

Sonarqube 가이드

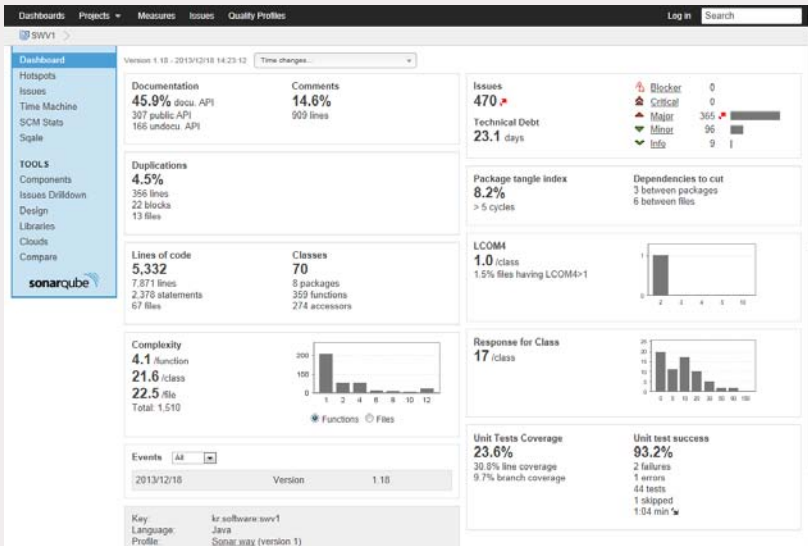
소프트웨어공학센터
SW공학기술팀

Sonarqube 개요

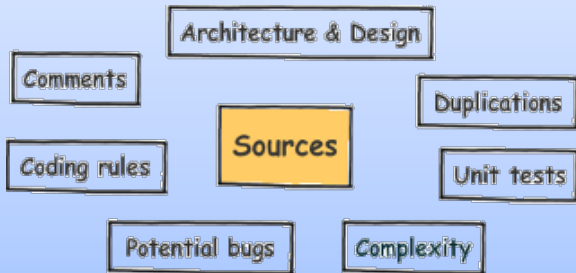
소개	소나큐브(Sonarqube)는 코드품질 검사를 위한 공개 소프트웨어 플랫폼이다.		
주요 기능	지속 빌드 품질 관리 (Continuous Build Code Quality Management)		
카테고리(용도)	–	세부 카테고리	–
커버리지	–	도구 난이도	상급
License 형태 / 비용	Lesser GNU General Public License	사전 설치 도구	–
사용 환경 / 개발 상태	운영체제	Multi-platform	
	Eclipse 환경	Multi-platform	
특징	• JAVA , C/C++ , Objective-C , C# , PHP , COBOL 등을 지원한다 (일부 언어는 상업용) • 중복코드 , 코딩 스탠다드 , 유닛테스트 , 코드 커버리지, 코드 복잡도, 잠재적 버그, 주석, 설계, 아키텍처에 대한 보고서를 생성한다 • 수치 통계를 저장하고 그래프를 제공한다		
적용 회사 / 프로젝트	–		
관련 도구	Jenkins , JIRA , Mantis , Maven , Ant , Gradle		
제작사	sonarsource		
공식 홈페이지	http://www.sonarqube.org/		
개발자	–		

Introduction

<http://www.sonarqube.org>



- ✓ 50개 이상의 plugin 활용 가능
- ✓ 20개 이상의 개발언어 지원



구분	내용
필수 조건	Java(Oracle JRE 6 onwards) Install
H/W 요구사항	Web Server (500MB RAM 이상)
제공 플랫폼	Java ✓ Java Oracle JRE Database ✓ Derby ✓ Microsoft SQL Server ✓ MySQL ✓ Oracle ✓ PostgreSQL Web Browser ✓ Microsoft Internet Explorer ✓ Mozilla Firefox ✓ Google Chrome ✓ Safari



Jenkins vs. SonarQube



Jenkins



지원언어

➤ 지원언어 제한
(대부분 Java 기준)

사용성

➤ 사용 어려움

UI

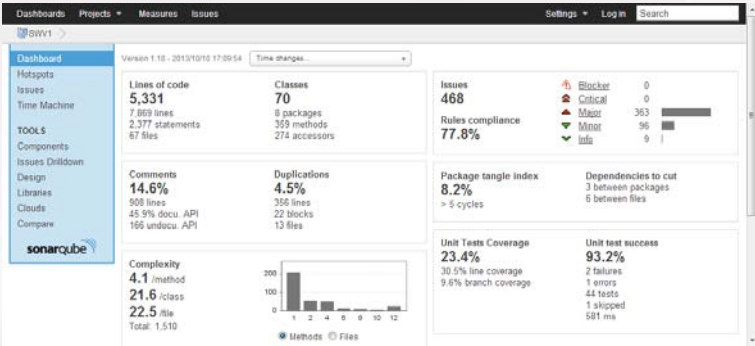
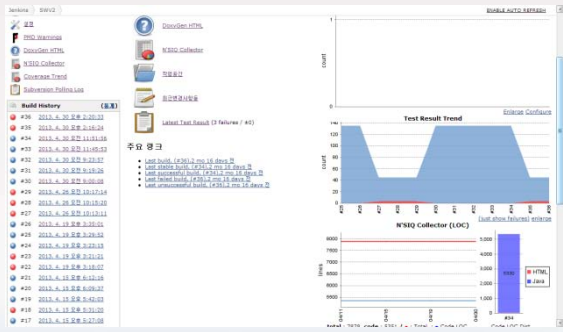
➤ 한눈에 파악 어려움

➤ 20여개 언어 지원
(Java 이외에는 부분적 유료)

.NET
ABAP
C/C++
Cobol
Delphi/Pascal
Drools
Erlang
Flex/ActionScript
Groovy
Java
JavaScript
Natural
Pacbase
PHP
PL/SQL
Python
Visual Basic
Web
XML

➤ Jenkins에 비해 쉬움

➤ 한눈에 파악 가능



Not Enough or Too Many Comments

Documentation	Comments
45.9% docu. API	14.6%
307 public API	909 lines
166 undocu. API	

Coding Standards Breach

Issues		Blocker	0
470		Critical	0
Technical Debt		Major	365
23.1 days		Minor	96
		Info	9

Spaghetti Design

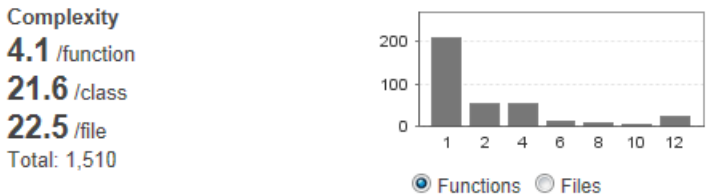
Package tangle index	Dependencies to cut
8.2%	3 between packages
> 5 cycles	6 between files

Duplications

Duplications
4.5%
356 lines
22 blocks
13 files

Developer's Seven Deadly Sins

Bad Distribution of Complexity



Bugs and Potential Bugs

Issues		Blocker	0
470		Critical	0
Technical Debt		Major	365
23.1 days		Minor	96
		Info	9

Lack of Unit Tests

Unit Tests Coverage	Unit test success
23.6%	93.2%
30.8% line coverage	2 failures
9.7% branch coverage	1 errors
	44 tests
	1 skipped
	1:04 min

Cyclomatic Complexity

- ✓ 프로그램의 논리적 복잡성
- ✓ 제어흐름이 분기될 때 마다 1씩 증가 (if, while, for, and, or case...)

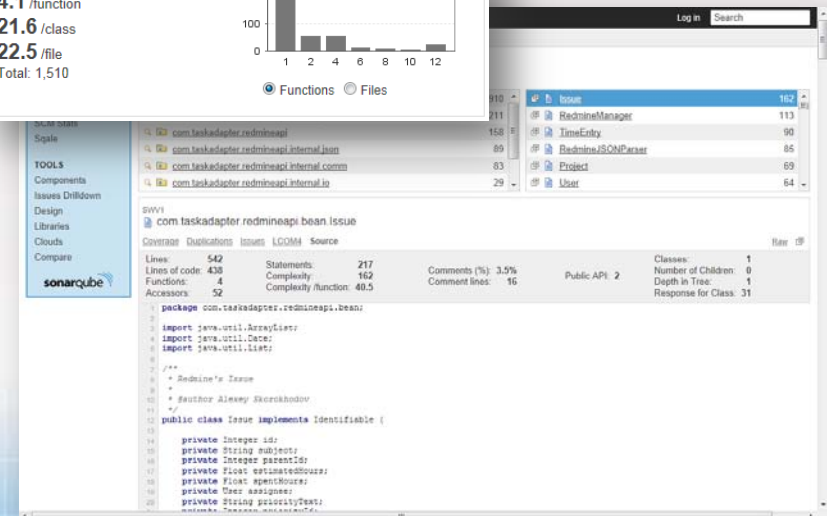
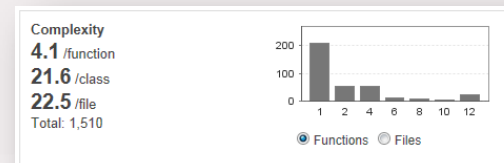
```
public void process(Car myCar) {
    if(myCar.isNotMine()) {
        return;
    }
    car.paint("red");
    car.changeWheel();
    while(car.hasGazol() && car.isNotStressed()) {
        car.drive();
    }
    return;
}
```

Cyclomatic Complexity : 5

CC	Risk
5-10	Low
11-20	Moderate
21-50	High
>50	Very high

CC	Bad fix probability
1-10	5%
20-30	20%
>50	40%
approaching 100	60%

출처 : <http://www.aivosto.com/project/help/pm-complexity.html>



SW Metric - Duplications

```
extern int array1[];  
extern int array2[];  
int sum1 = 0;  
int sum2 = 0;  
int average1 = 0;  
int average2 = 0;
```

```
for (int i = 0; i < 4; i++) {  
    sum1 += array1[i];  
}  
average1 = sum1/4;
```

```
for (int i = 0; i < 4; i++) {  
    sum2 += array2[i];  
}  
average2 = sum2/4;
```

```
int calcAverage (int* Array_of_4) {  
    int sum = 0;  
    for (int i = 0; i < 4; i++) {  
        sum += Array_of_4[i];  
    }  
    return sum/4;  
}
```

```
extern int array1[];  
extern int array2[];
```

```
int average1 = calcAverage(array1);  
int average2 = calcAverage(array2);
```

중복도 증가 원인

- ✓ 개발자 변경에 따른 기존 개발자 코드 사장
- ✓ Copy and Paste Programming
- ✓ 리턴 타입이 다른 동일 메소드
- ✓ 유사 비즈니스 로직 처리로 인한 중복
- ✓ 재사용 가능성이 있는 코드의 임시보관

The screenshot displays the SonarQube web interface for a project named 'SWV1'. The 'Duplications' section is highlighted, showing a 4.5% duplication rate across 356 lines, 22 blocks, and 13 files. A detailed view of a specific duplication is shown, comparing two versions of the 'com.taskadapter.redmineapi.bean.Changeset' class. The first version (line 14) shows a method 'result = prime * result + ((user == null) ? 0 : user.hashCode());'. The second version (line 62) shows a similar method but with a different implementation. The interface also shows a table of duplicated blocks and lines.

Blocks	Nb Lines	From line	File	Details
2	14	54	Changeset	com.taskadapter.redmineapi.bean.Changeset
	12	62	Membership	com.taskadapter.redmineapi.bean.Membership

출처 : http://en.wikipedia.org/wiki/Duplicate_code

SW Metric - Documentation

- ✓ 사내 주석 표준 정의
- ✓ public function 위주
- ✓ 주석 같은 소스코드

SWV1

Dashboard

Hotspots

Issues

Time Machine

SCM Stats

Sqale

TOOLS

Components

Issues Drilldown

Design

Libraries

Clouds

Compare

sonarqube

Public documented API (%)

45.9%

Drilldown on 166 Public undocumented API

com.taskadapter.redmineapi	58
com.taskadapter.redmineapi.internal	58
com.taskadapter.redmineapi.bean	20
com.taskadapter.redmineapi.internal.comm	18
com.taskadapter.redmineapi.internal.comm.redmine	4
com.taskadapter.redmineapi.internal.io	4

SWV1

com.taskadapter.redmineapi.RedmineManager

Coverage Duplications Issues LCOM4 Source

Lines: 877	Statements: 144	Comments (%): 40.6%	Public documented API (%): 62.7%	Classes: 2
Lines of code: 391	Complexity: 113	Comment lines: 267	Public undocumented API: 25	Number of Children: 0
Functions: 69	Complexity /function: 1.6		Public API: 67	Depth in Tree: 1
Accessors: 0				Response for Class: 76

```
1 /*
2  Copyright 2010-2012 Alexey Skorokhodov.
3
4  Licensed under the Apache License, Version 2.0 (the "License");
5  you may not use this file except in compliance with the License.
6  You may obtain a copy of the License at
7
8      http://www.apache.org/licenses/LICENSE-2.0
9
10 Unless required by applicable law or agreed to in writing, software
11 distributed under the License is distributed on an "AS IS" BASIS,
12 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
13 See the License for the specific language governing permissions and
14 limitations under the License.
15 */
16
17 package com.taskadapter.redmineapi;
```

Documentation	Comments
45.9% docu. API	14.6%
307 public API	909 lines
166 undocu. API	

$$\frac{\text{Comment lines}}{(\text{Lines of code} + \text{Comment lines})} \times 100 = \text{Comments}(\%)$$

$$\frac{267}{267 + 391} \times 100 = 40.6$$

$$\frac{\text{Public API} - \text{Public undocumented API}}{\text{Public API}} \times 100 = \text{Public documented API}(\%)$$

$$\frac{67 - 25}{67} \times 100 = 62.7$$

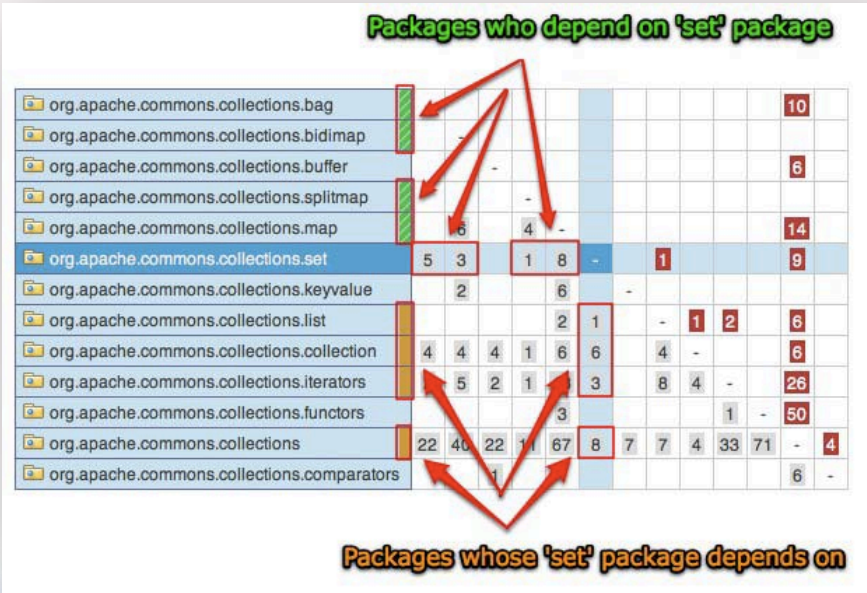
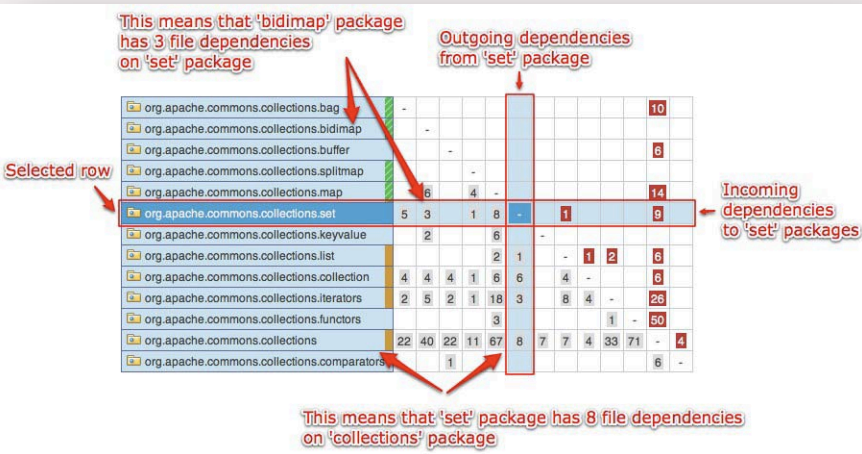
Package Tangle Index

- ✓ 디렉토리 간의 상호의존성 수준
- ✓ 100%에 가까워질 수록 상호의존성이 높다.

Package tangle index
8.2%
> 5 cycles

Dependencies to cut
3 between packages
6 between files

$$2 \times \frac{\text{File dependencies to cut}}{\text{Number of file dependencies between directories}} \times 100$$



DSM (Dependency Structure Matrix)

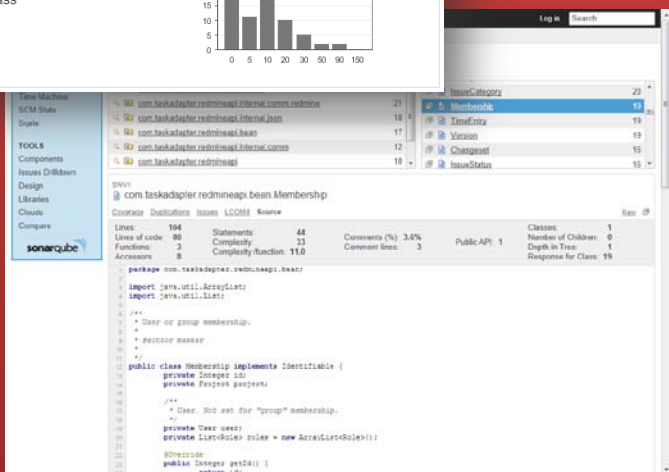
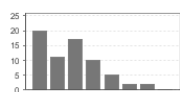
Coupling (결합도)

- ✓ 모듈간의 상호 의존하는 정도
- ✓ 하나의 변경이 다른 것의 변경을 요구하는 정도
- ✓ 독립적인 모듈이 되기 위해서는 각 모듈간의 결합도가 약해야 함

RFC (Response For Class)

- ✓ Number of Distinct Methods and Constructors invoked by a Class
- ✓ 일반적으로 50이 넘으면 리팩토링 대상

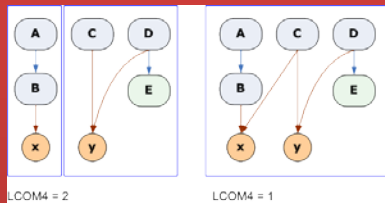
Response for Class
17/class



Cohesion (응집도)

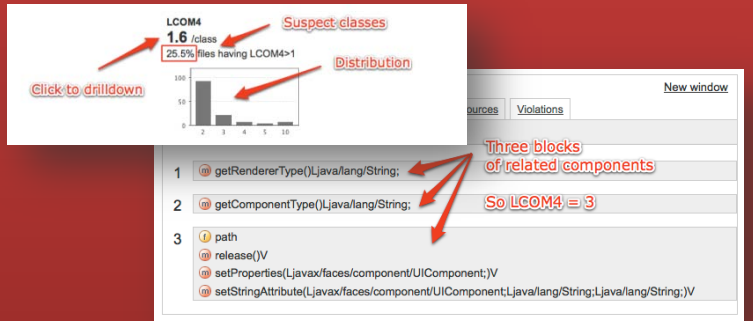
- ✓ 모듈의 구성 요소들 사이의 관련성 정도
- ✓ 변경이 필요할 때 하나의 요소에서 변화하는 부분의 정도
- ✓ 응집도가 높을수록 소프트웨어에 대한 이해 및 유지보수가 용이하다

LCOM4

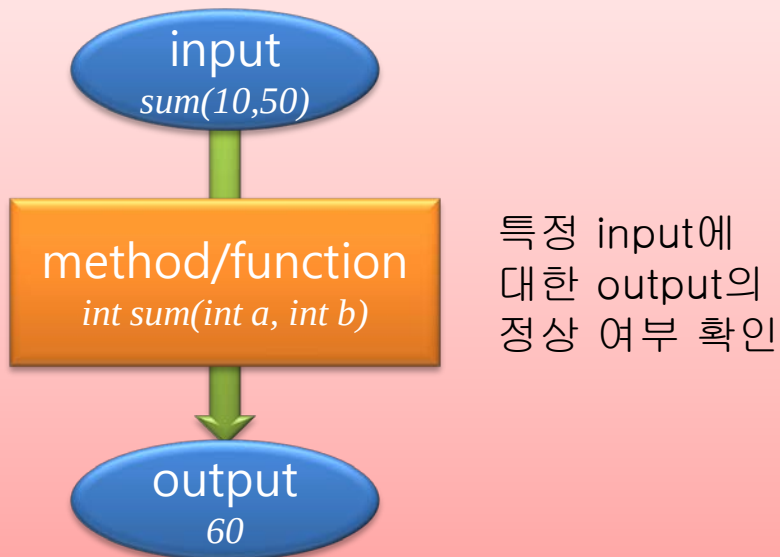


LCOM 4	해석 기준
1	응집도가 높다
>=2	응집도가 낮다
0	Method가 없다

출처 : <http://www.aivosto.com/project/help/pm-oo-cohesion.html>



- ✓ 코드의 각 모듈이 의도대로 작동하는지 확인하는 절차
- ✓ 소스코드의 method 혹은 function 단위로 정상 동작 여부를 테스트한다.
- ✓ 별도의 테스트 코드 작성이 필요하다.



```
public class Calcurator {  
    public int sum(int a, int b){  
        return a + b;  
    }  
}
```

```
public class CaluratorTest {  
    @Test  
    public void testSum(){  
        Calcurator c = new Calcurator();  
        int result = c.sum(10, 50);  
        assertEquals(60, result, 0);  
    }  
}
```

출처 : <http://using.tistory.com/entry/JUnit-테스트-하기>

SW Metric - Test

```
public int test(int input, boolean case1, boolean case2) {  
    int x = input;  
    if (case1)  
        x++;  
    if (case2)  
        x--;  
    return x;  
}
```

Unit Tests Coverage	Unit test success
23.6%	93.2%
30.8% line coverage	2 failures
9.7% branch coverage	1 errors
	44 tests
	1 skipped
	1:04 min

$$Coverage = \frac{CT+CF+LC}{2 \times B+EL} \times 100$$

Line Coverage

Branch Coverage

Path Coverage

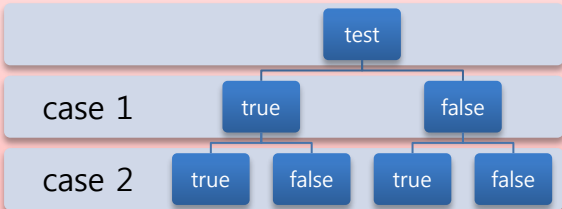
- ✓ 테스트에 의해 실행된 라인의 비율
- ✓ *Line Coverage*
$$= \frac{LC}{EL} \times 100$$

test(x, true, true);
: 100% statement coverage

- ✓ 모든 분기에 대해 참 / 거짓이 테스트된 비율
- ✓ *Branch Coverage*
$$= \frac{CT+CF}{2 \times B} \times 100$$

test(x, true, true);
test(x, false, false);
: 100% branch coverage

- ✓ 모든 실행 경로에 대해 테스트된 비율



test(x, true, true);
test(x, true, false);
test(x, false, true);
test(x, false, false);
: 100% path coverage

출처 : <http://testersthoughtsuncombed.blogspot.kr/2013/02/statement-coverage-vs-branch-coverage.html>

Dashboards Projects Measures Issues Quality Profiles Log in Search

Quality Profiles > Java > Sonar way >

Coding rules Alerts Projects Permalinks Profile inheritance Changelog

Name/Key Repository Severity Activation Status Search

Any Any Active Any

Download

Active/Severity Name [expand / collapse] Sort by: Rule name

☒ Major	Anon Inner Length	DEPRECATED
☒ Major	Avoid Array Loops	
☒ Major	Avoid Assert As Identifier	
☒ Major	Avoid Calling Finalize	DEPRECATED
☒ Major	Avoid Catching NPE	
☒ Critical	Avoid Catching Throwable	DEPRECATED
☒ Major	Avoid commented-out lines of code	
☒ Major	Avoid cycle between java packages	
☒ Major	Avoid Decimal Literals In Big Decimal Constructor	
☒ Major	Avoid Duplicate Literals	DEPRECATED
☒ Major	Avoid Enum As Identifier	DEPRECATED
☒ Minor	Avoid Instanceof Checks In Catch Clause	DEPRECATED
☒ Major	Avoid Print Stack Trace	DEPRECATED
☒ Major	Avoid Rethrowing Exception	DEPRECATED
☒ Major	Avoid Throwing Null Pointer Exception	

SW Metric - Issues

Issues
470

Technical Debt
23.1 days

Blocker0

Critical0

Major365

Minor96

Info9

Technical Review

- Confirm
- False positive
- Change severity

Dispositioning

- Assign
- Plan

General

- Comment

Endgame

- Resolve

SWV1

com.taskadapter.redmineapi.CopyBytesHandler

Magic Number | Open | Updated: 7일 | Technical debt: 30 minutes

'4096' is a magic number.

Comment | Assign (to me) | Plan | Confirm | More actions

Change severity

Resolve

False positive

```
24 }
25
26 @Override
27 public void processContent(BasicHttpResponse content)
28     throws RedmineException {
29     final byte[] buffer = new byte[4096 * 4];
30     int readed = 0;
31     try {
32         final InputStream input = content.getInputStream();
33         try {
34             while ((readed = input.read(buffer)) > 0)
```

Close

Sonar

Version 3.6-SNAPSHOT - Jun 03 2013 23:58

Time changes...

Manage dashboards

Issues 1,334 Open 670 Reopened 11 Confirmed 653

My Unresolved Issues No issue.

Unresolved Issues per Assignee

Simon Brandhof	25	
Evgeny Mandrikov	22	
Julien Lancelot	10	
Julien Henry	7	
Fabrice Belliard	7	
Jean Baptiste Vilain	2	
Not assigned	1261	

Open action plans

Sonar 3.6 17 of 24 issues solved

Long term 2 of 284 issues solved

Short term 4 of 117 issues solved

Accepted 32 of 276 issues solved

False Positives Issues

Removed field SCM_LAST_COMMIT_DATE about 11 hours

simon.brandhof: Removed in earlier versions

출처 : <http://docs.codehaus.org/display/SONAR/Seeing+Issues>

