THE
DEVELOPER'S
CONFERENCE

Melhorando o design de código através de **Metáforas**

Alessandro Dias
e-Core - UniRitter
@alessandro_dias

Guilherme Lacerda
Wildtech – Unisinos
@guilhermeslac

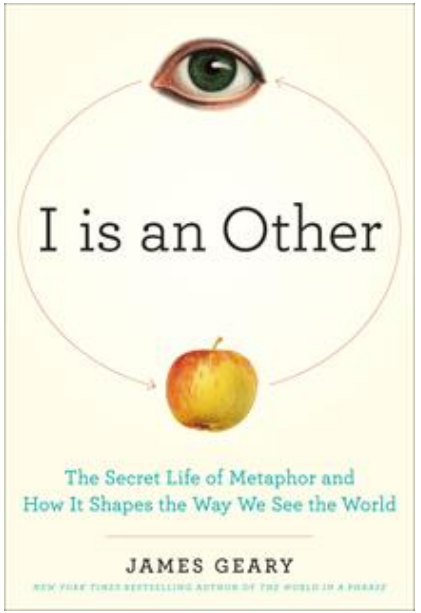
Quem somos?

“Todas as religiões, artes e ciências são ramos da mesma árvore.”

Albert Einstein

O que é metáfora?

O Uso



A metáfora não é só uma figura de linguagem

Ela está presente de forma intensa porém imperceptível em tudo que os humanos fazem

O Uso

Nós falamos uma metáfora a cada 10 a 25 palavras

Ou seja, cerca de seis metáforas por minuto

A metáfora é uma maneira de pensar muito antes de ser um estilo com palavras

“No silver bullet.”
Fred Brooks Jr.

Metáforas no desenvolvimento de software

Metáforas no **Design de Código**

THE EVOLUTION OF

SOFTWARE ARCHITECTURE

1990's

SPAGHETTI-ORIENTED
ARCHITECTURE
(aka Copy & Paste)

2000's

LASAGNA-ORIENTED
ARCHITECTURE
(aka Layered Monolith)

2010's

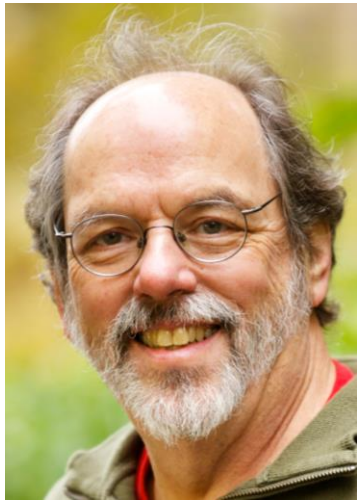
RAVIOLI-ORIENTED
ARCHITECTURE
(aka Microservices)

WHAT'S NEXT?

PROBABLY PIZZA-ORIENTED ARCHITECTURE

By @benorama

Technical Debt

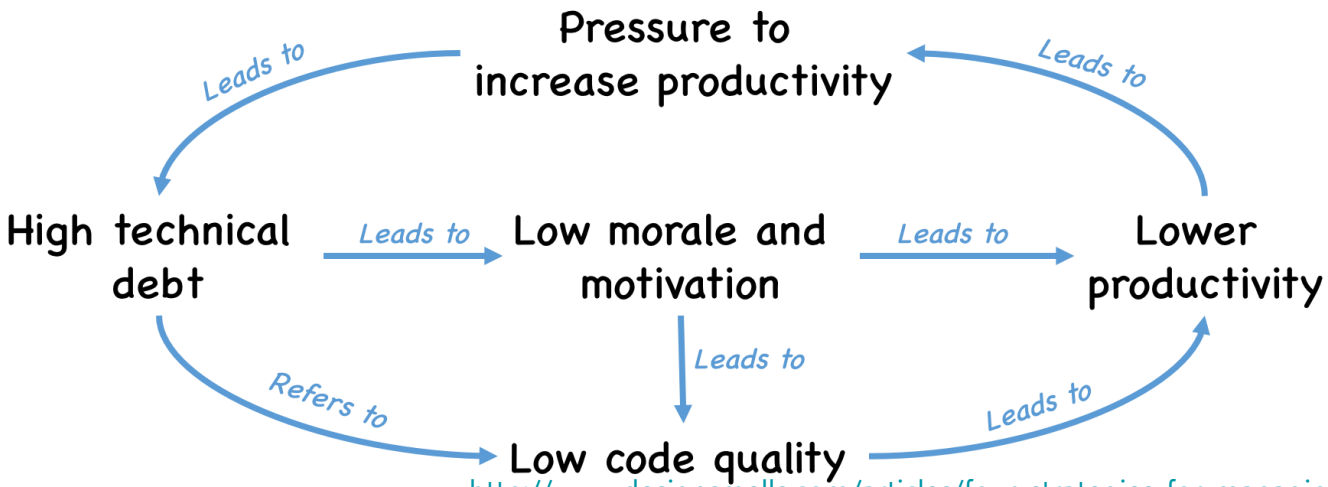


Criada por Ward Cunningham

Originária do setor financeiro

Alguns Tipos

Arquitetura/Design, Código, Testes e Build



http://www.designsmells.com/articles/four-strategies-for-managing-technical-debt/

Atividades

Identificação, medição, priorização, monitoramento, pagamento, documentação, comunicação, prevenção

Ferramentas

SonarQube, Better Code Hub, PMD, Checkstyle, Ndepend, Jdepend, FXCop, Spotbugs

Helicopter View
Activity
Java Projects
Javascript Projects
Languages Panel

TOOLS
Dependencies
Compare

All Projects

SQALE Rating: A

Technical Debt: 28,874.9 days

Lines of Code: 10,508K

All Projects

Issues: 668,081

Technical Debt: 28,874.9 days

Blocker	4,362
Critical	35,468
Major	503,908
Minor	107,690
Info	16,653

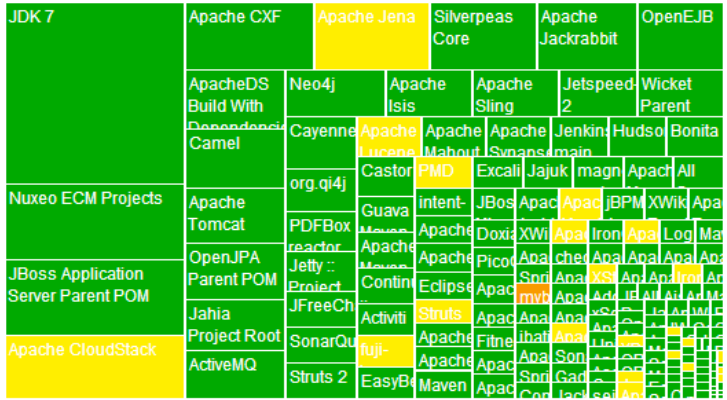
Forges

Name	LOCs	SQALE Rating
Forges	7,662,358	A
Apache	4,209,599	A
Others	1,966,714	A
JBoss	560,876	A
Sourceforge	344,112	A
Codehaus	267,568	C
OW2	164,683	A
OPS4J	71,965	A
SpringSource	51,192	A
GoogleCode	25,649	A

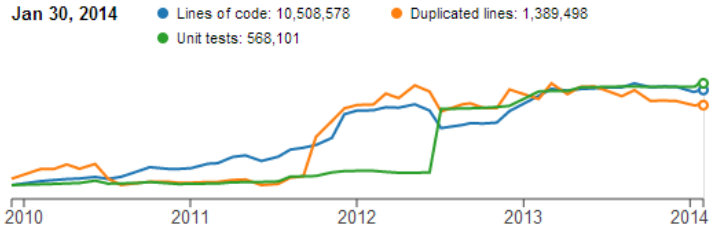
9 results

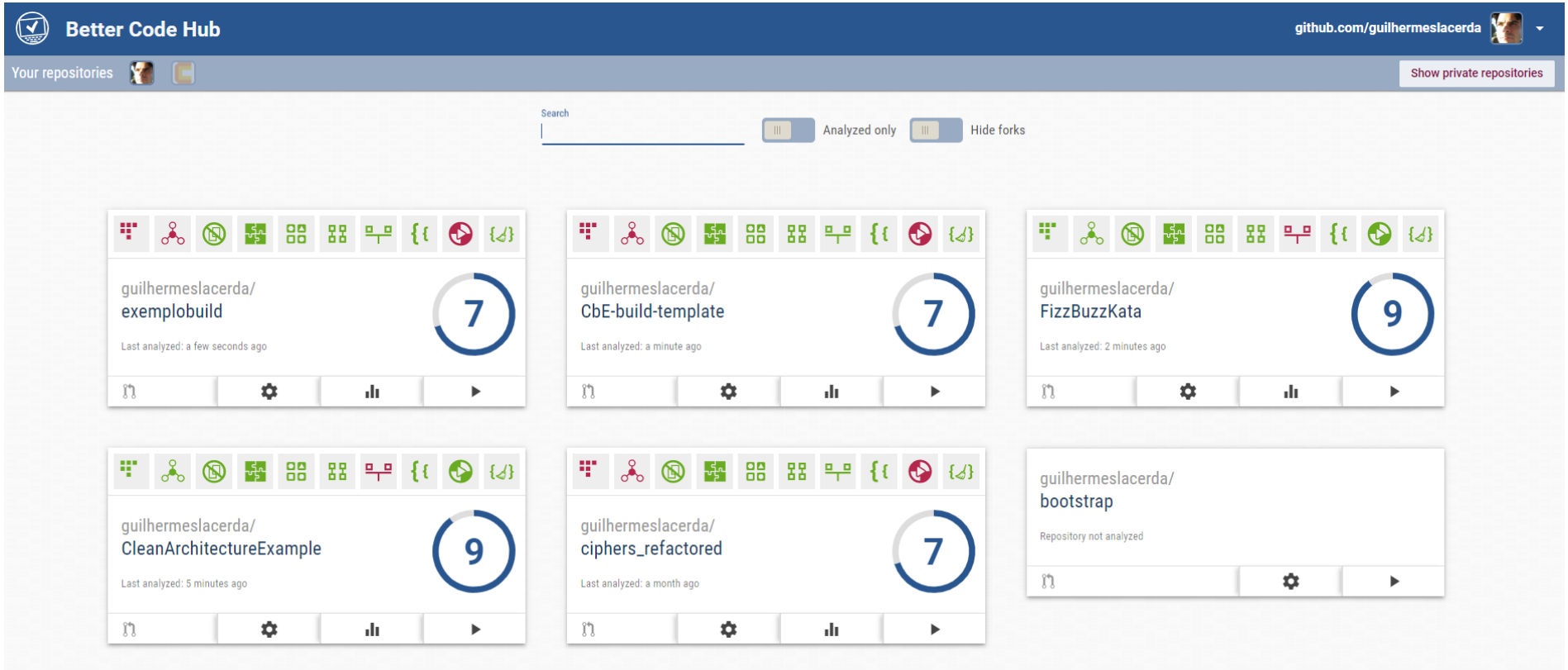
All Projects

Size: Lines of code Color: Rules compliance 0.0% 100.0%

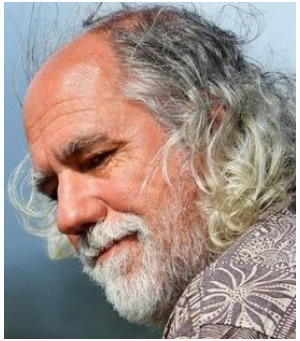


All Projects




Better Code Hub
github.com/guilhermeslacerda
Your repositories
Show private repositories
Search
Analyzed only
Hide forks
guilhermeslacerda/
exemplobuild
7
Last analyzed: a few seconds ago
guilhermeslacerda/
CbE-build-template
7
Last analyzed: a minute ago
guilhermeslacerda/
FizzBuzzKata
9
Last analyzed: 2 minutes ago
guilhermeslacerda/
CleanArchitectureExample
9
Last analyzed: 5 minutes ago
guilhermeslacerda/
ciphers_refactored
7
Last analyzed: a month ago
guilhermeslacerda/
bootstrap
Repository not analyzed

Software as a City



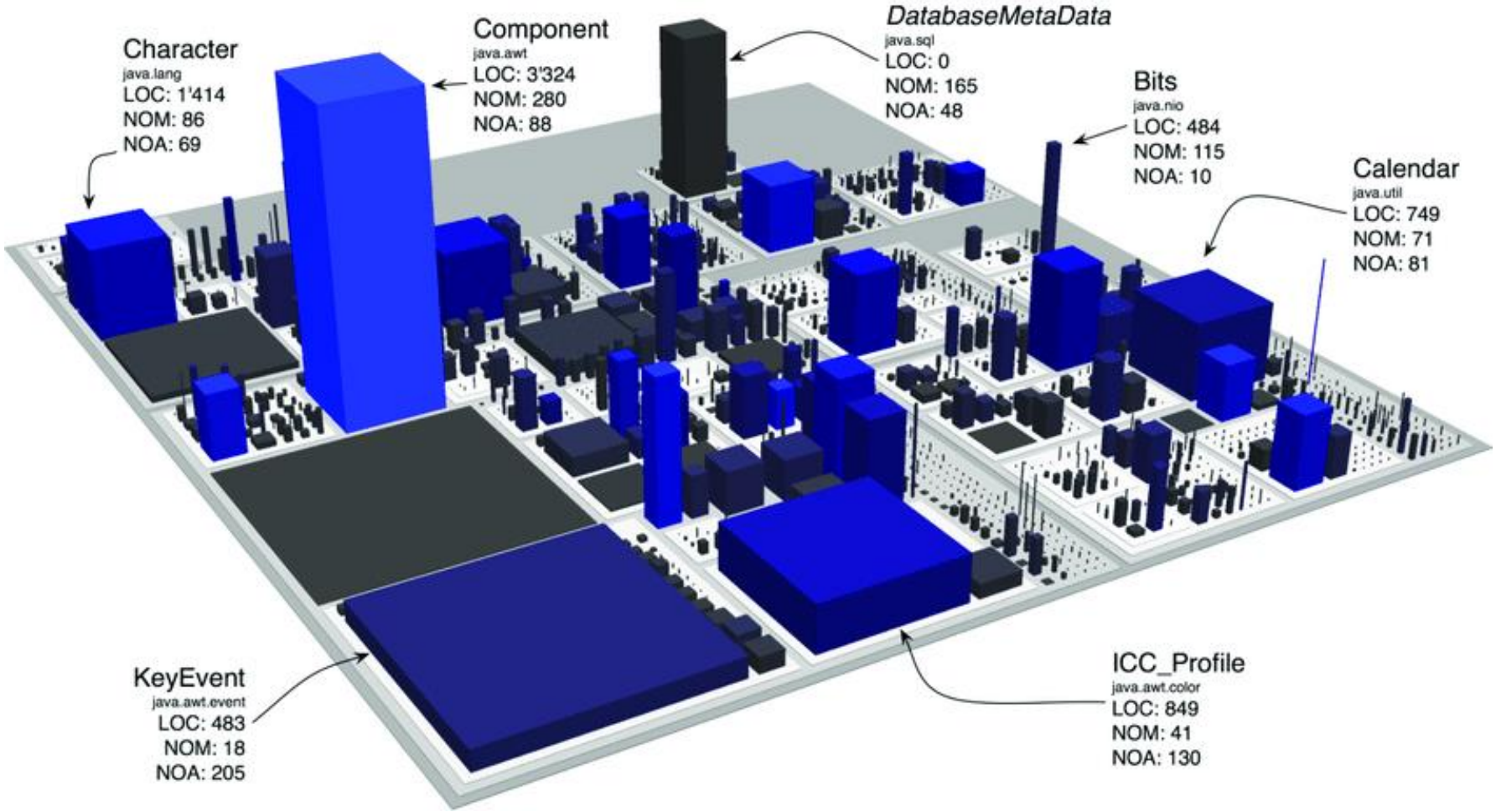
Definida por Grady Booch

Explorada na academia por Michele Lanza e Richard Wettel

Evolução do Software como uma cidade

Ferramenta

Code City


Character
java.lang
LOC: 1'414
NOM: 86
NOA: 69
Component
java.awt
LOC: 3'324
NOM: 280
NOA: 88
DatabaseMetaData
java.sql
LOC: 0
NOM: 165
NOA: 48
Bits
java.nio
LOC: 484
NOM: 115
NOA: 10
Calendar
java.util
LOC: 749
NOM: 71
NOA: 81
KeyEvent
java.awt.event
LOC: 483
NOM: 18
NOA: 205
ICC_Profile
java.awt.color
LOC: 849
NOM: 41
NOA: 130

Your Code as a Crime Scene

Criada por Adam Tornhill

Usa técnicas de psicologia forense

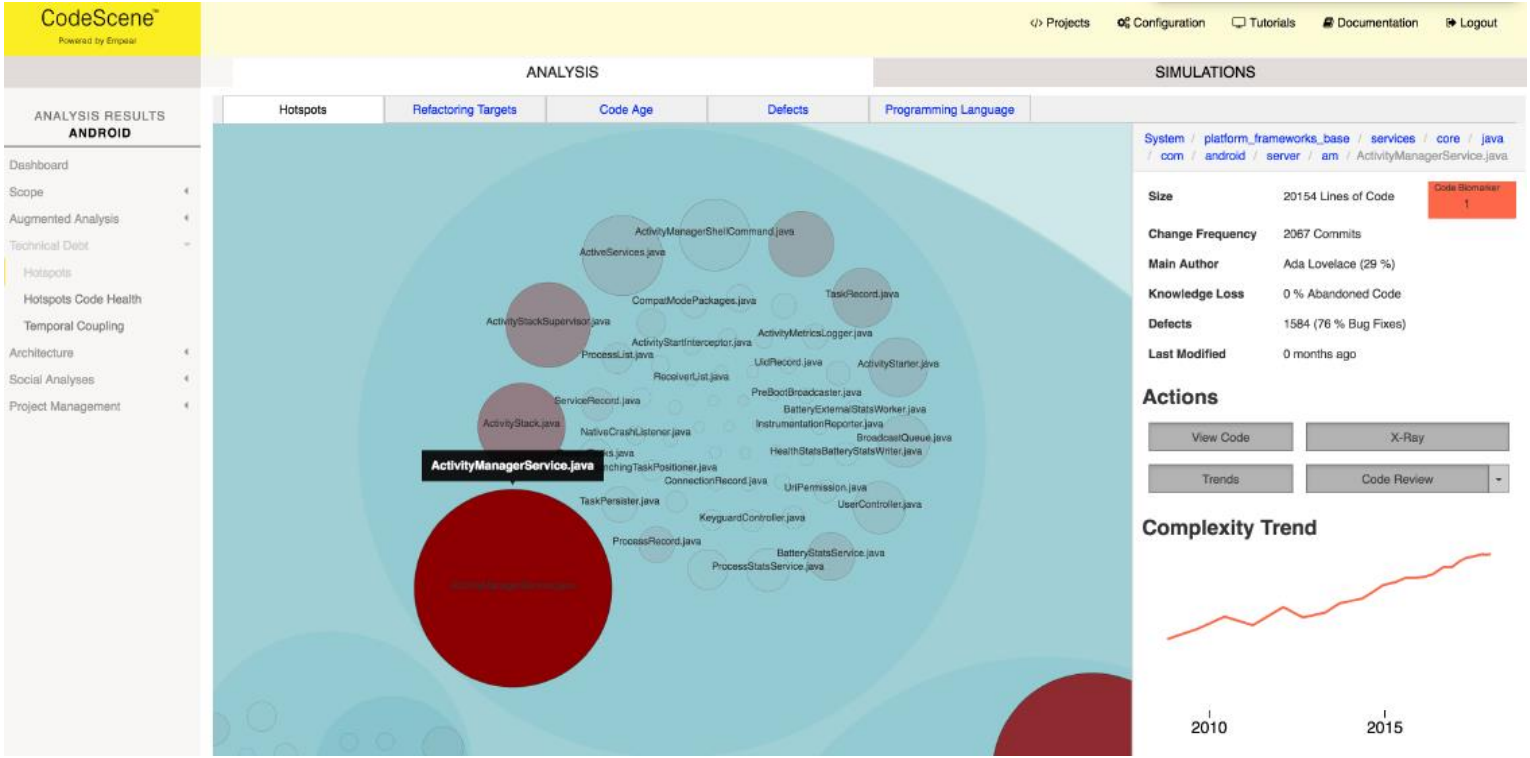
Técnicas

análise geográfica dos crimes, definição de perfil, uso do código como testemunha, mineração de hotspots

Ferramentas

code-maat, Code Scene

```
prompt> maat -l maat_evo.log -c git -a summary
statistic,value
number-of-commits,88
number-of-entities,45
number-of-entities-changed,283
number-of-authors,2
```



Dentistry Metaphor

Criada por Emerson Murphy-Hill

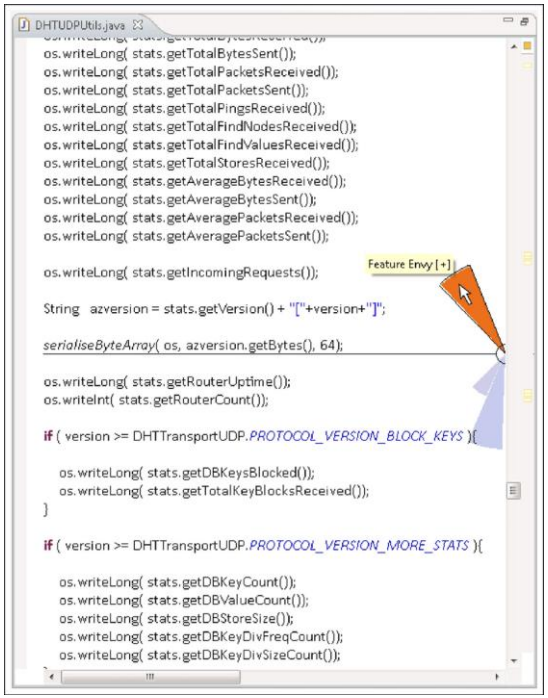
Refactorings

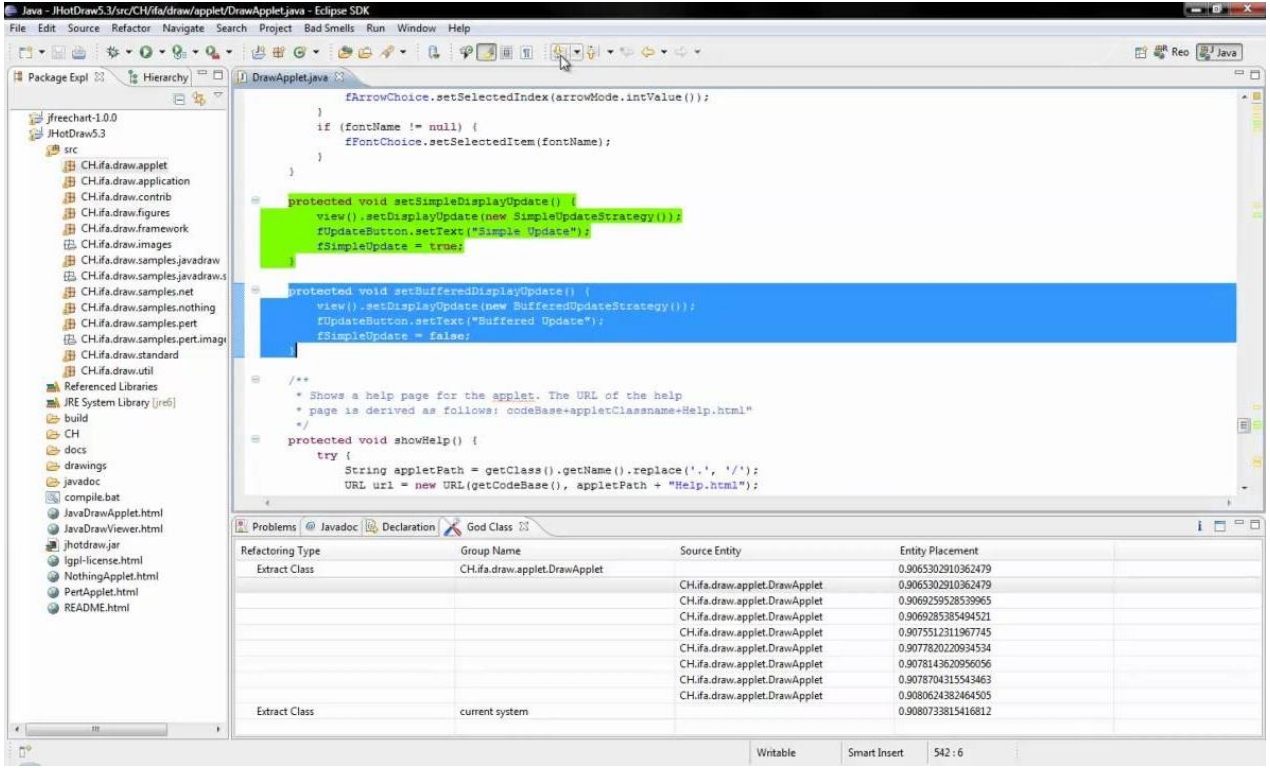
Root Canal x Floss

Ferramentas

Stench Blossom

Jdeodorant, Refactoring Miner (Tsantalis)


DHTUDPUtils.java
os.writeLong(stats.getTotalBytesSent());
os.writeLong(stats.getTotalPacketsReceived());
os.writeLong(stats.getTotalPacketsSent());
os.writeLong(stats.getTotalPingsReceived());
os.writeLong(stats.getTotalFindNodesReceived());
os.writeLong(stats.getTotalFindValuesReceived());
os.writeLong(stats.getTotalStoresReceived());
os.writeLong(stats.getAverageBytesReceived());
os.writeLong(stats.getAverageBytesSent());
os.writeLong(stats.getAveragePacketsReceived());
os.writeLong(stats.getAveragePacketsSent());
Feature Envy [+]
os.writeLong(stats.getIncomingRequests());
String azversion = stats.getVersion() + "["+version+"]";
serialiseByteArray(os, azversion.getBytes(), 64);
os.writeLong(stats.getRouterUptime());
os.writeInt(stats.getRouterCount());
if (version >= DHTTransportUDP.PROTOCOL_VERSION_BLOCK_KEYS){
os.writeLong(stats.getDBKeysBlocked());
os.writeLong(stats.getTotalKeyBlocksReceived());
}
if (version >= DHTTransportUDP.PROTOCOL_VERSION_MORE_STATS){
os.writeLong(stats.getDBKeyCount());
os.writeLong(stats.getDBValueCount());
os.writeLong(stats.getDBStoreSize());
os.writeLong(stats.getDBKeyDivFreqCount());
os.writeLong(stats.getDBKeyDivSizeCount());


Java - JHotDraw5.3/src/CH/ifa/draw/applet/DrawApplet.java - Eclipse SDK
File Edit Source Refactor Navigate Search Project Bad Smells Run Window Help
Package Expl
Hierarchy
jfreechart-1.0.0
JHotDraw5.3
src
CH.ifa.draw.applet
CH.ifa.draw.application
CH.ifa.draw.contrib
CH.ifa.draw.figures
CH.ifa.draw.framework
CH.ifa.draw.images
CH.ifa.draw.samples.javadraw
CH.ifa.draw.samples.net
CH.ifa.draw.samples.nothing
CH.ifa.draw.samples.pert
CH.ifa.draw.standard
CH.ifa.draw.util
Referenced Libraries
JRE System Library [jre6]
build
CH
docs
drawings
javadoc
compile.bat
JavaDrawApplet.html
JavaDrawViewer.html
jhotdraw.jar
lgpl-license.html
NothingApplet.html
PertApplet.html
README.html
DrawApplet.java
fArrowChoice.setSelectedIndex(arrowMode.intValue());
fFontChoice.setSelectedItem(fontName);
protected void setSimpleDisplayUpdate() {
view().setDisplayUpdate(new SimpleUpdateStrategy());
fUpdateButton.setText("Simple Update");
fSimpleUpdate = true;
protected void setBufferedDisplayUpdate() {
view().setDisplayUpdate(new BufferedUpdateStrategy());
fUpdateButton.setText("Buffered Update");
fSimpleUpdate = false;
/**
* Shows a help page for the applet. The URL of the help
* page is derived as follows: codeBase+appletClassname+Help.html"
*/
protected void showHelp() {
try {
String appletPath = getClass().getName().replace('.', '/');
URL url = new URL(getCodeBase(), appletPath + "Help.html");
Problems Javadoc Declaration God Class
Refactoring Type Group Name Source Entity Entity Placement
Extract Class CH.ifa.draw.applet.DrawApplet 0.9065302910362479
CH.ifa.draw.applet.DrawApplet 0.9065302910362479
CH.ifa.draw.applet.DrawApplet 0.9069259528539965
CH.ifa.draw.applet.DrawApplet 0.9069285385494521
CH.ifa.draw.applet.DrawApplet 0.9075512311967745
CH.ifa.draw.applet.DrawApplet 0.9077820220934534
CH.ifa.draw.applet.DrawApplet 0.9078143620956056
CH.ifa.draw.applet.DrawApplet 0.9078704315543463
CH.ifa.draw.applet.DrawApplet 0.9080624382464505
Extract Class current system 0.9080733815416812
Writable
Smart Insert
542 : 6

5S em Código

Usada por Guilherme Lacerda

5 sensos japoneses

Seiri, Seiton, Seiso, Seiketsu, Shitsuke

Materiais e ferramentas

www.codingbyexample.org

Metáfora da Medicina

DR-Tools

Usada por Guilherme Lacerda

Relação de sintomas no código com doenças e possíveis tratamentos

Ferramentas e materiais

drtools.site

```
D:\programs\drtools-metric
```

TYPES	SLOC	NOM	NPM	WMC	DEP	I-DEP	FAN-IN	FAN-OUT	NOA
org.hibernate.persister.entity.AbstractEntityPersister	4529	398	220	877	138	110	17	134	114
org.hibernate.test.legacy.FooBarTest	4490	110	109	221	79	44	0	54	0
org.hibernate.boot.model.source.internal.hbm.ModelBinder	3602	151	77	421	144	125	2	142	31
org.hibernate.test.hql.ASTParserLoadingTest	3318	127	118	170	82	50	0	67	4
org.hibernate.cfg.AnnotationBinder	3101	61	13	504	160	37	11	43	2

METHODS	MLOC	CYCLO	CALLS	NBD	PARAM
nSecondPass, MetadataBuildingContext context, Map<XClass,InheritanceState> inheritanceStatePerClass)	762	99	362	8	10
org.hibernate.hql.internal.ast.SqlASTFactory.getASTNodeType(int tokenType)	122	66	0	2	1
, XClass returnedClass, String declaringClassName, ConverterDescriptor attributeConverterDescriptor)	175	55	95	4	4
r, XProperty property, PropertyHolder parentPropertyHolder, MetadataBuildingContext buildingContext)	372	53	185	7	14
org.hibernate.hql.internal.classic.FromParser.token(String token, QueryTranslatorImpl q)	244	52	44	4	2

```
       Processing time: 37 seconds

D:\programs\drtools-metric
λ
```

```
C:\Program Files\cmder
λ drtools-metric

Usage: drtools-metric <project-directory> <OPTIONS> <OUTPUT> [--top <number>]
OPTIONS = <-a|-ac|-s|-n|-t|-m|-d|-cd|-id|-nc|-tc|-mt|-i|-mv> OUTPUT = <--console|--csv|--json|--save>

        Where
-a      list ALL metrics (namespaces/types/methods)             --console       show the results to console
-ac     list ALL metrics about COUPLING/DEPENDENCIES            --csv           generate results in CSV format
-s      list SUMMARY of project                                 --json          generate results in JSON format
-n      list information about NAMESPACES (packages)            --save          generate file results to drtools-metric-visualization tool
-t      list information about TYPES (classes)
-m      list information about METHODS (functions)              --top           list top 'number' records, based on used format
-d      list information about DEPENDENCIES of types
-cd     list information about CYCLIC DEPENDENCIES of types
-id     list information about INTERNAL DEPENDENCIES of types
-nc     list information about NAMESPACE COUPLING
-tc     list information about TYPE COUPLING
-mt     list information about METRIC THRESHOLDS
-i      list INFORMATION about tool development team
-mv     generate files to drtools-metric-visualization tool (use only with --save output option)

        Metrics
CA    - Number of types outside this component that depends on types inside this component (Afferent Coupling)
CE    - Number of types inside this component that depends on types outside this component (Efferent Coupling)
I     - Instability of namespace (range between 0=Maximally stability and 1=Maximally instability)
A     - Abstractness degree of namespace (range between 0=Minimally abstractness and 1=Maximally abstractness)
D     - Normalized distance of namespace                        NOM    - Number of methods of a type
NAC   - Number of abstract types inside namespaces              WMC    - Weighted methods per types (sum the CYCLO of each method)
NOC   - Number of types inside namespaces                       SLOC   - Number of lines of source code
DEP   - Number of external types dependencies                   I-DEP  - Number of internal types dependencies
FAN-IN- Number of other types that depend on a given type       FAN-OUT- Number of other types referenced by a type
NOA   - Number of attributes/variables                          NPM    - Number of public methods of a type
NBD   - Number of nested block depth of a method                MLOC   - Number of lines of a method
PARAM - Number of parameters of a method                        CYCLO  - Cyclomatic complexity (McCabe) of a method
CALLS - Number of invocations made from within a method

 Usage examples:
        Example 1 : # drtools-metric \Project\Java\src -a --console
        Example 2 : # drtools-metric \Project\Java\src -t --csv
        Example 3 : # drtools-metric \Project\Java\src -m --console --top 10

C:\Program Files\cmder
λ
```

```
C:\Program Files\cmder
λ drtools-metric D:\JavaApps\Doutorado\repos\findbugs-3.0.1\src -c --console | more
```

NAMESPACES	CA	CE	I	A	D
edu.umd.cs.findbugs.detect	5	24	0,828	0,031	0,142
edu.umd.cs.findbugs	393	30	0,071	0,273	0,656
edu.umd.cs.findbugs.ba	364	25	0,064	0,287	0,649
edu.umd.cs.findbugs.gui2	4	14	0,778	0,095	0,128
edu.umd.cs.findbugs.classfile.engine.bcel	1	15	0,938	0,022	0,041
edu.umd.cs.findbugs.classfile	263	5	0,019	0,629	0,353
edu.umd.cs.findbugs.util	135	6	0,043	0,147	0,810
edu.umd.cs.findbugs.ba.jsr305	14	8	0,364	0,219	0,418
edu.umd.cs.findbugs.ba.npe	21	11	0,344	0,107	0,549
edu.umd.cs.findbugs.filter	27	4	0,129	0,107	0,764
edu.umd.cs.findbugs.workflow	3	9	0,750	0,037	0,213
edu.umd.cs.findbugs.annotations	0	0	0,000	0,000	1,000
edu.umd.cs.findbugs.ba.bcp	3	3	0,500	0,240	0,260
edu.umd.cs.findbugs.classfile.impl	3	9	0,750	0,095	0,155
edu.umd.cs.findbugs.ba.type	44	8	0,154	0,100	0,746
edu.umd.cs.findbugs.graph	16	0	0,000	0,500	0,500
edu.umd.cs.findbugs.ba.obl	4	7	0,636	0,059	0,305
edu.umd.cs.findbugs.cloud	30	5	0,143	0,357	0,500
edu.umd.cs.findbugs.ba.vna	63	4	0,060	0,154	0,786
edu.umd.cs.findbugs.ba.ch	49	6	0,109	0,250	0,641
edu.umd.cs.findbugs.classfile.engine	4	9	0,692	0,333	0,026
edu.umd.cs.findbugs.anttask	0	2	1,000	0,091	0,091
edu.umd.cs.findbugs.plan	3	3	0,500	0,091	0,409
edu.umd.cs.findbugs.visitclass	52	5	0,088	0,273	0,640
edu.umd.cs.findbugs.xml	54	1	0,018	0,500	0,482
edu.umd.cs.findbugs.sourceViewer	2	2	0,500	0,000	0,500
edu.umd.cs.findbugs.classfile.analysis	33	7	0,175	0,111	0,714
edu.umd.cs.findbugs.ba.generic	9	4	0,308	0,000	0,692
edu.umd.cs.findbugs.jaif	0	1	1,000	0,125	0,125
edu.umd.cs.findbugs.model	8	4	0,333	0,250	0,417
edu.umd.cs.findbugs.ba.heap	3	1	0,250	0,143	0,607
edu.umd.cs.findbugs.config	23	4	0,148	0,143	0,709
edu.umd.cs.findbugs.ba.constant	4	1	0,200	0,000	0,800
edu.umd.cs.findbugs.ba.interproc	15	4	0,211	0,500	0,289
edu.umd.cs.findbugs.bcel	85	5	0,056	0,833	0,111
edu.umd.cs.findbugs.cloud.username	6	4	0,400	0,167	0,433
edu.umd.cs.findbugs.props	14	4	0,222	0,500	0,278
edu.umd.cs.findbugs.tools	0	4	1,000	0,000	0,000
edu.umd.cs.findbugs.ba.ca	3	1	0,250	0,000	0,750

```
C:\Program Files\cmder
λ drtools-metric D:\JavaApps\Doutorado\repos\hibernate-orm-master\ -s --console
------------------
SUMMARY OF METRICS
------------------
           Total of Namespaces: 1371
                Total of Types: 9954 - 7,26 (number of types/namespaces)
                 Total of SLOC: 738683 - 74,21 (number of SLOC/types)
              Total of Methods: 75730 - 7,61 (number of methods/types)
                 Total of CYCLO: 106331 - 10,68 (number of CYCLO/types)
       Processing time: 38 seconds

C:\Program Files\cmder
λ drtools-metric D:\JavaApps\Doutorado\repos\hibernate-orm-master\ -mv --save
Generating files to drtools-metric-visualization tool
Please, wait...

Summary info (CSV).................[DONE]
Namespaces info (CSV)..............[DONE]
Types info (CSV)...................[DONE]
Methods info (CSV).................[DONE]
Namespace coupling info (CSV).......[DONE]
Internal dependencies info (JSON)...[DONE]
Cyclic dependencies info (CSV)......[DONE]
Metric thresholds info (CSV)........[DONE]
Type coupling info (CSV)............[DONE]

To use the data with drtools-metric visualization, you need:
1 - create a folder of your project within the datasets folder
2 - copy the generated files (CSV and JSON) to the created folder
3 - do the setup on dr-tools-properties.js and you're done!

C:\Program Files\cmder
λ
```

Metric Visualization

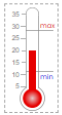
A tool quality suite to help the developers to maintain health and code evolution

Thermometer Visualization

Project: Findbugs 3.0.1

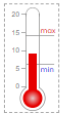
Back

Types (types/namespaces)



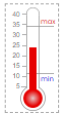
Total of Namespaces: 57
Total of Types: 1161
Types/namespaces: 20
Total of SLOC: 130364

Methods (methods/types)



Total of Methods: 10753
Methods/Types: 9

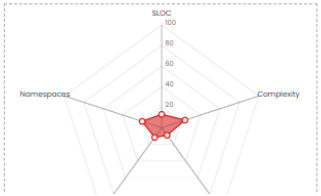
Complexity (WMC/types)



Total of Complexity: 28446
Complexity/Types: 24
Complexity/Methods: 3

Distribution of SLOC, Namespaces, Types, Methods, and Complexity

Scale used: 10,000 SLOC (Total SLOC/Scale)



Type Visualization

Types with Number of Methods/Functions (NOM - y-axis), Lines of Code (SLOC - x-axis), Complexity (WMC - bubble color), and Dependencies (DEP - bubble size)

Project: Findbugs 3.0.1

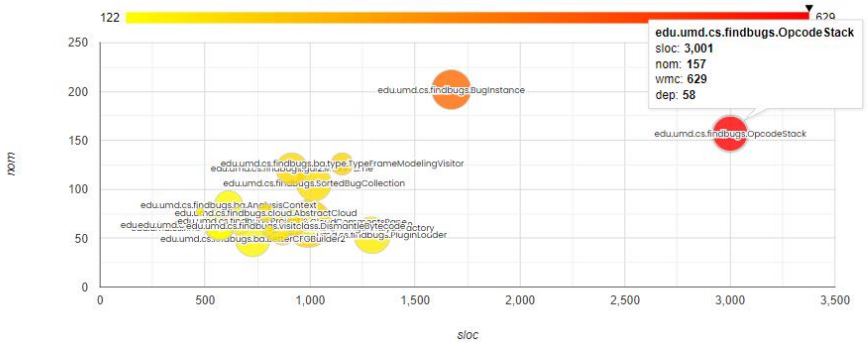
Back

Number of Methods (NOM) 43.0 202.0

Lines of Code (SLOC) 331.0 3,001.0

Complexity (WMC) 115.0 629.0

Dependencies (DEP) 13.0 103.0



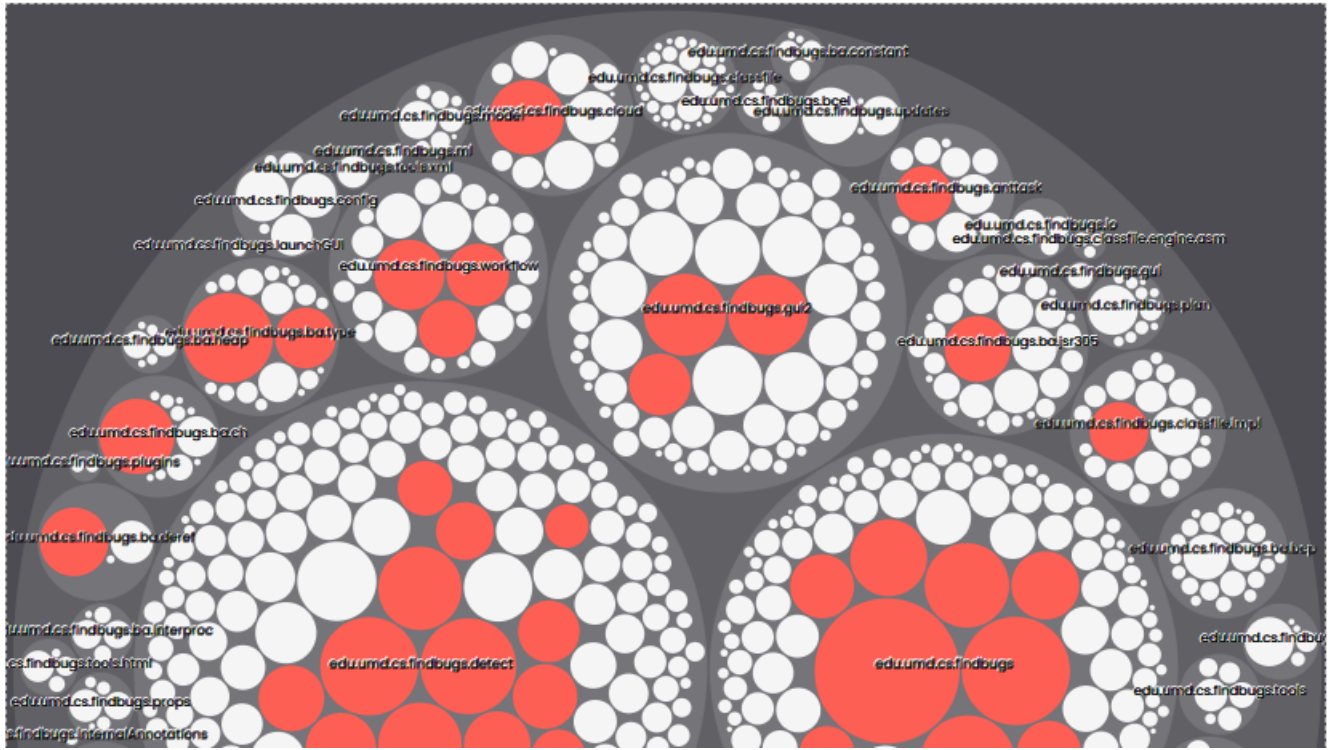
Code Resonance

Bubble Size (SLOC - Lines of Code) and Bubble Color (The most complex classes are red bubbles, with high WMC)

Project: Findbugs 3.0.1

Back

☐ Disable name of namespaces/types

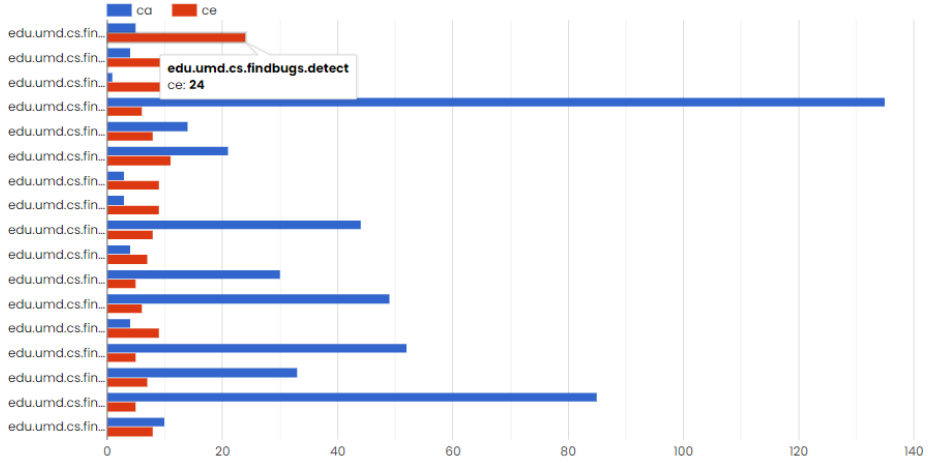


Namespace Coupling Visualization

CA (Afferent Coupling) and CE (Efferent Coupling)
Project: Findbugs 3.0.1

Back

Afferent Coupling (CA) 0.0 224.0 Efferent Coupling (CE) 5.0 30.0

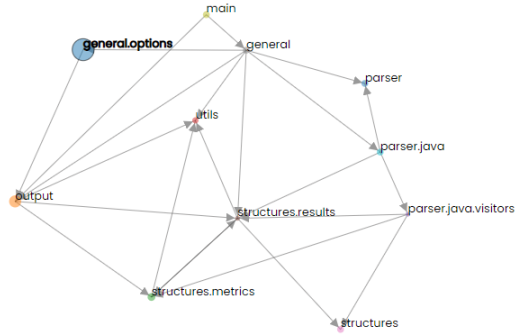


Namespace Dependencies

Bubble Size (NOC - Number of Classes/Types)
Project: DR Tools Metric

Back

Disable name of namespaces



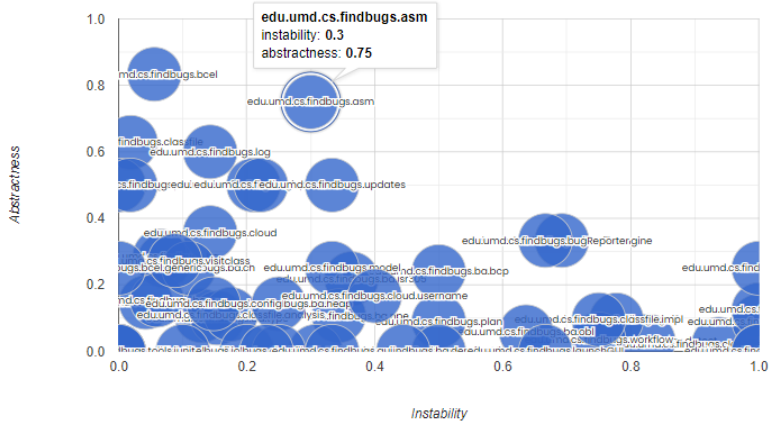
Instability and Abstractness Visualization

Abstractness degree (y-axis) and Instability (x-axis)

Project: Findbugs 3.0.1

Back

Abstractness 0.0 1.0 Instability 0.0 1.0

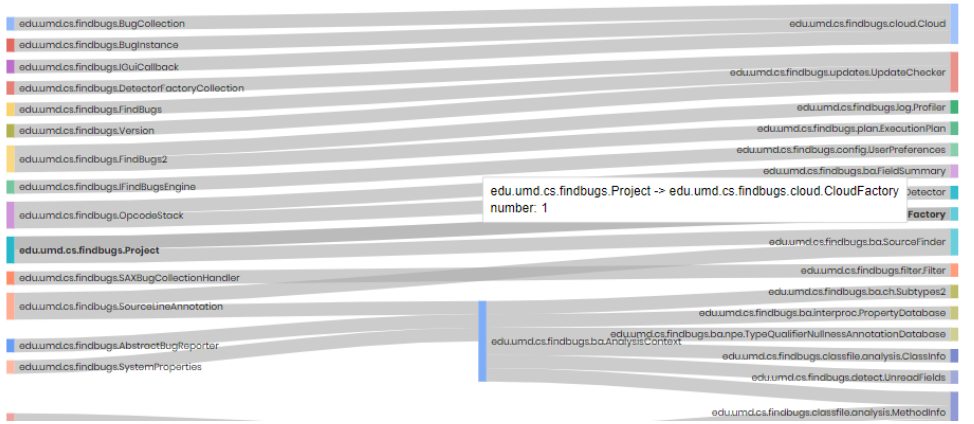


Cyclic Dependencies Visualization

Cyclic Dependencies Visualization

Project: Findbugs 3.0.1

Back



Considerações Finais

As metáforas têm um grande valor

Ampliam o poder de comunicação

Compreensão compartilhada e colaborativa

Uso de alto e baixo nível, diferentes níveis de abstração

Entenda a metáfora, colete material, use a criatividade e use sem moderação (By Émerson Hernandez)

Questões??

THE
DEVELOPER'S
CONFERENCE

Melhorando o design de código através de **Metáforas**

Alessandro Dias
e-Core - UniRitter
@alessandro_dias

Guilherme Lacerda
Wildtech – Unisinos
@guilhermeslac