

Mesurer la qualité du code en continu avec

sonarqube

Plan

- **Qualité du code**
- Présentation de SonarQube
- Installation
- Configuration
- Utilisation

- But
 - ▣ Fiabilité
 - ▣ Maintenabilité
 - ▣ Réutilisabilité

Critères	Métriques
Respect d'un standard de codage	
Documentation du code	Taux de documentation
Tests	Taux de couverture
Non duplication du code	Taux de duplication
Règles de responsabilité unique	Taille du code Nombre de lignes de code par fichier Nombre de lignes de code par classe Nombre de méthodes par classe Complexité cyclomatique

- Complexité cyclomatique = nombre de chemins dans le code

Complexité	Basse	Modérée	Haute	Très haute
valeur	1 à 4	5 à 7	8 à 10	≥ 11

Calcul approximé par les outils : addition de certains mots clés et opérateurs

```
+ 1  function fact(nb) {  
2  
++ 3  if ((nb === 0) || (nb === 1)){  
+ 4      return 1;  
5  }  
6  
7      return (nb * fact(nb-1));  
8  }
```

- Complexité cyclomatique = nombre de chemins dans le code

Complexité	Basse	Modérée	Haute	Très haute
valeur	1 à 4	5 à 7	8 à 10	≥ 11

```

+ if (scope == SCOPE.GLOBAL) {
    //ALL non team members gets only READ Permission
+ if (permission.getImpliedBy() == Permission.READ) {
+ return true;
    }
    // Member of any of the team with JOB CREATE Permission can create Job
+ if (permission == Item.CREATE) {
+ for (Team userTeam : teamManager.findUserTeams(userName)) {
+ if (isTeamAwareSecurityRealm()) {
+ return true; // for now give full permission to all team members
    }
    TeamMember member = userTeam.findMember(userName);
++ if ((member != null) && member.hasPermission(Item.CREATE)) {
+ return true;
    }
    }
    }
    // Member of any of the team with View CREATE Permission can create View
+ if (permission == View.CREATE) {
+ for (Team userTeam : teamManager.findUserTeams(userName)) {

```

□ Problème de la complexité cyclomatique

```
int sumOfPrimes(int max) { // +1
    int total = 0;
    OUT: for (int i = 1; i <= max; ++i) { // +1
        for (int j = 2; j < i; ++j) { // +1
            if (i % j == 0) { // +1
                continue OUT;
            }
        }
        total += i;
    }
    return total;
} // Cyclomatic Complexity 4
```

```
String getWords(int number) { // +1
    switch (number) {
        case 1: // +1
            return "one";
        case 2: // +1
            return "a couple";
        default: // +1
            return "lots";
    }
} // Cyclomatic Complexity 4
```

- Complexité cognitive (v6)
 - ▣ Basée sur la complexité cyclomatique
 - ▣ Mesure la difficulté de compréhension du code

```
String getWords(int number) { // Cyclomatic Complexity    Cognitive Complexity
  switch (number) { // +1
    case 1: // +1
      return "one";
    case 2: // +1
      return "a couple";
    default: // +1
      return "lots";
  }
} // =4    =1
```

Critères	Métriques	Outils
Standard de codage		phpcs  
Documentation du code	Taux de documentation	Cloc
Tests	Taux de couverture	Junit, PHPUnit, Cobertura
Non duplication du code	Taux de duplication	CPD, Phpcpd
Règles de responsabilité unique	Taille du code : Nombre de lignes par fichier Nombre de lignes par classe Nombre de méthodes par classe Complexité cyclomatique	Cloc JDepend
Absence de bogue, sécurité		FindBugs 

Plan

- Qualité du code
- **Présentation de SonarQube**
- Installation
- Configuration
- Utilisation

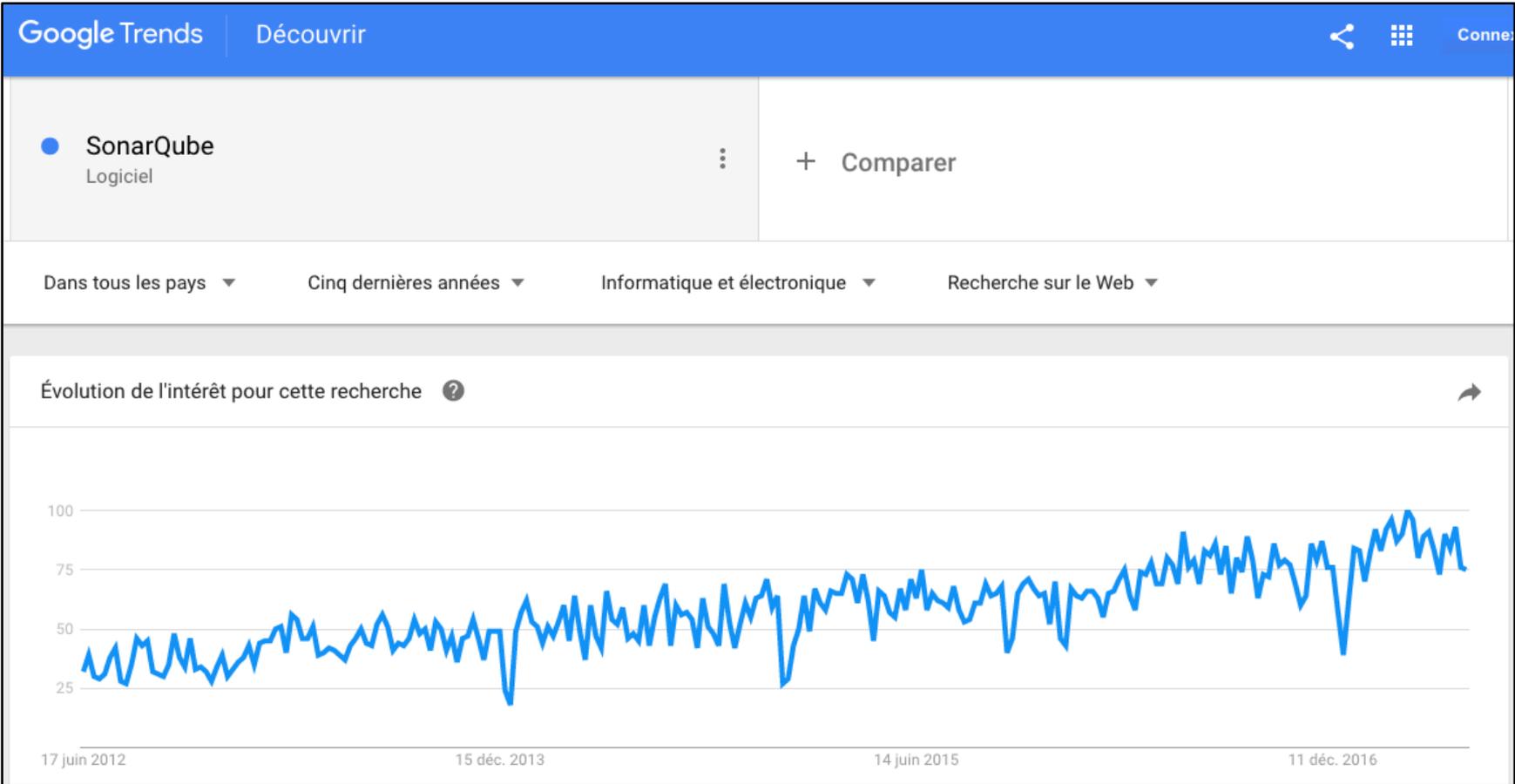
□ Introduction

- Mesure de la qualité du code source
- ~ 20 langages : C, C++, PHP, Java, Python, JavaScript, XML, ...
- Open Source, licence GNU GPLv3
- Développé depuis 2007
- Périmètre : code source, design, tests unitaires



□ Clients SonarSource





<https://trends.google.fr/trends/>

□ **Fonctionnalités**

▣ **Mesure la qualité**

- Taille du projet
- Densité des commentaires
- Taux de couverture
- Respect des conventions de nommage et de codage
- Détection des bogues
- Détection de code mort
- Détection de code dupliqué
- Complexité
- Score de maintenabilité, fiabilité et sécurité
- Dette technique

Measures Code

All Reliability Security **Maintainability** Coverage Duplications Size Complexity Issues Leak Period: last 30 days

41,695 Code Smells	5,750 New Code Smells	 Maintainability Rating	Technical Debt	1018d
			Added Technical Debt	127d
			Technical Debt Ratio	13.3%
			Technical Debt Ratio on New Code	0.0%
			Effort to Reach Maintainability Rating A	636d

- Fonctionnalités
 - ▣ Informations au niveau projet, fichier, classe, méthode
 - ▣ Historique des analyses
 - ▣ Résultat accessible via le web
 - ▣ Annotation du code source

Indication sur le problème

duplication

```
78 jerem...  public boolean authenticate(final Request request, final HttpServletResponse response, final LoginConfig loginConfig) {  
79 dbloc...  
80  
81 jerem...  
82     this.log.warn("missing context/realm");  
83 jerem...  
84 dbloc...  
85     }  
86  
87 jerem...  
88 jerem...  
89 dbloc...  
90 jerem...  
    final boolean negotiateCheck = request.getParameter("j_negotiate_check") != null;
```

The Cyclomatic Complexity of this method "authenticate" is 15 which is greater than 10 authorized. ... il y a 6 mois L78  
Code Smell  Major  Open Jeremy Landis 15min effort  brain-overload

Code partiellement testé

Not covered by tests.

Code testé non testé

□ Différents types de problèmes (issues)

□ Vulnerabilities 

```

public function setPasswordSetsPassword()
{
    $password = 'f00Bar';
}
    
```

Remove this hard-coded password. ... 2 years ago ▾ L65 

 Vulnerability  Blocker  Open Not assigned 30min effort  cwe, owasp-a2, sans-top25-porous

```

$this->subject->setPassword($password);
$this->assertSame($password, $this->subject->getPassword());
    
```

□ Bug 

```

henri. // Skip any extra separator chars
while (++i < end && ((c = path.charAt(i)) == '/' || c == '\\')) {
}
winst. // Add token for separator unless we reached the end
    
```

Either remove or fill this block of code. ... 3 years ago ▾ L241 

 Bug  Major  Open Not assigned 5min effort  No tags

- Différents types de problèmes (issues)

- ▣ Code smell 

```
public static final int CHARACTERS = 7;
```

Rename field "CHARACTERS" to prevent any misunderstanding/clash with method "characters" defined on line 162. ... 2 years ago ▾ L36 

 Code Smell  Blocker  Open Not assigned 10min effort  confusing

```
public static final int IGNORABLE_WHITESPACE = 8;
```

Refactor this function to reduce its Cognitive Complexity from 34 to the 15 allowed. ... 5 months ago ▾ L1528 **15** 

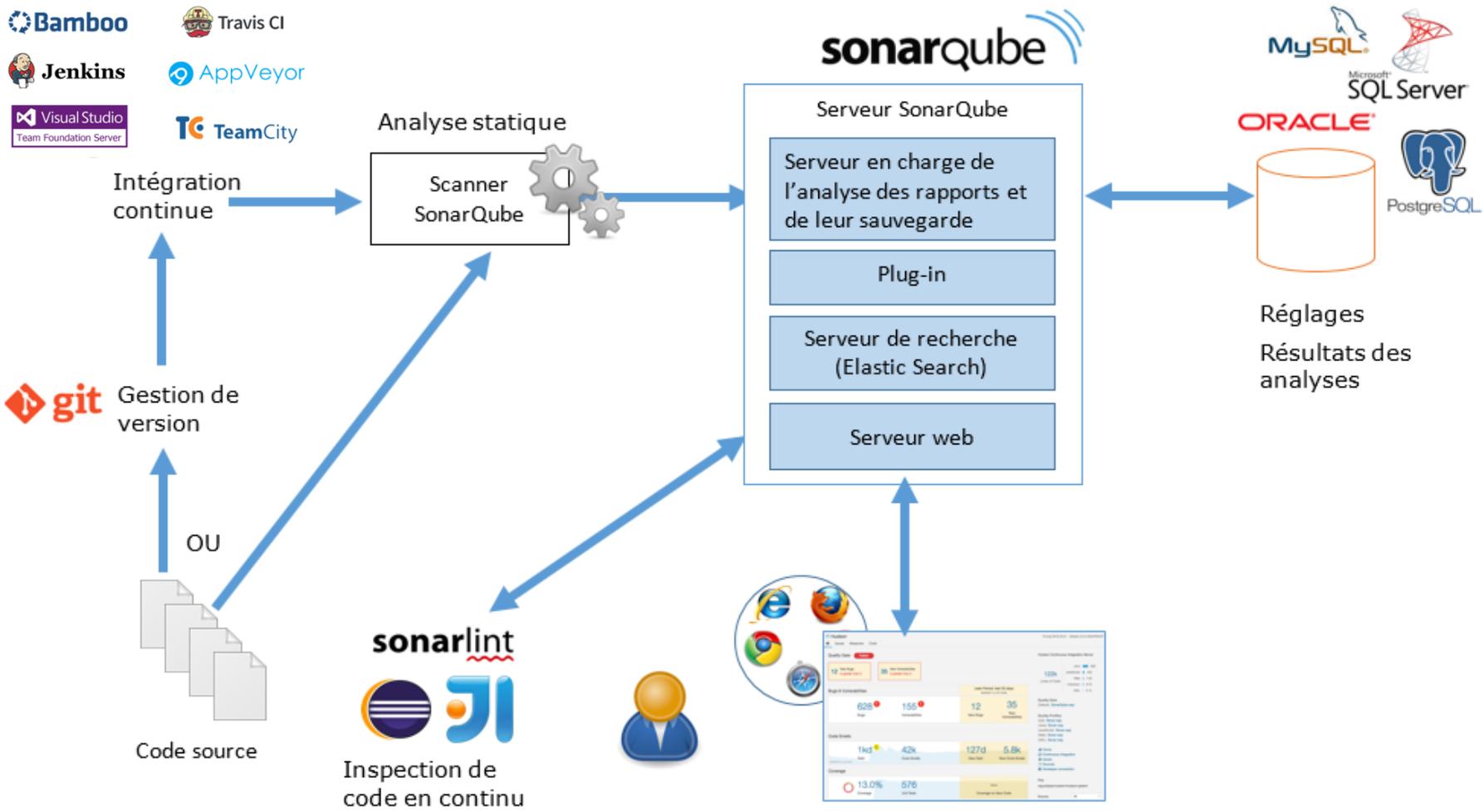
 Code Smell  Critical  Open Q2aki 24min effort  brain-overload

```

"""Calculates inverse (fast assumes is_scale_rotation_translation is True)."""
def adjoint(m, ii, jj):
    result = []
    5 for i, row in enumerate(m):
        12 if i == ii: continue
        result.append([])
        13 for j, x in enumerate(row):
            2 if j == jj: continue
            result[-1].append(x)
    return result
def determinant(m):
    9 if len(m) == 2:
        return m[0][0]*m[1][1] - m[1][0]*m[0][1]

```

Architecture de la plateforme



Plan

- Qualité du code
- Présentation de SonarQube
- **Installation**
- Configuration
- Utilisation

- Pré-requis (version 6.4)
 - ▣ Côté serveur
 - Java : Oracle JRE 8 ou OpenJDK 8 (conseillé pour mac OS X)
 - SGBD :
 - MySQL \geq 5.6
 - Oracle \geq 11G
 - PostgreSQL \geq 8
 - Microsoft SQL Server \geq 11.0
 - RAM : 2GO

 - ▣ Côté client
 - Navigateur web récent
 - JavaScript activé

- ❑ Télécharger Sonarqube
- ❑ Créer une base et un compte Sonar dans MySQL
- ❑ Configuration sonar.properties

- ❑ URL d'accès `http://localhost:7223`

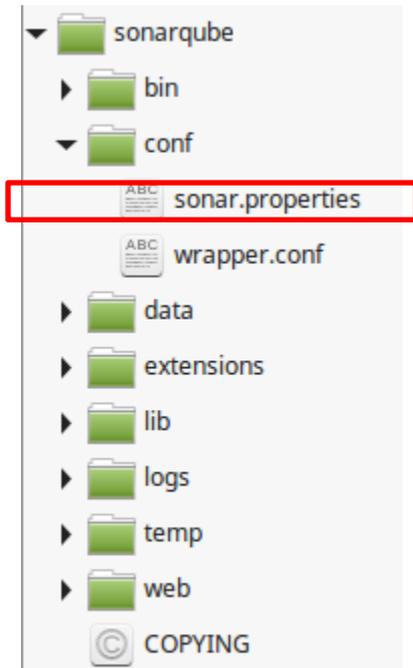
```
# 0.0.0.0 = n'importe quelle interface réseau
sonar.web.host= 0.0.0.0
# port (valeur par défaut 9000)
sonar.web.port=7223
# si /sonar/ => URL = http://localhost:7223/sonar/
sonar.web.context=/
```

- ❑ Login et mot de passe de la base de données

```
sonar.jdbc.username=sonar
sonar.jdbc.password=sonar
```

- ❑ MySQL

```
#----- MySQL 5.x
sonar.jdbc.url=jdbc:mysql://localhost:3306/sonar?useUnicode=true&character
Encoding=utf8&rewriteBatchedStatements=true&useConfigs=maxPerformance
```



□ Démarrer Sonarqube

▣ Sous linux/mac os :

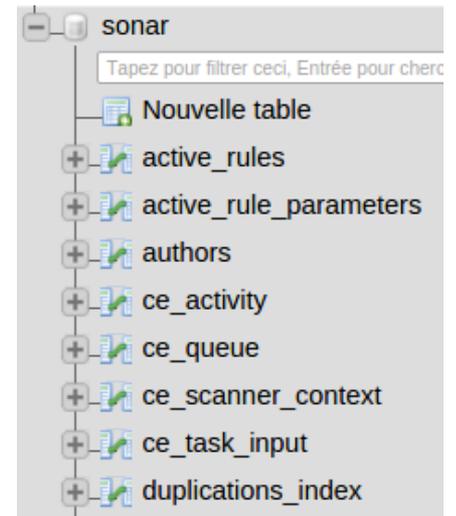
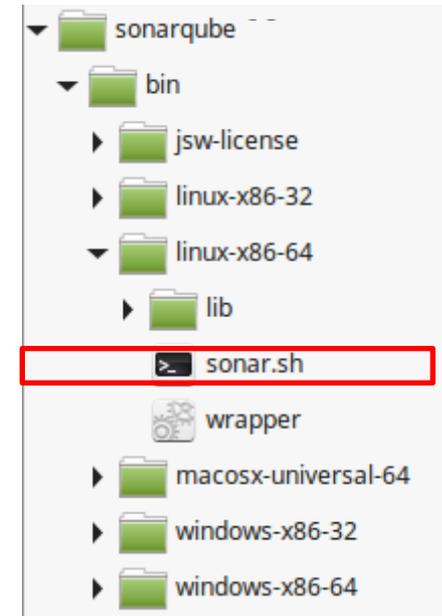
```
sudo ./sonar.sh start
```

▣ Sous Windows

```
StartSonar.bat
```

▣ Les tables sont créées dans la base sonar de MySQL

▣ L'interface web de sonar est accessible sur le port 7223

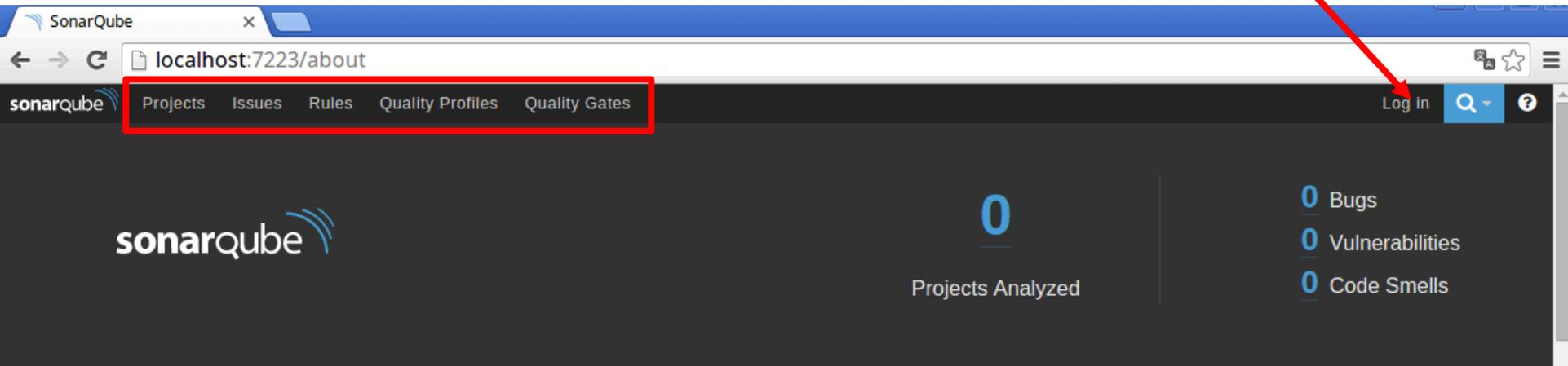


Plan

- Qualité du code
- Présentation de SonarQube
- Installation
- **Configuration**
- Utilisation

- Interface web de sonarQube
 - ▣ <http://localhost:7223/>

Login : admin
Mdp : admin



Keep your code clean by fixing the leak

By fixing new issues as they appear in code, you create and maintain a clean code base. Even on legacy projects, focusing on keeping new code clean will eventually yield a code base you can be proud of.

Understanding the Leak Period

The leak metaphor and the default Quality Gate are based on the leak period - the recent period against which you're tracking issues. For some previous_version makes the most sense, for others the last 30 days is a good option.

SonarQube Quality Model

Bugs

Bugs track code that is demonstrably wrong or highly likely to yield unexpected behavior.

Vulnerabilities

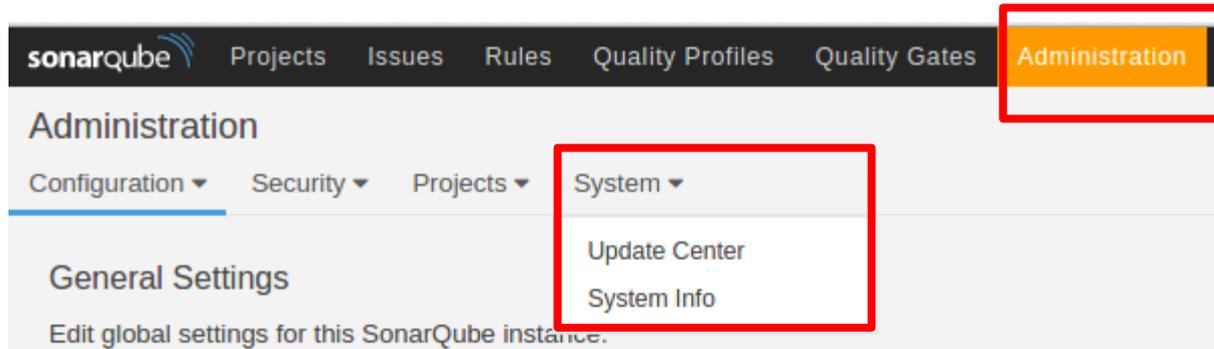
Vulnerabilities are raised on code that is potentially vulnerable to exploitation by hackers.

Code Smells

□ Interface d'administration

The screenshot shows the SonarQube Administration interface. At the top, there is a navigation bar with the following items: sonarqube, Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration (highlighted with a red box), Administrator, a search icon, and a help icon. Below the navigation bar, the page title is 'Administration', followed by a sub-menu with 'Configuration', 'Security', 'Projects', and 'System'. The main content area is titled 'General Settings' and contains the text 'Edit global settings for this SonarQube instance.' On the left side, there is a sidebar with a list of categories: Analysis Scope, C#, General, Java, and JavaScript. The main content area displays the 'Database Cleaner' settings, including a section titled 'Keep only one snapshot a day after' with a text description, a key 'sonar.dbcleaner.hoursBeforeKeepingOnlyOneS...', and a text input field containing the value '24' with '(default)' below it. Below this, there is a section titled 'Clean directory/package history' with a partially visible text description.

- Interface d'administration
 - ▣ Centre de mises à jour



- ▣ Listes des plugins

Administration

Configuration ▾ Security ▾ Projects ▾ System ▾

Update Center

Install, uninstall and delete plugins. You can also download SonarQube updates from the System Updates tab on this page.

Installed Updates Only **Available** System Upgrades 🔍 Search

C# Languages Code analyzer for C# projects	5.5.1.522 installed Updates: 5.5.2 Bug fix for 0-length issue parsing ...	Homepage Issue Tracker Licensed under GNU LGPL 3 Developed by SonarSource	Update to 5.5.2 Uninstall
------------------------------------------------------	-----------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------

□ Profil Qualité

- ▣ Création d'un nouveau profil en copiant celui de SonarWay

Quality Profiles

Quality Profiles are collections of rules to apply during an analysis.
For each language there is a default profile. All projects not explicitly assigned to some other profile will be analyzed with the default

PHP Profiles ▾

PHP, 3 profile(s)	Projects	Rules	Updated	Used
Drupal	0	20	Never	Never <input type="button" value="▾"/>
PSR-2	0	20	Never	Never <input type="button" value="▾"/>
Sonar way	Default	64	Never	Never <input type="button" value="▾"/>

- Activate More Rules
- Back up
- Compare
- Copy**
- Rename

- Profil Qualité
 - ▣ Configurer un profil

Quality Profiles / PHP

My way

Updated: il y a quelques secondes

Used: Never

[Changelog](#)

[Actions](#) ▾

Rules	Active	Inactive
Total	64	62
Bugs	15	11
Vulnerabilities	3	7
Code Smells	46	44

[Activate More](#)

Inheritance [Change Parent](#)

My way 64 active rules

Projects [Change Projects](#)

No projects are explicitly associated to the profile.

- Profil Qualité
 - ▣ Ajout ou suppression de règles à un profil

The screenshot shows the SonarQube interface for configuring rules. The top navigation bar includes 'sonarqube', 'Projects', 'Issues', 'Rules', 'Quality Profiles', 'Quality Gates', and 'Administration'. The user is logged in as 'Administrator'. The 'Rules' section is active, showing a list of 65 rules. The rule 'Functions should not be too complex' is highlighted with a red box. The 'active' status of the 'My way PHP' quality profile is also highlighted with a red box.

Rules	1 / 65 rules	Reload	New Search	Bulk Change
Functions should not be too complex	PHP Code Smell	brain-overload		Deactivate
Functions should not contain too many return statements	PHP Code Smell	brain-overload		Deactivate
Functions should not have too many lines	PHP Code Smell	brain-overload		Deactivate
Functions should not have too many parameters	PHP Code Smell	brain-overload		Deactivate
Generic exceptions RuntimeException, RuntimeException and Exception should not be thrown	PHP Bug	cert, cwe, error-handling		Deactivate
Identical expressions should not be used on both sides of a binary operator	PHP Bug	cert		Deactivate
Interface names should comply with a naming convention	PHP Code Smell	convention		Deactivate
Jump statements should not be followed by other statements	PHP Bug	cert, cwe, misra, unused		Deactivate

Quality Profile: My way PHP (active inactive)

- Profil Qualité
 - ▣ Configuration d'une règle

The screenshot shows the SonarQube interface for configuring a rule. The top navigation bar includes 'sonarqube', 'Projects', 'Issues', 'Rules', 'Quality Profiles', 'Quality Gates', and 'Administration'. The user is logged in as 'Administrator'. The 'Rules' section is active, showing 27 / 65 rules. A search bar is present. On the left, there are filters for Language, Type, Tag, Repository, Default Severity, Status, Available Since, and Template. The 'Quality Profile' section is expanded, showing 'Drupal PHP' and 'My way PHP' (active). The main content area displays the rule 'Functions should not be too complex' with a 'Code Smell' icon circled in red. A red arrow points from this icon to a red-bordered box containing the text 'Bug Code smell Vulnerability'. The rule is marked as 'Critical' and 'brain-overload'. The description states: 'The cyclomatic complexity of functions should not exceed a defined threshold. Complex code can perform poorly and will in any case be difficult to understand and therefore to maintain.' There is an 'Extend Description' button. The 'Parameters' section shows 'threshold' with a description 'The maximum authorized complexity.' and a 'Default Value: 20'. A red box highlights 'threshold: 20' in the 'Quality Profiles' section. A dropdown menu is open, showing severity levels: Blocker, Critical, Major, Minor, and Info. The 'Change' button is highlighted with a red box. The 'My way' profile is shown with a 'Critical' status and a 'threshold: 20' value.

□ Barrière qualité

▣ Profil par défaut

- Taux de couverture tests > 80
- Score de maintenabilité : A
- Score de fiabilité : A
- Score de sécurité : A

Quality Gate: Passed

Quality Gate: Failed

Score	Fiabilité	Sécurité	Maintenabilité
A	0 bug	0 vulnérabilité	$0.00 \leq \text{ratio dette technique} \leq 0.05$
B	≥ 1 bug mineur	≥ 1 vulnérabilité mineure	$0.06 \leq \text{ratio dette technique} \leq 0.1$
C	≥ 1 bug majeur	≥ 1 vulnérabilité majeure	$0.11 \leq \text{ratio dette technique} \leq 0.20$
D	≥ 1 bug critique	≥ 1 vulnérabilité critique	$0.21 \leq \text{ratio dette technique} \leq 0.5$
E	≥ 1 bug bloquant	≥ 1 vulnérabilité bloquante	$0.51 \leq \text{ratio dette technique} \leq 1$
Issue	Bug	Vulnerability	Code smell

sonarqube
Projects
Issues
Rules
Quality Profiles
Quality Gates
Administration
Administrator ▾
🔍
?

Quality Gates [Rename](#) [Copy](#) [Unset as Default](#) [Delete](#)

SonarQube way [Update](#) [Delete](#)

SonarQube way [Default](#)

Conditions

Only project measures are checked against thresholds. Sub-projects, directories and files are ignored. [More](#)

Metric	Over Leak Period	Operator	Warning	Error	
Coverage on New Code	Always	is less than ▾	<input type="text"/>	<input type="text" value="80"/>	Update Delete
Maintainability Rating on New Code	Always	is worse than	<input type="text"/>	<input type="text" value="A X ▾"/>	Update Delete
Reliability Rating on New Code	Always	is worse than	<input type="text"/>	<input type="text" value="A X ▾"/>	Update Delete
Security Rating on New Code	Always	is worse than	<input type="text"/>	<input type="text" value="A X ▾"/>	Update Delete

▾

Plan

- Qualité du code
- Présentation de SonarQube
- Installation
- Configuration
- **Utilisation**

- Installation du scanner de sources

- Télécharger SonarQube Scanner

- Configurer le fichier de configuration `conf/sonar-scanner.properties`

```
sonar.host.url=http://localhost:7223
sonar.sourceEncoding=UTF-8
```

- Ajout du chemin du répertoire *bin* de `sonar-scanner` au `PATH`

```
export PATH=$PATH:chemin vers sonar scanner
```

- Création d'un fichier `sonar-project.properties` dans le projet

```
# cle unique
sonar.projectKey=IBDM:aliquot
# nom et version du projet affichés par l'interface graphique
sonar.projectName=Aliquot
sonar.projectVersion=1.0
# chemin relatif des fichiers sources, séparateur de chemins = virgule
sonar.sources=src/AppBundle,web/js
# encodage des fichiers sources
sonar.sourceEncoding=UTF-8
```

- Lancer `sonar-scanner` depuis le répertoire du projet

Visualiser les résultats dans l'interface web

Démo

Hudson

Quality Gate: **Failed**

E
Reliability

E
Security

C
Maintainability

13.0%
Coverage

2.7%
Duplications

L 122k
Java, JavaScript

Tableau de bord

Hudson

Page du projet

19 mai 2016 22:51 Version 3.3.4

Issues Measures Code

Quality Gate **Failed**

12 New Bugs
is greater than 0

35 New Vulnerabilities
is greater than 0

Bugs & Vulnerabilities

628 **E**
Bugs

155 **E**
Vulnerabilities

Leak Period: last 30 days
started il y a 9 mois

12 New Bugs 35 New Vulnerabilities

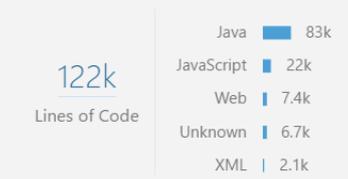
Code Smells

1kd **C**
Debt

42k
Code Smells

127d New Debt 5.8k New Code Smells

Hudson Continuous Integration Server



Quality Gate (Default) [SonarQube way](#)

- Quality Profiles (css) [Sonar way](#)
- (Java) [Sonar way](#)
- (JavaScript) [Sonar way](#)
- (Web) [Sonar way](#)
- (XML) [Sonar way](#)

sonarlint = détection directement dans le code



```
170 public void setProduct(int id, String designation, SubCategory subCategory, Integer warningPeriod, double minQuantity,
171 double price, boolean visibility, String picture, double conditioning) {
172     setProductTd(id);

```

Console SonarLint On-The-Fly SonarLint Rule Description

7 items

Date	Description	Resource
	Remove this empty statement.	Product.java
	Replace the type specification in this constructor call with the diamond operator ("<>").	Product.java
	Replace the type specification in this constructor call with the diamond operator ("<>").	Product.java
	Method has 9 parameters, which is greater than 7 authorized.	Product.java
	This block of commented-out lines of code should be removed.	Product.java
	Add a nested comment explaining why this method is empty, throw an UnsupportedOperationException or c...	Product.java
	Refactor this method to reduce its Cognitive Complexity from 29 to the 15 allowed.	Product.java

```
170 public void setProduct(int id, String designation, SubCategory subCategory, Integer warningPeriod, double minQuantity
171 double price, boolean visibility, String picture, double conditioning) {
172     setProductTd(id);

```

Console SonarLint On-The-Fly SonarLint Rule Description

Methods should not have too many parameters (squid:S00107)

A long parameter list can indicate that a new structure should be created to wrap the numerous parameters or that the function is doing too many things.

SonarLint Analyze changed files Bind to a SonarQube project...

- Intégrer la qualité au plus tôt dans le développement
 - ▣ Configurer les IDE pour respecter un standard de codage
 - ▣ Utiliser SonarLint
- Créer un profil qualité adapté
 - ▣ Choisir les règles
 - ▣ Paramétrer les seuils et les niveaux de criticité
- Forcer un développeur à respecter strictement les seuils peut mener à un comportement contre-productif
 - ▣ Taux de documentation => ajout de lignes de commentaires inutiles
 - ▣ Taille maximale d'une classe => code découpé sans logique
 - ▣ Complexité cyclomatique d'une fonction => code déplacé sans logique

- SonarQube <https://docs.sonarqube.org>
- SonarLint <http://www.sonarlint.org>
- Plug-in <https://docs.sonarqube.org/display/PLUG/Plugin+Library>
- Inspection continue
 <https://www.sonarsource.com/resources/white-papers/continuous-inspection.html>
- <https://blog.sonarsource.com/cognitive-complexity-because-testability-understandability/>

Merci

36

Questions ?