École De Technologie Supérieure

MGL805-01, Vérification et assurance qualité de logiciels

Automne 2019

Analyse de qualité de code source avec SonarQube

Équipe N 1

Par

Mohamed Yassine Katiri

Ilyass CHAHID

Yassine RABBOUH

Remis à

Alain April

Historique des changements

Version	Date	Réviseur	Description du changement
V 0.1	2019-09-22		Version initiale
V 1.0	2019-10-25		Version finale

Préface

Ce rapport détaille les étapes d’installation et configuration de SonaQube et aussi comment effectuer l’analyse du code pour les langages de programmation PHP, Java, .Net (C#), JavaScript et Angular, WinDev/WebDev, RPG et CL.

TABLE DES MATIÈRES

1. Vue générale

1.1 Problématique

Ce travail s'inscrit dans une démarche d'amélioration des pratiques et de la performance TI chez SOGEBANK.

Le diagnostic qui a été effectué couvre l'ensemble des pratiques de gestion et les processus rattachés au cycle de vie des solutions TI, allant de la définition des besoins jusqu'à la maintenance post-livraison.

Une analyse de la performance suite à l'élaboration d'un plan de mesure ainsi que des entrevues avec les différentes parties prenantes ont permis de constater les faits suivants :

- Perte de temps lors de la mise en production entraînant des retards au niveau des échéanciers, ainsi que des coûts non prévus (Overtime pour stabilisation).
- Perte de temps pour mettre en place les correctifs liés aux régressions aux niveau des fonctionnalités livrées.
- Certains cas limites s'avèrent non fonctionnels suite à la mise en production.
- Après le déploiement de la solution en production, le coût de la maintenance s'avère très élevé en comparaison avec la valeur d'affaire livrée ainsi qu'avec les concurrents œuvrant dans le domaine bancaire, malgré les tests effectués.
- La relation d'affaire entre le département TI, et les autres départements clients est devenue conflictuelle.
- Le climat de travail à l'intérieur du département TI se détériore à cause de la pression engendrée, ce qui impacte les taux d'assurance salaire.
- Difficulté d'identifier les codes puants et les antipatrons.
- L'ensemble de ces points font que la qualité des livrables TI soit ponctuellement remise en question, particulièrement celle du code source. L'équipe TI n'a pas de métriques pour mesurer la performance et la qualité et ainsi prendre des actions pour les améliorer.

1.2 Solution

Afin de traiter ces causes, une recommandation de mise en place de l'outil d'analyse de code SonarQube est formulée. Ceci permettra de:

- Avoir des métriques mesurables qui vont nous permettre de diagnostiquer la qualité du code.
- Être capable de se fixer des objectifs mesurables et réalisables.
- Prendre les mesures nécessaires pour atteindre une meilleure qualité.
- Avoir une meilleure traçabilité de l'évolution du projet en termes de qualité
- Améliorer et optimiser les essais.
- Donner à l'équipe du projet un meilleur focus sur les défauts à corriger. et les mauvais pratiques appliqués dans le projet.

2. Intégration et configuration de SonarQube

2.1 Qu'est-ce que SonarQube ?

SonarQube est un outil Open source qui fait l'analyse dynamique d'un projet afin d'inspecter la qualité de code d'un projet.

L'inspection de SonarQube permet de faire des review automatique grâce a elle qu'il peut détecter des bugs, vulnérabilité et plusieurs code puants.

L'outil est déjà compatible avec plus de 20 langages de programmation est-il est facilement intégrable avec les projets logiciels.

2.2 Les bénéfices d'utilisation de SonarQube ?

Les avantage de l'utilisation de SonarQube c'est qu'il donne la possibilité de générer plusieurs rapports de la qualité de logiciel et celle de niveau de concept de qualités dans le projet (Fiabilité, sécurité, maintenabilité).

2.2.1 Intégration

2.2.2 Intégration de SonarQube sur Eclipse

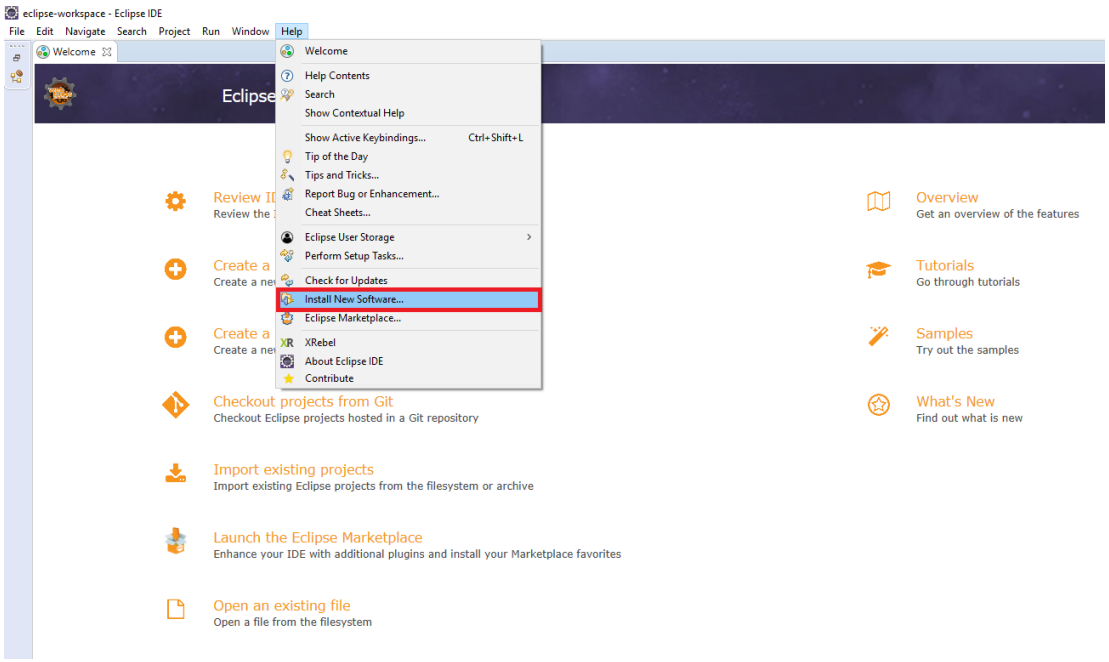
Eclipse est un IDE qui permet produire et fournir des outils qui facilite la réalisation de logiciels de différents langages, mais principalement, il est utilisé pour le langage JAVA.

L'avantage d'intégrer SonarQube dans Eclipse est la possibilité de voir directement la qualité de son code sans besoin d'ouvrir un navigateur Web à chaque fois. [Ref-1].

2.2.2.1 Installation de plugin SonarQube dans Eclipse

Voice les étapes à suivre pour intègre le plugin de SonarQube dans Eclipse :

- Ouvrir Eclipse IDE et cliquer sur l'onglet Help, Install New Software.



- Entrer le lien suivant: **http://downloads.sonarsource.com/eclipse/eclipse/** et cliquer sur la touche Entrée, la liste des plugins SonarQube seront affichés.

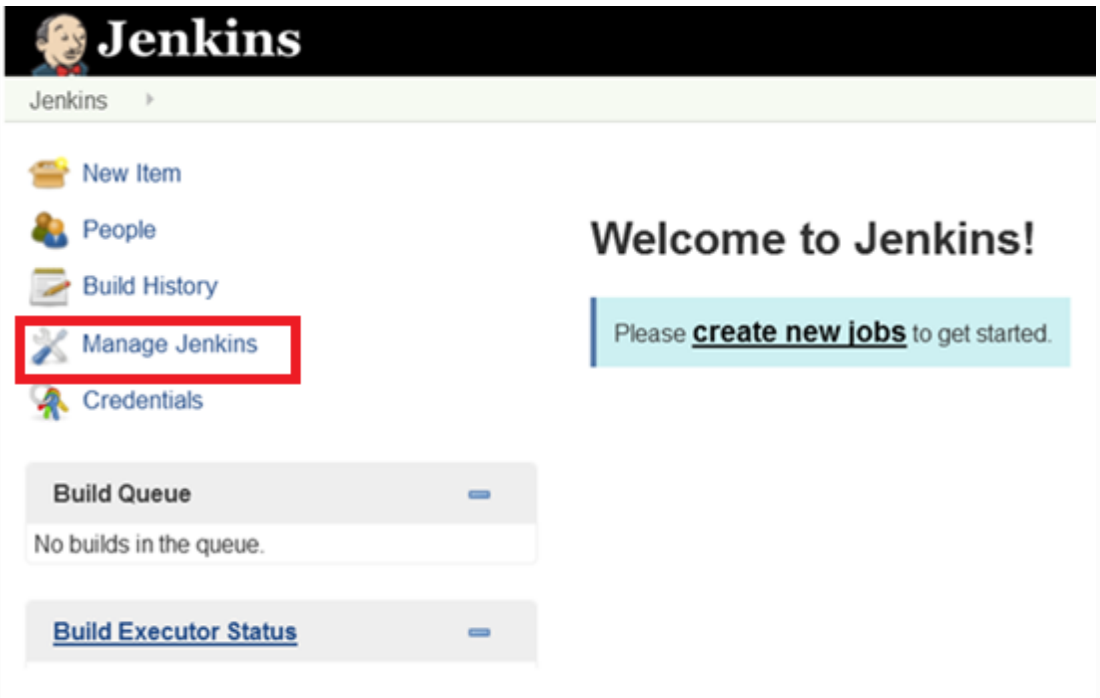
- Cocher ‘ **SonarQube Eclipse Integration** ‘ et taper le bouton Next
- Cliquer sur **Finish**, redémarrer Eclipse après avoir terminé l’installation.

2.2.3 Intégration de SonarQube avec Jenkins

2.2.3.1 Installation de Jenkins

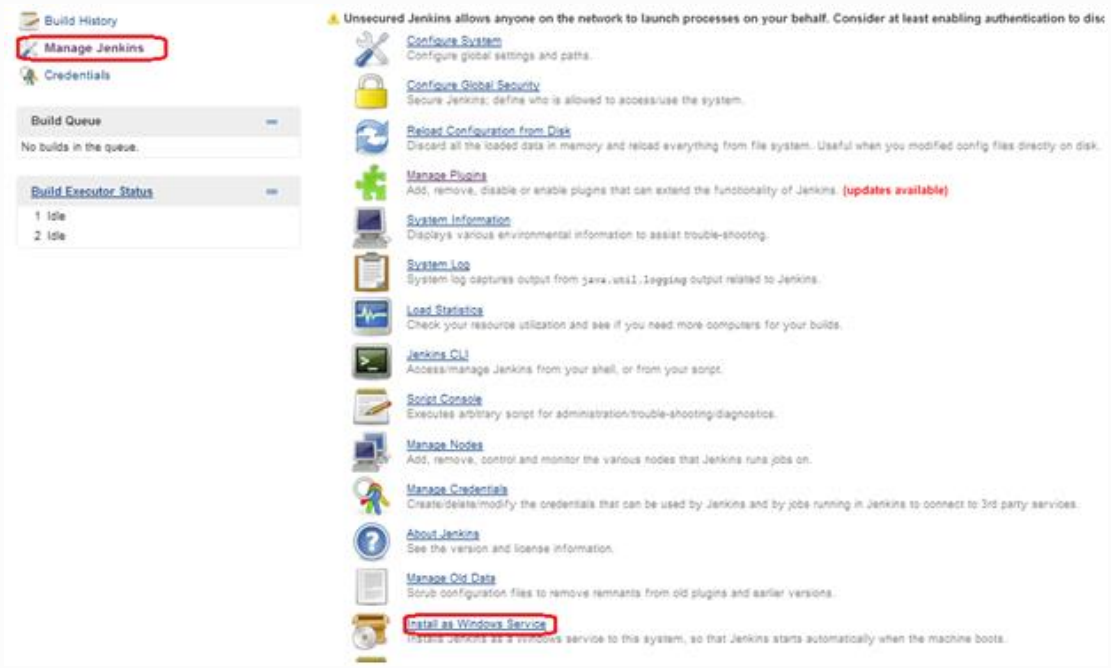
Voice les étapes a suivre pour installer Jenkins sur une machine Windows :

- Télécharger jenkins.war sur le site **https://jenkins.io/index.html**
- Copier le dans un répertoire C:\jenkins
- Ouvrer cmd (invites de commandes) et lancer la commande :
  java -jar jenkins.war
- Ouvrer votre navigateur web et, connectez-vous sur **http://localhost:8080**



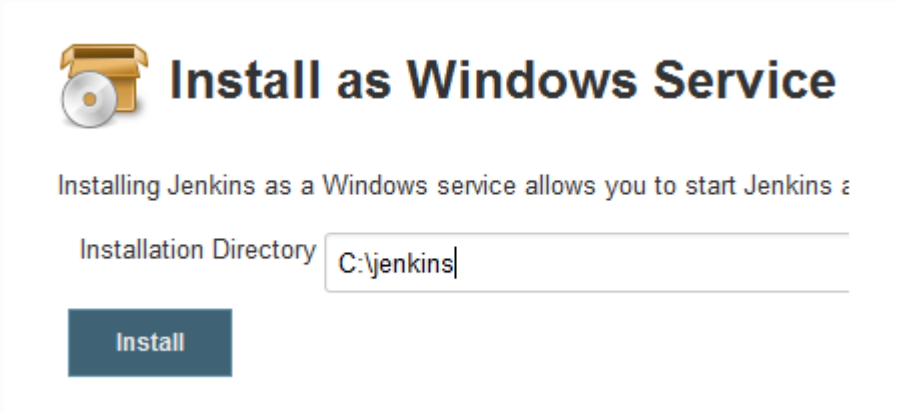
[*]

- Dans la barre de gauche, cliquer sur le lien **Manage Jenkins** > **« Install as Windows Service »**



[*]

- Spécifier maintenant le répertoire d'installation



- Cliquez sur **Install,** si tout se passe parfaitement, l'installation de Jenkins sera terminée.

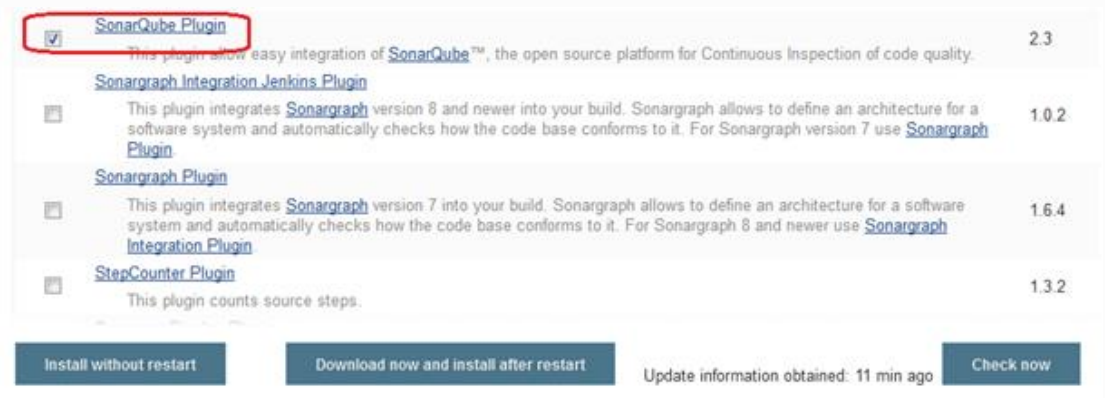
2.2.4 Installation de plugin SonarQube

Jenkins nous donne la possibilité d'intégré SonarQube avec grâce à un plugin qu'il faut l'ajouter, pour ce faire :

- Cliquer sur le lien '**Manage Jenkins' -> 'Manage Plugins'.**

[*]

- Puis cliquer sur l'onglet **« Available »** et sélectionner **SonarQube Plugin**

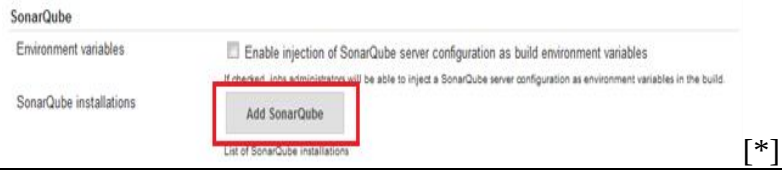
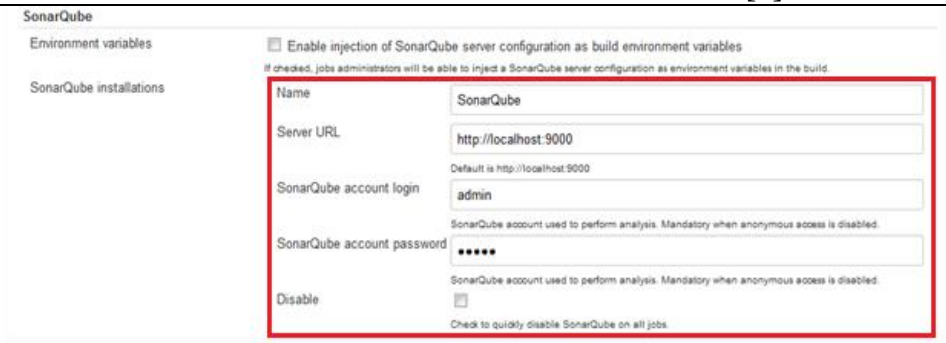


[*]

- Après avoir cocher le plugin, cliquer sur le bouton **Install without restart.**
- Si l'installation se passe sans problème, le plugin de SonarQube sera intègre dans notre conteneur Jenkins.

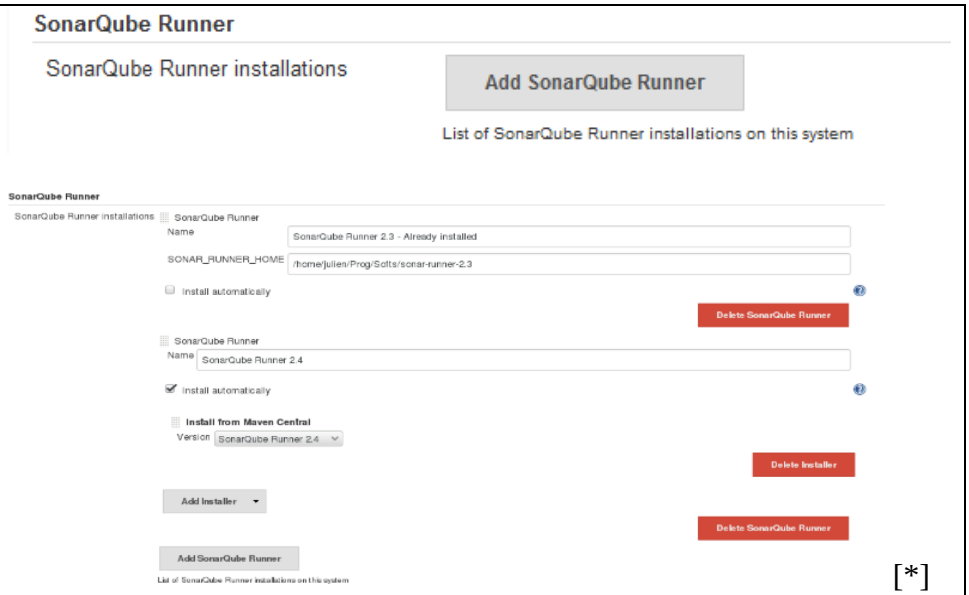
2.2.5 Installation du serveur SonarQube

L'installation du serveur SonarQube s'effectue en trois étapes :

Étapes	Description	Illustration
Étape 1	Dans le menu à gauche, cliquer sur **Manage Jenkins**	 [*]
Étape 2	Cliquer sur le bouton **Add SonarQube**	 [*]
Étape 3	Introduire les paramètres de configuration	

2.2.6 Installation du scanner SonarQube

Étapes	Description	Illustration
Étape 1	Dans le menu à gauche, cliquer sur **Manage Jenkins**	 [*]

Étape 2	Cliquer sur le bouton **Add SonarQube Runner** et procéder à l'installation	 [*]

3. Installation de SonarQube

Dans cette section, on va d'abord détailler les prérequis nécessaires à vérifier,

Ensuite, l'installation des différents composants de l'architecture de Sonarqube :

3.1 Prérequis

Les prérequis nécessaires pour le fonctionnement de Sonarqube sont regroupés ci-dessous sous deux catégories, logiciels et matériels :

- Prérequis logiciels :
  - Machine virtuelle Java : Oracle JRE 11, ou OpenJDK11
  - Base de données:
    - Configuration du codage en UTF-8.
    - Configuration de la langue : Anglais.
    - PostGreSQL : 9.3 et plus.

        - MS SQL Server :12.0 et plus.
        - Oracle : 11G et plus, Éditions XE.
    - Navigateurs :
        - MS Internet Explorer : IE 11.
        - MS Edge : Dernière mise à jour.
        - Mozilla Firefox : Dernière mise à jour.
        - Google Chrome : Dernière mise à jour.
        - Safari : Dernière mise à jour.
    - Intégration avec les services web de gestion du développement logiciel :
        - GitHub : GitHub Entreprise version 2.14 et plus.
        - Bitbucket : Bitbucket Server version 5.15 et plus.

- Prérequis matériels :
    - Mémoire RAM :
        - Un minimum de de 2G pour une petite équipe de développement.
        - Un minimum de 18G pour une grande équipe de développement.
    - Espace disque : Dépend de la quantité de lignes de code à analyser et la BD utilisée. Le disque dur doit présenter de hautes performances en lecture/écriture.

3.2 Installation par défaut de SonarQube

L'installation se fait en deux étapes :

- Installation du serveur SonarQube
- Installation du scanner SonarQube

3.2.1 Installation du serveur SonarQube

Ci-dessous les étapes d'installation de SonarQube :

Étapes	Description
Étape 1	Télécharger SonarQube : http://www.sonarsource.org/downloads/
Étape 2	Extraire le fichier Zip dans le lecteur C:/Sonarqube
Étape 3	Démarrer le serveur SonarQube : Taper la commande : C:\sonarqube\bin\windows-x86-xx\StartSonar.bat

3.2.2 Installation du serveur SonarQube

Étapes	Description
Étape 1	Télécharger le scanner SonarQube : https://sonarsource.bintray.com/Distribution/sonar-scanner-cli/sonar-scanner-2.5.1.zip
Étape 2	Extraire le fichier Zip dans le lecteur C:/Sonar/runner

3.3 Installation d'une base de données

Tel que mentionné dans la section X, SonarQube peut s'intégrer à plusieurs Bases de données. D'une façon générale, il va falloir créer un schéma vide et un utilisateur qui a les droits en lecture et écriture.

3.4 Installation du serveur Web

Les étapes suivantes sont à effectuer :

- Configurer l'accès à la base de données
    - L'accès se configure dans le fichier **<Root>/conf/sonar.properties**

- Chaque type de BD a ses propres configurations. Mettre les BD non utilisées en commentaire et saisir les informations d'accès pour la Bd utilisée.

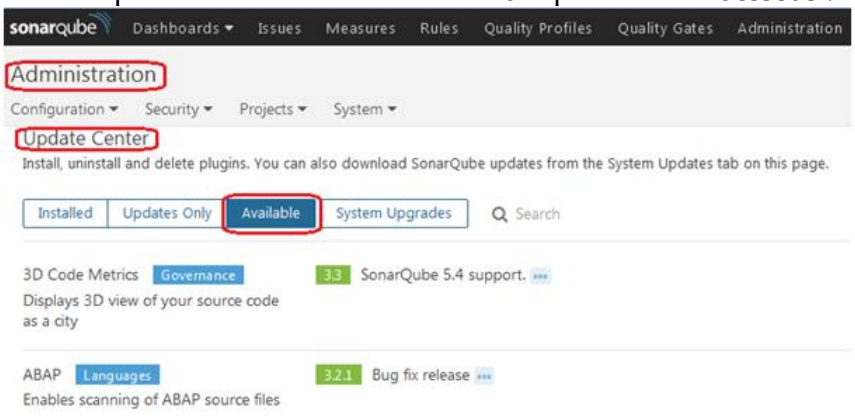
- Ajouter le Driver JDBC
  - Si la Base de données utilisée est Oracle, le driver JDBC est à copier dans **<Root>//*extensions/jdbc-driver/oracle.***
  - *Pour les autres types de base de données, les drivers sont déjà fournis par défaut.*
- Démarrer le serveur
  - Pour Linux, excéuter la commande suivante : **bin//sonar.sh start**
  - Pour Windows, exécuter la commande suivante : **bin/windows-x86-XX/StartSonar.bat**
  - La console utilisateur est accessible dans le lien **: http://localhost :9000**.

3.5 Installation des plugins

L'installation des plugins peut se faire d'une façon automatique, ou manuelle.

3.5.1 Installation automatique

- Se fait à partir de l'interface utilisateur tel qu'illustré ci-dessous :



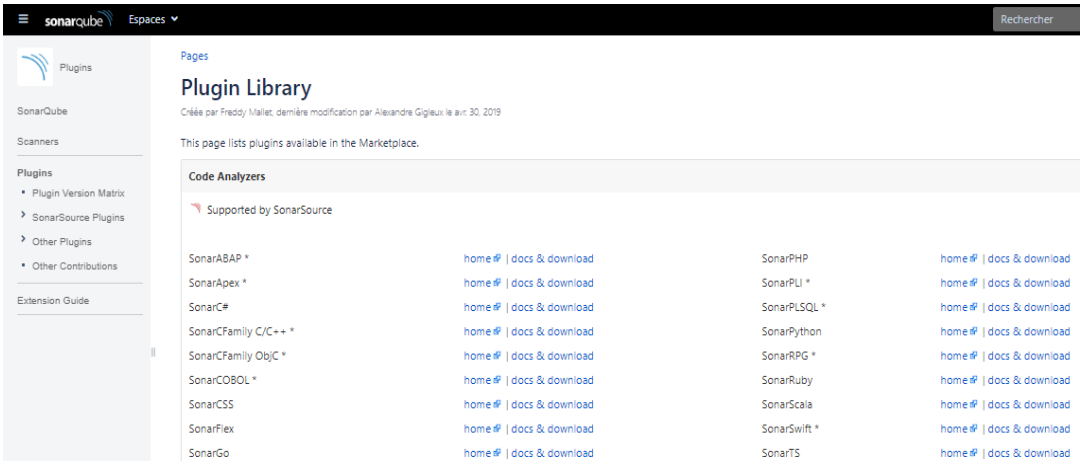
[*]

- Une liste des plugins est affichée sous l'onglet Available. On choisi le plugin et on clique sur Installer.

- Un redémarrage de SonarQube est requis suite à l'installation. Le plugin est donc affiché sous l'onglet Installed.
-

3.5.2 Installation manuelle

- Les plugins sont disponibles dans le lien : https://docs.sonarqube.org/display/PLUG/Plugin+Library



[*]

- Une fois le plugin choisi, on le télécharge.
- Le plugin est disponible dans le répertoire : **ROOT/extensions/plugins**
- Tel que c'est mentionné pout l'installation automatique, un redémarrage du SonarQube est requis.

4. Analyse de code source

Maintenant que SonarQube est installé, nous allons voir les différentes possibilités à analyser un code source. Avant d'analyser un code source, il faut s'assurer d'abord que le plugin pour le langage du code à analyser soit installé dans SonarQube.

4.1 Analyse de code pour Java et JavaScript avec Sonar Runner

4.1.1 Exécuter l'analyse de code

Afin d'analyser le code à travers Sonar Runner, il faut suivre les étapes suivantes :

- Pour analyser le code Java, il faut exécuter la commande suivante :

```
cd C:\sonar-examples-master\projects\languages\java\sonar-
runner\java-sonar-runner-simple
C:\sonar-scanner-2.5.1\bin\sonar-runner.bat
```

Voici une capture d'écran de la fin de l'analyse de code :

[*]

- Pour analyser le code JavaScript, il faut exécuter la commande suivante :

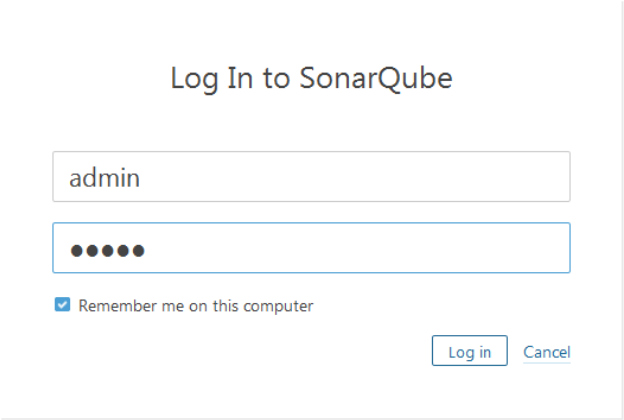
```
C:\sonar-examples master\projects\languages\javascript\javascript-
sonar-runnere
C:\sonar-scanner-2.5.1\bin\sonar-runner.bat
```

Voici une capture d'écran de la fin de l'analyse de code :

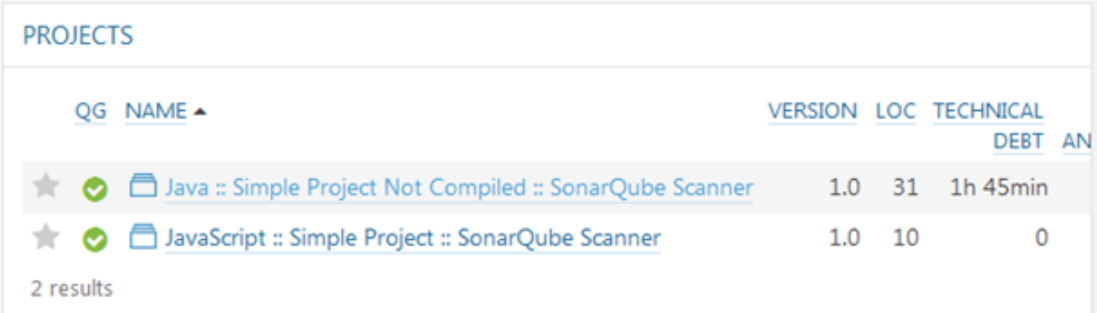
[*]

4.1.2 Analyse des résultats

Maintenant nous allons voir les résultats de l'analyse sur l'interface utilisateur à travers le lien : http://localhost:9000 (user : admin/ mot de passe : admin) :

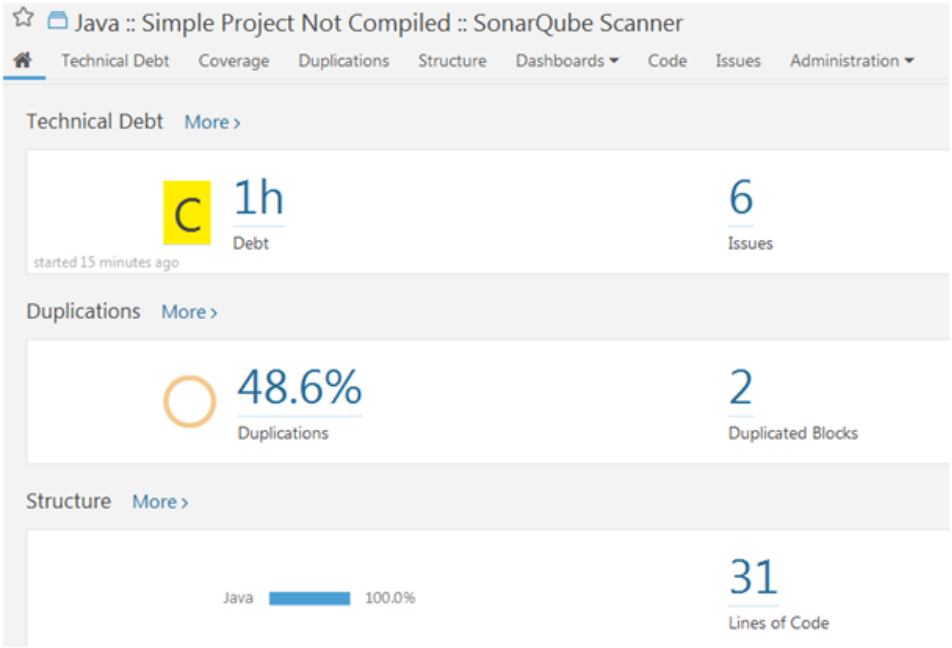


Après l'authentification, la page web affiche les résultats des deux analyses que nous avons effectué :



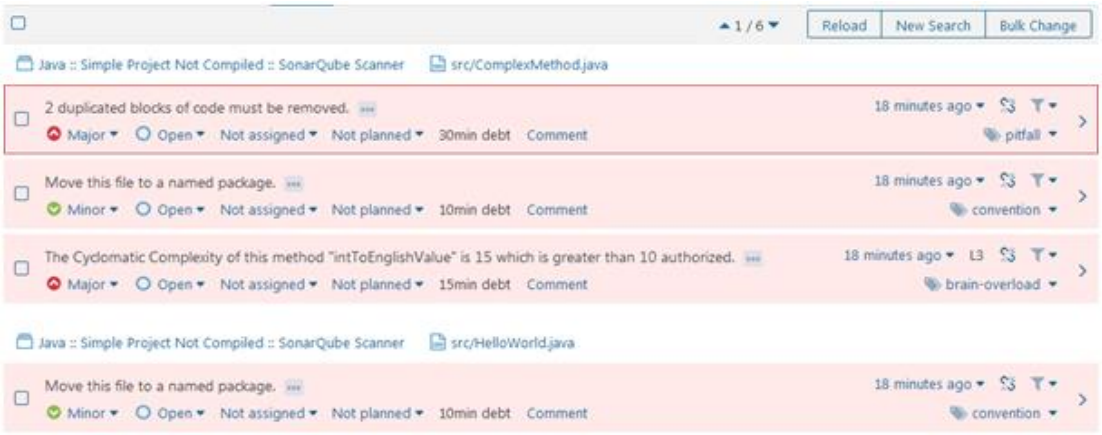
[*]

Cliquer sur la première analyse pour voir les résultats en détails :

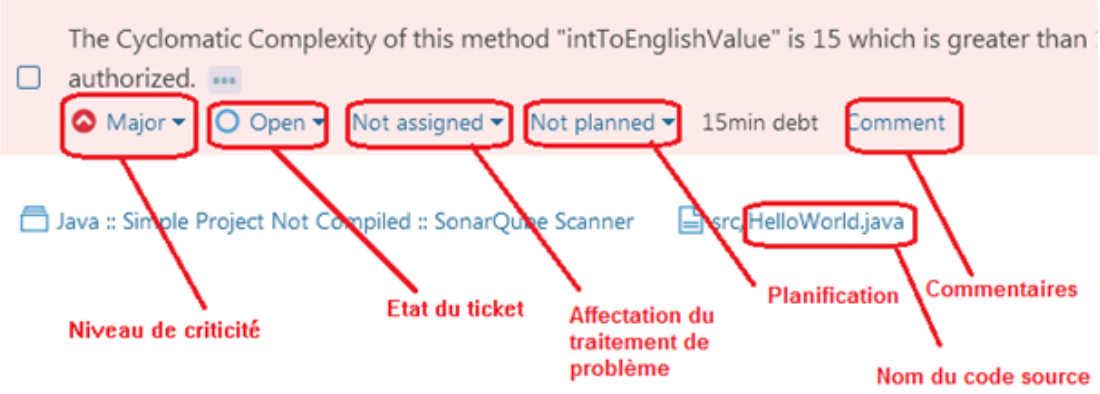


[*]

Cliquez par exemple sur les « issues » pour voir leur détail :



Les problèmes sont classifiés par type de criticité (Majeur / Mineur) et la même interface nous pouvons ouvrir un ticket pour leurs traitements.



[*]

4.2 Analyse de code C# avec MSBuild SonarQube Runner

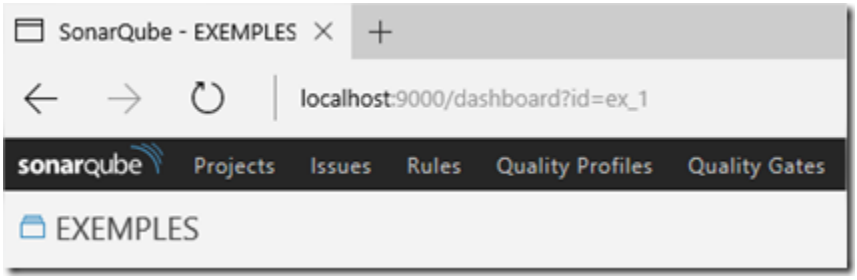
4.2.1 Exécuter l'analyse de code

Pour analyser le code C# avec SonarQube, il faut utiliser le scanner lié à MSBuild qui donne la possibilité d'analyser un projet ou tous les projets d'une solution.

Afin d'utiliser ce scanner, il faut tout d'abord l'installer et le configurer pour utiliser le serveur SonarQube. Puis démarrer l'invite de commande msbuild et exécuter le code suivant pour initialiser le scanner :

```
c:\<installation scanner>\sonarqube.scanner.msbuild begin /k:"ex_1"
/n:"EXEMPLES" /v:"1.0"
msbuild <solution>.sln
c:\<installation scanner>\sonarqube.scanner.msbuild end
```

Le projet "EXEMPLES" va être créé sur le serveur SonarQube avec l'identifiant est ex_1.

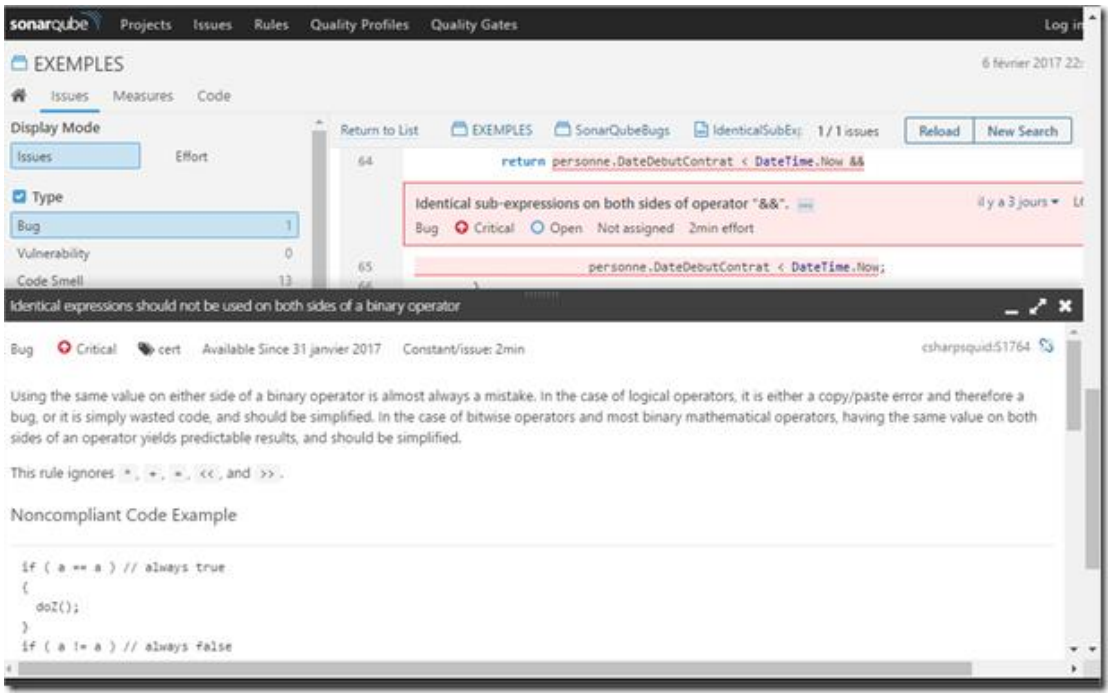


[*]

Les résultats évolue dans SonarQube évolue à chaque exécution d'une analyse avec l'identifiant « ex_1 ».

4.2.2 Analyse des résultats

Pour analyser les résultats, il faut suivre les mêmes étapes indiquées dans la section 4.1.2 Analyse des résultats pour Java.



[*]

4.3 Analyse de code PHP avec Sonar Runner

4.3.1 Paramétrage du projet

Tout d'abord il faut créer le fichier « sonar-project.properties » dans le répertoire de projet PHP à analyser, afin de le paramétrer.

Voici le contenu du fichier :

```
# required metadata
sonar.projectKey=my:project
sonar.projectName=My project
sonar.projectVersion=1.0
# path to source directories (required)
sources=srcDir1,srcDir2
# path to test source directories (optional)
tests=testDir1,testDir2
# path to project binaries (optional), for example directory of Java
bytecode
binaries=binDir
# optional commaseparated list of paths to libraries. Only path to
JAR file and path to directory of classes are
supported.
libraries=path/to/library.jar,path/to/classes/dir
# Uncomment this line to analyse a project which is not a java
project.
# The value of the property must be the key of the language.
#sonar.language=cobol
```

4.3.2 Exécuter l'analyse de code

Après avoir paramétrer le projet, nous pouvons lancer l'analyse à partir du répertoire où est situé le fichier « sonar-project.properties » avec la commande suivante :

```
cd /home/mika/www/mkframework/
/home/mika/bin/sonarrunner/bin/sonarrunner
```

L'analyseur va tourner pendant un bon moment, une fois qu'il aura terminé, vous commencer l'analyse des résultats.

4.3.3 Analyse des résultats

Sur la page web de SonarQube, et comme les autres langages de programmation (Java, C#...) Les problèmes sont classifiés par type de criticité (Majeur / Mineur) et la même interface nous permet pouvons ouvrir un ticket pour leurs traitements.

4.4 Analyse de code Angular avec Karma et Sonar Scanner

4.4.1 Exécuter l'analyse de code

Créez un fichier appelé « sonar-project.properties » dans votre répertoire racine angulaire et ajoutez les attributs ci-dessous :

```
sonar.host.url=http://localhost:9000
sonar.login=admin
sonar.password=admin
sonar.projectKey=test-app
sonar.projectName=test-app
sonar.projectVersion=1.0
sonar.sourceEncoding=UTF-8
sonar.sources=src
sonar.exclusions=**/node_modules/**
sonar.tests=src
sonar.test.inclusions=**/*.spec.ts
sonar.typescript.lcov.reportPaths=coverage/lcov.info
```

4.4.2 Analyse des résultats

Nous avons à la fois une couverture de code Karma et un serveur Sonar prêts, nous allons maintenant les intégrer à l'aide du sonar-scanner que nous avons installé à l'étape précédente.

Tout d'abord, ajoutez un script appelé sonar à votre package.json,

```
"scripts": {
    "sonar": "sonar-scanner"
}
```

Enfin, exécutez la commande ci-dessous pour intégrer la couverture Karma au serveur Sonar :

```
npm run sonar
```

vous obtiendrez le résultat directement sur le serveur Sonar en accédant à http://localhost:9000/ projects.

4.5 Analyse de code RPG avec Sonar Scanner

Après avoir installé le plugin Sonar RPG, vous pouvez exécutez l'analyse avec le scanner SonarQube en exécutant la commande suivante à partir du répertoire racine du projet:

```
sonar-scanner -Dsonar.projectKey=xxx -Dsonar.sources=.
```

vous obtiendrez le résultat directement sur le serveur Sonar en accédant à http://localhost:9000/ projects.

5. Conclusion

Enfin, ce projet nous a permis de connaitre l'importance de SonarQube et son utilité dans l'assurance qualité des projets de développement. Nous avons aussi appris l'installation et la configuration de SonarQube pour différents langages de programmation (Java, C#, PHP, JavaScript, Angular et RPG) et aussi l'exécution de l'analyse et comment analyser les résultats.

6. Références

[*] : http://www.ismanager.fr/initiation-a-sonarqube/

https://pragma3.wordpress.com/2017/02/11/analyse-statique-de-c-avec-sonarqube/

https://imikado.developpez.com/tutoriels/php/sonar/

https://medium.com/@learning.bikash/angular-code-coverage-with-sonarqube-d2283442080b

https://docs.sonarqube.org/pages/viewpage.action?pageId=4784221