation du code source, surlignant les violations des règles de codage qui s'y trouvent.
* Fonction "Time machine" permettant de comparer plusieurs versions d'une même application.
* Identification des points faibles d'un projet.
* Support des plugins.

Par ailleurs, Sonar est multi-langage ; et disponible sous plusieurs formes :

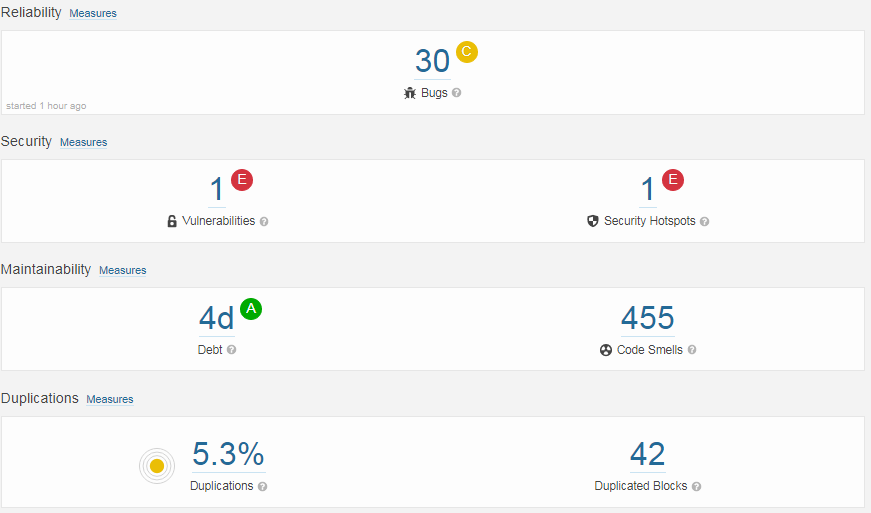
* Sous forme de plugin (**sonarLint**)
* Sous forme d’application indépendante (**SonarQube**)
* Sous forme d’application web (**sonarCloud**)

En outre, il peut être utilisé comme un outil d’intégration continu puisqu’il peut être relié à un dépôt comme **github**, **bitbuket** et bien d’autres. Et à chaque commit, il analyse le code afin de nous permettre de conserver la qualité de l’application durant son développement.

* 1. **Résultat de l’analyse du projet avec SonarCloud**

Nous avons opté pour notre analyse d’utiliser la version online de l’outil sonar (**sonarQube**) pour ne pas nous encombrer d’une installation locale. Nous l’avons directement relié avec notre dépôt github où le projet à analyser a été logé. Ainsi, une nouvelle analyse est lancée automatique à chaque commit.

L’analyse a produit un résultat dont la synthèse est représentée sur la figure suivante :

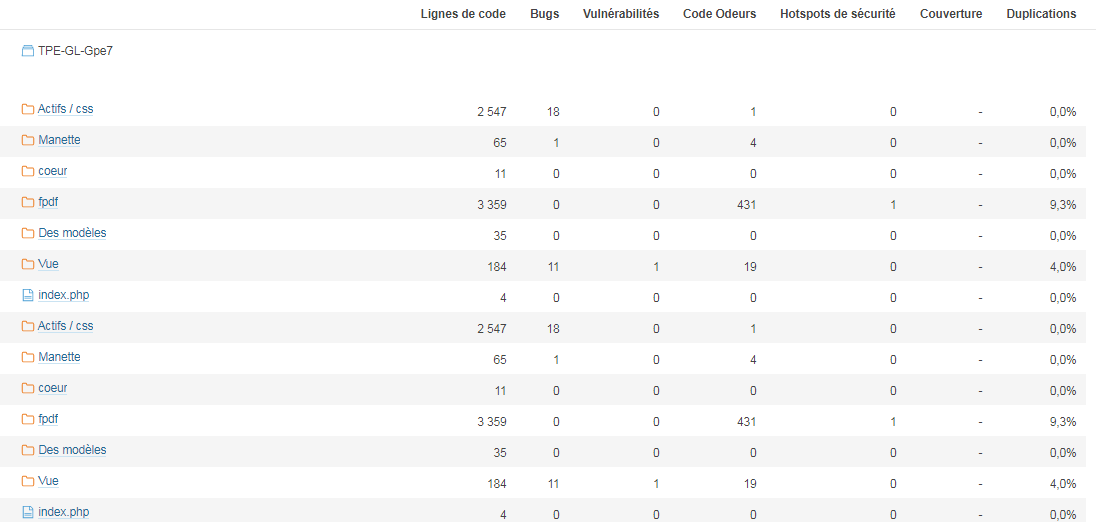


***Figure 2****: Résultat d’analyse du projet avec SonarCloud*

Il en ressort que

* Le projet contient **30 bugs**. La cote de fiabilité est **C** c’est-à-dire « *qu’il y a au moins un bug majeur ».*
* En termes de sécurité, sonar juge que l’application contient une vulnérabilité de blocage (la mention **E**).
* La mesure de la maintenabilité montre 4 jours de dette technique nécessaires pour réparer les **455 codes odeurs** (codes déroutant et difficiles à maintenir). La mention **A** signifie que « *le taux d’endettement technique est inférieur à 5% »*
* Enfin, **5,3%** de codessont **identique**, soit **42 bocks dupliqués**.

La capture suivante montre les statistiques (**lignes de code, bugs, vulnérabilités, codes odeurs, taux de duplication**) par dossier du projet.



***Figure 3****: Résultat d’analyse du projet par dossier avec SonarCloud*