



Menschen. Innovationen. Lösungen.



CODENARC
Less Bugs Better Code

Statische Code-Analyse für Groovy & Grails mit CodeNarc

CamelCaseCon 2011
Vortrag von Stefan Glase
am 07.09.2011



OPITZ CONSULTING

Stefan Glase, OPITZ CONSULTING



Software-Entwickler

Java EE, Spring, Groovy, Grails

Trainer und Coach

Sprecher und Autor

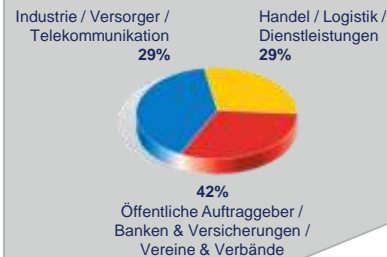


Märkte

- Java
- SOA
- ORACLE
- BI/DWH
- Outtasking

Kunden

- Branchen-
übergreifend
- Über 600
Kunden



Leistungsangebot

- IT-Strategie
- Beratung
- Implementierung
- Betrieb
- Training



Fakten

- Gründung 1990
- 400 Mitarbeiter
- 8 Standorte in D/PL



Agenda

- **Was ist CodeNarc?**
- **CodeNarc Regeln und Regelwerke**
- **CodeNarc im Browser**
- **Eigene CodeNarc Regeln erstellen**
- **Integration in Gradle-basierte Groovy-Anwendungen**
- **Integration in Grails-Anwendungen**
- **Weitere Integrationsmöglichkeiten**
- **Fazit**

1

Was ist CodeNarc?

Was ist CodeNarc?

**CodeNarc analyzes Groovy code for defects,
bad practices, inconsistencies, style issues
and more.**

<http://codenarc.sourceforge.net/>

Was ist CodeNarc?

- **Aktuelle Version 0.15**
- **Bibliothek ist im Maven Central Repository verfügbar**

```
<dependency>  
  <groupId>org.codenarc</groupId>  
  <artifactId>CodeNarc</artifactId>  
  <version>0.15</version>  
</dependency>
```

- **264 Regeln in kategorisierten Regelwerken**
- **Verschiedene Arten von Reports: XML, HTML, Text**
- **Kontakt zu den Entwicklern Chris Mair und Hamlet D'Arcy via Mailing-Liste, BugTracker oder Twitter**

Was ist CodeNarc?

CodeNarc Report

Report title: Sample Project
Date: Mar 5, 2011 10:01:23 PM
Generated with: CodeNarc.v0.13

Summary by Package

Package	Total Files	Files with Violations	Priority 1	Priority 2	Priority 3
All Packages	7	4	-	9	5
org.codenarc/sample/domain	2	1	-	2	3
org.codenarc/sample/other	1	-	-	-	-
org.codenarc/sample/service	4	3	-	7	2

Package: org.codenarc.sample.domain

↳ SampleDomain.groovy

Rule Name	Priority	Line #	Source Line / Message
EmptyElseBlock	2	24	<pre> } else { //MDC: The else block is empty</pre>

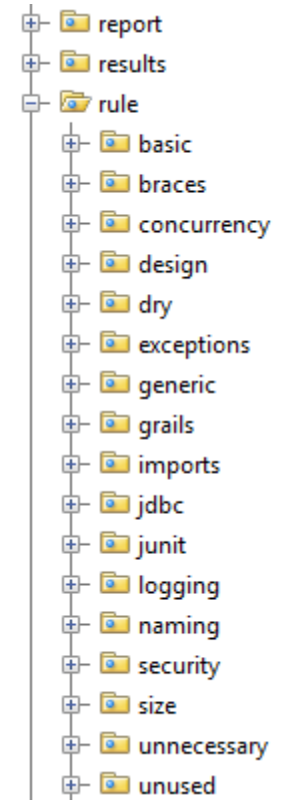
Beispiel HTML-Report

2

CodeNarc Regeln und Regelwerke

CodeNarc Regeln und Regelwerke

- Regeln (**Rule**) werden zu Regelwerken (**RuleSet**) zusammengefasst und liefern Verstöße (**Violation**) im untersuchten Code (**SourceCode**)
- Analyse des Abstract Syntax Tree (AST) mittels Visitor-Objekten (**AstVisitor**)
 - Inklusive Hilfsklasse **AstUtil** für übliche Operationen auf dem AST
- Erzeugung von Verstößen (**Violation**) mittels einfacher API im **AstVisitor**



CodeNarc Regeln und Regelwerke

■ 264 Regeln... und die Zahl ist stetig wachsend!

ConstantTernaryExpressionRule

RemoveAllOnSelfRule

BigDecimalInstantiationRule

CloneableWithoutCloneRule

SerializableClassMustDefineSerialVersionUIDRule

EqualsAndHashCodeRule

EmptySynchronizedStatementRule

BrokenOddnessCheckRule

EmptySwitchStatementRule

ConsecutiveLiteralAppendsRule

EmptyTryBlockRule

EmptyForStatementRule

BooleanGetBooleanRule

DuplicateCaseStatementRule

EmptyElseBlockRule

EmptyFinallyBlockRule

BooleanMethodReturnsNullRule

ConstantIfExpressionRule

... und viele weitere!

CodeNarc Regeln und Regelwerke

■ Ein paar Beispiele...

```
def a = " + 123
```

Rule Name	Priority	Line #	Source Line / Message
AddEmptyString	2	6	[SRC] def a = " + 123 [MSG] Concatenating an empty string is an inefficient way to convert an object to a String. Consider using toString() or String.valueOf(Object)
ConsecutiveStringConcatenation	3	6	[SRC] def a = " + 123 [MSG] String concatenation in class smellicious.CodeDumpster can be joined into the literal '123'
ConsecutiveStringConcatenation	3	6	[SRC] def a = " + 123 [MSG] String concatenation in class smellicious.CodeDumpster can be joined into the literal '123'

CodeNarc Regeln und Regelwerke

```
abstract class CodeMonster {  
    static { }  
    int hashCode() { return 1 }  
}
```

Rule Name	Priority	Line #	Source Line / Message
EmptyStaticInitializer	2	5	[SRC] static { } [MSG] The class smellicious.CodeMonster has an empty static initializer. It is safe to delete it
EqualsAndHashCode	2	4	[SRC] abstract class CodeMonster { [MSG] The class smellicious.CodeMonster defines hashCode() but not equals(Object)
AbstractClassWithoutAbstractMethod	2	4	[SRC] abstract class CodeMonster { [MSG] The abstract class smellicious.CodeMonster contains no abstract methods
UnnecessaryReturnKeyword	3	7	[SRC] return 1 [MSG] The return keyword is not needed and can be removed

CodeNarc Regeln und Regelwerke

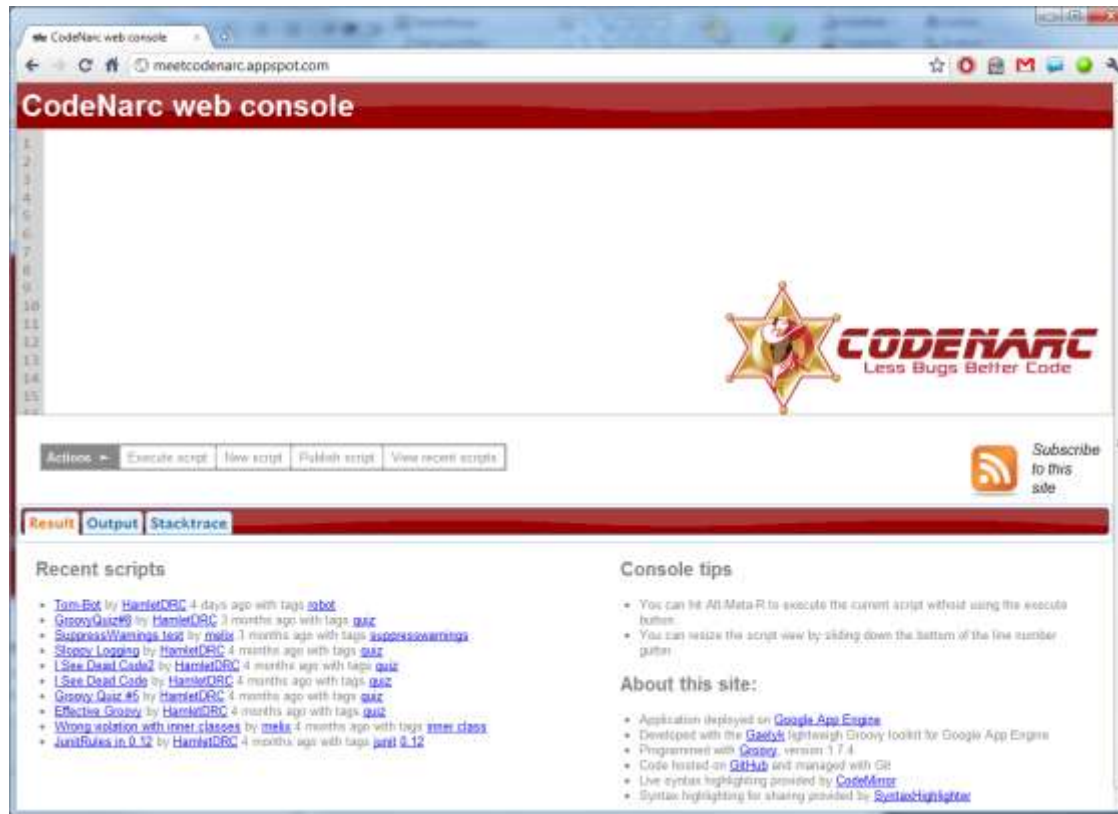
- Regeln liegen in Form von XML-Dokumenten zur einfachen Einbindung und Konfiguration der anzuwendenden Regeln vor

```
SourceCode.groovy x AvoidPrintStackTraceRule.groovy x basic.xml x concurrency.xml x design.xml x grails.xml x
1 <ruleset xmlns="http://codenarc.org/ruleset/1.0"
2     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3     xsi:schemaLocation="http://codenarc.org/ruleset/1.0 http://codenarc.org/ruleset-schema.xsd"
4     xsi:noNamespaceSchemaLocation="http://codenarc.org/ruleset-schema.xsd">
5
6     <description>
7         Grails rule set.
8         These rules implement standards and best practices related to the Grails framework (http://grails.org).
9     </description>
10
11     <rule class='org.codenarc.rule.grails.GrailsPublicControllerMethodRule' />
12     <rule class='org.codenarc.rule.grails.GrailsSessionReferenceRule' />
13     <rule class='org.codenarc.rule.grails.GrailsServletContextReferenceRule' />
14     <rule class='org.codenarc.rule.grails.GrailsStatelessServiceRule' />
15
16 </ruleset>
```

3

CodeNarc im Browser

CodeNarc im Browser

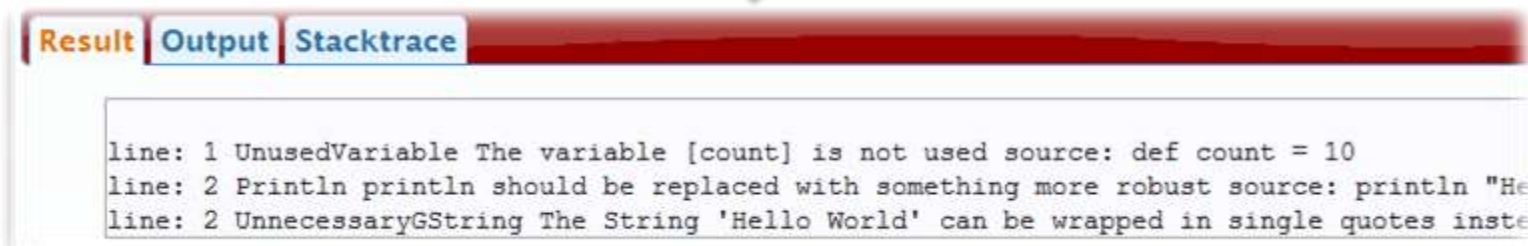
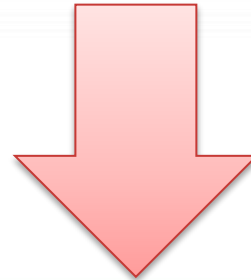


<http://meetcodenarc.appspot.com/>

CodeNarc im Browser



```
← → ↻ 🏠 🌐 meetcodenarc.appspot.com/edit/27001
CodeNarc web console
1  def count = 10
2  println "Hello World"
3
4  |
5
6
```



```
Result Output Stacktrace
line: 1 UnusedVariable The variable [count] is not used source: def count = 10
line: 2 Println println should be replaced with something more robust source: println "He
line: 2 UnnecessaryGString The String 'Hello World' can be wrapped in single quotes inste
```

4

Eigene CodeNarc Regeln erstellen

Eigene CodeNarc Regeln erstellen

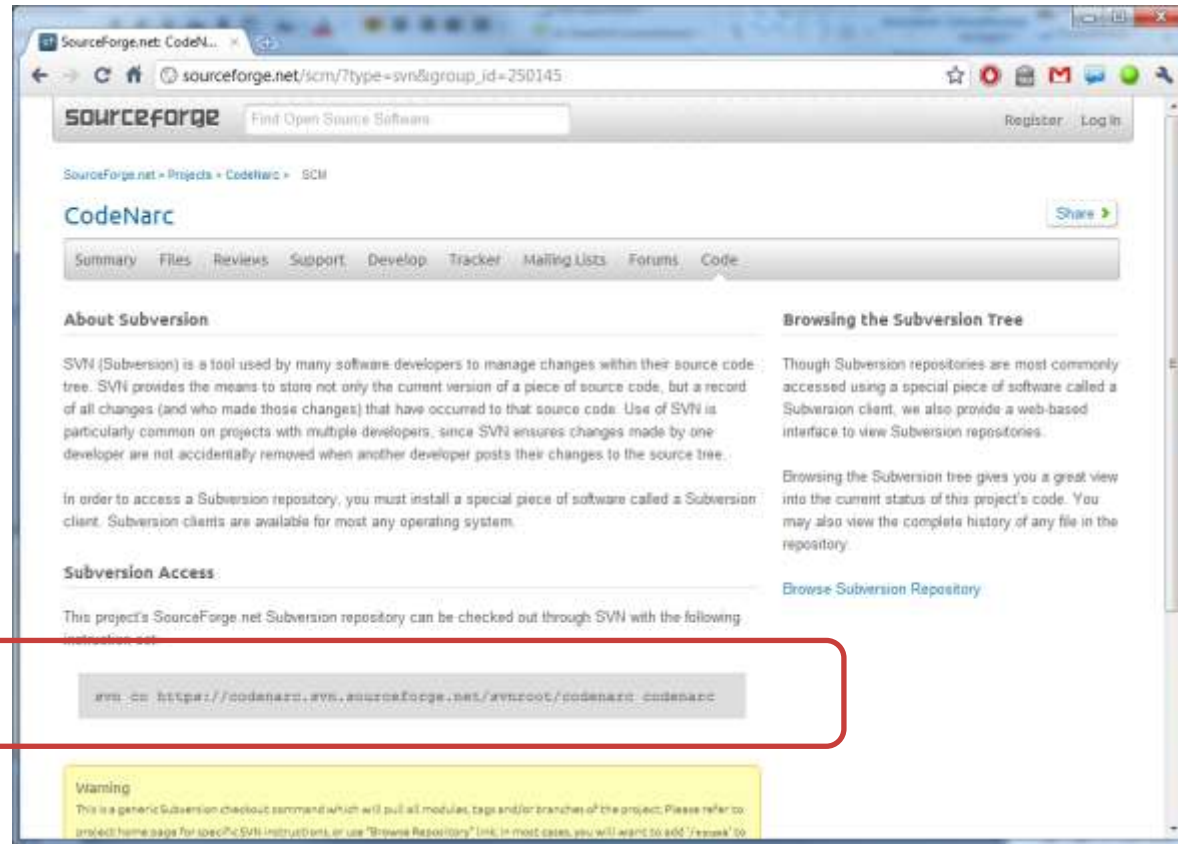


maven

Anforderungen



Eigene CodeNarc Regeln erstellen



Checkout der Sourcen von SourceForge

`svn checkout https://codenarc.svn.sourceforge.net/svnroot/codenarc codenarc`

Eigene CodeNarc Regeln erstellen

```
cs. C:\Windows\system32\cmd.exe
[INFO] --- maven-compiler-plugin:2.3.2:compile (default-compile) @ CodeNarc ---
[INFO] Nothing to compile - all classes are up to date
[INFO] --- gmaven-plugin:1.3:compile (default) @ CodeNarc ---
[INFO] Compiled 835 Groovy classes
[INFO] --- gmaven-plugin:1.3:generateTestStubs (default) @ CodeNarc ---
[INFO] Generated 320 Java stubs
[INFO] --- maven-resources-plugin:2.4.3:testResources (default-testResources) @ CodeNarc ---
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 32 resources
[INFO] --- maven-compiler-plugin:2.3.2:testCompile (default-testCompile) @ CodeNarc ---
[INFO] Nothing to compile - all classes are up to date
[INFO] --- gmaven-plugin:1.3:testCompile (default) @ CodeNarc ---
[INFO] Compiled 583 Groovy classes
[INFO] --- maven-surefire-plugin:2.7.2:test (default-test) @ CodeNarc ---
[INFO] Tests are skipped.
[INFO] --- maven-jar-plugin:2.3.1:jar (default-jar) @ CodeNarc ---
[INFO] Building jar: D:\Temp\codenarc\trunk\target\CodeNarc-0.14.jar
[INFO] --- maven-install-plugin:2.3.1:install (default-install) @ CodeNarc ---
[INFO] Installing D:\Temp\codenarc\trunk\target\CodeNarc-0.14.jar to D:\Java\maven-repository\org\codenarc\CodeNarc\0.14\CodeNarc-0.14.jar
[INFO] Installing D:\Temp\codenarc\trunk\pom.xml to D:\Java\maven-repository\org\codenarc\CodeNarc\0.14\CodeNarc-0.14.pom
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 18.992s
[INFO] Finished at: Mon Jun 06 15:29:57 CEST 2011
[INFO] Final Memory: 77M/634M
[INFO] -----
```

Bauen des Projektes mit Maven

mvn install

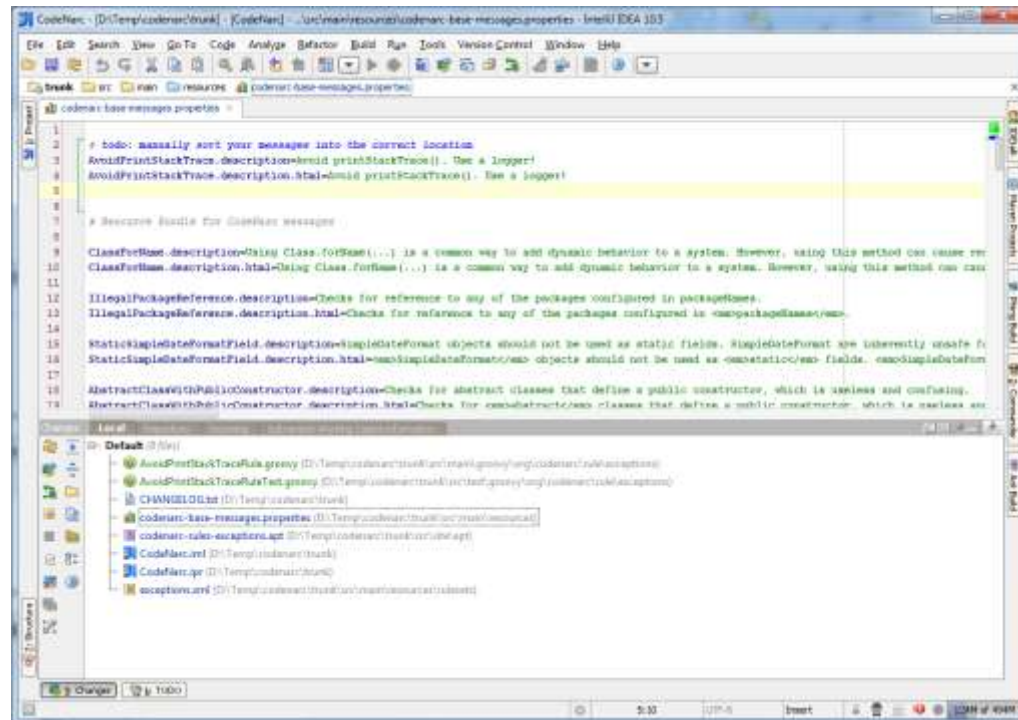
Eigene CodeNarc Regeln erstellen

```
cs. C:\Windows\system32\cmd.exe
D:\Temp\codenarc\trunk>
D:\Temp\codenarc\trunk>
D:\Temp\codenarc\trunk>groovy codenarc.groovy create-rule
Enter your name:Stefan Glase
Enter the rule name:AvoidPrintStackTrace
Enter the rule category. Valid categories are:
basic
braces
concurrency
design
dry
exceptions
generic
grails
imports
jdbc
junit
logging
naming
security
size
unnecessary
unused
exceptions
Enter the rule description:Avoid printStackTrace(). Use a logger!
Created ./src/main/groovy/org/codenarc/rule/exceptions/AvoidPrintStackTraceRule.groovy
Created ./src/test/groovy/org/codenarc/rule/exceptions/AvoidPrintStackTraceRuleTest.groovy
Updated ./src/main/resources/codenarc-base-messages.properties
Updated ./src/main/resources/rulesets/exceptions.xml
Updated ./src/site/apt/codenarc-rules-exceptions.ap
Updated ./CHANGELOG.txt
adding to svn... A      src\main\groovy\org\codenarc\rule\exceptions\AvoidPrintStackTraceRule.groovy
adding to svn... A      src\test\groovy\org\codenarc\rule\exceptions\AvoidPrintStackTraceRuleTest.groovy
Finished

D:\Temp\codenarc\trunk>
D:\Temp\codenarc\trunk>
```

Erstellen der Regel mit CodeNarc eigenem Groovy Script groovy codenarc.groovy create-rule

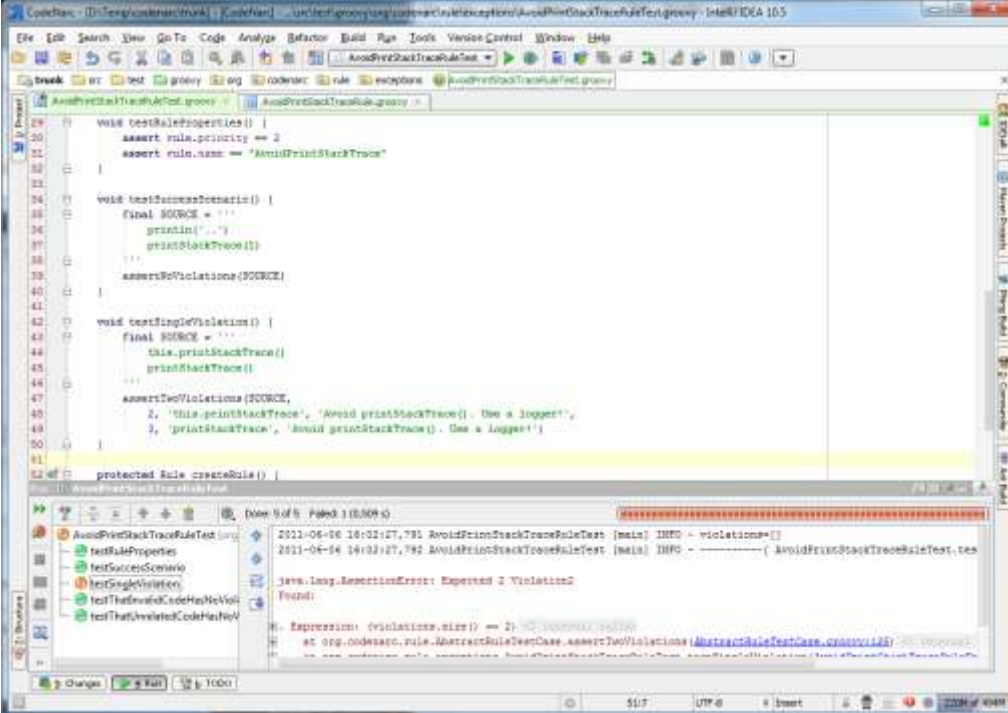
Eigene CodeNarc Regeln erstellen



Editieren des MessageBundle (korrekte Sortierung)

src/main/resources/codenarc-base-messages.properties

Eigene CodeNarc Regeln erstellen



The screenshot displays the IntelliJ IDEA 10.5 IDE. The main editor shows a Groovy test file named `AvoidPrintStackTraceRuleTest.groovy`. The code includes several test methods: `testRuleProperties()`, `testSuccessScenario()`, `testSingleViolation()`, and `testTwoViolations()`. The `testTwoViolations()` method uses `assertTwoViolations()` to verify that two violations are detected. The bottom pane shows the test execution results, indicating that the test passed with no violations found.

```
void testRuleProperties() {
    assert rule.priority == 2
    assert rule.name == "AvoidPrintStackTrace"
}

void testSuccessScenario() {
    final SOURCE = '''
    println('.')
    printStackTrace()
    '''
    assertNoViolations(SOURCE)
}

void testSingleViolation() {
    final SOURCE = '''
    this.printStackTrace()
    printStackTrace()
    '''
    assertTwoViolations(SOURCE,
        2, 'this.printStackTrace', 'Avoid printStackTrace(). Use a logger!',
        2, 'printStackTrace', 'Avoid printStackTrace(). Use a logger!')
}

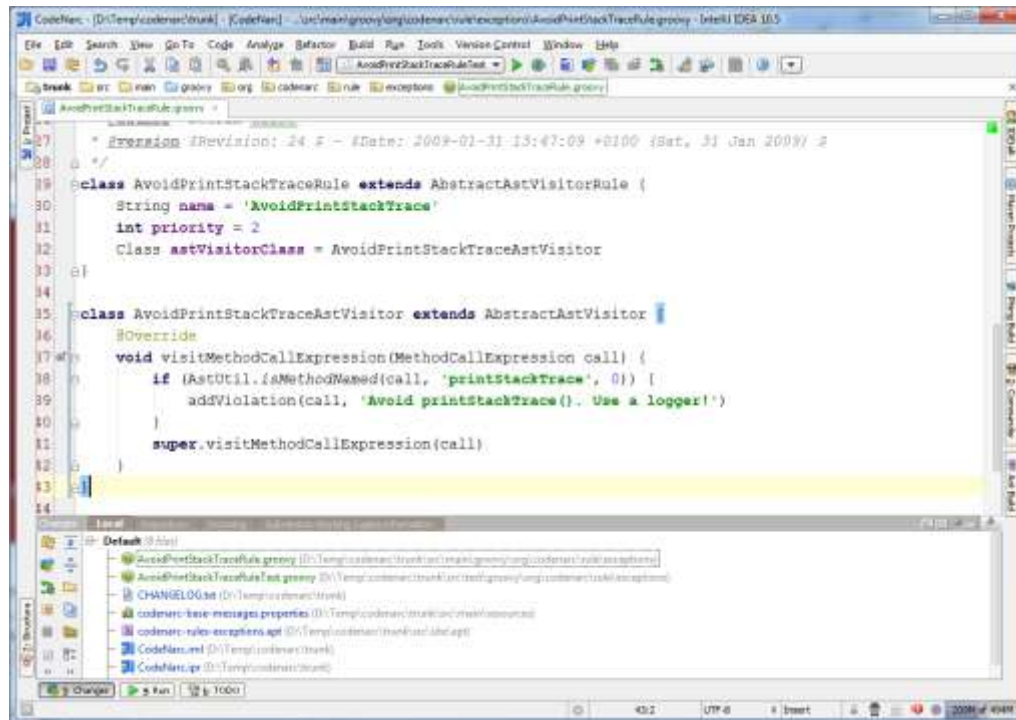
protected Rule createRule() {
```

Test Results:

```
Done: 0 of 5 Passed: 1 (8.00%)
2011-06-16 16:02:27,791 AvoidPrintStackTraceRuleTest [main] INFO - violations=[]
2011-06-16 16:02:27,792 AvoidPrintStackTraceRuleTest [main] INFO - -----( AvoidPrintStackTraceRuleTest.testTwoViolations)
java.lang.AssertionError: Expected 2 Violations2
Found:
- Expressions: (violations.size() == 2)
at org.codenarc.rule.AbstractRuleTestBase.assertTwoViolations(AbstractRuleTestBase.groovy:145)
```

Implementierung des Tests zur Absicherung der Regel `src/test/groovy/org.codenarc.rule.exceptions.AvoidPrintStackTraceRuleTest`

Eigene CodeNarc Regeln erstellen



```
CodeNarc - [D:\Temp\codenarc\trunk] - [Kotlin] - ...src/main/groovy/org.codenarc.rule.exceptions.AvoidPrintStackTrace.groovy - IntelliJ IDEA 10.5

File Edit Search View Go To Code Analyze Refactor Build Run Tools Version Control Window Help
src trunk main groovy org codenarc rule exceptions AvoidPrintStackTrace.groovy

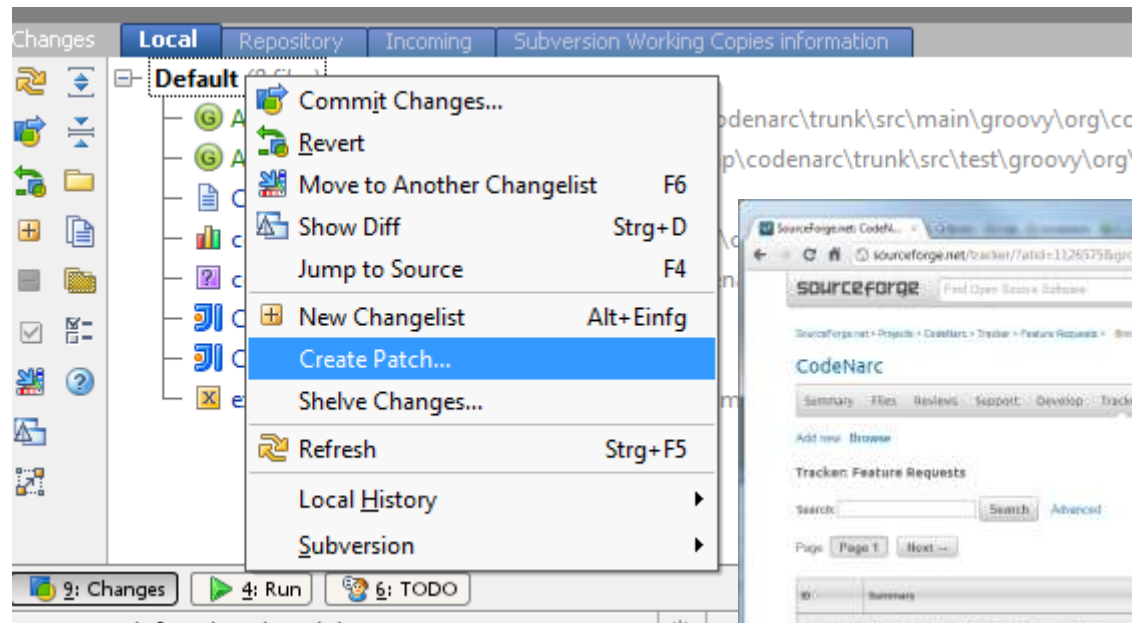
* Revision: 24 # - #Date: 2009-01-31 13:47:09 +0100 (Sat, 31 Jan 2009) #
*/
class AvoidPrintStackTraceRule extends AbstractAstVisitorRule {
    String name = 'AvoidPrintStackTrace'
    int priority = 2
    Class astVisitorClass = AvoidPrintStackTraceAstVisitor
}

class AvoidPrintStackTraceAstVisitor extends AbstractAstVisitor {
    @Override
    void visitMethodCallExpression(MethodCallExpression call) {
        if (AstUtil.isMethodNamed(call, 'printStackTrace', 0)) {
            addViolation(call, 'Avoid printStackTrace(). Use a logger!')
        }
        super.visitMethodCallExpression(call)
    }
}

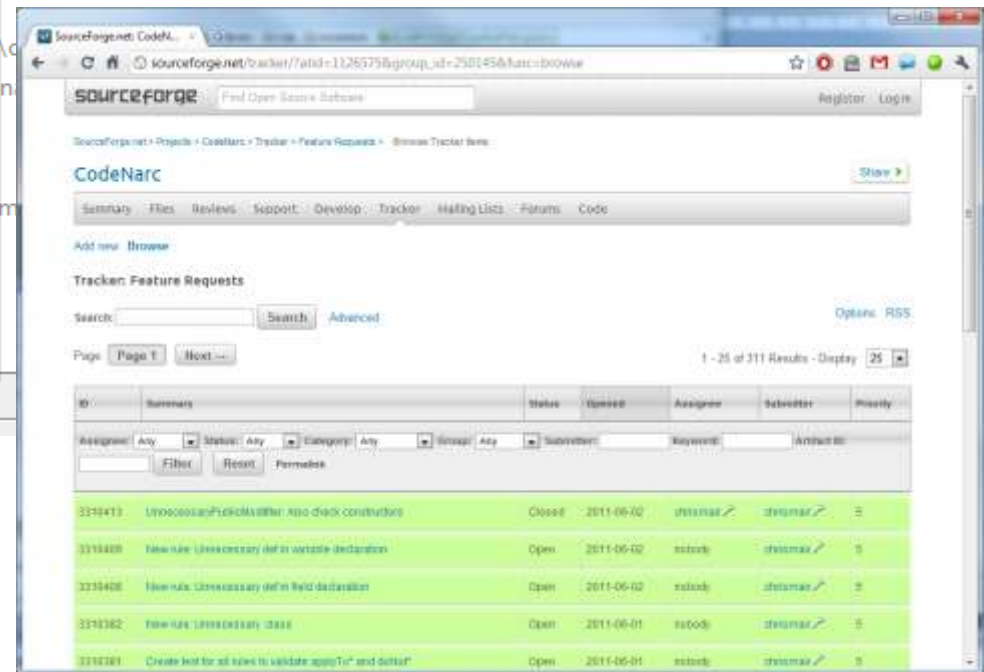
Local
Default [0 text]
AvoidPrintStackTrace.groovy [D:\Temp\codenarc\trunk\src\main\groovy\org\codenarc\rule\exceptions]
AvoidPrintStackTraceTest.groovy [D:\Temp\codenarc\trunk\src\test\groovy\org\codenarc\rule\exceptions]
CHANNELLOG.txt [D:\Temp\codenarc\trunk]
codenarc-base-messages.properties [D:\Temp\codenarc\trunk\src\main\resources]
codenarc-rule-exceptions.groovy [D:\Temp\codenarc\trunk\src\test\groovy]
CodeNarc.xml [D:\Temp\codenarc\trunk]
CodeNarc.groovy [D:\Temp\codenarc\trunk]
CodeNarc.groovy [D:\Temp\codenarc\trunk]
```

Implementierung der Regel und des AST-Visitors
src/main/groovy/org.codenarc.rule.exceptions.AvoidPrintStackTraceRule

Eigene CodeNarc Regeln erstellen



1. Patch erstellen



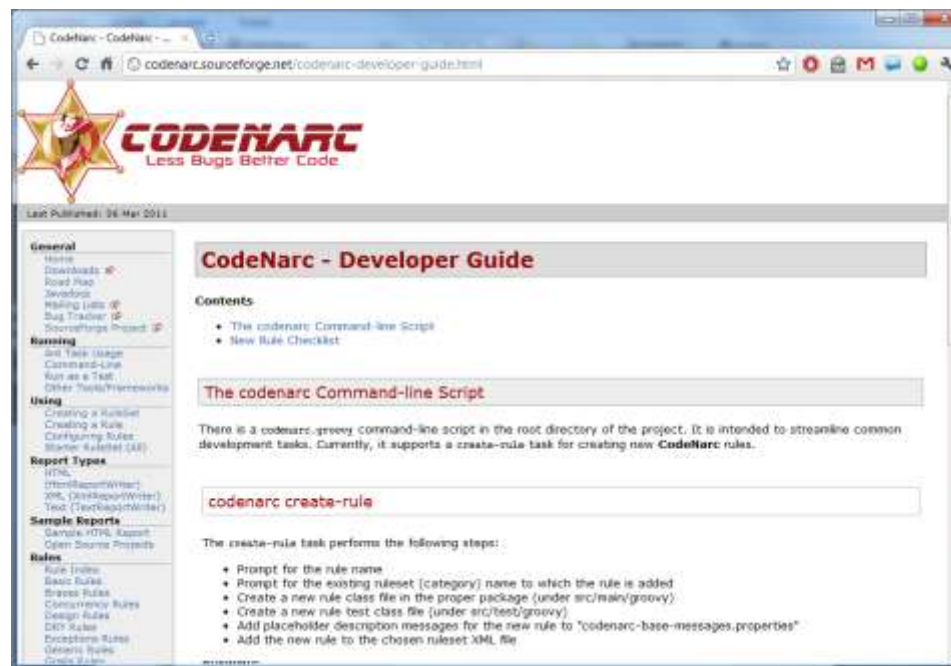
2. Eintrag im BugTracker mit angehängtem Patch erstellen

Bereitstellen der eigenen Regel

Eigene CodeNarc Regeln erstellen

Mehr Informationen für Entwickler auf der CodeNarc Webseite

<http://codenarc.sourceforge.net/codenarc-developer-guide.html>



5

Integration in Gradle-basierte Groovy-Anwendungen

Integration in Gradle-basierte Groovy-Anwendungen

■ CodeNarc ist fest integriert im Gradle Code Quality Plugin

- Gradle 1.0 Milestone 3 enthält CodeNarc 0.13
- http://www.gradle.org/code_quality_plugin.html

■ Code Quality Plugin verwendet

- Checkstyle für Java-Code
- CodeNarc für Groovy-Code



VS.



■ CodeNarc Plugin als eigenständiges Plugin

- Version von CodeNarc ist frei konfigurierbar
- <http://evgeny-goldin.com/wiki/Gradle-CodeNarc-plugin>

Integration in Gradle-basierte Groovy-Anwendungen



Live-Demo... bitte Daumen drücken!

6

Integration in Grails- Anwendungen

Was ist Grails?



Was ist Grails?

Grails is an advanced and innovative **open source web application platform** that delivers new levels of developer productivity by applying principles like **Convention over Configuration**. Grails helps development teams **embrace agile methodologies**, deliver quality applications in reduced amounts of time, and focus on what really matters: creating high quality, easy to use applications that delight users. Grails naturally complements Java application development since it **is built on Spring** and **based on Groovy**, the leading dynamic language for the Java platform.

<http://www.springsource.com/developer/grails>

Integration in Grails-Anwendungen



Zur Integration von CodeNarc in Grails existiert das CodeNarc Plugin
<http://grails.org/plugin/codenarc>

Integration in Grails-Anwendungen



Live-Demo... bitte noch einmal Daumen drücken!

7

Weitere Integrationsmöglichkeiten

Weitere Integrationsmöglichkeiten

■ IDEs

- IntelliJ..... (IDEA CodeNarc Plugin)

■ Application Frameworks

- Grails (Grails CodeNarc Plugin)
- Griffon (Griffon CodeNarc Plugin)

■ Build- und Code-Analyse-Werkzeuge

- Gradle (Gradle Code Quality Plugin)
- Maven (Maven CodeNarc Plugin)
- Hudson/Jenkins (Hudson Violations Plugin)
- Sonar (Sonar Groovy Plugin)

<http://codenarc.sourceforge.net/codenarc-other-tools-frameworks.html>

Fazit



Im Netz...

CodeNarc:

<http://codenarc.sourceforge.net>

<http://meetcodenarc.appspot.com>

<http://www.youtube.com/watch?v=ZPu8FaZZwRw>

Beispiele:

<https://github.com/codescape/presentations>

Twitter:

[@stefanglase](https://twitter.com/stefanglase)

Fragen und Antworten



Kontakt

Stefan Glase

Senior Consultant

OPITZ CONSULTING Gummersbach GmbH

Kirchstr. 6, 51647 Gummersbach

Tel. +49 (2261) 6001 – 0

stefan.glase@opitz-consulting.com

