



## SOFTWARE ANALYTICS REPORT

Application: amadio5/SensorTool

Analysis label: no label

Analysis date: Jan 29, 2018, 6:37 PM

Report date: Jan 29, 2018, 6:39 PM



## TABLE OF CONTENTS

Find in  
this report

Introduction: methodology

Application & analysis information

Risk index

Quality

Reparation efforts

Main metric values

Quality distribution in files

Metric distribution in files

10 top repair first defects

About us



## SOFTWARE EVALUATION METHODOLOGY

checkKING Quality Model for Software (CQM) is Kiuwan methodology for evaluating the internal quality of a software product. **CQM is ISO-25000 based.** It defines us the internal quality scope and characteristics. CQM simplifies ISO 25000 focusing on internal quality. CQM proposes indicators for the following software characteristics:

**Security.** The capability of the software product to protect information and data so that unauthorised persons or systems cannot read or modify them and authorised persons or systems are not denied access to them.

**Reliability.** The capability of the software product to maintain a specified level of performance.

**Efficiency.** The capability of the software product to provide appropriate performance relative to the amount of resources used under stated conditions.

**Maintainability.** The capability of the software product to be modified. Modifications may include corrections, improvements or adaptability of the software to changes in environment and in requirements and functional specifications.

**Portability.** The capability of the software product to be transferred from one environment to another.

CQM indicators are normalised to represent these regions:

**0-30 region.** The characteristic pointed to by the indicator is in the RED zone. Improvements are needed.

11



**30-70 region.** Represented by YELLOW and means that you have to keep your mind on this indicator. Your next moves will depend on your requirements.

57



**70-100 region.** The GREEN zone. This is the zone where all indicators must be. No critical defects founded.

87



This normalisation allows the comparison of different characteristics between them; this means that you can say if the software is more maintainable than it is efficient or reliable. You are going to compare different version of the same application over time because the meaning of the indicator does not change. You can even compare two different technology applications.



## APPLICATION & ANALYSIS INFORMATION

Kiuwan've analyzed for you a set of source files that is called application. In order to put in context this analysis and your results, kiuwan is going to give you some statistics:

### Notes:

- unparsed files are the source code files of your application that kiuwan engine couldn't read during the analysis process.
- unrecognized files are the ones that were in your analyzed path but they aren't source code or they contains code in languages that Kiuwan doesn't support yet.

### Analysis Info

label	no label
date	2018/01/29 06:37
ordered by	amadio5
encoding	UTF-8
analyzed path	archive.zip
languages found	javascript, php
analyzed files	191
unparsed files	0
unrecognized files	6

### Application Info

name	amadio5/SensorTool
Bussiness value	critical
Times analyzed	1

### Model Info

model name	CQM
model version	v1.2.13
engine version	-
active rules	245
active metrics	123



## RISK INDEX

Risk index represents the potential problems that you are assuming for not paying attention to the quality of your source code. So far as you are (measured in effort) to get an acceptable quality level.

It is a number that concentrates all the evidence found in the source code of your application. It has been used your global indicator and the effort that you need to spend to reach the quality level set as goal for you. So if you have poor quality, but if the effort needed to get better is low you are not assuming a high risk in this application because you are going to repair your problems easily. But if your effort needed to get better is very high your risk index will be high too.

Pay attention to risk index evolution in time.

**RISK INDEX**

**0**



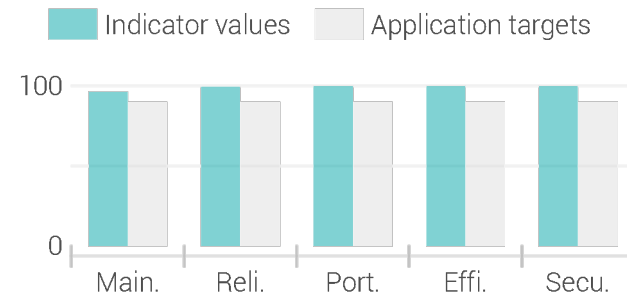
## MAIN INDICATORS

You get your global indicator at the right and a breakdown in software characteristics.

Pay attention to the **target values** that have been set when you configure your application. It is important that you know if your quality is better or not that these values and you modify them according to your requirements.

The quality evolution is another important point. **Are you improving?**

GLOBAL  
INDICATOR  
99.1



### Characteristic Indicator

Maintainability	96.43
Reliability	99.68
Portability	100
Efficiency	100
Security	100

### App. target

90
90
90
90
90

### Global indicator evolution



## REPARATION EFFORTS

Now you know your quality level, you'll want to know how much it will cost to reach your goal.

We've calculated for you the minimum set of defects will be corrected to achieve it. Here's what you need to invest. You can configure (for accuracy) the effort needed to correct each defect type.

**EFFORT TO  
TARGET**  
**0h**

■ Maintainability  
■ Reliability  
■ Efficiency

### Characteristic

### Effort to target

Maintainability	0 h
Reliability	0 h
Efficiency	0 h

### Effort to target evolution



## MAIN METRIC VALUES

We've computed some metrics of your source code. Below are shown the most important ones.

**Lines of code.** Excluding commented lines and blank lines.

**Function points.** Functional size calculated by backfiring strategy.

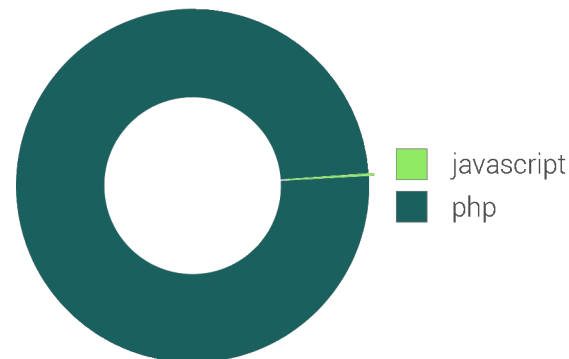
**Avg complexity.** Average of each function (or method).

**Dup code.** The ratio of the duplicated code.

Lines of code	8,313
Function points	83.68
Average complexity	1.57
Duplicated code	3.55%



### Lines of code by technology







# INDICATOR DISTRIBUTION IN FILES

It's important to know if your issues are distributed homogeneously through files. In the chart we've used **quintiles** dividing the file indicator value in five equal sized subsets, and then we tell you how many files fall in each subset.

In the table, you have a list of the files (the worst ones) that have reduce their indicator from previous analysis. If this is your first analysis, you will get the files with the lower indicator values.

File name	Indicators	Previous	Delta
.../DetectionController.php	71.82	-	-
.../passwords.php	76.74	-	-
.../DetectionTenantScope.php	76.81	-	-
.../EnterpriseTenantScope.php	76.81	-	-
.../MessageTenantScope.php	76.81	-	-
.../SensorTenantScope.php	76.81	-	-
.../SiteTenantScope.php	76.81	-	-
.../DemoSeeder.php	77.29	-	-
.../MakeSendorData.php	78.92	-	-
.../2018_01_17_172639_create_roles_seeder.php	87.76	-	-
.../EnterpriseCrudController.php	89.29	-	-
.../Sensor.php	91.64	-	-
.../auth.php	91.66	-	-
.../2018_01_18_160010_create_sensors_table.php	92.28	-	-
.../SensorCrudController.php	93.4	-	-
.../2018_01_25_221100_add_guest_role_to_roles_table.php	94.16	-	-
.../SensorCatalogCrudController.php	94.92	-	-
.../SiteCrudController.php	95.42	-	-
.../DetectionCrudController.php	95.6	-	-
.../channels.php	96.61	-	-

## Global indicator





## METRIC DISTRIBUTION IN FILES

It's important to know if your metrics are distributed homogeneously through files. We've used **quintiles** dividing the file level metric values in five equal sized subsets, and then we tell you how many files fall in each subset.

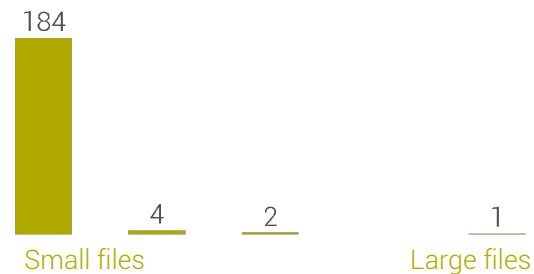
We give you the most three popular metrics:

**Size** as lines of code of each file.

**Complexity** as average cyclomatic complexity of each function per file.

**Dup code** as the ratio of the duplicated code of each file.

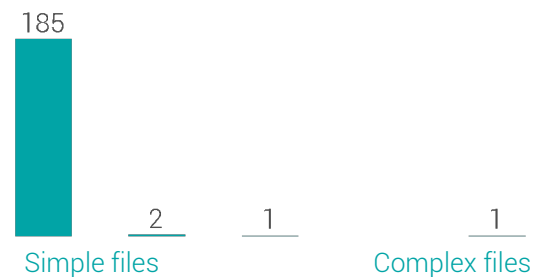
### Size



### Duplicated code



### Complexity





## TOP 10 REPAIR FIRST DEFECTS

Here you have a list of top 10 defect types that you have in your application. These defects are the ones that once eliminated give more benefit by unit time of effort. If you want to start fixing quality issues, you must start with this to maximize your time.

Rank	Defects	Files	Rule name	Lang.	Characteristic	Priority	Effort
1	6	6	Avoid unused function parameters	php	Reliability	2	18m
2	1	1	Avoid unused local variables	php	Reliability	2	3m
3	1	1	Use default clause at the end of the switch statement	php	Maintainability	2	6m
4	2	2	Duplicated code: medium block	php	Maintainability	2	1h
5	1	1	Function call return value ignored	php	Reliability	2	4h
6	23	15	Do not use if (\$var) to check if a variable is initialized	php	Reliability	3	2h 18
7	201	49	Use simple quotes to demarcate string literals, except when containing apostrophes, escaped chars or variable substitutions	php	Maintainability	3	20h
8	2	2	Variable initialization	php	Reliability	3	12m
9	2	2	Avoid concatenations in echo statement	php	Efficiency	3	12m
10	4	2	Avoid unnecessary string replacements in loops	php	Efficiency	3	2h

If you want a complete list of defects to repair in order to reach a quality target or to spend a bag of budgeted hours you can take an **action plan** report from kiuwan.com in **what if** function.



# THANK YOU!

## About us

Kiuwan is a SaaS Software Analytics platform specialized in security, quality assurance and application inventory management products.

Kiuwan provides end-to-end Software Analytics to bring you objective data so you can make informed decisions regarding the cost, effort, activity, quality, security, maintainability, efficiency and dependencies of your applications. Kiuwan makes tough decisions easier and this translates in cost reduction, risk mitigation, objective measurement, software technical security certifications and outsourcing agreement management, to name but a few of the possibilities brought about by Kiuwan.

Kiuwan supports multiple application technologies, covering over 20 programming languages. Kiuwan proves to be of key importance when companies industrialize the Software Development Life Cycle, when they want to secure their applications from cyber threats, going as far as covering SLA compliance all within the most relevant IT frameworks and standards.

[contact@kiuwan.com](mailto:contact@kiuwan.com) | Partnership: [partners@kiuwan.com](mailto:partners@kiuwan.com) | +1 9045123050

© Optimyth Software Technologies 2017 – All rights reserved

