

TP5 - MEDEV Sonar

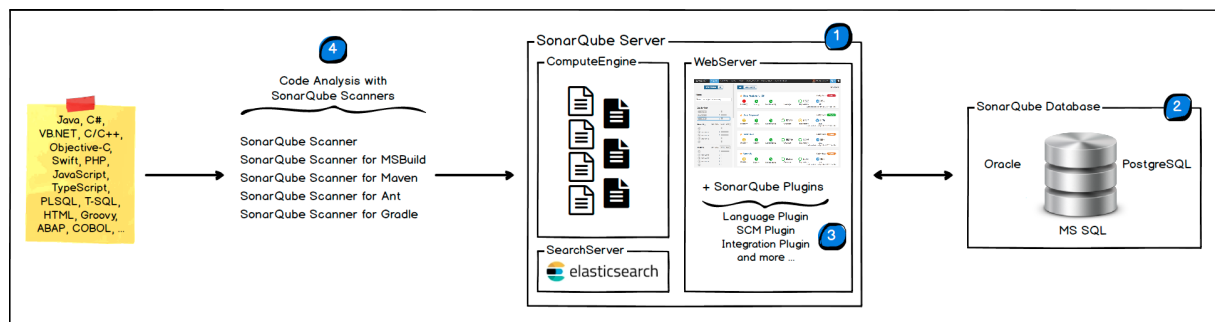
Ce TP est orienté sur le “Software Quality” et l'utilisation d'un des outils disponibles pour le mesurer : **SonarQube** (anciennement **Sonar**). Il existe de nombreux autres outils répertoriés sur cette page Wikipédia: [List of tools for static code analysis](https://en.wikipedia.org/wiki/List_of_tools_for_static_code_analysis#Multi-language)[1].

Comme d'autres domaines de la qualité, la qualité logicielle est encadrée par un ensemble de normes. Nous nous concentrons ici sur la norme ISO/CEI 9126 (redéfinie par la norme ISO/CEI 25010). Nous vous recommandons de lire cette introduction aux normes 9126 et 25010 pour vous familiariser avec ces concepts : [Les indicateurs de la qualité de la norme ISO 9126](http://www.test-recette.fr/generalites/qualite-logicielle/indicateurs.html)[2].

SonarQube (outil d'analyse statique de code) utilise le modèle de qualité SQALE, conforme à la norme ISO 9126. Il présente en sus de nombreux avantages (opensource, multilanguage, etc.) celui d'être – relativement – aisé à utiliser.

1) Architecture SonarQube/SonarScanners³

La plateforme SonarQube se compose de 4 éléments :



1. Un serveur SonarQube démarre 3 processus principaux :
 - a. Serveur Web pour les développeurs et les gestionnaires pour parcourir les vues résumant la qualité des projets et configurer l'instance de SonarQube
 - b. Serveur de recherche basé sur Elasticsearch pour sauvegarder les recherches de l'interface utilisateur
 - c. Un serveur dédié à l'analyse de code (Compute Engine) pour faire puis traiter les rapports d'analyse de code et ensuite les enregistrer dans la base de données SonarQube
2. Une base de données SonarQube pour stocker :
 - a. la configuration de l'instance de SonarQube (sécurité, paramètres des plugins, etc.)
 - b. les instantanés de qualité des projets, des vues, etc.
3. Plusieurs plugins SonarQube installés sur le serveur, incluant éventuellement des plugins de gestion des langues, de versionnement, d'intégration, d'authentification et de gouvernance (à paramétrer dans la fenêtre des plugins)
4. Un ou plusieurs SonarScanners fonctionnant sur les serveurs d'intégration continue ou de compilation pour analyser les projets

1 https://en.wikipedia.org/wiki/List_of_tools_for_static_code_analysis#Multi-language

2 <http://www.test-recette.fr/generalites/qualite-logicielle/indicateurs.html>

3 <https://docs.sonarqube.org/latest/architecture/architecture-integration/>

2) Téléchargement et Lancement

Nous allons utiliser un serveur sonar en local sur vos machines. Vous devez donc télécharger puis installer le serveur sonarQube pour ensuite pouvoir analyser vos codes à l'aide du SonarQube Scanner et afficher les résultats de l'analyse.

ATTENTION. Télécharger [SonarQube 8.1](https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-8.1.0.31237.zip)

(<https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-8.1.0.31237.zip>)

[SonarQube Scanner 4.2](https://docs.sonarqube.org/latest/analysis/scan/sonarscanner/) (<https://docs.sonarqube.org/latest/analysis/scan/sonarscanner/>).

Suivez les instructions de la page suivante : “[Get started in two minutes](https://docs.sonarqube.org/latest/analysis/scan/sonarscanner/)”[4] pour mettre en place l'environnement server en local. **Attention, le démarrage du serveur nécessite Java 11.** Dans un navigateur web à l'adresse **http://localhost:9000**, assurez-vous de vous connecter en tant qu'administrateur (admin / admin), pour pouvoir plus tard configurer les profils de qualité.

Installez à partir de l'onglet « Administration>Marketplace » les plug-ins suivants :

- Code Smells
- Findbugs
- French Pack

3) Analyser son code

Au besoin référez-vous à cette ressource documentaire : [Analyse avec le SonarQube Scanner](https://docs.sonarqube.org/latest/analysis/scan/sonarscanner/)[5] (partie Prerequis + partie Usage). A noter que par défaut, vous avez un plugin SonarQube pour le langage Java, il n'est donc pas nécessaire d'en ajouter un pour scanner un projet Java.

a) Directement à partir de SonarQube Scanner

Téléchargez [SonarQube Scanner 4.2](https://docs.sonarqube.org/latest/analysis/scan/sonarscanner/)

(<https://docs.sonarqube.org/latest/analysis/scan/sonarscanner/>).

A la racine de votre projet, créez un fichier de configuration similaire à celui ci-dessous nommé `sonar-project.properties`. Puis lancer la commande `sonar-scanner`.

```
# must be unique in a given SonarQube instance

sonar.projectKey=Dames:project

# this is the name displayed in the SonarQube UI

sonar.projectName=JeuDeDames

sonar.projectVersion=1.0
```

4 [http://docs.sonarqube.org/latest/analysis/scan/sonarscanner/](https://docs.sonarqube.org/latest/analysis/scan/sonarscanner/)

5 [http://docs.sonarqube.org/latest/analysis/scan/sonarscanner/](https://docs.sonarqube.org/latest/analysis/scan/sonarscanner/)

```
# Path is relative to the sonar-project.properties file. Replace "\" by "/"
on Windows.

# Since SonarQube 4.2, this property is optional if sonar.modules is set.

# If not set, SonarQube starts looking for source code from the directory
containing

# the sonar-project.properties file.

sonar.sources=src

sonar.java.binaries=target/classes

#sonar.java.binaries=build/classes

# Encoding of the source code. Default is default system encoding

#sonar.sourceEncoding=UTF-8
```

Vérifiez les résultats de l'analyse. Soyez attentif aux différents indices, tel que le Rating, la dette technique, etc. La [documentation de SonarQube](https://docs.sonarqube.org/latest/analysis/scan/sonarscanner-for-maven/) vous aidera à assimiler ces indices. **Avant de poursuivre, vérifiez vos « issues », mais ne les corrigez pas encore.**

Vous trouverez dans la partie « User Guide » de la documentation une description des métriques et concepts utilisés.

b) Scanner pour projet Maven

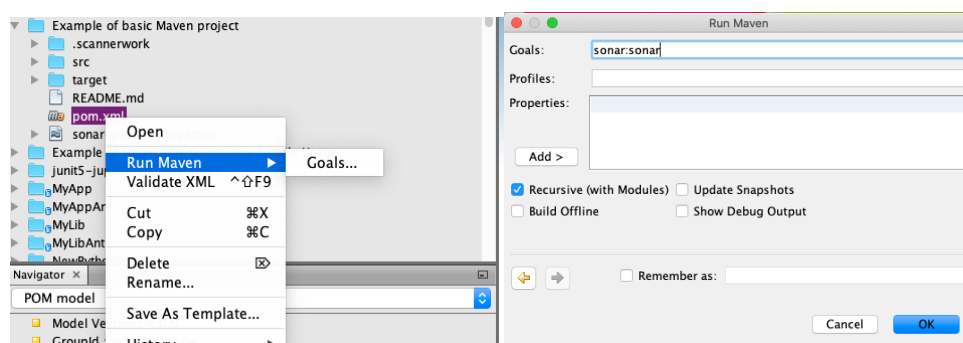
Il est possible de lancer scanner sonar directement via Maven :

<https://docs.sonarqube.org/latest/analysis/scan/sonarscanner-for-maven/>

Pour cela il vous faut éditer et compléter le fichier **settings.xml** de maven comme précisé dans l'URL précédente.

Remarque : ce fichier pourra être trouvé dans l'arborescence de fichiers NetBeans/java/maven/conf

Ensuite tout scanner d'un projet maven pourra être fait en lançant le *goal* sonar:sonar comme montré dans les captures d'écrans suivantes (clic droit sur le fichier pom.xml).



c) Scanner pour projet ant

Pour un projet Ant, commencez par télécharger le jar nécessaire : <https://binaries.sonarsource.com/Distribution/sonarqube-ant-task/sonarqube-ant-task-2.7.0.1612.jar>

et modifiez le build.xml de votre projet en conséquence comme indiqué ici :

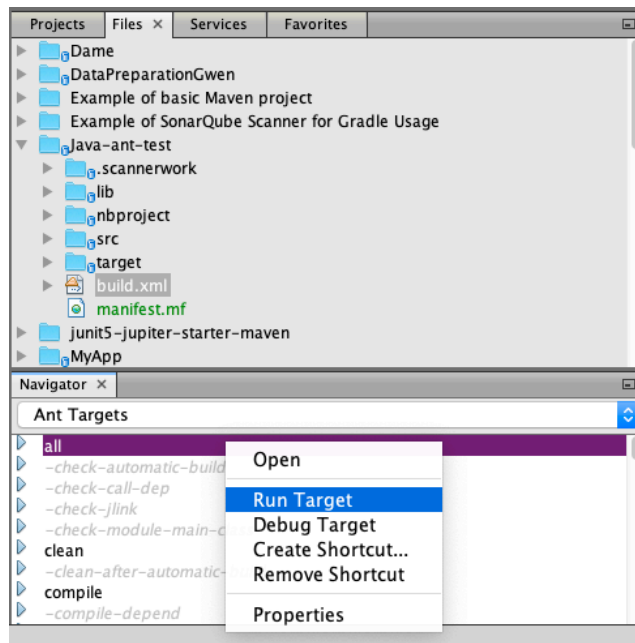
<https://docs.sonarqube.org/latest/analysis/scan/sonarscanner-for-ant/>

ou dans l'exemple suivant :

<https://github.com/SonarSource/sonar-scanning-examples/tree/master/sonarqube-scanner-ant>

Vous pourrez ensuite exécuter votre cible sonar de ant après avoir compilé.

L'exemple ci-dessous montre l'exécution manuelle de la cible « all » qui dépend des cibles « clean,compile,sonar » dans le projet sonarqube-scanner-ant d'exemple.



4) Ajout d'un outil de « couverture de tests » - exemple pour un projet maven

Pour ajouter une mesure de la couverture de vos tests à votre projet, nous allons utiliser l'outil JaCocoverage avec maven.

Pour cela télécharger et exécuter l'exemple de projet maven avec couvertures de tests de la page suivante :

<https://github.com/SonarSource/sonar-scanning-examples/tree/master/sonarqube-scanner-maven/maven-basic>

Après exécution et lancement du goal sonar:sonar, le projet apparaît sur le serveur sonarqube avec la couverture de test associée.

La couverture de test donne un pourcentage du code source pris en charge par les tests. Une valeur de 0 % signifie qu'il n'y a pas de code pris en charge par les tests ; une valeur de 100 % que l'ensemble est pris en charge. Une valeur haute de couverture de tests est conseillée, même s'il n'y a pas de « valeur type » à atteindre.

5) Correction du code

Analysez les rapports correspondants à votre code et faites votre possible pour augmenter votre SQALE Rating. Gardez toutefois à l'esprit que l'objectif n'est pas tant d'avoir un « Rating » « A », mais est d'avoir un code correctement testé et qui fonctionne (cf. TP précédents). Exécutez une analyse sonar après des modifications majeures de votre code pour voir l'évolution des métriques.