



Paris
JUG

www.parisjug.org

Contrôle qualité « Open Source »

www.parisjug.org





Paris
JUG

Romain Pelisse

PMD, XRadAr, QSOS, KDE,
CKJM, Sonar...



Atos OriginTM

François Le Droff

Fna, FlexPMD, QSOS,
XRadAr



Adobe[®]



www.parisjug.org

Copyright © 2008 ParisJug. Licence CC – Creative Commons 2.0 France – Paternité – Pas d'Utilisation Commerciale – Partage des Conditions Initiales à l'Identique



Intervenants

▪ Romain Pelisse

- belaran-<at>-gmail.com
- <http://belaran.eu/>
- belaran@jabber.fr
- <http://twitter.com/rpelisse>
- https://www.xing.com/profile/Romain_PELISSE
- <http://www.linkedin.com/in/rpelisse>

▪ François Le Droff

- ✓ francois.le.droff-<at>-gmail.com
- ✓ <http://www.droff.com>

*“Any fool can write code that a computer can understand.
Good programmers write code that humans can understand.”*

~M. Fowler

“You can't control what you can't measure.”

~Tom DeMarco

“Controlling complexity is the essence of computer programming.”

~Brian Kernighan.

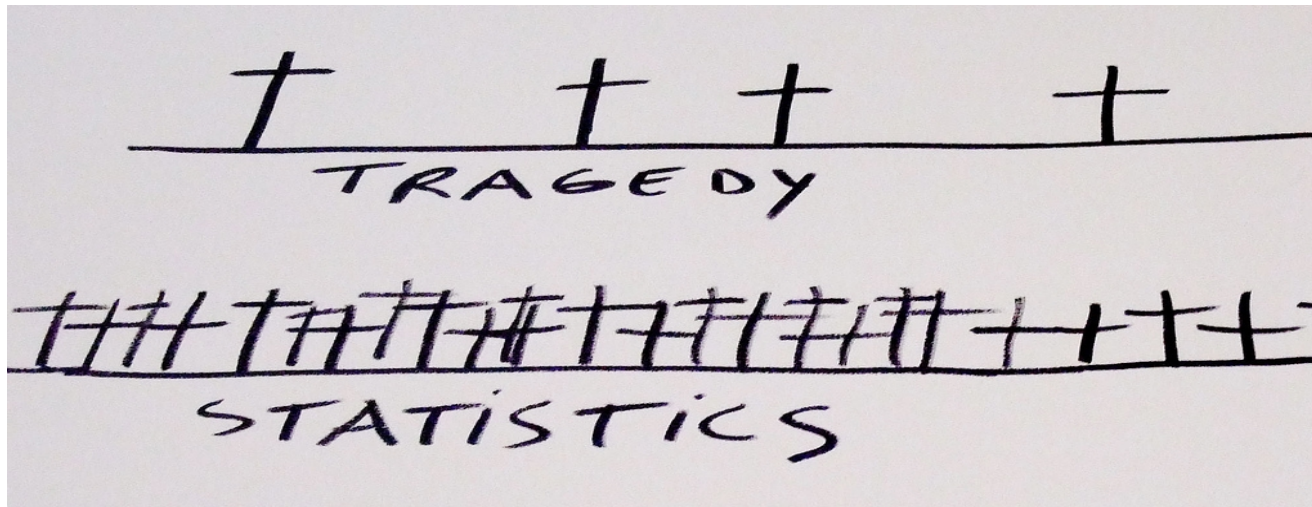
*“There does not now, nor will there ever exist, a programming language in
which it is the least bit hard to write bad programs.”*

~Lawrence Flon

“Programs for sale: fast, reliable, cheap - choose two.”

~Author Unknown

Pourquoi la qualité logicielle ?



- **80% des entreprises négligent la qualité logicielle**
- **1 projet sur 3 n'est pas livré, 2/3 ne passent pas la recette**
- **9/10 sont livrés en retard**
- **70% du budget : maintenance**
- **40% de cette maintenance à la réécriture**

Une obligation professionnelle



... Et parce qu'on n'aime plus concevoir que corriger...

Pourquoi ? La qualité c'est fun !



*... La qualité c'est fun, enrichissant, c'est à la mode, ~~c'est sexy~~... c'est agile, c'est scrum et tdt friendly, ~~c'est web2.0~~ c'est SOA nom mieux ! **c'est la QOA !***

Code, réputation, et publicité

- ✓ Programmer c'est « publier » (une API)
 - ✓ « APIs are for ever »
- ✓ Une fois adoptée, on refuse d'en changer !

- ✓ Une API de qualité :
 - ✓ Simple
 - ✓ Addictif



La qualité rendra vos clients accro



Pourquoi s'en priver ?



En un mot : la qualité c'est bon, mangez-en !

Pour qui ?

- **Le développeur :**
 - Assisté par les outils, intégré à l'IDE
 - Qualité garantie à la source (démarche proactive)
- **L'intégrateur/recetteur/QA :**
 - veille au respect des règles au quotidien (intégration continue)
 - assure la recevabilité technique.
- **Le chef de projet de la MOE :**
 - planifie et prend les mesures nécessaires au respect des exigences qualité
- **L'architecte**
 - Facilite le respect de l'ensemble des règles ;
 - Définit les contraintes d'architecture spécifiques ;



Quand ?

- Tout au long du cycle de vie du projet
 - Eviter l'effet tunnel
 - Contractualiser les exigences
 - Mode itératif

- Contrôle qualité continu
 - Au cours du développement
 - Build, intégration continue
 - Recette



Tour d'horizon des outils OS

- ant, maven1, maven2, eclipse, PMD, CheckStyle, (bluecheckstyle, jjguidelines), Hammurapi, FindBugs, PMD-CPD, JavaNCSS, squid, JDepend, Classycle, CKJM, Clirr, JUnit, Cobertura, Emma, XRadar
- et aussi PatternTesting, lint4j, jcsc, QJ-Pro, Spoon, TPTP, Netbeans, Jackpot, Simian, Condenser, CAP, Metrics, ByeCycle, JarDiff, JCoverage, EclEmma, CVSStat, QALab, maven dashboard



Types d'outils

- ✓ **Les outils d'analyse statique de code source Java (*.java)**
 - ✓ contrôle de validité javadoc
 - ✓ contrôle de règles de style, de codage et d'anti-pattern
 - ✓ indicateurs métriques
- ✓ **Les outils d'analyse statique de byte code Java (*.class)**
 - ✓ indicateurs métriques
- ✓ **Les outils d'analyse statique d'archive Java (*.jar)**
 - ✓ Les outils de vérification de compatibilité binaire, compatibilité API et de rapport de changement
- ✓ **Test unitaires**
 - ✓ rapport de test
 - ✓ rapport de couverture et taux de couverture de code
- ✓ **Autres**
 - ✓ activité GCL (CVS, SVN)

Quoi ? Faites votre marché



Analyse statique de source

✓ Les Contrôleur de code :

- ✓ PMD, CheckStyle (bluecheckstyle, jjguidelines), Hammurapi, FindBugs, PatternTesting, lint4j, jcs, QJ-Pro
- ✓ Contrôle et refactoring:
 - ✓ Eclipse : Spoon, TPTP
 - ✓ Netbeans : Jackpot
- ✓ Pré requis unique : un code syntaxiquement correct
 - ✓ ré-utilise l'arbre syntaxique abstrait (AST)
- ✓ Fonctionnalités: contrôlent
 - ✓ le style et les règles de nommage,
 - ✓ les erreurs d'encapsulation,
 - ✓ les erreurs de programmations, de design,
 - ✓ les erreurs de gestion et l'utilisation des exceptions,
 - ✓ les erreurs de gestion mémoire,
 - ✓ les erreurs de synchronisation défectueuses
 - ✓ et détecte une partie du code mort/inutile

Zoom sur quelques exemples



Règles de nommages



On ne retient que les bons mots...

La gestion des exceptions



✓ Préserver la *stacktrace*

- ✓ Générer une exception depuis un bloc `catch` sans passer la première exception à la nouvelle exception entraîne la perte de la *stacktrace*, ce qui pourra poser des problèmes lors du débogage de l'application.



```
log(1.00 - 9*.10); // surprise ! Il s'affiche : 0.099999999999999998
```

- ✓ **Eviter float et double si un résultat exact (monétaire) est requis**

- ✓ Les types float et double ont été conçus pour les calculs scientifiques et l'ingénierie. Ils réalisent de l'arithmétique binaire à virgule flottante qui a été conçue pour fournir des approximations efficaces et rapides sur une large échelle. Ils ne fournissent pas cependant de résultats exacts et ne **devraient** pas être utilisés quand ils ne sont pas requis.






Choisissez vos règles! Faites votre marché



Construisez votre référentiel

Communiquer et formaliser vos exigences qualité !

Qualification	Sigle	Explication
<u>Erreur</u>		Le non respect d'une règle de cette qualification est une erreur de programmation ou de « design ».
<u>Avertissements</u>		Les infractions de ce type ne sont pas systématiquement des erreurs, mais plutôt des points douteux ou des choix qui doivent nécessairement être validés par l'architecte.
<u>Bonne pratique</u>		Les infractions à ce type de règles indiquent généralement le non respect d'une bonne pratique.

Announce FlexPMD



Adobe Open Source

Adobe.com Home | Developer Connection

Go

[Home](#) [Projects](#) [Source](#) [Documentation](#) [Forums](#) [About](#)

Welcome Guest | [Sign In](#)

FlexPMD

FlexPMD uses a part of [PMD](#), its Java well known counterpart.

Overview

FlexPMD is a tool that helps to improve code quality by auditing any AS3/Flex source directory and detecting common bad practices, such as:

- Unused code (functions, variables, constants, etc.)
- Inefficient code (misuse of dynamic filters, heavy constructors, etc.)
- Over-complex code (nested loops, too many conditionals, etc.)
- Over-long code (classes, methods, etc.)
- Incorrect use of the Flex component lifecycle (commitProperties, etc.)

A report is produced describing the violations of a given rule set. FlexPMD includes a rule set that is broad ranging and continually growing. It is also straightforward to create new rules and users are encouraged to do so.

FlexPMD can currently be invoked from:

- The command line
- Maven
- Ant
- Automator on Mac OS X

Currently, FlexPMD produces a XML report by default, plus an HTML report when invoked by the Maven site plugin. The XML report can then be consumed by any PMD reports pretty printers (like the PMD hudson plugin).

An Eclipse plugin is under consideration.

Here is the kind of report you can have as an output of FlexPMD (violations in given file):



FlexPMD Project

[Home](#)
[About](#)
[Downloads](#)
[Source](#)
[Bug Database](#)
[Submitting a Patch](#)
[Developer Documentation](#)
[Forums](#)
[License](#)

Other Projects

[Cairngorm](#)
[Cairngorm Plugin](#)
[BlazeDS](#)
[Flex SDK](#)
[FlexUnit](#)
[Corelib](#)

[More related projects >](#)
[More Adobe projects >](#)

Exemple de rapport PMD

SOURCEFORGE.NET®



Last Published: 01/22/2008

SourceForge.net Project Page | Hosted by SourceForge

Introduction

- Motivation
- Reports
- What's Different?
- Vision
- Credits
- References

Installation and Execution

- ▶ Ant Configuration
- Maven Configuration
- XRadar-Config.xml
- Continuous Build

Documentation

- The XRadar Architecture
- ▶ Developer Documentation
- ▶ Plugins

Analysis Library

- Example Analysis

Project

Documentation

- ▶ Project Information
- ▼ Project Reports
 - CPD Report
 - Dashboard Report
 - PMD Report**



PMD Results

The following document contains the results of PMD 3.9.

Files

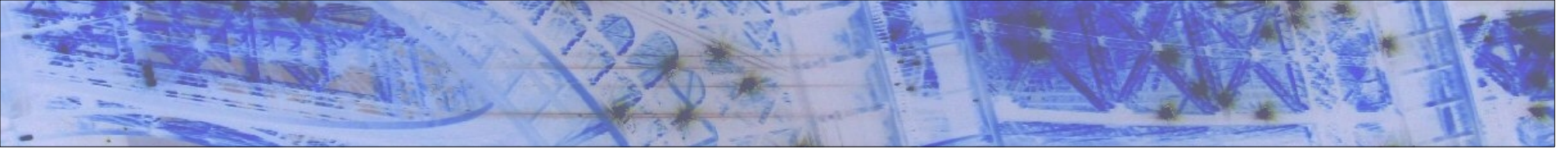
net/sourceforge/pmd/cpd/XMLRenderer.java

Violation	Line
Avoid appending characters as strings in StringBuffer.append.	27
Avoid appending characters as strings in StringBuffer.append.	38

org/sourceforge/xradar/AbstractProcess.java

Violation	Line
Avoid unused imports such as 'org.apache.tools.ant.BuildException'	24
Unnecessary wrapper object creation	123

org/sourceforge/xradar/config/ReleaseDataLoader.java



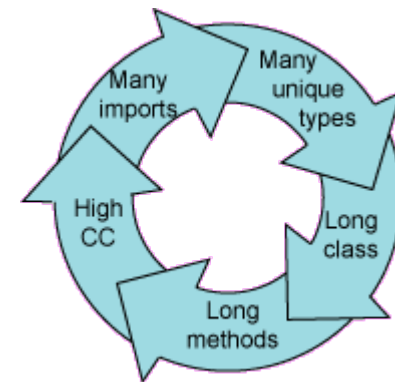
Taille du code : le pire ennemi ?

- Nombre de lignes de code
 - Total
 - Par fichier/classes
 - Par méthode
- Détection du copié-collé
- La complexité cyclomatique



Autre type d'analyse statique de code source

- ✓ **Détecteurs de copié-collé : PMD-CPD, Simian, Condenser :**
 - ✓ Pré requis unique : un code syntaxiquement correct
 - ✓ ré-utilise l'arbre syntaxique abstrait (AST) fourni par javacc



- ✓ **Calcul de métriques : ~~JavaNCSS~~ : Squid**
 - ✓ Pré requis unique : un code syntaxiquement correct
 - ✓ ré-utilise l'arbre syntaxique abstrait (AST) fourni par javacc
 - ✓ Fonctionnalités: calcul de métriques
 - ✓ calcul de complexité cyclomatique
 - ✓ de lignes de code : ncsc, javadoc, commentaires

Exemple de rapport JavaNCSS

SOURCEFORGE.NET®



Last Published: 01/24/2008

SourceForge.net Project Page | Hosted by SourceForge

About

[Quick history](#)
[Purpose](#)
[Download](#)

How to run

[Ant Usage](#)

Project

Documentation

- ▶ [Project Information](#)
- ▼ [Project Reports](#)
 - [Checkstyle](#)
 - [CPD Report](#)
 - [JavaDocs](#)
 - [JavaNCSS Report](#)**
 - [JDepend](#)
 - [PMD Report](#)
 - [Source Xref](#)



JavaNCSS Metric Results

[[package](#)] [[object](#)] [[method](#)] [[explanation](#)]

The following document contains the results of a JavaNCSS metric analysis.
[JavaNCSS web site.](#)

Packages

[[package](#)] [[object](#)] [[method](#)] [[explanation](#)]

Packages sorted by NCSS.

Package	Classes	Methods	NCSS	Javadocs	Javadoc lines	Single lines comment	Multi lines comment
org.sourceforge.mbeanmonitoring.report.castor.descriptors	7	63	811	63	455	179	42
org.sourceforge.mbeanmonitoring.report	5	29	548	6	33	47	115
org.sourceforge.mbeanmonitoring.report.castor	7	152	457	23	237	102	860
org.sourceforge.mbeanmonitoring.report.castor.types	2	14	136	2	16	24	254
org.sourceforge.mbeanmonitoring.report.castor.types.descriptors	2	18	56	18	130	24	12
Classes total	Methods total	NCSS total	Javadocs	Javadoc lines	Single lines comment	Multi lines comment	
23	276	2008	112	871	376	1283	

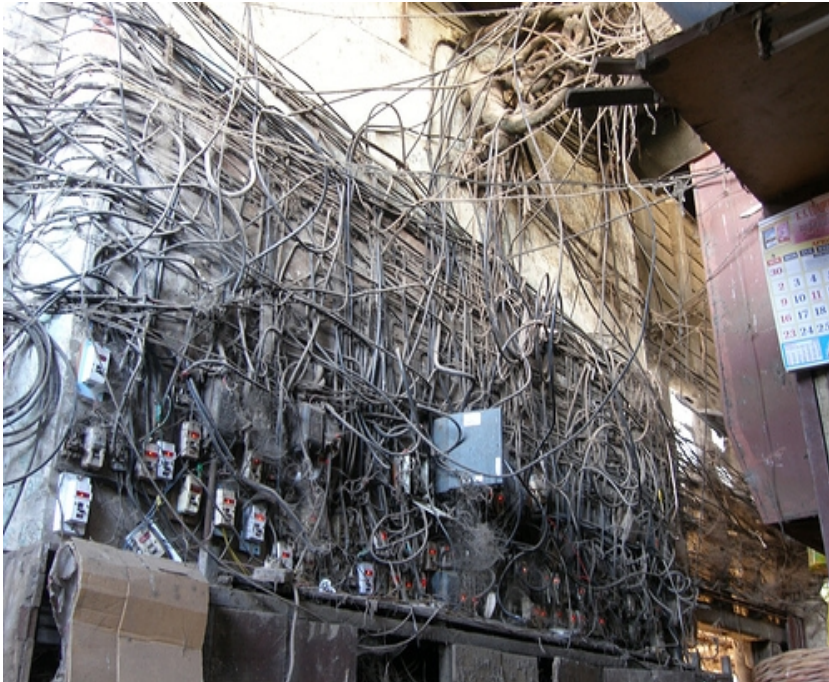
Objects

[[package](#)] [[object](#)] [[method](#)] [[explanation](#)]

TOP 30 classes containing the most NCSS.

Object	NCSS	Methods	Classes	Javadocs
org.sourceforge.mbeanmonitoring.report.castor.descriptors.ServerParamDescriptor	280	9	10	9
org.sourceforge.mbeanmonitoring.report.castor.descriptors.AttributeDescriptor	128	9	4	9

Règles d'encapsulation



Rendre les choses aussi inaccessibles que possible

Analyse de byte code Java

✓ Indicateurs métriques OO et architecture :

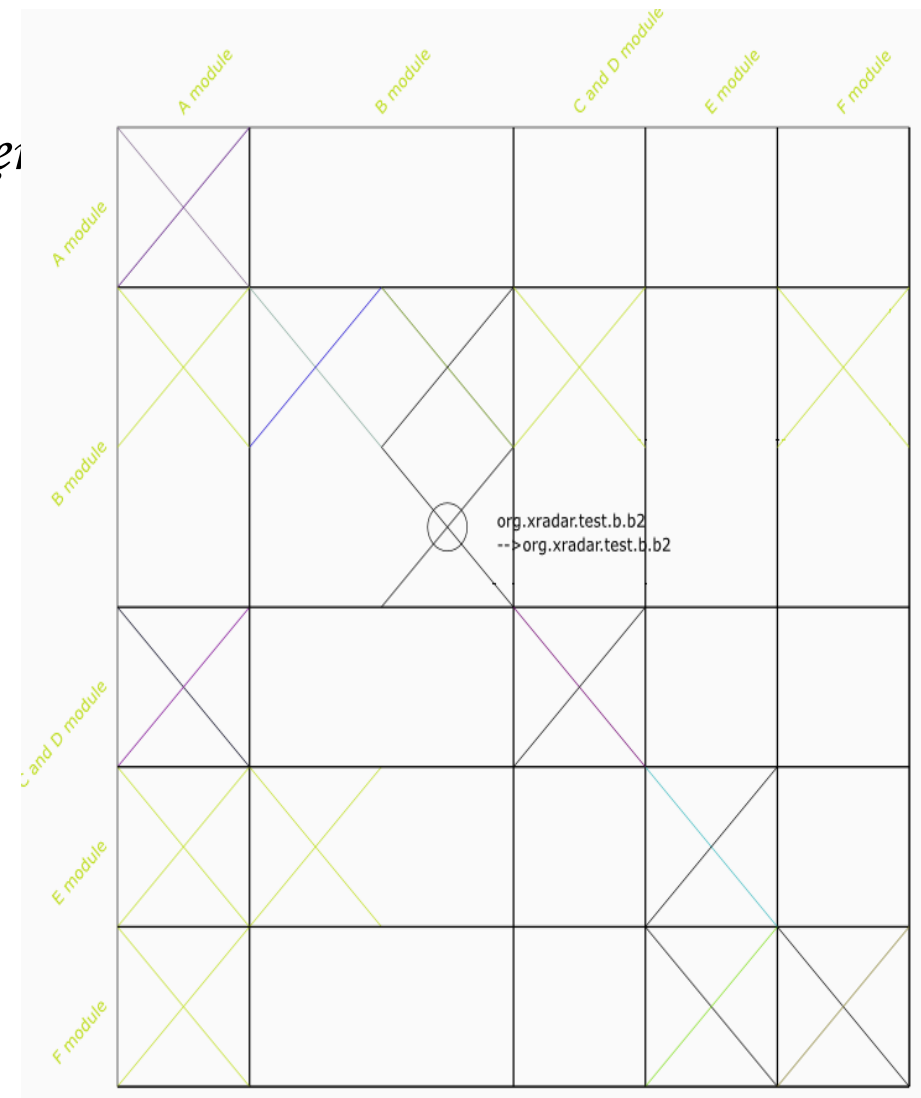
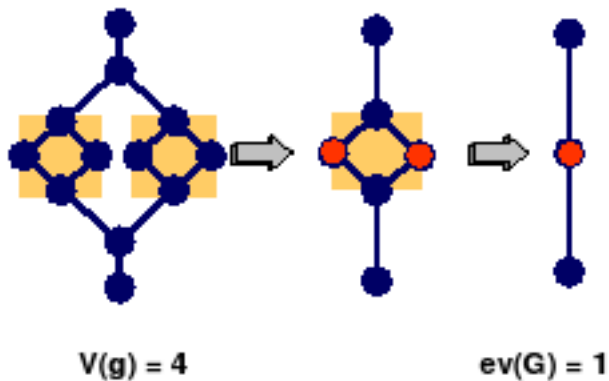
✓ *Squid, JDepend, Classycle, CKJM, Macken*

✓ *Eclipse plugins : CAP, Metrics, ByeCycle*

✓ Exemple :

✓ *calcul et graphe de dépendances*

✓ *Complexité cyclomatique*



Analyse de byte code Java

✓ Métriques de package

✓ Nombres de classes concrètes, abstraites, publiques

✓ Nombres d'interfaces

✓ Graphe de dépendances de packages :

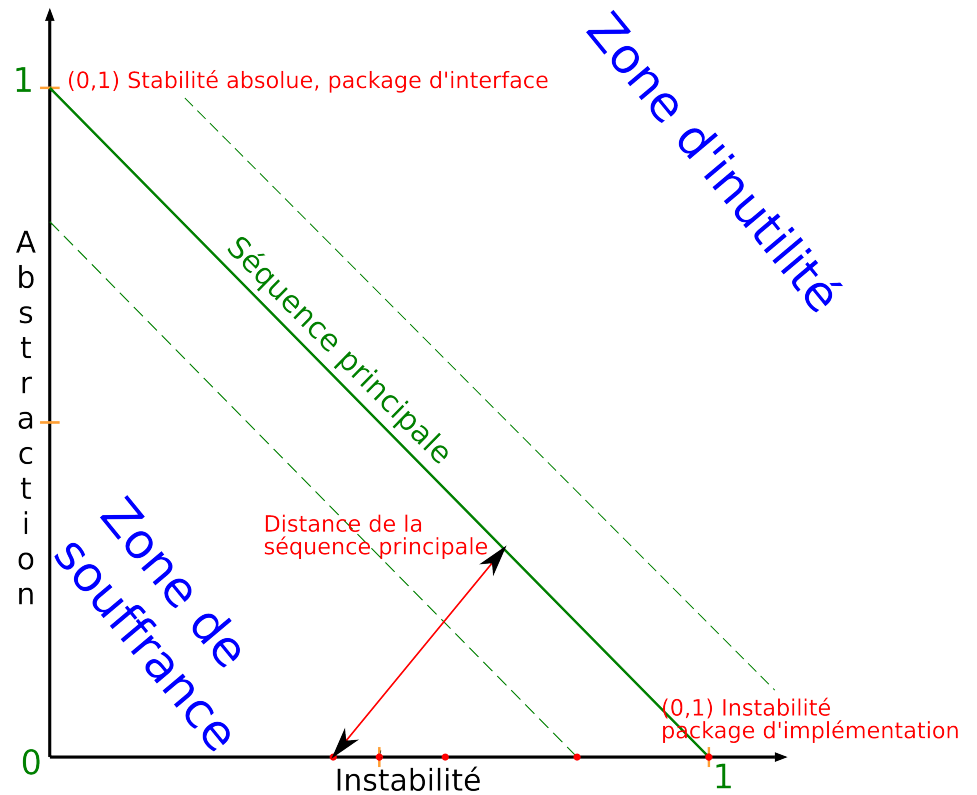
✓ couplage afférent et efférent

✓ Détection de dépendances cycliques

✓ Degré d'abstraction

✓ Degré d'instabilité

✓ Distance normale

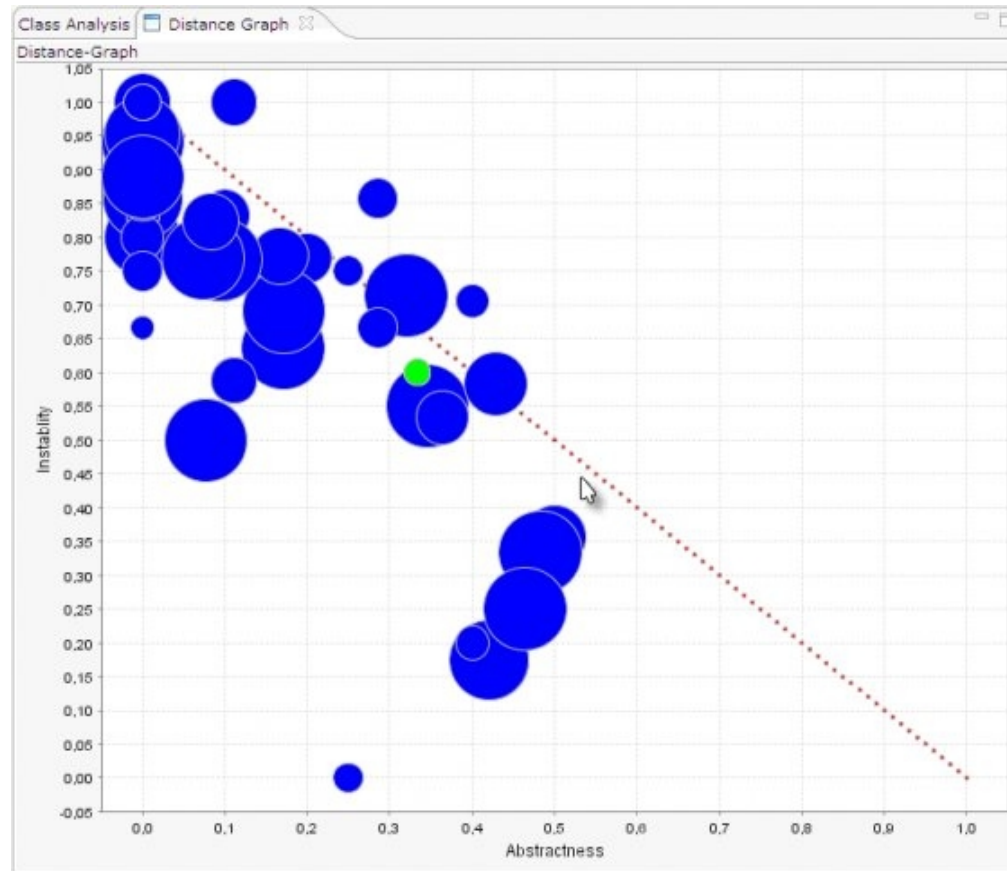


Analyse de byte code Java - 3/4

✓ Métriques de classes

- ✓ Nombres d'attributs, de méthodes, d'inner-class
- ✓ Nombre de classes enfants
- ✓ Profondeur d'héritage (le nombre d'ancêtres de la classe en ligne directe)
- ✓ Nombre de méthodes implémentées (Weighted methods per class)
- ✓ *Response for a Class* (RFC)
 - ✓ *Nombre de Méthodes accessibles par la classe, qu'elles soient implémentées directement, surchargées ou disponibles via héritage (méthodes publiques, protégées et du package pour Java)*
- ✓ Manque de cohésion des méthodes (LCOM)
 - ✓ *évalue la cohérence d'une classe entre ses attributs et ses méthodes et de vérifier qu'il n'y a pas d'ensembles disjoints.*
- ✓ Couplage efférent (Ce) ou *Coupling between object classes* (CBO)
 - ✓ *Une classe est couplée à une autre selon Chidamber et Kemerer, si les méthodes de la classe utilisent des attributs ou des méthodes d'une autre classe.*

Exemple plugin CAPS



Dépendances et compatibilités

✓ Gestion des dépendances externes

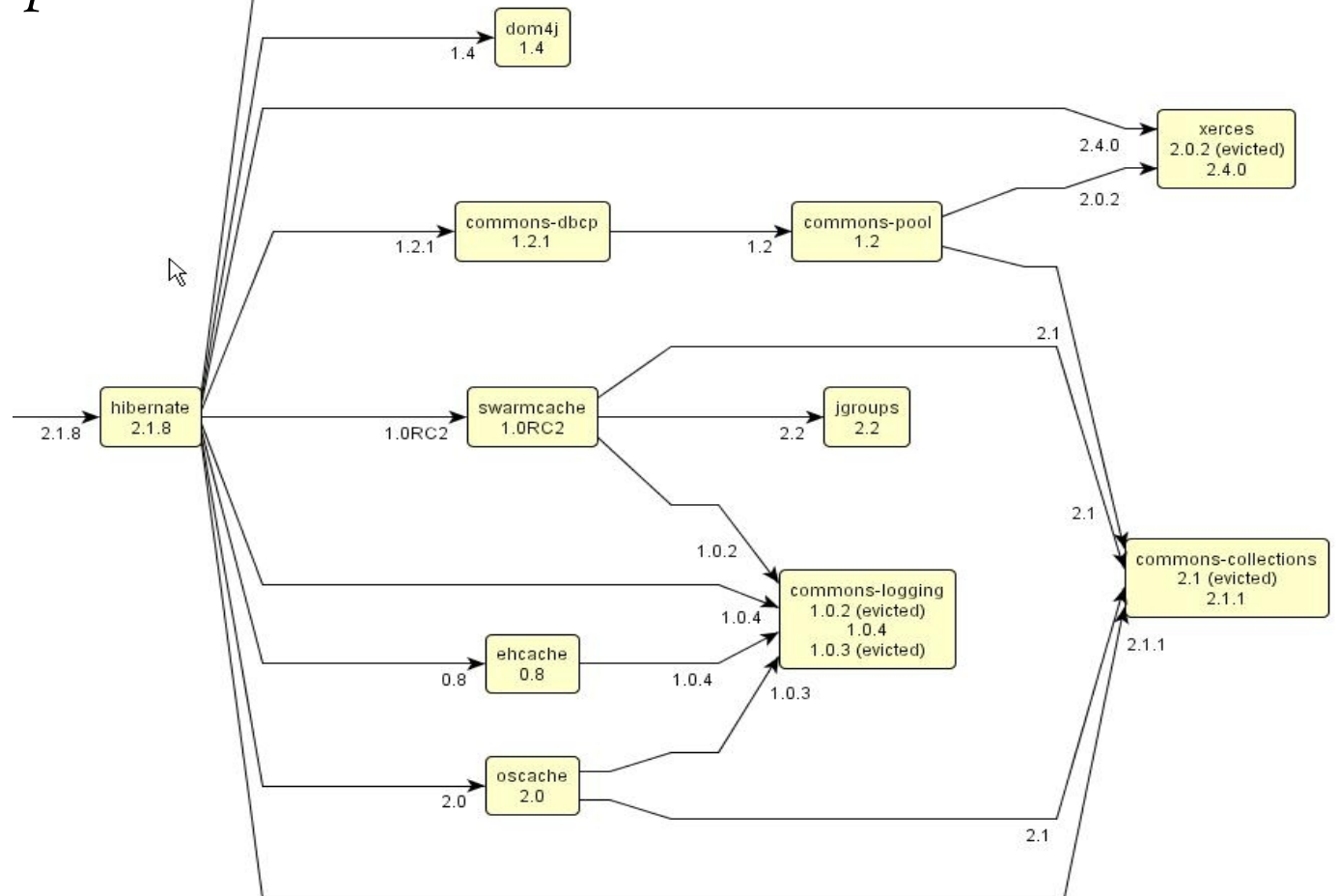
✓ Connaissance et maîtrise des dépendances

✓ transitivité

✓ Détection des incompatibilités

✓ Quelques outils :

✓ Clirr, Maven, Ivy

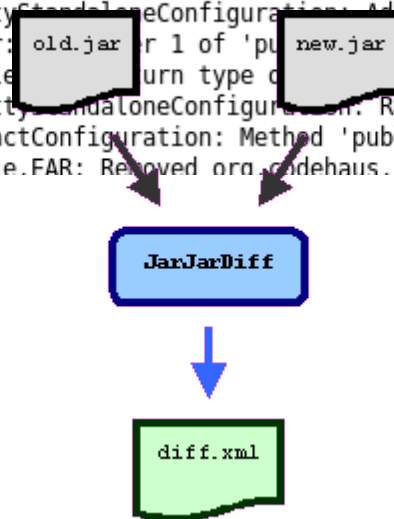


Outils d'analyse statique d'archive Java (*.jar)

- ✓ **Vérification de compatibilité binaire, rapport de changement d'API**

Clirr, JarDiff

```
ERROR: 8001: org.codehaus.cargo.deployment.DefaultJarArchive: Class org.codehaus.cargo.deployment.DefaultJarArchive removed
INFO: 8000: org.codehaus.cargo.module.DefaultJarArchive: Class org.codehaus.cargo.module.DefaultJarArchive added
ERROR: 7002: org.codehaus.cargo.container.Container: Method 'public void addDeployable(org.codehaus.cargo.container.deployable.Deployable)' has been removed, but an inheritance exists
INFO: 7011: org.codehaus.cargo.ant.ConfigurationElement: Method 'public void addConfiguredEar(org.codehaus.cargo.ant.EARElement)' has been removed, but an inheritance exists
INFO: 4000: org.codehaus.cargo.container.jetty.JettyStandaloneConfiguration: Added org.codehaus.cargo.container.configuration.StandaloneConfiguration for 1 of 'public void setConfiguration(org.codehaus.cargo.container.Configuration)'
ERROR: 7005: org.codehaus.cargo.container.Container: Method 'public void setConfiguration(org.codehaus.cargo.container.Configuration)' has been removed, but an inheritance exists
ERROR: 7006: org.codehaus.cargo.ant.ConfigurationElement: Method 'public org.codehaus.cargo.container.Configuration createConfiguration()' has been removed, but an inheritance exists
ERROR: 4001: org.codehaus.cargo.container.jetty.JettyStandaloneConfiguration: Removed org.codehaus.cargo.container.Configuration from the list of superclasses
INFO: 7003: org.codehaus.cargo.container.spi.AbstractConfiguration: Method 'public void configure()' has been removed, but an inheritance exists
ERROR: 5001: org.codehaus.cargo.container.deployable.FAR: Removed org.codehaus.cargo.util.MonitoredObject from the list of superclasses
```



Autres outils

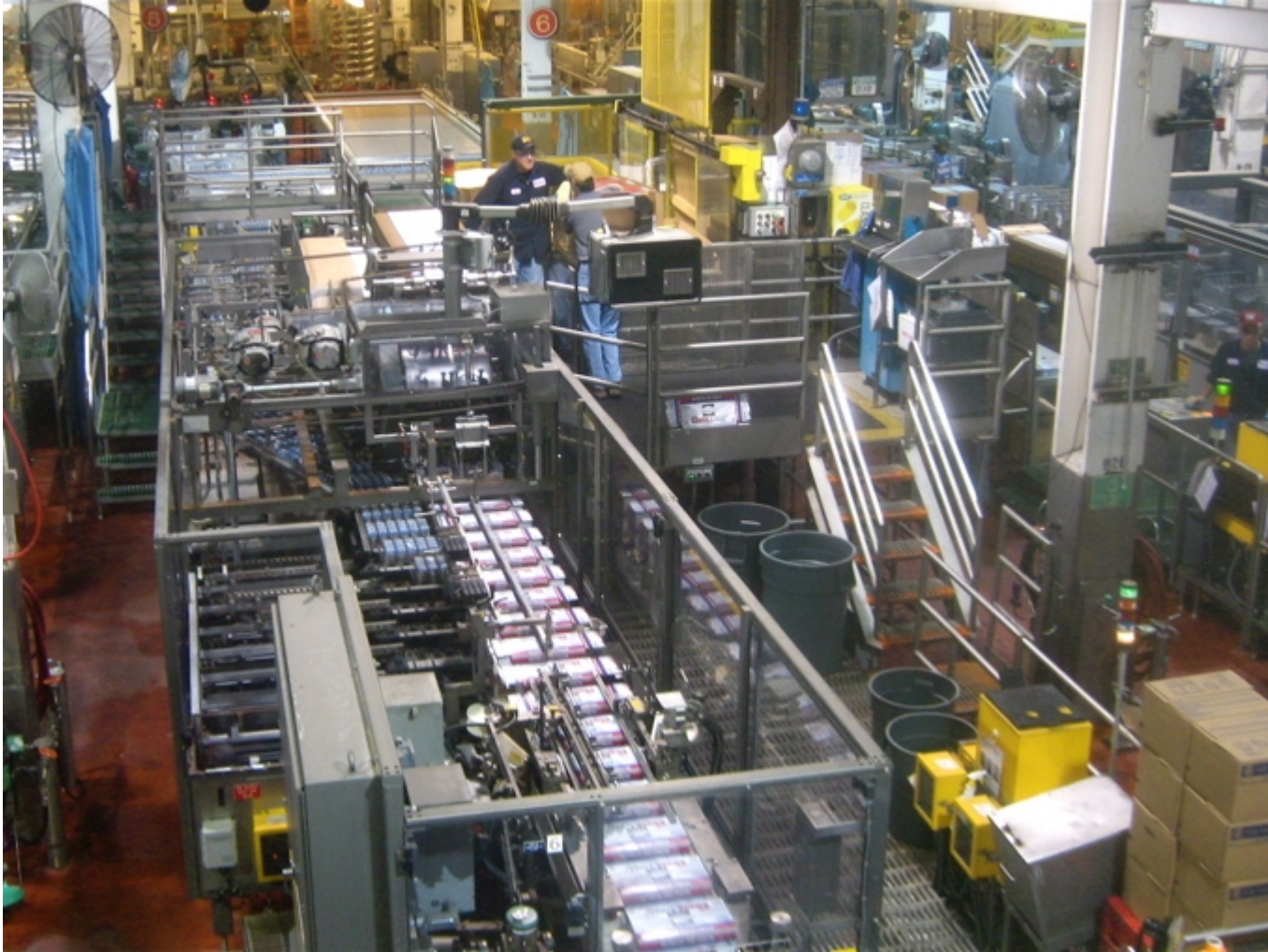
✓ Test

- ✓ Emma, Cobertura avec ant et maven,
- ✓ de nombreux plugins eclipse : EclEmma par exemple
- ✓ rapport de test
- ✓ rapport de couverture et taux de couverture de code

✓ Autres

- ✓ CVSStat
 - ✓ activité GCL (CVS, SVN)

Comment industrialiser ?



Intégration & Qualité continue

Hudson

search ?

Hudson » FlexPMD

ACTIVER LE RAFFRAICHISSEMENT AUTOMATIQUE

Ajouter une description

Nouveau job

Administrer Hudson

Personnes

Historique des builds

Editer la vue

Supprimer la vue

Relations entre les projets

Vérifier les empreintes numériques

Leader board

AlivePDF	CSAT	Cairngorm	FlexPMD	Fna-V1	Hendrix	MAX	Radon V1.0	TSO Reusable assets	Tous	Under_FlexPMD	+
S	W	Job ↓	Dernier succès		Dernier échec		Dernière durée				
		flex-pmd-ruleset-creator	3 h 9 mn (#91)		3 h 39 mn (#90)		4 mn 16 s				
		FlexPMD	3 h 29 mn (#453)		3 j 15 h (#446)		9 mn 18 s				
		FlexPMD sonar	7 j 1 h (#57)		19 j (#48)		4 mn 20 s				
		FlexPMDlight	3 h 29 mn (#315)		1 j 0 h (#310)		5 mn 23 s				

Icônes: [S](#) [M](#) [L](#)

Légende tous les builds tous les échecs pour les derniers builds seulement

XRadar

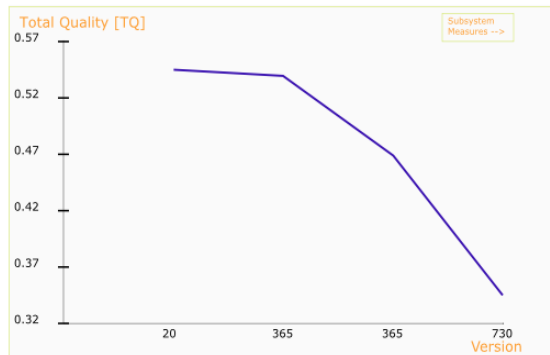
✓ « Killing Features »

- ✓ Agrège les résultats de nombreux outils d'analyses de code OSS
- ✓ Intégration dans les outils de builds Ant et Maven
- ✓ Historisation, analyse de la tendance qualité du projet:

Scorecard

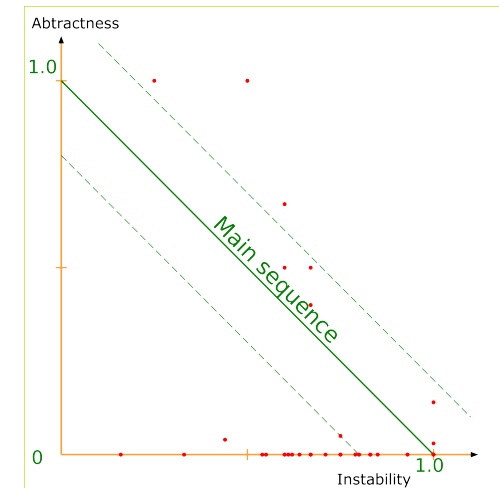
The graphs below gives the head defined metric in the system and shows how it has trended in the last versions. In the table on its side, it is shown what metric components it is made up of. Each of the subsequent graph and table is a identical view of that metric with the associated metric breakdown.

For metric details press the various categories in the submenu.



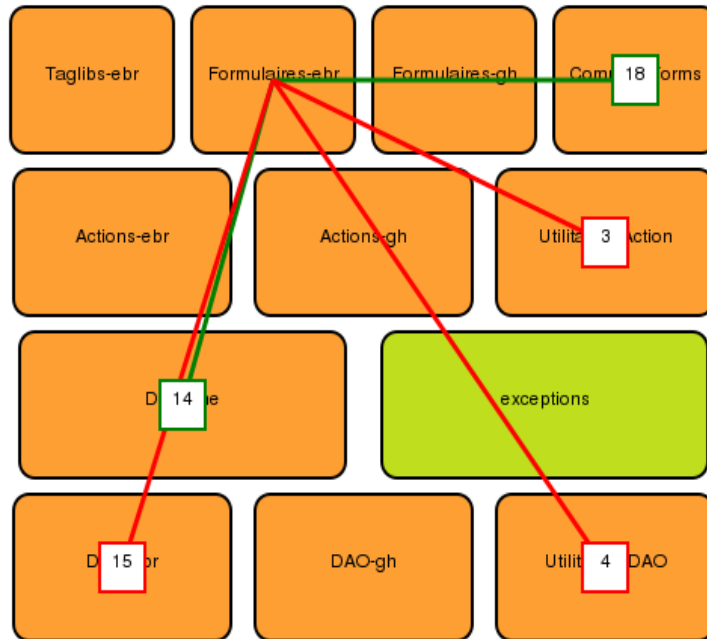
Version	Date	Total Quality [TQ]	Unit Test Suite [TS]	Architecture [ARCH]	Design [DES]	Code Quality [CODE]
20	2000.01.01	0.55	0	0.6	1	0.61
365	2001.01.01	0.55	0	0.6	1	0.59
365	2002.01.01	0.47	0.14	0.3	1	0.45
730	2004.01.01	0.34	0.07	0.09	0.94	0.26

✓ Synthèse graphique:



- ✓ Analyse complète, du plus bas niveau (code) au plus haut (architecture)

XRadar



Snapshot - Formulaires-ebr
Formulaires struts ebr

Core Metrics
 Packages = 18 (14%)
 Classes = 30 (9.7%)
 Source Statements = 1544 (9.6%)
 Cyclomatic Complexity = 585 (13.4%)

Subsystem Dependencies	Taglibs-ebr	Formulaires-ebr	Formulaires-gh	Common-Forms	Actions-ebr	Actions-gh	Utilitaires-Action	Domaine	exceptions	DAO-ebr	DAO-gh	Utilitaires-DAO	External Packages
Taglibs-ebr	0	0	0	0	0	0	1	3	0	0	0	1	11
Formulaires-ebr	0	0	0	18	0	0	3	14	0	15	0	4	30
Formulaires-gh	0	0	1	8	0	0	0	5	0	0	14	2	16
Common-Forms	0	0	0	4	0	0	1	3	0	0	2	1	10
Actions-ebr	0	24	0	26	0	0	25	52	6	58	0	19	89
Actions-gh	0	0	8	10	0	0	9	29	6	0	26	10	46
Utilitaires-Action	0	0	0	7	0	0	3	8	2	1	0	7	15
Domaine	0	0	0	0	0	0	0	22	0	0	0	0	0
exceptions	0	0	0	0	0	0	0	0	0	0	0	0	0
DAO-ebr	1	2	0	0	0	0	1	22	6	9	1	20	27
DAO-gh	0	0	1	0	0	0	2	25	5	0	7	14	29
Utilitaires-DAO	0	0	0	0	0	0	0	5	2	0	0	6	15

Aggrégateur : Sonar

Home Search projects Adobe Flex PMD

Dashboard

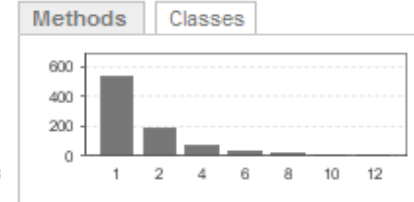
- Violations drilldown
- Time machine
- Clouds
- Hotspots
- Settings



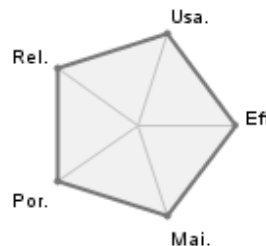
Version 0.5.6-SNAPSHOT on Jul. 21, 2009 00:09 , using profile [Sonar way](#).

Lines of code 10'115 ▲ 17'729 lines ▲ 38 packages 173 classes ▲ 856 methods ▲ 69 accessors	Comments 4.7% ▼ 501 lines ▼ 8.9% docu. API	Duplications 0.0% 0 lines 0 blocks 0 files
----------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------	----------------------------------------------------------------------

Complexity
2.1 / method
10.4 / class
1'807 cmpx ▲
2'910 statements ▲



Rules compliance
99.2%



Violations
26 ▼
Efficiency |
Maintainability |
Portability |
Reliability |
Usability |
 include opt. rules

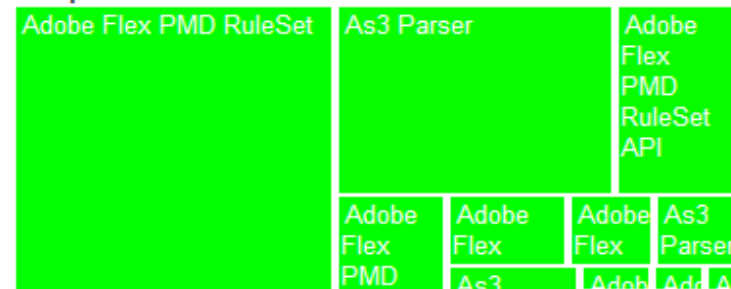
Code coverage
90.6%
355 tests ▲
+0 skipped
18.6 sec ▼

Test success
100.0%
0 failures
0 errors

Technical Debt
\$ 2'499
5 man days



Components



Size: Lines of code
Color: 0% to 100% Rules compliance

Events All ▼

Jul. 21, 2009 Version 0.5.6-SNAPSHOT

Key : com.adobe.ac.flex-pmd
Language : java

[Developer connection](#)

[Home](#)
[Sources](#)

Conclusion



Ne pas mesurer pour mesurer

- ✓ Usabilité / Lisibilité
- ✓ Maintenabilité
- ✓ Efficacité / Performance
- ✓ Portabilité
- ✓ Fiabilité
- ✓ Testabilité

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Clint	1	54	78	10	0	12	73	23	27	5	0	0	0	5	22	26	0	0	1	19
Carolyn	2	81	45	8	0	32	92	6	7	12	0	0	0	0	29	7	0	0	0	0
Clint (L/P)	3	0	0	37	24	0	0	0	0	1	19	0	0	0	0	0	0	0	0	0
Brent (L/P)	4	0	0	43	64	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Computer	5	30	78	0	0	20	1	0	0	6	0	0	0	0	0	0	0	0	0	0
Dave	6	21	12	0	0	1	39	5	0	0	0	0	0	0	0	0	0	0	0	0
Roger	7	5	3	0	0	0	1	24	0	0	0	0	0	0	0	0	0	0	0	2
Brent (P)	8	12	2	0	0	0	0	0	29	0	0	0	0	0	0	0	0	0	0	0
Patrick	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Dave (L/P)	10	0	0	2	0	0	0	0	0	0	44	2	0	0	0	0	0	0	0	0
Carolyn (L/P)	11	0	0	29	0	0	0	0	0	4	63	0	0	7	0	0	0	0	0	0
Roger (L/x)	12	0	0	1	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
Heather	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Chris	14	0	7	0	0	0	0	0	0	0	2	0	0	33	0	18	0	0	0	0
Warlord	15	4	2	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0
Computer	16	0	0	0	0	0	0	0	0	0	0	0	14	0	44	0	0	0	0	0
Computer	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Computer	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Dummy	19	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50
Computer	20	14	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	42

4 arrows - move, C - change player, P - change password, Enter - play Kentris!

✓ Ne pas mesurer pour mesurer

Facteurs clés de réussite:

- ✓ *Implication des sponsors*
- ✓ *Adhésion des développeurs,*
 - ✓ *chacun des acteurs du projet*
- ✓ *Ne pas repousser les tests et mesures en fin de projet*
- ✓ *Formaliser ses exigences*
 - ✓ *ne pas mesurer pour mesurer*
 - ✓ *intégrer la qualité à la recette aux contrats*
- ✓ *Outillage intégré et adapté*
- ✓ *Communiquer et fournir des rapports synthétiques aux décideurs.*

Références : projets et outils OS

- ✓ Ant : <http://ant.apache.org/>
- ✓ Maven 2 : <http://maven.apache.org/>
- ✓ Ivy : <http://www.jaya.free.fr/ivy/>
- ✓ Xradar: <http://xradar.sourceforge.net/>
- ✓ Sonar: <http://sonar.codehaus.org/>
- ✓ PMD : <http://pmd.sourceforge.net/>
- ✓ CheckStyle : <http://checkstyle.sourceforge.net/>
- ✓ Metrics : <http://metrics.sourceforge.net/>

- ✓ FindBugs : <http://findbugs.sourceforge.net/>
- ✓ JavaNCSS : <http://www.kçlee.de/clemens/java/javancss/>
- ✓ Jdepend : <http://clarkware.com/software/JDepend.html>
- ✓ Macker : <http://innig.net/macker/index.html>
- ✓ Ckjm : <http://www.spinellis.gr/sw/ckjm/doc/indexw.html>
- ✓ Classycle : <http://classycle.sourceforge.net/>

- ✓ ... et encore d'autre sur :
http://en.wikipedia.org/wiki/List_of_tools_for_static_co

- ✓ Junit: <http://www.junit.org/>
- ✓ Unitils: <http://www.unitils.org/summary.html>
- ✓ Emma : <http://emma.sourceforge.net/>
- ✓ Cobertura : <http://cobertura.sourceforge.net/>
- ✓ Dbunit: <http://www.dbunit.org/>
- ✓ HttpUnit: <http://httpunit.sourceforge.net/>

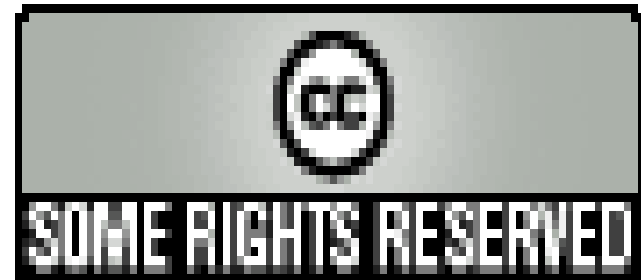
Références : biblio

- ✓ « *Effective Java: Programming Language Guide* », Joshua Bloch
- ✓ « *OO Design Quality Metrics* », Robert Martin
- ✓ « *Langage Java* », Philippe Prados

- ✓ <http://lil.univ-littoral.fr/~oumoumsack/qualite/>
- ✓ http://www.objectmentor.com/resources/articles/Principles_and_Patterns.pdf
- ✓ <http://www.infoq.com/presentations/effective-api-design>
- ✓ <http://www.infoq.com/news/Ken-Schwaber-Sacrificing-Quality>
- ✓ <http://www.quotegarden.com/programming.html>

Les photos

- ✓ <http://flickr.com/photos/slimcoincidence/594599028/>
- ✓ <http://flickr.com/photos/jeanmi/210374309/>
- ✓ <http://flickr.com/photos/cheesebikini/804099102/>
- ✓ <http://flickr.com/photos/jurvetson/101046514/>
- ✓ <http://flickr.com/photos/jeremybrooks/1399474069/>
- ✓ <http://flickr.com/photos/el%2Dcatalejo/1474055306/>
- ✓ <http://flickr.com/photos/cunaldo/271683015/>
- ✓ http://flickr.com/photos/denial_land/2094163680/
- ✓ <http://flickr.com/photos/clintjcl/382107734/>
- ✓ <http://flickr.com/photos/jurvetson/128666829/>
- ✓ <http://flickr.com/photos/opendemocracy/1417040538/>
- ✓ <http://www.flickr.com/photos/purplemattfish/3918004964/>
- ✓ <http://www.flickr.com/photos/stevendepolo/3096471361/>
- ✓ <http://www.flickr.com/photos/jamesjordan/3859026695/>
- ✓ <http://www.flickr.com/photos/summachphoto/3380254405/>
- ✓ <http://flickr.com/photos/lesloups/1907354379/>
- ✓ <http://flickr.com/photos/marysia/514900457/>
- ✓ <http://flickr.com/photos/77945684@N00/310887851/>
- ✓ <http://flickr.com/photos/invad3r/236233328/>
- ✓ http://flickr.com/photo_zoom.gne?id=1039909856&size=l





Questions / Réponses

www.parisjug.org



Copyright © 2008 ParisJug. Licence CC – Creative Commons 2.0 France – Paternité – Pas d'Utilisation Commerciale – Partage des Conditions Initiales à l'Identique



Licence



Paternité-Pas d'Utilisation Commerciale-Partage des Conditions Initiales à l'Identique
2.0 France

▪ <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>