



**PHP project (group-2\_magento) analyzed  
using SonarQube  
2.0**

*2016-07-19*



# Table of contents

1. PHP project (group-2_magento) analyzed using SonarQube	Page 1
<hr/>	
1.1. Report Overview	
1.2. Issues Analysis	
1.3. Issues Details	

# 1. PHP project (group-2\_magento) analyzed using SonarQube

This chapter presents an overview of the project measures. This dashboard shows the most important measures related to project quality, and it provides a good starting point for identifying problems in source code.

## 1.1. Report Overview

### Static Analysis

#### Lines of code

**682,472**

N/A packages

7,444 classes

43,216 methods

14.9% duplicated lines

#### Comments

**35.0%**

367,118 comment lines

#### Complexity

**3.2**

18.3 /class

146,774 decision points

### Dynamic Analysis

#### Code Coverage

**N/A**

N/A tests

#### Test Success

**N/A**

N/A failures

N/A errors

### Coding Rules Issues

#### Technical Debt

**915d**

#### Issues

**54,090**

## 1.2. Issues Analysis

Most violated rules	
"\$this" should not be used in a static context	31
Method arguments with default value should be last	31
Credentials should not be hard-coded	22
Objects should not be created to be dropped immediately without being used	11

Code should not be dynamically injected and executed to prevent Eval Injection vulnerability	9
--	---

Most violated files	
Date.php	818
Date.php	814
Data.php	440
QueryLexer.php	341
mysql4-upgrade-1.5.9.9-1.6.0.0.php	265

Most complex files	
Date.php	1,226
Date.php	1,220
Mysql.php	615
Form.php	610
Client.php	584

Most duplicated files	
mysql4-upgrade-1.5.9.9-1.6.0.0.php	7,676
mysql4-upgrade-1.5.9.9-1.6.0.0.php	5,897
Date.php	4,926
Date.php	4,922
install-1.6.0.0.php	3,186

### 1.3. Issues Details

Rule	"\$this" should not be used in a static context	
File	Line	
PEAR.php	564, 563, 541, 551, 541, 563, 553, 541, 541, 552, 551, 553, 541, 551, 551, 541, 541, 564, 541, 552	

Rule	Method arguments with default value should be last	
File	Line	
Container.php	138, 169	
Price.php	632, 119, 43, 43, 74, 296, 178	
Productimg.php	177	
Locale.php	567	
Price.php	43	
Entity.php	192	
Abstract.php	200	
Quote.php	1558, 1601	
Relation.php	106	
Order.php	310	
CentinelClient.php	77	
Value.php	236	

Rule	Credentials should not be hard-coded	
File	Line	
Data.php	32, 39, 49, 48, 46, 47, 85, 79	
Collection.php	41	
Customer.php	68	
Ftp.php	60	
Instance.php	47	
Attribute.php	41, 37, 39, 40, 38	
Soap.php	85, 91	
DbAdapter.php	93	

Rule	Objects should not be created to be dropped immediately without being used	
File	Line	
V1.php	72	
Api.php	158	
Profile.php	651	
Fs.php	71, 72, 70, 73, 71, 70, 73, 72	

Rule	Code should not be dynamically injected and executed to prevent Eval Injection vulnerability	
------	--	--

File	Line
Data.php	751
Application.php	840
PEAR.php	917, 917
Client.php	54, 56
WSDL.php	789
Filesystem.php	447
PhpCode.php	59

Rule	Multiline blocks should be enclosed in curly braces	
File	Line	
Collection.php	345	
Object.php	104, 78	
Blob.php	377, 375, 379	

Rule	Related "if/else if" statements and "cases" in a "switch" should not have the same condition	
File	Line	
SalesController.php	411	
Info.php	564	
element.phtml	67	
DefaultValue.php	216	
QueryToken.php	217	

Rule	Identical expressions should not be used on both sides of a binary operator	
File	Line	
Tiff.php	389	

Rule	Static members should be referenced with "static::"	
File	Line	
Session.php	144, 89, 480, 413, 92, 332, 242, 543, 122, 494, 113, 437, 396, 81, 183, 630, 340, 92, 610, 87	



Rule	Sections of code should not be "commented out"	
File	Line	
api.php	50	
Acl.php	151, 155, 158, 148	
Feed.php	146, 135, 148	
User.php	145	
Rolesedit.php	103, 52	
Roles.php	36, 89	
Grid.php	102, 153, 120	
Form.php	101	
General.php	72	
Tabs.php	164	
Tree.php	241	