

Suivi de la qualité  
Du  
Projet Easy Language

## Sommaire

1- Langage de programmation EasyLanguage.....	3
1. Fiabilité : .....	4
2. Sécurité : .....	5
3. Maintenabilité: .....	5
4. Duplication : .....	6
5. Taille : .....	7
6. Complexité : .....	7
7. Documentation : .....	8
2- Site Web ( Easy Language ) .....	8
1. Fiabilité : .....	9
2. Sécurité : .....	10
3. Maintenabilité: .....	11
4. Duplication : .....	12
5. Taille : .....	13
6. Documentation : .....	14

# Projet Easy Language

---

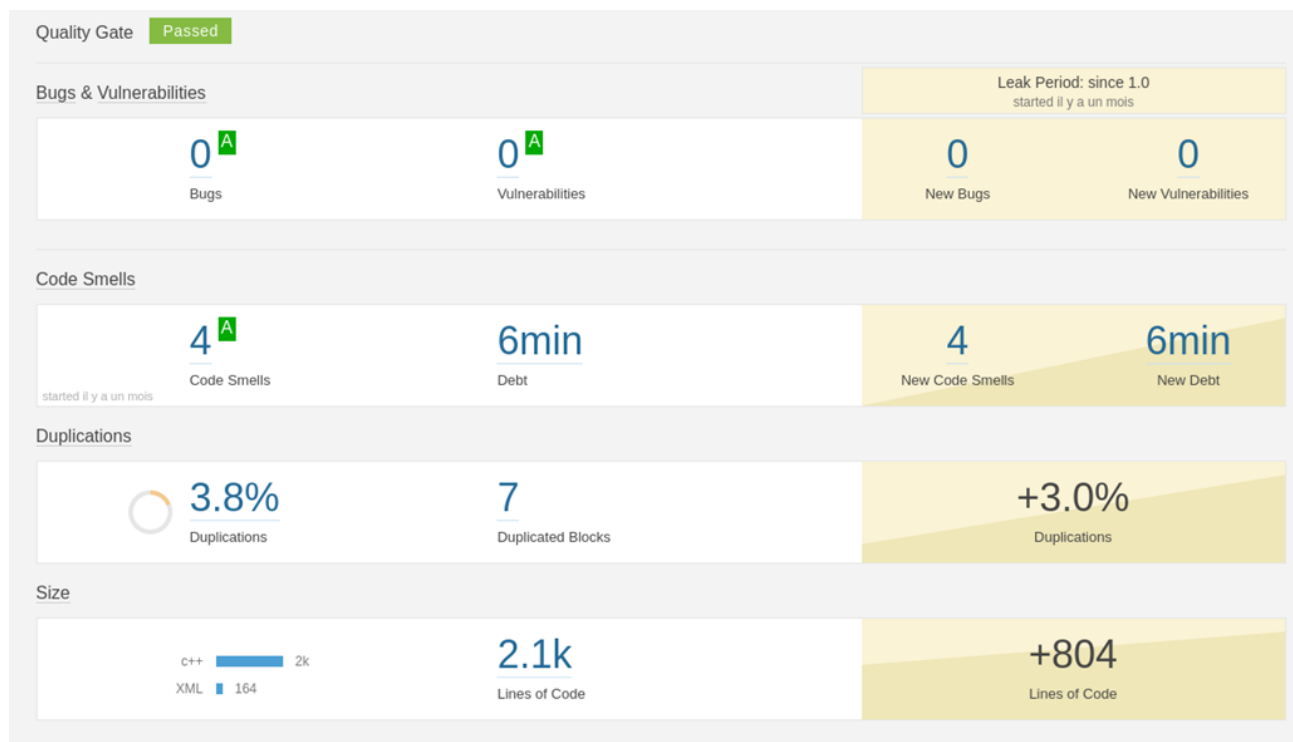
Afin de pouvoir analyser la qualité de ce projet, nous avons utilisé l'outil SonarQube 6.0 (sur le serveur de la faculté) ainsi que SonarQube 5.6.4 sur ma machine Local (vu que le plugin c++ gratuit n'est pas compatible avec la version 6.0) et au niveau du serveur de l'université nous avons prévu de tester seulement le code développé par nous même en liaison avec Jenkins ( le code source des Frameworks n'est pas pris en considération ).

Ce projet est composé de deux parties, la première représente le nouveau langage de programmation EasyLanguage et la deuxième représente le site Web de ce langage.

PROJECTS						
QG	NAME	VERSION	LOC	BUGS	VULNERABILITIES	CODE SMELLS
★	📁 Easy Language	2.0	2 126	0	0	4
★	📁 Easy Language ( Site Web )	2.0	44 340	2 968	13	29 540

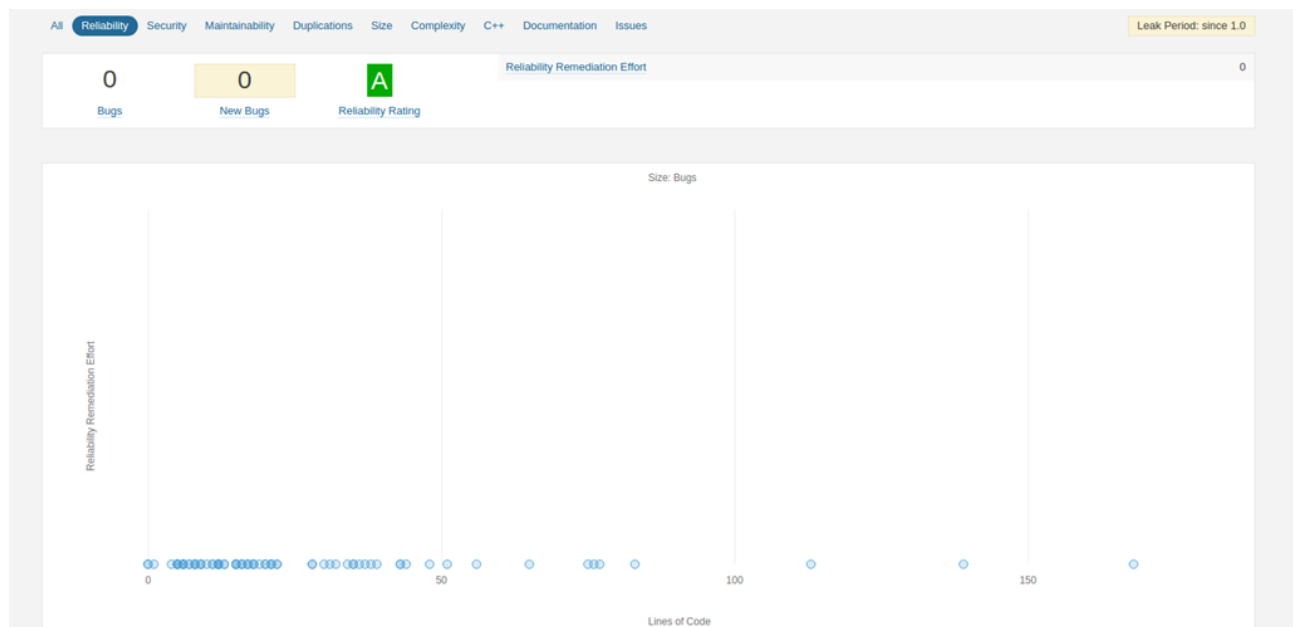
## 1- Langage de programmation EasyLanguage

La capture ci-dessous permet de donner une idée générale sur la qualité de notre langage de programmation (version 2.0).



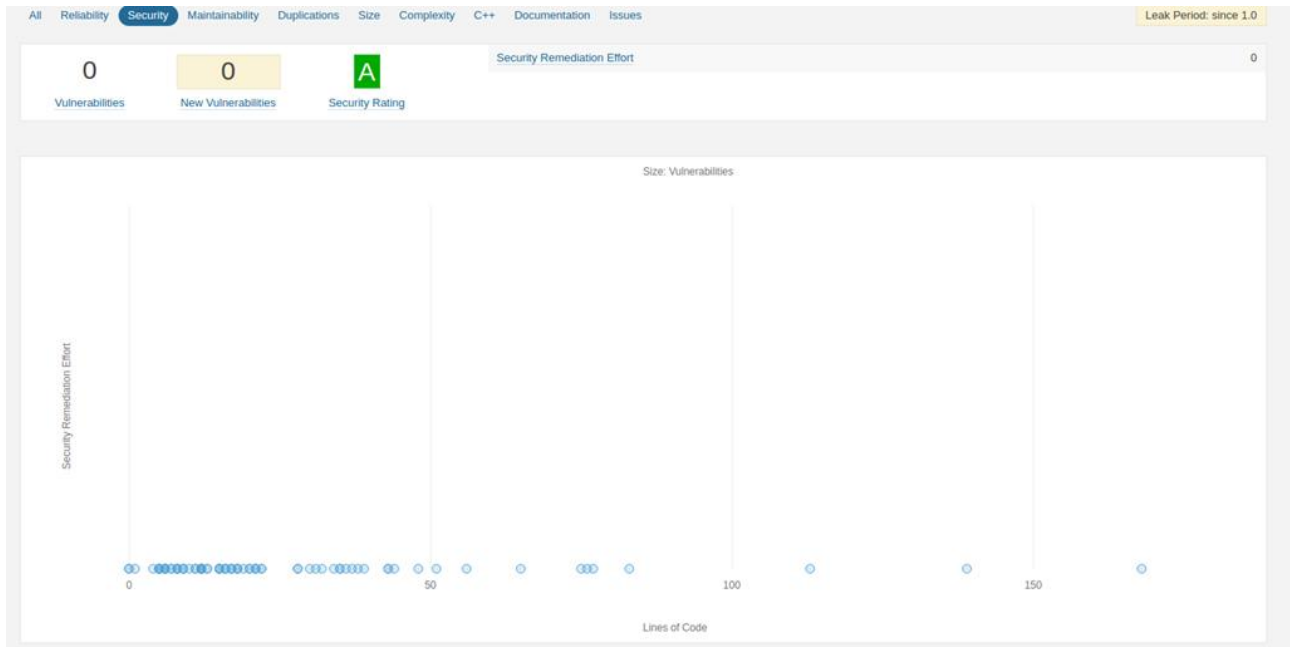
## 1. Fiabilité :

Dans le langage EasyLanguage, la note de fiabilité est A, car nous n'avons pas de bugs au niveau de c++.



## 2. Sécurité :

Comme vous pouvez remarquer sur la capture ci-dessous la note de sécurité est A donc nous n'avons pas de vulnérabilité

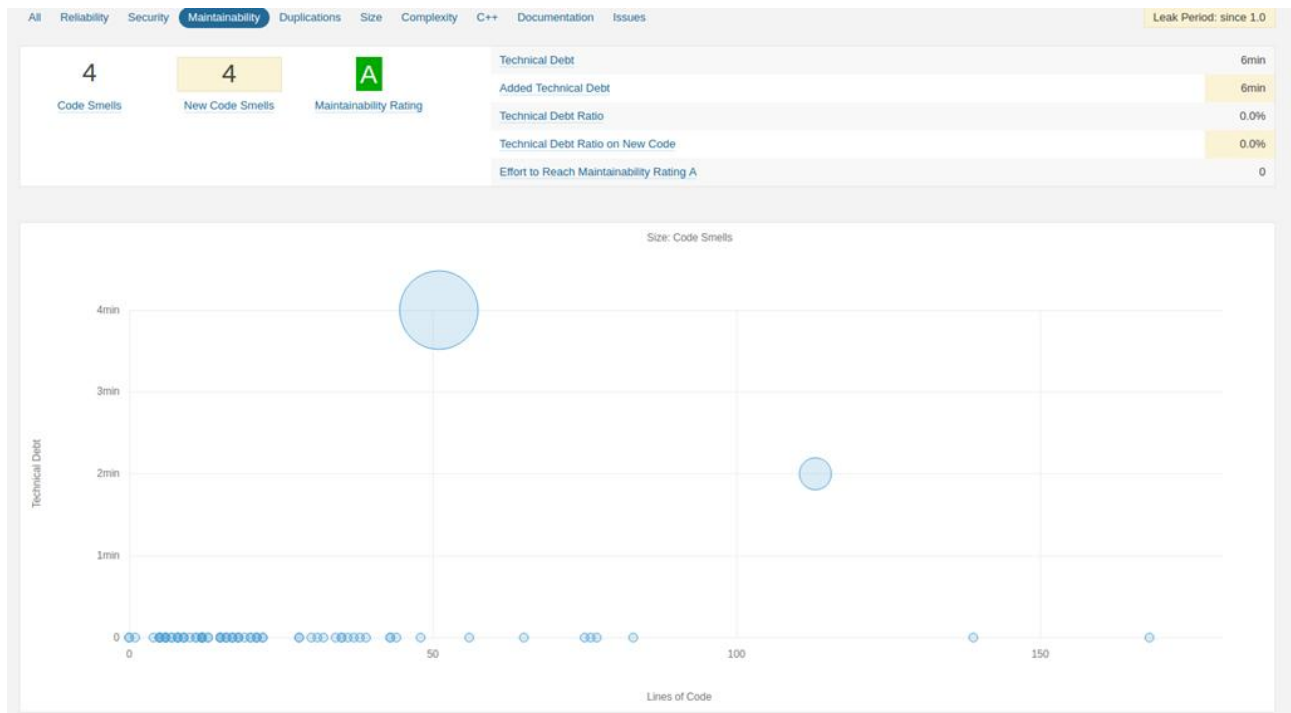


## 3. Maintenabilité:

En génie logiciel, la mauvaise pratique de conception logicielle conduit à l'apparition de défauts (code smell) qui complexifient le code source et la maintenance et l'évolutivité.

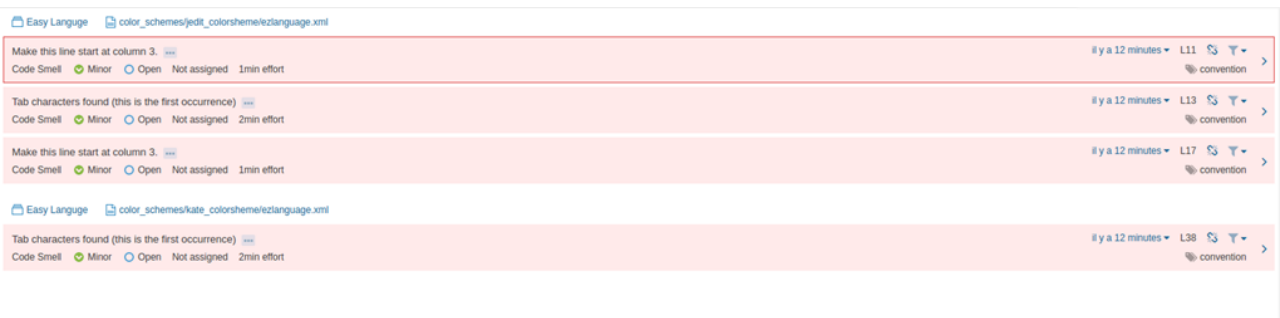
Dans notre projet, nous avons certains code smells (voir les deux captures ci-dessous), mais afin de les corriger, il est nécessaire de procéder à un réusinage du code source.

Malgré ça, la note de maintenabilité est toujours A, car le nombre de code smells (4) est ignorable.



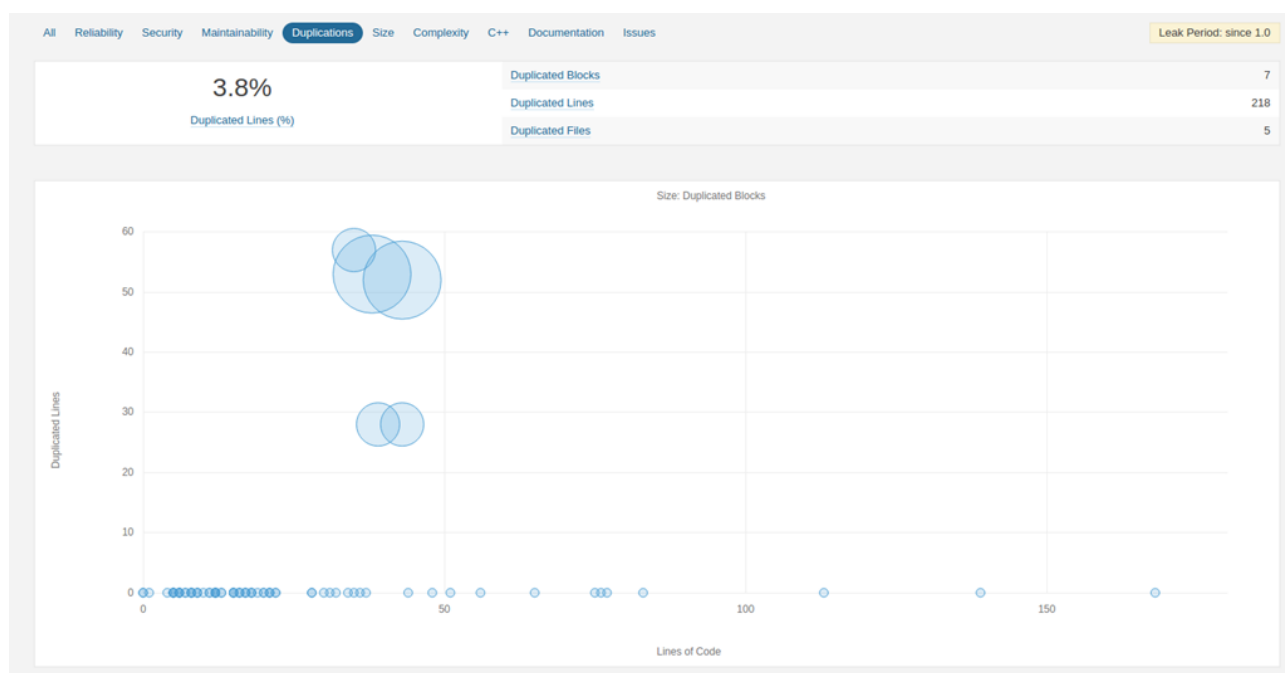
La capture ci-dessous cite la liste des codes smells.

Comme vous pouvez remarquer, pour chaque défauts, nous avons besoin seulement d'une ou deux minutes maximum pour le corriger.



## 4. Duplication :

Le pourcentage de duplication de code est de 3,8 %, y compris duplication de blocs, de lignes et de fichiers.



Ci-dessous la liste des fichiers avec le nombre et le pourcentage de lignes dupliquées, triés par ordre descendant.

All Measures / Duplications Measures

Duplicated Lines (%)

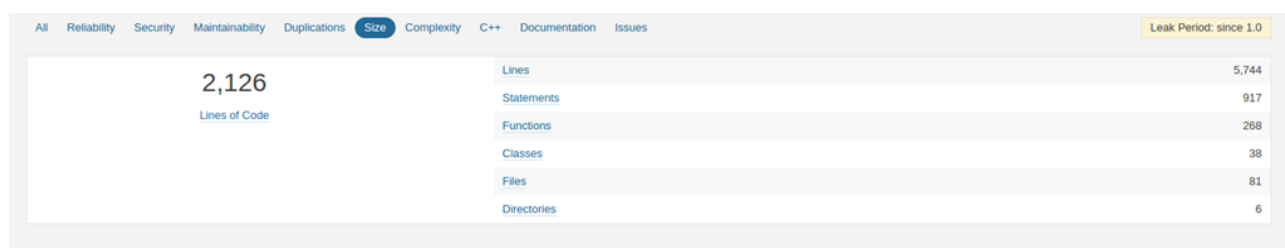
3.8%

List Tree Treemap History

	Duplicated Lines (%)	Duplicated Lines
src/modules/DeclarationList.cpp	70.7%	53
src/modules/DeclarationVector.cpp	67.1%	57
src/modules/DeclarationSet.cpp	65.8%	52
src/modules/DeclarationProcedure.cpp	50.0%	28
src/modules/DeclarationFunction.cpp	45.2%	28

## 5. Taille :

Notre projet contient 2126 lignes de codes, 917 instructions, 268 fonctions, 38 classes, 81 fichiers et 6 répertoires.



## 6. Complexité :

La capture d'écran suivante représente le niveau de complexité de l'application, notamment au niveau des fonctions, des fichiers et des classes.

All	Reliability	Security	Maintainability	Duplications	Size	Complexity	C++	Documentation	Issues	Leak Period: since 1.0
456							Complexity / Function		1.7	
Complexity							Complexity / File		5.8	
							Complexity / Class		10.0	

Ci-dessous la liste des fichiers contenant les morceaux de code compliqués, par ordre descendant.

All Measures / Complexity Measures

Complexity

456

List Tree History

src/modules/Operator.cpp	40
src/main.cpp	40
src/modules/Class.cpp	39
src/hash_table/Function.cpp	25
src/hash_table/ScopeHashTable.cpp	24
src/addons/String_addon.cpp	23
src/hash_table/HashTable.h	22
src/modules/It.cpp	15
src/modules/DeclarationMap.cpp	13
src/modules/DeclarationContainer.cpp	12
src/hash_table/ClassHashTable.cpp	12
src/hash_table/Variable.cpp	11
src/modules/Node.cpp	11
src/modules/ArrayDeclaration.cpp	11
src/modules/DeclarationSet.cpp	9
src/modules/Declaration.cpp	9
src/hash_table/ClassDeclaration.cpp	9
src/modules/ArrayAccess.cpp	9
src/modules/DeclarationVector.cpp	8
src/modules/DeclarationList.cpp	8
src/modules/DeclarationFunction.h	8
src/hash_table/Variable.h	6
src/modules/Iterative_instruction.cpp	6
src/modules/Holder.cpp	6
src/hash_table/HashElement.cpp	6

## 7. Documentation :

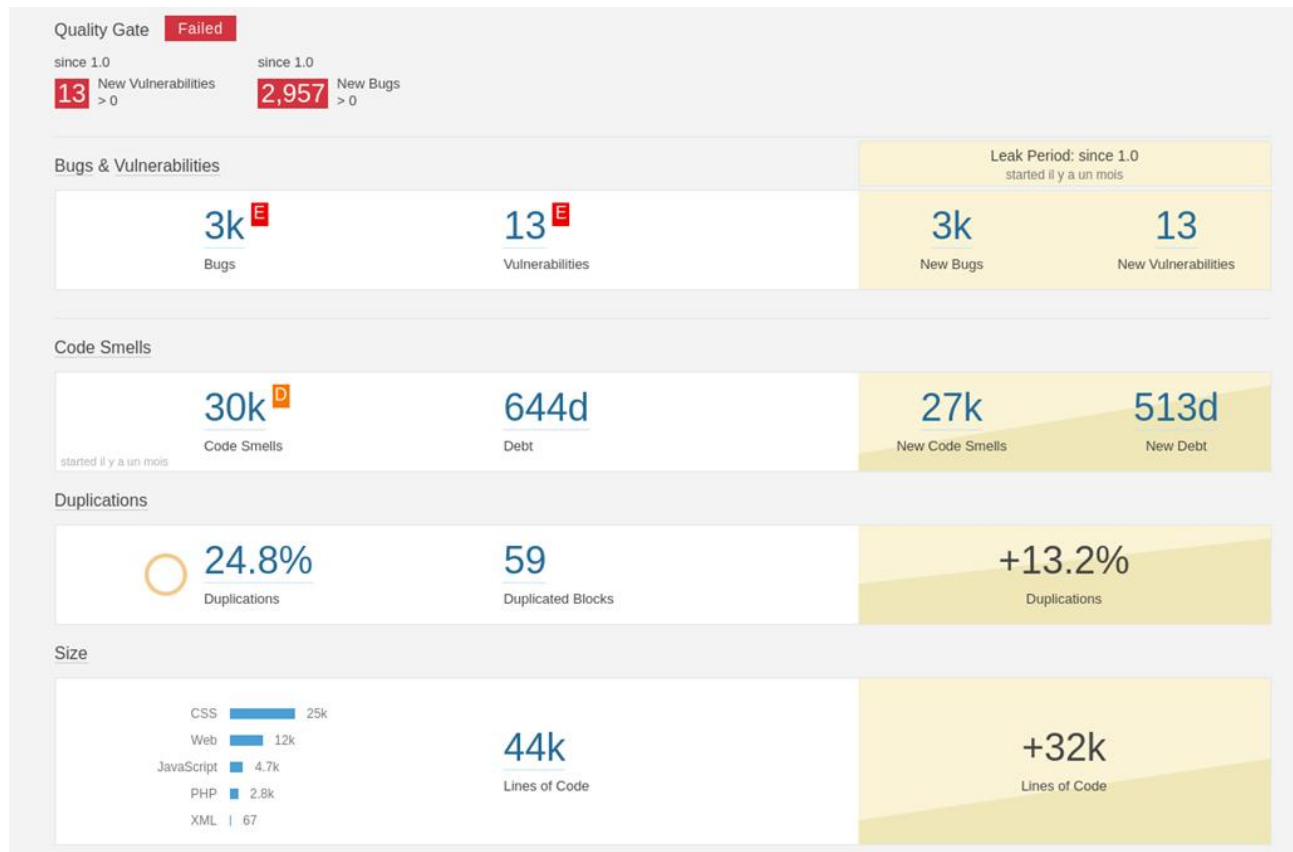
Afin de faciliter la compréhension du code, la documentation du code est une tâche indispensable. Le niveau de documentation de notre projet est acceptable mais il faut penser à documenter davantage notre code au fur et à mesure du développement.

## 2- Site Web ( Easy Language )



La capture ci-dessous permet de donner une idée générale sur la qualité du site web de notre langage de programmation (version 2.0).

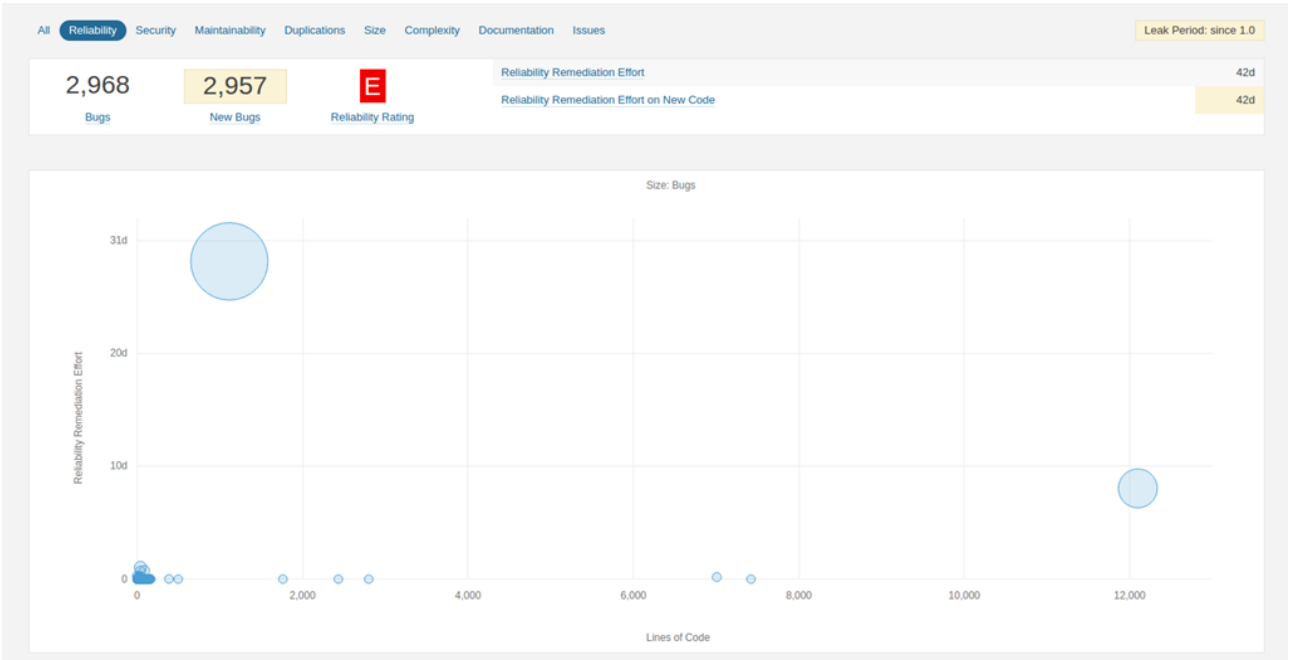
Dans notre analyse de qualité, nous avons prévu d'analyser même le code source des deux frameworks « Symfony » et « bootstrap », vu qu'ils font partie de notre siteweb.



## 1. Fiabilité :

Dans le site web du langage EasyLangue, la note de fiabilité est E, car nous avons vraiment beaucoup de bugs.

Le temps nécessaire pour corriger ces bugs est estimé à 42 jours.



Ci-dessous la liste des fichiers contenant les bugs, et on peut déduire que 90% des bugs sont des bugs de JavaScript et de css.

All Measures / Reliability Measures

Reliability Rating

E

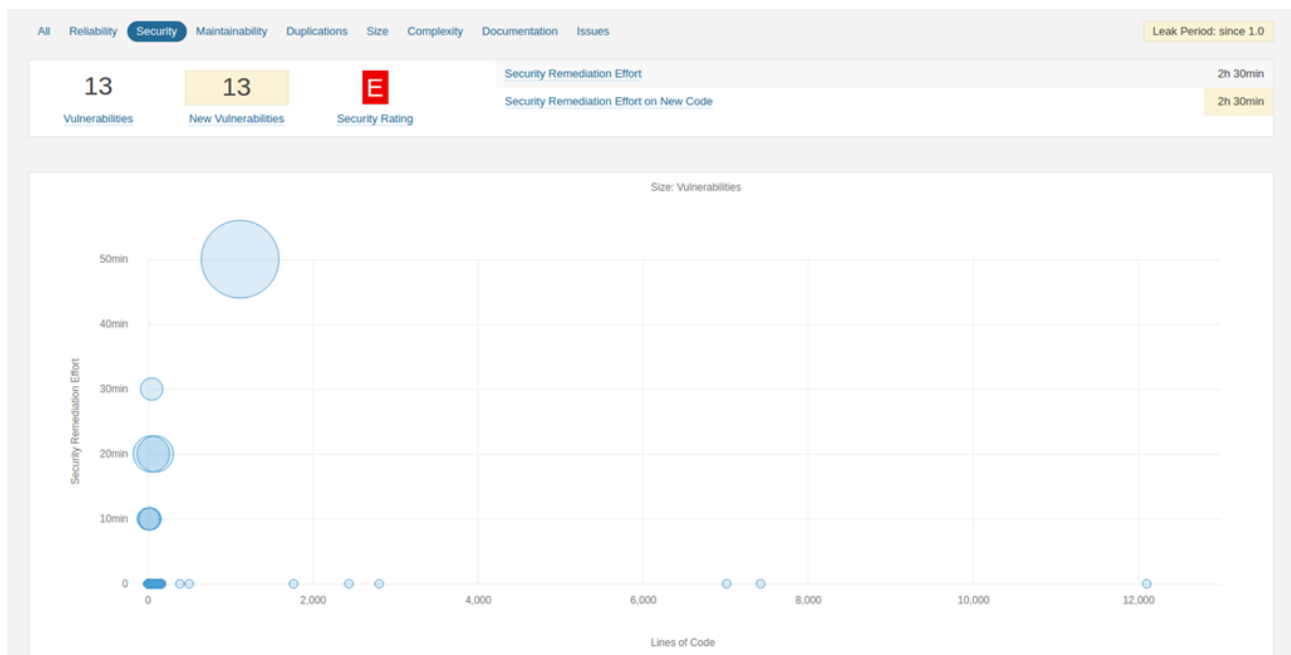
List Tree Treemap History

src/AppBundle/Resources/public/ckeditor/ckeditor.js	E
src/AppBundle/Resources/public/ckeditor/skins/moono-lisa/editor_iequirks.css	D
src/AppBundle/Resources/public/ckeditor/skins/moono-lisa/editor_ie8.css	D
src/AppBundle/Resources/public/ckeditor/skins/moono-lisa/editor_ie.css	D
src/AppBundle/Resources/public/ckeditor/skins/moono-lisa/editor_gecko.css	D
src/AppBundle/Resources/public/ckeditor/skins/moono-lisa/editor.css	D
src/AppBundle/Resources/public/ckeditor/skins/moono-lisa/dialog_iequirks.css	D
src/AppBundle/Resources/public/ckeditor/skins/moono-lisa/dialog_ie8.css	D
src/AppBundle/Resources/public/ckeditor/skins/moono-lisa/dialog_ie.css	D
src/AppBundle/Resources/public/ckeditor/plugins/pastefromword/filter/default.js	D
src/AppBundle/Resources/public/css/external/custom.css	D
src/AppBundle/Resources/public/css/external/bootstrap.css	D
src/AppBundle/Resources/public/ckeditor/plugins/wsc/dialogs/wsc_ie.js	C
src/AppBundle/Resources/public/ckeditor/plugins/wsc/dialogs/wsc.js	C
src/AppBundle/Resources/public/ckeditor/plugins/wsc/dialogs/tmpFrameset.html	C
src/AppBundle/Resources/public/ckeditor/plugins/tabletools/dialogs/tableCell.js	C
src/AppBundle/Resources/public/ckeditor/plugins/table/dialogs/table.js	C
src/AppBundle/Resources/public/ckeditor/plugins/specialchar/dialogs/specialchar.js	C
src/AppBundle/Resources/public/ckeditor/plugins/indentblock/plugin.js	C
src/AppBundle/Resources/public/ckeditor/plugins/clipboard/dialogs/paste.js	C
src/AppBundle/Resources/public/ckeditor/plugins/scayt/dialogs/options.js	C
src/AppBundle/Resources/public/ckeditor/plugins/link/dialogs/link.js	C
src/AppBundle/Resources/public/js/jquery.tubular.1.0.js	C
src/AppBundle/Resources/public/ckeditor/adapters/jquery.js	C

## 2. Sécurité :

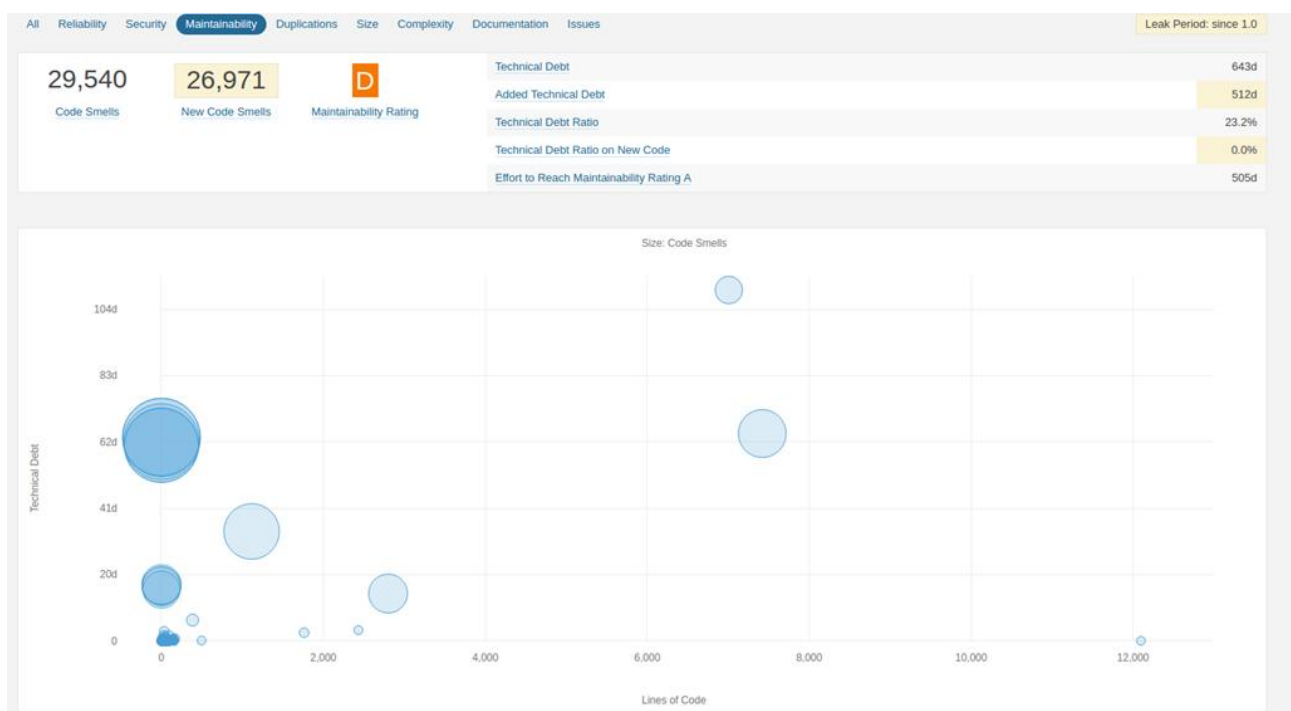
Comme vous pouvez remarquer sur la capture ci-dessous la note de sécurité est toujours E vu le nombre de vulnérabilité que nous avons.

Afin de corriger ces 13 vulnérabilités, il ne faut minimum 2 heures et demi.



### 3. Maintenabilité:

Le niveau de maintenabilité est vraiment catastrophique, ce qui explique la note obtenue « D », C'est vrai que nous avons fait le maximum en corrigeant presque 2569 code smells pendant une période de 20 jours. Mais cela n'est pas suffisant parce que nous avons 26971 code smells à corriger et il ne faut minimum 643 jours ce qui me semble impossible avec les contraintes que nous avons (50 heures par personne)

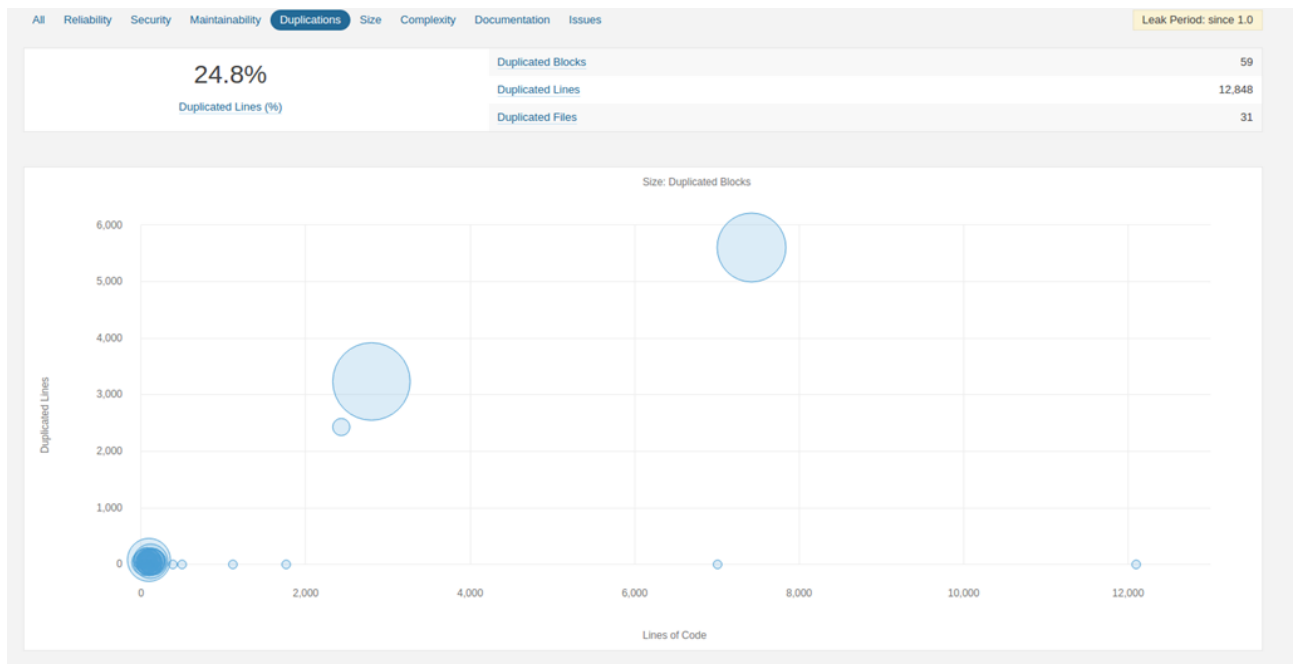


Voici quelques exemples de code smells ( crituque et majeur ) qu'il fallait corriger.

Easy Language ( Site Web ) app/SymfonyRequirements.php		1 / 32,521	Reload	New Search
The Cyclomatic Complexity of this function "___construct" is 37 which is greater than 20 authorized.	Code Smell Critical Open Not assigned 27min effort	il y a un mois L388	brain-overload	>
This function "___construct" has 365 lines, which is greater than the 150 lines authorized. Split it into smaller functions.	Code Smell Major Open Not assigned 20min effort	il y a un mois L388	brain-overload	>
Define a constant instead of duplicating this literal "5.4.0" 3 times.	Code Smell Critical Open Not assigned 8min effort	il y a un mois L500	design	>
Define a constant instead of duplicating this literal "\$cfgValue" 4 times.	Code Smell Critical Open Not assigned 10min effort	il y a un mois L520	design	>
Reduce the number of conditional operators (11) used in the expression (maximum allowed 3).	Code Smell Critical Open Not assigned 13min effort	il y a un mois L704	brain-overload	>
Reduce the number of returns of this function 4, down to the maximum allowed 3.	Code Smell Major Open Not assigned 20min effort	il y a un mois L762	brain-overload	>
Easy Language ( Site Web ) app/check.php				
Define a constant instead of duplicating this literal "green" 4 times.	Code Smell Critical Open Not assigned 10min effort	il y a un mois L13	design	>
Define a constant instead of duplicating this literal "yellow" 5 times.	Code Smell Critical Open Not assigned 12min effort	il y a un mois L15	design	>
Define a constant instead of duplicating this literal "error" 5 times.	Code Smell Critical Open Not assigned 12min effort	il y a un mois L26	design	>
Define a constant instead of duplicating this literal "warning" 3 times.	Code Smell Critical Open Not assigned 8min effort	il y a un mois L37	design	>
Define a constant instead of duplicating this literal "title" 4 times.	Code Smell Critical Open Not assigned 10min effort	il y a un mois L64	design	>
Immediately return this expression instead of assigning it to the temporary variable "\$errorMessage".	Code Smell Minor Open Not assigned 7min effort	il y a un mois L83	clutter	>

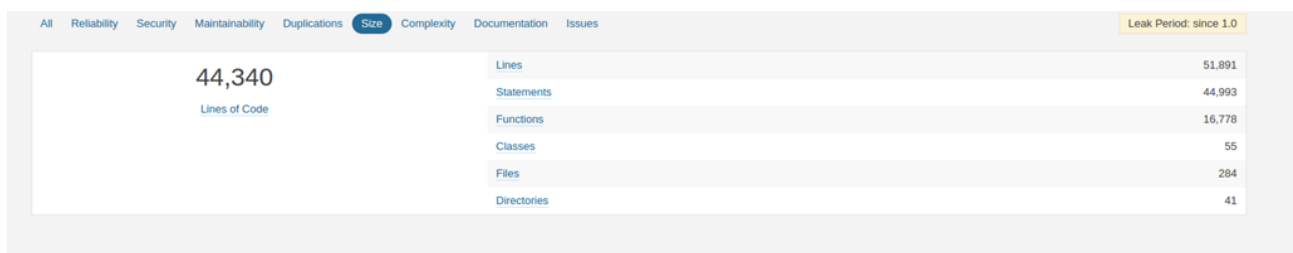
## 4. Duplication :

Le pourcentage de duplication de code est de 24,8 %, y compris duplication de blocs, de lignes et de fichiers.



## 5. Taille :

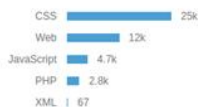
Notre projet contient 51891 lignes de codes, 44993 instructions, 16778 fonctions, 55 classes, 284 fichiers et 41 répertoires.



En regardant la capture ci-dessous, nous pouvons remarquer que notre site est basé sur PHP, XML, JavaScript, css et html.

## Lines of Code

44,340



List Tree History

src/AppBundle/Resources/public/fonts/icomoon/icomoon/demo.html	12,098
src/AppBundle/Resources/public/css/style.css	7,421
src/AppBundle/Resources/public/css/external/bootstrap.css	7,008
src/AppBundle/Resources/public/css/external/animate.css	2,800
src/AppBundle/Resources/public/fonts/icomoon/icomoon/style.css	2,433
src/AppBundle/Resources/public/bootstrapdatepicker/bootstrap-datepicker.js	1,763
src/AppBundle/Resources/public/ckeditor/ckeditor.js	1,115
app/SymfonyRequirements.php	497
src/AppBundle/Resources/public/css/external/custom.css	385
src/AppBundle/Resources/public/ckeditor/contents.css	165
src/AppBundle/Resources/public/js/external/ezlanguage.highlight.js	159
src/AppBundle/Resources/public/ckeditor/plugins/indentblock/plugin.js	153
src/AppBundle/Resources/public/fonts/icomoon/icomoon/demo/demo.css	153
src/AppBundle/Resources/public/ckeditor/plugins/codesnippet/lib/highlight/styles/tailecasts.css	148
src/AppBundle/Controller/EZFunctionController.php	148
src/AppBundle/Resources/public/ckeditor/plugins/codesnippet/lib/highlight/styles/arta.css	145
src/AppBundle/Resources/public/ckeditor/plugins/codesnippet/lib/highlight/styles/default.css	138
src/AppBundle/Resources/public/ckeditor/plugins/codesnippet/lib/highlight/styles/xcode.css	133
src/AppBundle/Entity/EZFunction.php	133
src/AppBundle/Resources/public/ckeditor/plugins/codesnippet/lib/highlight/styles/obsidian.css	132
src/AppBundle/Resources/public/ckeditor/plugins/codesnippet/lib/highlight/styles/sunburst.css	130
src/AppBundle/Resources/public/ckeditor/plugins/codesnippet/lib/highlight/styles/monokai_sublime.css	130

## 6. Documentation :

Après avoir analysé le site web, nous pouvons dire que notre site web n'est pas bien documenté, ce qui sera vraiment gênant pour ceux qui visent une maintenance de notre site web par la suite. Et même pour ceux qui cherche à comprendre les fonctionnalités de chaque partie de code dans notre site web.

