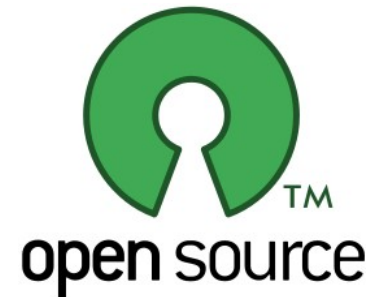


Présentation outils de Qualimétrie Open Source pour Java

Club Qualimétrie – 25/09/2007

Fabrice Bellingard - Qualixo



L'open source s'outille de lui même pour sa propre qualité

qualixo

Plan

- Les outils de qualimétrie
- Critères d'analyse
- Présentation des outils
 - Vérification de règles
 - Production de métriques
 - Automatisation
 - Détection de patterns
- Synthèse

Les grandes familles d'outils de qualimétrie

- De la qualité du produit...
 - Vérification de règles (aka « Rules Checking »)
 - programmation
 - Nommage
 - Mise en forme
 - Documentation
 - Respect de l'architecture
 - Production de métriques pour l'analyse statique
 - Production de métriques pour l'analyse dynamique
 - Détection de patterns
- ... à la qualité du processus
 - Automatisation
 - Intégration continue
 - Gestion des dépendances

Sélection d'outils abordés

- Vérification de règles
 - Checkstyle
 - PMD
 - Macker
- Production de métriques (statiques)
 - Metrics
 - JavaNCSS
 - CKJM
 - JDepend
- Production de métriques (dynamique)
 - JUnit/TestNG
 - Cobertura et Emma
- Détection de patterns
 - CPD
 - Détection de copier-coller
 - Clirr
 - Analyse de compatibilité d'API
- Automatisation
 - Ant
 - Maven

Les points à regarder

- Maturité / popularité
- Activité / réactivité de la communauté open-source
- Extensibilité et configurabilité
- Exploitation des données générées
- Couplage avec un IDE / outil de build
 - Fort : impossibilité d'en extraire la technologie
 - Faible : intégrations possibles existantes
 - Dans quels IDEs ?
 - Dans quels outils de build ?
- Type de licence
 - Dans notre cas, seulement des licences approuvées par l'OSI

Vérification de règles

Checkstyle

PMD

Macker





- Rules Checking Java
 - Outil couvrant un spectre de règle étendu
 - Intégrable à de nombreux IDE dont Eclipse

Avantages



- Spectre de règle relativement large
 - Mise en forme
 - Nommage
 - Documentation
 - Programmation
- Connus et reconnus
 - Outil très largement employé au delà du monde de l'Open Source
- Communauté très réactive
- Plugin Eclipse très avancé et réactif
- Nombreuses extensions et personnalisations possibles

Inconvénients



- Création de nouvelles règles requiert un développement Java
- Règles plus génériques que PMD



Checkstyle : illustration

```
/**
 * This class provides the functionality to check a set of files.
 * @author Oliver Burn
 * @author <a href="mailto:stephane.bailliez@wanadoo.fr">Stephane Bailliez</a>
 * @author lkuehne
 */
public class Checker extends AutomaticBean
    implements MessageDispatcher
{
    /** maintains error count */
    private final int counter =
        new Set<Integer>(new Integer[] { Integer.valueOf(Integer.TYLevel.ERROR)});

    /** vector of listeners */
    private final ArrayList mListeners = new ArrayList();

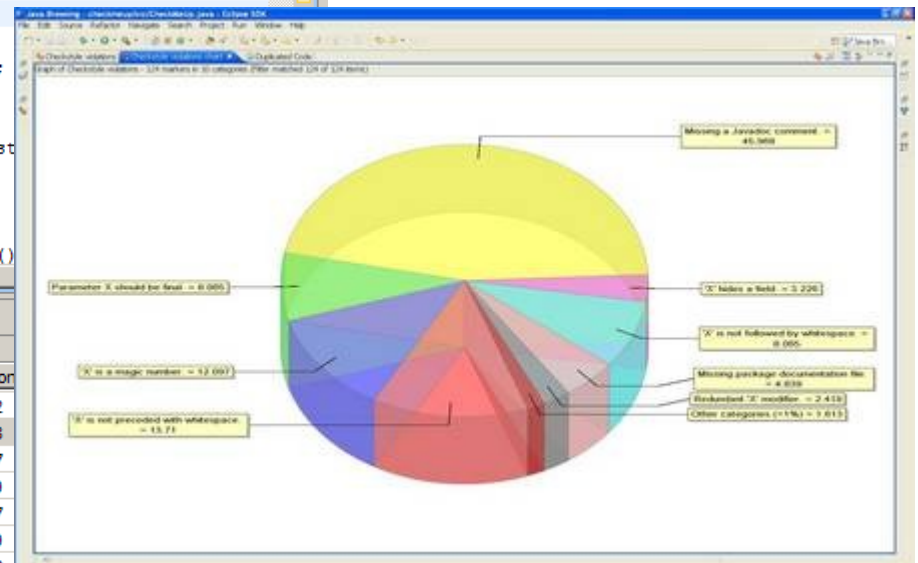
    /** vector of fileset checks */
    private final ArrayList mFileSetChecks = new ArrayList();

    /** class loader to resolve classes with. */
    private ClassLoader mLoader =
        Thread.currentThread().getContextClassLoader();
}
```



Problems | Javadoc | Declaration | Error Log

11 errors, 89 warnings, 0 infos (Filter matched 100 of 8,682 items)

Description	Resource	In Folder	Location
'/' should be on the previous line.	Checker.j...	checkstyle/src/checkstyle/...	line 52
First sentence should end with a peri...	Checker.j...	checkstyle/src/checkstyle/...	line 53
First sentence should end with a peri...	Checker.j...	checkstyle/src/checkstyle/...	line 57
First sentence should end with a peri...	Checker.j...	checkstyle/src/checkstyle/...	line 60
First sentence should end with a peri...	Checker.j...	checkstyle/src/checkstyle/...	line 67
First sentence should end with a peri...	Checker.j...	checkstyle/src/checkstyle/...	line 70
First sentence should end with a peri...	Checker.j...	checkstyle/src/checkstyle/...	line 72
First sentence should end with a peri...	Checker.j...	checkstyle/src/checkstyle/...	line 75





- Rules Checking Java/JSP/JSF
 - Outil
 - Polyvalent
 - Dans l'esprit d'un détecteur de mauvaises pratiques
 - Intégrable à de nombreux IDE dont Eclipse
- Avantages
 -  – Nombreuses règles mais spectre plus réduit
 - Règles souvent plus pointues et plus précises
 - Possibilité d'adapter des règles à ses propres règles
 - Création de nouvelles règles sans nouveau développement
 - XSL / XPATH
- Inconvénients
 - Pas de règles de mise en forme
 - Règles de nommage limitées
 - Nécessité de connaître XPath pour l'écriture de nouvelles règles
 - Requête XPath sur l'arbre du code source
 -  – Plugin Eclipse moins avancé
 - bien intégré, mais réactivité plus limité



PMD : illustration

The screenshot shows the Eclipse IDE interface. The Package Explorer on the left shows a project structure for 'pmd' with a sub-project 'pmd-eclipse'. The main editor window displays the source code for 'PMDPlugin.java'. The code defines a class 'PMDPlugin' that extends 'AbstractUIPlugin'. It includes a static instance 'plugin', a resource bundle, and a plugin ID. The constructor 'PMDPlugin()' calls 'super()' and assigns 'this' to 'plugin'. The Problems view at the bottom shows 21 errors, 0 warnings, and 0 infos. The errors are listed in a table below.

Description	Resource	In Folder	Location
Possible unsafe assignment	PMDPlugin	pmd-eclipse/src/net/s	line 17
Avoid excessively long variable names	PMDPlugin	pmd-eclipse/src/net/s	line 19
Found non-transient, non-static field	PMDPlugin	pmd-eclipse/src/net/s	line 19
A signature (constructor or method) is redundant	PMDPlugin	pmd-eclipse/src/net/s	line 33
A signature (constructor or method) is redundant	PMDPlugin	pmd-eclipse/src/net/s	line 40
Assigning an Object to null is allowed only in an if-statement	PMDPlugin	pmd-eclipse/src/net/s	line 42
Assigning an Object to null is allowed only in an if-statement	PMDPlugin	pmd-eclipse/src/net/s	line 43
A method argument that is redundant	PMDPlugin	pmd-eclipse/src/net/s	line 57

- Respect Architecture

- Une fois définie, vérifie que l'architecture définie sur un projet est respecté et que les appels entres classes sont autorisés

- Avantages



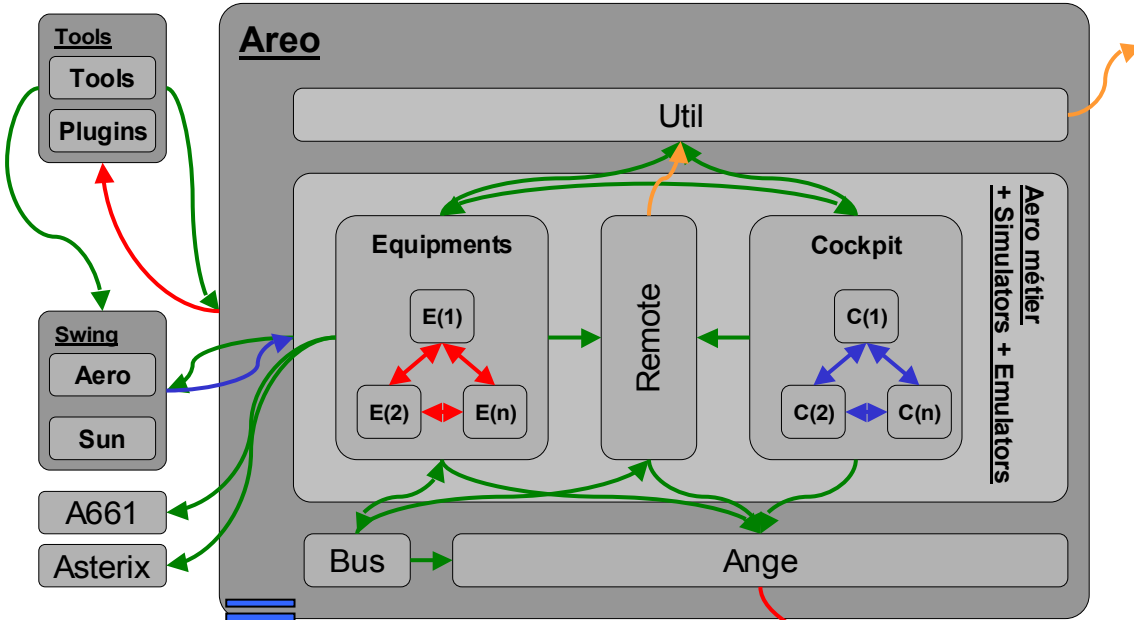
- Outil puissant avec peu d'équivalent
- Configuration relativement aisée
- Automatisable via Ant

- Inconvénients

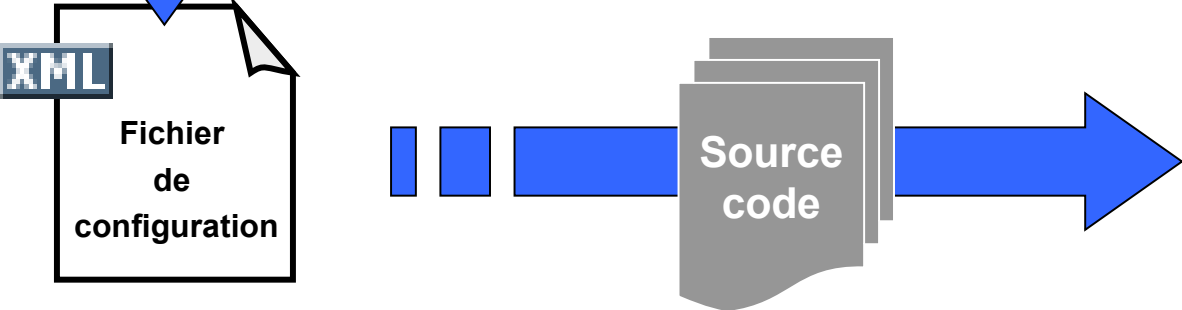


- Ne descend pas plus bas que le niveau classe
- Fichier de configuration à maintenir à jour en fonction de l'évolution du projet
- Pas de plugin dans les IDE

Macker : illustration



Organisation de l'architecture en couche





Appels incorrects		
Depuis	Vers	NB
Bus		1
	plugins	1
emulators		131
	plugins	3
	remote	3
	swing	125
equipments		14
	plugins	1
	tools	13
remote		78
	equipments	27
	tools	9
	util	42
simulators		48
	Bus	3
	plugins	14
	remote	4
	swing	27
swing		26
	cockpit	16
	remote	10
tools		6
	Autre couche	6
util		184
	Ange	11
	Bus	2
	cockpit	2
	equipments	8
	remote	120
	swing	41
Total		488

Synthèse des signalements obtenus avec si besoin la référence aux méthodes incriminées

Production de métriques

Statique

Dynamique

- Calcul de métriques procédurales
 - Calcule un set restreint de métriques basées essentiellement sur la volumétrie
- Avantages
 - Répandu et fiabilité éprouvée
 -  – Robuste et tolérant aux erreurs
 - Java 1.5 supporté
 - Calcul peu de métriques mais les calcule bien
 - Existe depuis les premières versions de Java 1997
 - Plug-ins à de nombreux outils
- Inconvénients
 -  – Ne travaille que sur le code source et pas sur le bytecode
 - Ouverture prévue au niveau classes et packages, mais ne prévoit pas de s'ouvrir à plus de métriques



JavaNCSS : illustration

Last published: 03 December 2004 | Doc for 1.0.3-dev Turbine | Fulcrum

Overview

- Main
- Overview
- ▶ Lifecycle
- ▶ Services
- How To
- Todo's

Project Documentation

- About Fulcrum's YAAFI
- ▶ Project Info
- ▶ Project Reports
 - Metrics
 - Change Log
 - Changes
 - Developer Activity
 - File Activity
 - Unit Tests
 - Dependency
 - Convergence
 - PMD Report
 - Source Xref
 - Test Xref
 - JavaDocs
 - JavaDoc Report
 - JavaDoc Warnings Report
 - JavaNCSS Development Process

built by maven

JavaNCSS Metric Results

[packages] [objects] [functions] [explanations]

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

Packages

[packages] [objects] [functions] [explanations]

Packages sorted by NCSS

Package	Classes	Functions	NCSS	Javadocs	Javadoc Lines	Single Comment Lines	Multi Comment Lines
org.apache.fulcrum.yaafi.framework.container	5	69	570	72	280	46	90
org.apache.fulcrum.yaafi.cli	2	13	65	13	56	6	32
org.apache.fulcrum.yaafi.framework.factory	1	3	50	4	15	11	18
org.apache.fulcrum.yaafi.service.baseservice	2	8	38	10	41	1	33
org.apache.fulcrum.yaafi.service.systemproperty	2	2	24	4	19	0	32
org.apache.fulcrum.yaafi.service.servicemanager	1	7	23	7	23	6	17
Classes total	13	102	770	110	434	70	222

Objects

[packages] [objects] [functions] [explanations]

TOP 30 classes containing the most NCSS

Object	NCSS	Functions	Classes	Javadocs
org.apache.fulcrum.yaafi.framework.container.ServiceContainerImpl	287	31	0	31
org.apache.fulcrum.yaafi.framework.container.ServiceComponentImpl	203	31	0	31
org.apache.fulcrum.yaafi.cli.Main	45	11	0	10
org.apache.fulcrum.yaafi.framework.factory.ServiceManagerFactory	44	3	0	4
org.apache.fulcrum.yaafi.service.baseservice.BaseServiceImpl	21	8	0	9
org.apache.fulcrum.yaafi.service.servicemanager.ServiceManagerService	17	7	0	7
org.apache.fulcrum.yaafi.service.systemproperty.SystemPropertyServiceImpl	17	2	0	3
org.apache.fulcrum.yaafi.framework.container.ServiceConstants	11	0	0	1
org.apache.fulcrum.yaafi.cli.Shutdown	10	2	0	3
org.apache.fulcrum.yaafi.framework.container.ServiceComponent	8	7	0	8
org.apache.fulcrum.yaafi.framework.container.ServiceContainer	1	0	0	1
org.apache.fulcrum.yaafi.service.baseservice.BaseService	1	0	0	1

- Calcul de métriques orientés objets
 - Calcule les métriques de Chidamber et Kemerer

- Avantages



- Travaille sur le byte code généré
 - Java 1.5 supporté
- Auteur réactif et reconnu dans le milieu de la qualité
 - Nombreux ouvrages à son actif
- Solution régulièrement maintenue
- Calcul peu de métriques mais les calcule bien
- Automatisable via Ant

- Inconvénients



- Moins connu que les autres outils
- Communauté plus faible
- Pas de plugins pour les IDE

CKJM : illustration

CKJM Chidamber and Kemerer Java Metrics

Designed for use with [CKJM](#) and [Ant](#).

Top 25: wmc

[\[wmc\]](#) [\[dit\]](#) [\[noc\]](#) [\[cbo\]](#) [\[rfc\]](#) [\[lcom\]](#) [\[ca\]](#) [\[npm\]](#) [\[explanations\]](#)

name	wmc	dit	noc	cbo	rfc	lcom	ca	npm
org.sqljdbc.jdbcDatabaseMetaData	172	1	0	6	197	14692	1	165
org.sqljdbc.jdbcResultSet	147	1	0	18	198	10333	5	139
org.sqljdbc.Table	142	2	2	41	330	8117	46	24
org.sqljdbc.Expression	90	1	0	22	190	581	18	2
org.sqljdbc.jdbc.CallableStatement	83	3	0	7	121	3397	1	81
org.sqljdbc.Server	72	1	1	23	187	1896	5	45
org.sqljdbc.DatabaseCommandInterpreter	72	1	0	44	301	0	2	1
org.sqljdbc.Library	71	1	0	6	139	2473	5	64
org.sqljdbc.Result	60	1	0	12	104	1054	36	27
org.sqljdbc.Parser	60	1	0	27	191	842	4	0
org.sqljdbc.Database	59	1	0	29	149	1387	35	18
org.sqljdbc.Session	59	1	0	29	157	1097	42	14
org.sqljdbc.jdbc.PreparedStatement	51	2	1	22	129	1111	2	45
org.sqljdbc.util.DatabaseManagerSwing	47	6	0	12	244	849	5	18
org.sqljdbc.jdbc.Connection	46	1	0	16	93	837	8	38
org.sqljdbc.DITableInfo	43	1	0	9	90	0	1	0
org.sqljdbc.DITypeInfo	42	1	0	3	63	0	4	0
org.sqljdbc.Column	42	1	0	14	131	769	23	7
org.sqljdbc.jdbc.Statement	41	1	1	8	62	654	3	37
org.sqljdbc.lib.ArrayUtil	40	1	0	1	55	778	19	36
org.sqljdbc.Index	40	1	0	12	67	396	16	2
org.sqljdbc.util.SqlFile	37	1	0	8	175	402	6	6
org.sqljdbc.HsqlDateTime	35	1	0	1	73	477	12	24
org.sqljdbc.util.TableSorter	34	2	0	8	79	377	5	19
org.sqljdbc.lib.DoubleIntIndex	34	1	0	2	40	0	2	21

Top 25: dit

[\[wmc\]](#) [\[dit\]](#) [\[noc\]](#) [\[cbo\]](#) [\[rfc\]](#) [\[lcom\]](#) [\[ca\]](#) [\[npm\]](#) [\[explanations\]](#)

name	wmc	dit	noc	cbo	rfc	lcom	ca	npm
org.sqljdbc.util.ConnectionDialogSwing	8	6	0	3	87	14	1	5
org.sqljdbc.util.ZaurusConnectionDialog	11	6	0	1	55	43	1	7
org.sqljdbc.util.DatabaseManagerSwing	47	6	0	12	244	849	5	18
org.sqljdbc.util.ZaurusDatabaseManager	11	6	0	7	105	25	0	7
org.sqljdbc.util.FontDialogSwing	8	6	0	9	68	2	8	6
org.sqljdbc.util.QueryTool	21	5	0	2	102	170	0	12
org.sqljdbc.util.Transfer	31	5	0	8	143	269	2	14
org.sqljdbc.rowio.RowInputTextQuoted	3	5	0	3	17	1	1	2
org.sqljdbc.util.ConnectionDialog	11	5	1	4	70	35	5	4
org.sqljdbc.rowio.RowOutputTextQuoted	3	5	0	2	10	3	1	1
org.sqljdbc.util.DatabaseManager	32	5	1	7	190	336	1	16
org.sqljdbc.rowio.RowInputTextLog	23	4	0	10	52	223	1	8
org.sqljdbc.util.SqlFile\$ContinueException	2	4	0	2	4	0	1	2

- Calcul des dépendances entre classes et détection des cycles
 - Couplages afférents/efférents, taux d'abstraction et d'instabilité, distance à la normale
 - Extensibilité, réutilisabilité et maintenabilité

- Avantages



- Données facilement exploitables
- Très bonne intégration dans Eclipse
- Automatisable via Ant et Maven

- Inconvénients



- Communauté plus faible



JDepend : illustration

JDepend - Eclipse Platform

File Edit Navigate Search Project Run Window Help

Run JDepend again

Dependencies

Selected object(s)

Package	CC(con...)	AC(abs...)	Ca(aff.)	Ce(eff.)	A	I	D	Cycle!
d.da	1	0	1	3	0.0	0.75	0.25	⚠

Packages with cycle

Package	CC(con...)	AC(abs...)	Ca(aff.)	Ce(eff.)	A	I	D	Cycle!
a	1	0	1	2	0.0	0.66	0.33	⚠
b	1	0	3	2	0.0	0.40	0.60	⚠
d	1	0	2	2	0.0	0.50	0.50	⚠
d.da	1	0	1	3	0.0	0.75	0.25	⚠
d.db	1	0	1	3	0.0	0.75	0.25	⚠

Depends upon - efferent dependencies

Package	CC(con...)	AC(abs...)	Ca(aff.)	Ce(eff.)	A	I	D	Cycle!
d	1	0	2	2	0.0	0.50	0.50	⚠
d.db	1	0	1	3	0.0	0.75	0.25	⚠



Used by - afferent dependencies

Package	CC(con...)	AC(abs...)	Ca(aff.)	Ce(eff.)	A	I	D	Cycle!
d.db	1	0	1	3	0.0	0.75	0.25	⚠

Instability

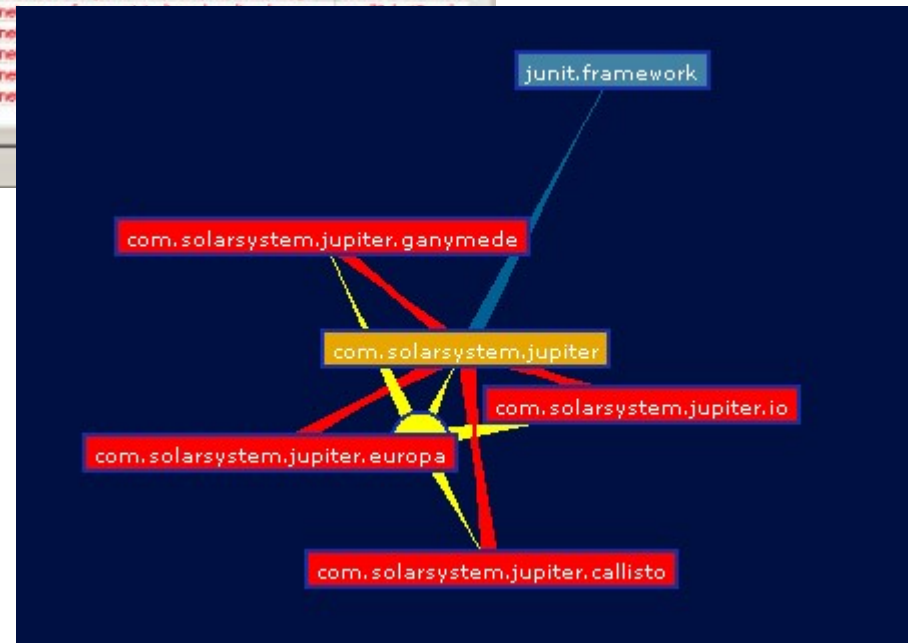
Abstra



- Plugin Eclipse permettant de calculer des métriques
 - Structurelles
 - Orientées Objets
- Avantages
 -  – Excellente intégration à Eclipse
 - Donne les moyennes et maximum au niveau projet
 - Permet de plonger jusqu'au attributs et méthodes de classes
 - Donne une représentation graphique de son analyse des dépendances
- Inconvénients
 -  – Plugin Eclipse seulement : impossibilité de récupérer les métriques automatiquement
 - Représentation graphique des dépendances pas évidente à exploiter

Metrics : illustration

Metric	Total	Mean	Std. Dev.	Maximum	Resource causing Maximum	Method
Number of Packages	16					
Number of Methods (avg/max per type)	1310	6.65	8.553	76	/net.sourceforge.metrics/tgsrc/com/touchgrap...	
+ tgsrc	489	7.191	11.544	76	/net.sourceforge.metrics/tgsrc/com/touchgrap...	
src	761	6.238	6.553	45	/net.sourceforge.metrics/src/net/sourceforge/...	
+ net.sourceforge.metrics.core.sources	108	15.429	12.129	45	/net.sourceforge.metrics/src/net/sourceforge/...	
+ net.sourceforge.metrics.ui	77	9.625	10.111	33	/net.sourceforge.metrics/src/net/sourceforge/...	
+ net.sourceforge.metrics.core	198	6.6	7.093	27	/net.sourceforge.metrics/src/net/sourceforge/...	
+ net.sourceforge.metrics.ui.preferences	52	6.5	7.467	26	/net.sourceforge.metrics/src/net/sourceforge/...	
+ net.sourceforge.metrics.ui.dependencies	95	5.588	3.727	15	/net.sourceforge.metrics/src/net/sourceforge/...	
+ net.sourceforge.metrics.internal.persistence	18	4.5	4.33	12	/net.sourceforge.metrics/src/net/sourceforge/...	
+ net.sourceforge.metrics.internal.prevayler.implementa...	54	5.4	2.871	10	/net.sourceforge.metrics/src/net/sourceforge/...	
+ net.sourceforge.metrics.internal.xml	41	4.1	2.022	9	/net.sourceforge.metrics/src/net/sourceforge/...	
+ net.sourceforge.metrics.calculators	79	4.158	2.254	8	/net.sourceforge.metrics/src/net/sourceforge/...	
+ net.sourceforge.metrics.propagators	31	5.167	1.067	7	/net.sourceforge.metrics/src/net/sourceforge/...	
+ net.sourceforge.metrics.internal.tests	8	2.667	1.886	4	/net.sourceforge.metrics/src/net/sourceforge/...	
+ net.sourceforge.metrics.internal.prevayler	0	0	0	0		
+ classycle	60	8.571	2.556	13	/net.sourceforge.metrics/classycle/classycle/g...	
Lines of Code (avg/max per type)	6593	33.467	49.02	339	/net.sourceforge.metrics/tgsrc/com/touchgrap...	
Number of Interfaces (avg/max per packageFragment)	16	1	1.414	4	/net.sourceforge.metrics/src/net/sourceforge/...	
Lines of Code (avg/max per method)	6593	4.812	7.355	69	/net.sourceforge.metrics/classycle/classycle/g...	calculateAttributes
+ classycle	324	5.4	9.94	69	/net.sourceforge.metrics/classycle/classycle/g...	calculateAttributes
+ tgsrc	2321	4.661	8.278	59	/ne	
src	3948	4.862	6.473	52	/ne	
+ net.sourceforge.metrics.ui	544	6.8	8.707	52	/ne	
+ MetricsTable.java	194	10.778	13.831	52	/ne	
+ MetricsTable	194	10.778	13.831	52	/ne	
+ setMetrics	52					





- Framework d'écriture de tests unitaires
 - Facilite la vérification de la non régression du code
 - Standard de facto
 - Incontournable du développement en Java
 - Copié sur d'autres plateformes (JUnit en .NET)
 - TestNG : challenger, créé à la base pour Java 5

- Avantages



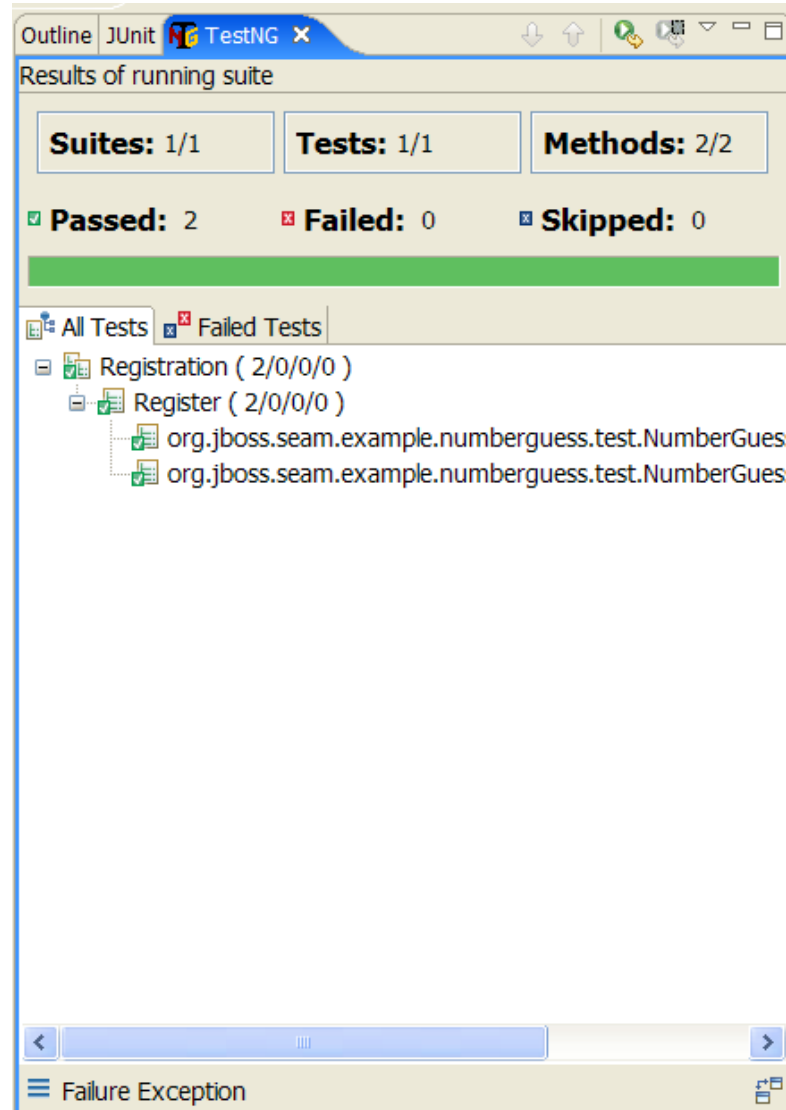
- Technologie très répandue et fiabilité éprouvée
- Principe très simple et minimaliste mais très puissant
- Intégration dans tous les IDEs Java
- De multiples extensions existent
 - Base de données, couche HTTP, couche Struts, ...



- Inconvénients

- L'écriture des tests peut se révéler plus ou moins complexe
 - Isolation du code à tester (bouchonnage)

JUnit & TestNG : illustration



Cobertura

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

- Calcul de la couverture des tests unitaires
 - Se greffe sur l'exécution de tests JUnit pour donner le pourcentage de code effectivement testé
 - Permet de vérifier la validité des tests unitaires
 - Indispensable
 - Autres technologies OSS : Emma, JCoverage

- Avantages



- Utilisation triviale
- Permet de détecter du code potentiellement mort
- Quelques intégrations au niveau des IDEs (Eclipse)

- Inconvénients



- Dérive possible : vouloir un 100% de couverture est une utopie



GPL

Cobertura : illustration

Coverage Report - All Packages

Package	# Classes	Line Coverage	Branch Coverage	Complexity
All Packages	9	60%	55%	1.75
com.beer.business.data	2	90%	100%	2.143
com.beer.business.domain	2	75%	N/A	1
com.beer.business.service	2	100%	100%	1.167
com.beer.common	2	67%		
com.beer.web	1	0%		

Report generated by Cobertura 1.8 on 1/21/07 3:48 PM.

Classes in this File

Class	Line Coverage	Branch Coverage	Complexity
HierarchyBuilder	59%	75%	2.667

```
1 package com.vanward.adana.hierarchy;
2 /**
3  *
4  */
5 public class HierarchyBuilder {
6
7     private HierarchyBuilder() {
8         super();
9     }
10
11     /**
12     * @param clzName
13     * @return Hierarchy
14     * @throws ClassNotFoundException
15     */
16     public static Hierarchy buildHierarchy(final String clzName)
17         throws ClassNotFoundException{
18         // don't initialize classes
19         final Class clzz = Class.forName(clzName, false,
20             HierarchyBuilder.class.getClassLoader());
21         return buildHierarchy(clzz);
22     }
23
24     /**
25     * @param clzz
26     * @return Hierarchy
27     */
28     public static Hierarchy buildHierarchy(Class clzz){
29
30         if(clzz == null){
31             throw new RuntimeException("Class parameter can not be null");
32         }
33
34         final Hierarchy hier = new Hierarchy();
35         hier.setBaseClass(clzz);
36
37         final Class superclass = clzz.getSuperclass();
38
39         if(superclass != null && superclass.getName().equals("java.lang.Object")){
40             return hier;
41         }
42     }
43 }
```

Détection de patterns



- Détection de copier coller
 - Support spécifique pour langages Java/JSP, C/C++
 - Possibilité d'ignorer les littérales, les noms de variables, etc.
 - Seuil minimum de détection
 - Détection générique dans tout fichier au format texte

- Avantages



- Intégration avec Ant
- Très rapide pour des résultats pertinents

- Inconvénients



- Essentiellement une analyse textuelle
- Moins puissant que des outils type Simian
 - Encore un peu jeune
- Pas d'intégration dans des IDE
- Requiert des fichiers bien formés pour certains langages
 - Ex : fichiers xml-compliant pour les JSP





- Comparaison de la compatibilité d'API entre 2 versions d'un JAR
 - Recommandé pour le développement de bibliothèques/frameworks qui exposent des APIs
 - Intéressant aussi dans tout développement pour vérifier les changements de structure interne
- Avantages
 - Permet d'assurer :
 - Pour les développeurs : une évolution sereine des APIs d'un composant (voire d'une application)
 - Pour les utilisateurs : une connaissance parfaite des changements d'API lors d'une montée de version d'un composant
 - utilisation d'une numérotation adaptée
 - Intégré dans les outils de build connus (Ant, Maven)
- Inconvénients
 - Ne permet pas de juger de la qualité du code au fil du temps
 - Utilisé généralement seulement entre 2 versions n et n-1



Automatisation



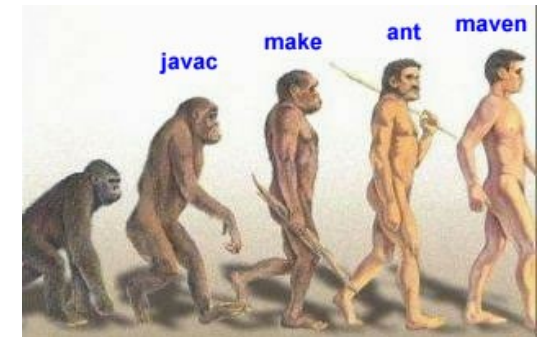
- Automatisation du build et du passage des outils
 - Standard Java de facto pour gérer
 - Le build
 - L'intégration
 - Le passage d'outils sur du code source
 - Équivalent de l'outil « make » en C/C++
 - Copié dans sur d'autres plateformes (NAnt sur .NET)
- Avantages
 - Technologie répandue et fiabilité éprouvée
 - Très massivement utilisé
 -  – Outil réactif et forte communauté
 - S'interface avec la très grande majorité des outils java du marché
- Inconvénients
 - Puissant mais scripts XML parfois verbeux
 -  – Manque de réutilisabilité
 - Gestion des dépendances manuelle
 - Transitivité et compatibilité des dépendances (← projet Ivy)

- Automatisation du build et du passage des outils
 - Basé sur la définition d'une carte d'identité du projet
 - Moteur d'exécution de plugins mis à disposition via un référentiel central



- Avantages

- Technologie répandue et fiabilité éprouvée
- Outil réactif et forte communauté
- Plus haut niveau qu'Ant
 - Gestion efficace des dépendances
- Forte réutilisabilité et partage des meilleures pratiques



- Inconvénients



- Prise en main plus complexe qu'Ant
 - Mais retour sur investissement plus rapide et important
- Moins bonne intégration dans les IDEs

On aurait pu aborder...

- D'autres outils
 - vérification de règles
 - Jlint
 - FindBugs
 - Production de métriques statiques
 - MetricsReloaded
 - Production de métriques dynamiques
 - JMeter, JMeasurement
 - Les analyseurs de dépendances
 - DependencyFinder
 - Classycle
- En marge de la qualimétrie
 - L'intégration continue / ordonnancement
 - CruiseControl
 - Continuum
 - Luntbuild
 - L'agrégation de résultats
 - XRadar : intranet statique à base de feuilles XSL sur les résultats
 - QALab : statique sur format propre

Récapitulatif

	Outil	maturité	Popularité	Activité	Réactivité	configuration	Extension	Couplage		Licence
V.Règles	Checkstyle	++	+++	++	++	++	+	++		LGPL
	PMD	++	++	++	++	++	++	++		BSD
	Macker	+	--	-	+	++	=	+		GPL
Métriques S.	Metrics	+	++	=	=	+	+	--		CPL
	JavaNCSS	++	+	++	+	=		+		LGPL
	CKJM	+	-	+	+	-	+	+		~BSD
	JDepend	++	++	++	+	=		++		BSD
Métriques D.	JUnit	++	+++	++	++	++	++	++		CPL
	TestNG	++	+	++	++	++	++	++		Apache
	Cobertura	+	++	+	+	+		++		GPL
	Emma	+	++	+	+	+		++		CPL
Patterns	CPD	~	+	++	+	+	++	+		BSD
	Clirr	+	--	=	=	+		+		LGPL
Auto.	Ant	+++	+++	+++	+++	+++	+++	++		Apache
	Maven	+++	+++	+++	++	+++	+++	+		Apache

Conclusion

- Monde riche
 - Couvre un large spectre du domaine de la qualité
 - Qualité du produit
 - Qualité des processus
 - Très orienté projets/développeurs
- En maturation
 - Manque de suivi
 - Efforts redondants
 - Manque de concertation/coordination
 - Efforts en cours
- Des manques
 - Outil d'agrégation de métriques
 - Outil de cartographie/valorisation du patrimoine applicatif



Merci de votre attention

- Questions
- Commentaires
- Remarques

