



# DR-Tools

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

**[drtools.site](https://drtools.site)**



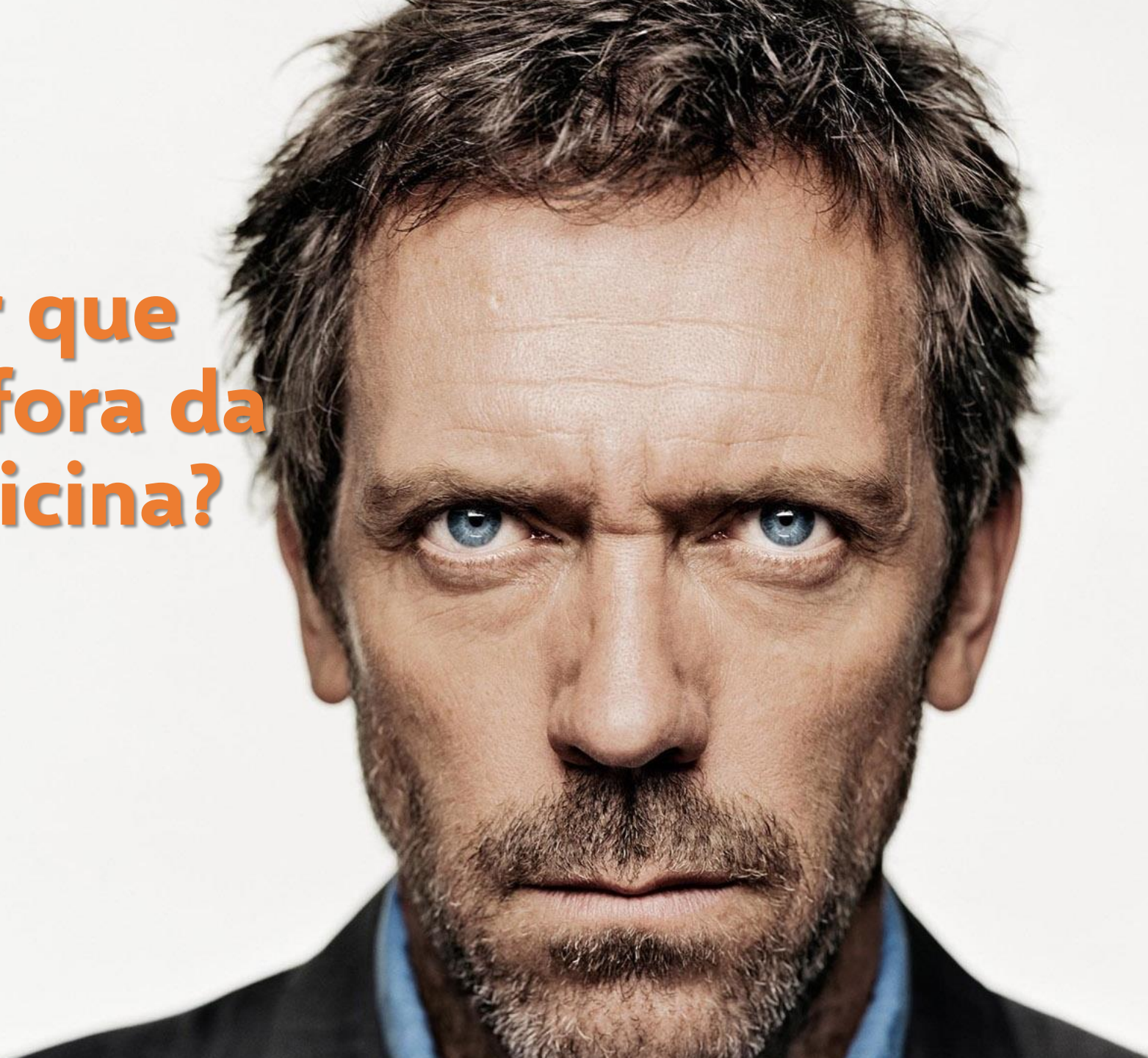
# Quem sou eu?

glacerda@wildtech.com.br  
@guilhermeslac

- ✓ Mestre e Doutorando em Ciência da Computação (UFRGS)
- ✓ Professor de Graduação (UniRitter) e Pós-Graduação (UniRitter, Unisinos, UFRGS)
- ✓ Consultor associado da Wildtech
- ✓ Pioneiro em Metodologias Ágeis no Brasil
- ✓ Fundador do XP-RS/GUMA
- ✓ Membro da ScrumAlliance , IASA, SBC e ACM



**Por que  
metáfora da  
medicina?**

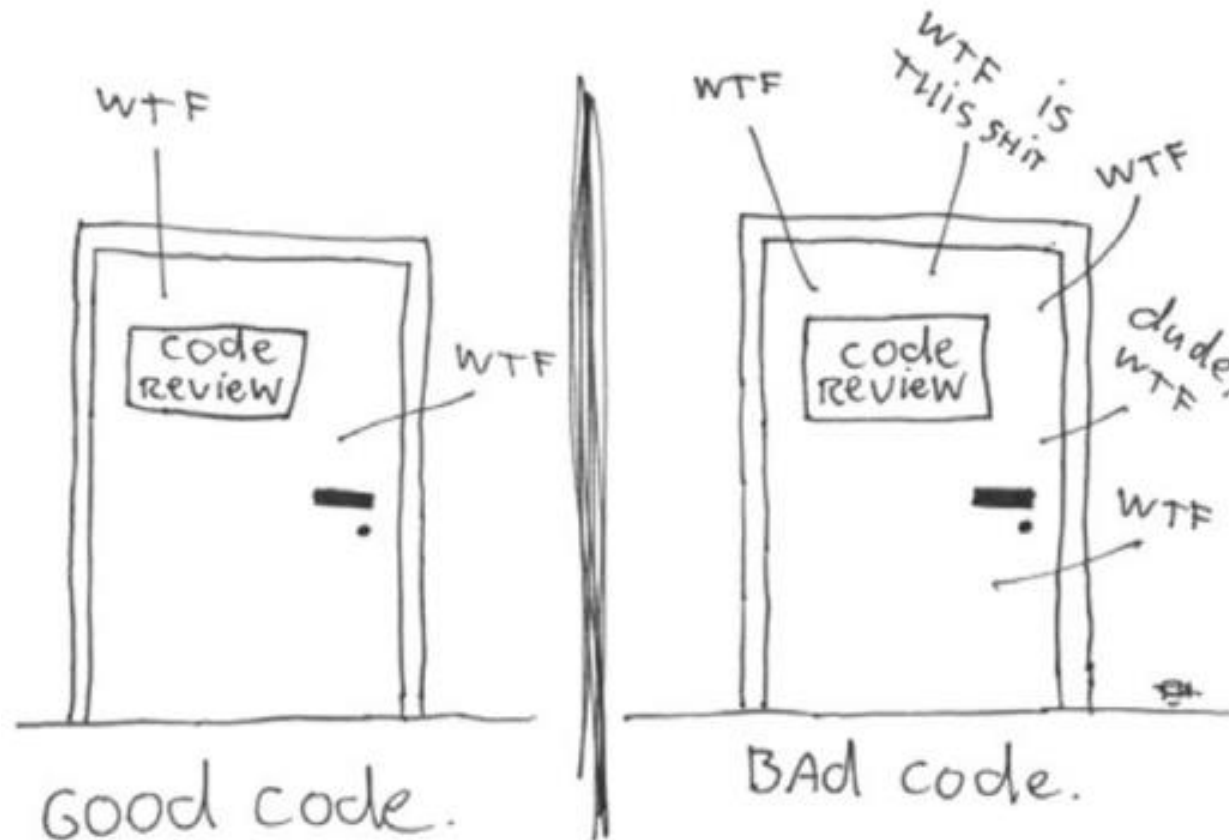




**metric**  
**DR-Tools**

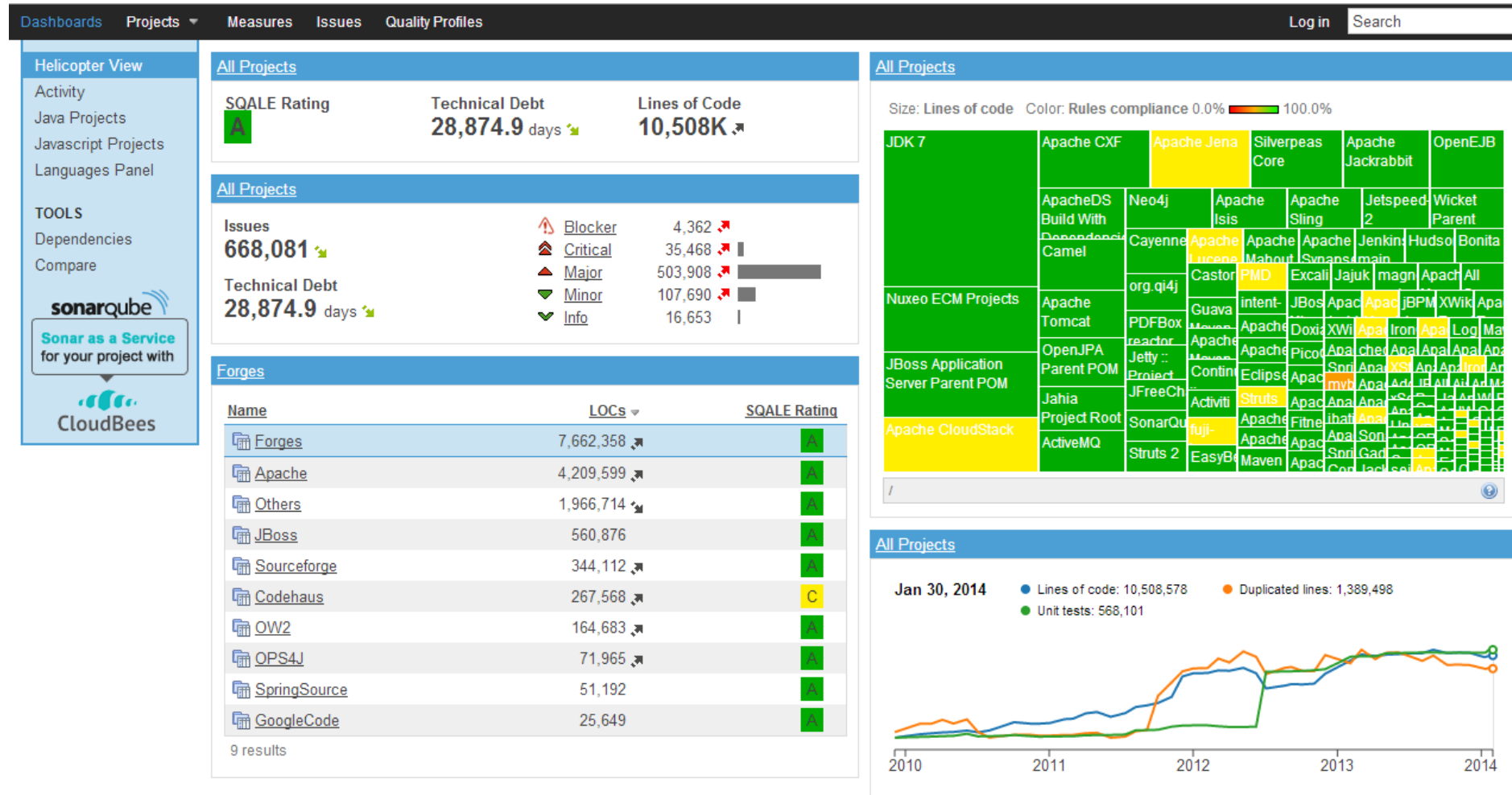
# Como você mede a qualidade do código?

The ONLY VALID MEASUREMENT  
OF CODE QUALITY: WTFs/MINUTE





# “ah... Nós temos o SonarQube”



**Se eu baixar um projeto do  
GitHub,  
onde estão os maiores  
problemas?**



# Características

- ✓ **Simplicidade, valor do XP**
- ✓ **CLI (*command line interface*)**
- ✓ **Versatilidade**
- ✓ **Download and run! (pré-requisito: JRE 8)**
- ✓ **Seleção de métricas, correlacionadas e ordenadas**
- ✓ **Visualizações usando Google Chart e D3.js**
- ✓ **Atualmente analisa projetos Java**





# Funções

- ✓ **Resultados em diferentes formatos**  
*Console, JSON, CSV*
- ✓ **Filtragem dos 'Top X'**
- ✓ **Resultados contextualizados**  
*Projeto, Namespaces, Classes, Métodos, Acoplamento, Dependências*



# Métricas Contextualizadas



## Sumário do Projeto

*Total de Namespaces, Classes, SLOC, Métodos e CYCLO*



## Namespaces

*NOC, NAC*



## Classes

*SLOC, NOM, WMC/CYCLO, DEP, I-DEP/Fan-Out, NPM, NOA*



## Métodos

*MLOC, CYCLO, CALLS, NBD, PARAM*



## Acoplamento

*CA, CE, Instability, Abstractness, Normalized Distance*



## Dependências

*Externas, Internas, Ciclic Dependency*



**DR-Tools**

# Uso

```
C:\Program Files\cmdr
```

```
λ drtools-metric
```

```
drtools-metric - helping you to improve the health of your source code and reduce technical debt!
```

```
Developed by Guilherme Lacerda (guilhermeslacerda@gmail.com)
```

```
Usage: drtools-metric <project-directory> <OPTIONS> <OUTPUT> [--top <number>]
```

```
OPTIONS = <-a|-ac|-s|-n|-t|-m|-d|-cd|-id|-c|-mt> OUTPUT = <--console|--csv|--json>
```

```
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 a SUMMARY of project                             --json       generate results in JSON format
-n      list information about NAMESPACES                      --top        list top 'number' records, based on used format
-t      list information about TYPES (classes)
-m      list information about METHODS (functions)
-d      list information about DEPENDENCIES of types/classes
-cd     list information about CYCLIC DEPENDENCIES of types/classes
-id     list information about INTERNAL DEPENDENCIES of types/classes
-c      list information about COUPLING of namespaces
-mt     list information about METRIC THRESHOLDS
```

```
Metrics
CA - Number of types/classes outside this component that depends on types/classes inside this component (Afferent Coupling)
CE - Number of types/classes inside this component that depends on types/classes 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 from the main sequence
NAC - Number of abstract types/classes of package/namespace
NOC - Number of types/classes of package/namespace
DEP - Number of type/classes external dependencies
NOA - Number of attributes/variables
NPM - Number of public methods/functions of a type
MLOC - Number of lines of a method/function
CYCLO - Cyclomatic complexity (McCabe) of a method/function

WMC - Weighted methods per types/classes (sum the CYCLO of each method)
SLOC - Number of lines of source code
I-DEP - Number of type/classes internal dependencies (Fan-Out)
NOM - Number of methods/functions of a type
NBD - Number of nested block depth of a method/function
PARAM - Number of parameters of a method/function
CALLS - Number of invocations made from within a method/function
```

```
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\cmdr
```

```
λ |
```

# Uso

```
C:\Program Files\cmdr
λ drtools-metric D:\JavaApps\Plugins\DrToolsMetric\src -s --console
```

## SUMMARY OF METRICS

```
-----
Total of Namespaces: 11
Total of Types: 39 - 3,55 (number of types/namespaces)
Total of SLOC: 2106 - 54,00 (number of SLOC/types)
Total of Methods: 278 - 7,13 (number of methods/types)
Total of CYCLO: 446 - 11,44 (number of CYCLO/types)
Processing time: 960 milliseconds
```

```
C:\Program Files\cmdr
λ drtools-metric D:\JavaApps\Plugins\DrToolsMetric\src -mt --console
```

## INFORMATION ABOUT METRIC THRESHOLDS

```
-----
Small Project (SMALL)                small project with < 50 KLOC or 200 < classes
Medium Project (MEDIUM)              medium project with (50 KLOC <= project <= 250 KLOC) or (200 <= classes <= 1000)
Large Project (LARGE)                 large project with > 250 KLOC or > 1000 classes
Number of Types/Classes (NOC)         Good: <= 11; Regular: between 11 and 28; Bad: > 28
Number of Abstract Types/Classes (NAC) without references
Type/Class Line of Code (SLOC)        Bad: > 500
Number of Functions/Methods (NOM)     Good: <= 6; Regular: between 6 and 14; Bad: > 14
Weighted Methods per Class (WMC)      Good: <= 11; Regular: between 11 and 34; Bad: > 34
Number of external types/classes dependencies (DEP) Bad: > 20
Number of internal types/classes dependencies (I-DEP) Bad: > 15
Number of Public Methods (NPM)        Good: <= 10; Regular: between 11 and 40; Bad: > 40
Number of Attributes/Fields (NOA)     Good: <= 3; Regular: between 3 and 8; Bad: > 8
Method Lines of Code (MLOC)           Good: <= 10; Regular: between 10 and 30; Bad: > 30
Cyclomatic Complexity (CYCLO)         Good: <= 2; Regular: between 2 and 4; Bad: > 4
Number of Invocations (CALLS)         without references
Nested Block Depth (NBD)              Good: <= 1; Regular: between 1 and 3; Bad: > 3
Number of Parameters (PARAM)          Good: <= 2; Regular: between 2 and 4; Bad: > 4
Afferent Coupling (CA)                Good: <= 7; Regular: between 7 and 39; Bad: > 39
Efferent Coupling (CE)                Good: <= 6; Regular: between 6 and 16; Bad: > 16
Package Instability (I)               range between 0=Maximally stability and 1=Maximally instability
Abstractness Degree (A)               range between 0=Minimally abstractness and 1=Maximally abstractness
Normalized Distance (D)               range between 0=exactly located in the main sequence and 1=far from the main sequence
```

```
C:\Program Files\cmdr
λ |
```



# Uso

```
C:\Program Files\cmdr
```

```
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: 55503 - 5,58 (number of methods/types)
Total of CYCLO: 76955 - 7,73 (number of CYCLO/types)
Processing time: 14 seconds
```

```
C:\Program Files\cmdr
```

```
λ drtools-metric D:\JavaApps\Doutorado\repos\hibernate-orm-master\ -c --console | more
```

NAMESPACES	CA	CE	I	A	D
org.hibernate.test.legacy	2	41	0,953	0,071	0,025
org.hibernate.annotations	7	1	0,125	0,000	0,875
org.hibernate.type	681	29	0,041	0,173	0,786
org.hibernate.dialect	731	51	0,065	0,059	0,876
org.hibernate.test.hql	3	38	0,927	0,055	0,018
org.hibernate.boot.model.source.spi	87	8	0,084	0,876	0,039
org.hibernate.boot.model.source.internal.hbm	7	31	0,816	0,186	0,002
org.hibernate	2047	27	0,013	0,329	0,658
org.hibernate.hql.internal.ast.tree	29	26	0,473	0,333	0,194
org.hibernate.event.spi	139	7	0,048	0,563	0,389
org.hibernate.engine.spi	1099	64	0,055	0,524	0,421
org.hibernate.cfg	851	41	0,046	0,186	0,768
org.hibernate.mapping	294	40	0,120	0,316	0,564
org.hibernate.type.descriptor.java	165	13	0,073	0,113	0,814
org.hibernate.criterion	103	12	0,104	0,196	0,700
org.hibernate.test.annotations.entity	0	12	1,000	0,045	0,045
org.hibernate.userguide.mapping.basic	1	12	0,923	0,023	0,054
org.hibernate.event.internal	11	35	0,761	0,163	0,076
org.hibernate.id	73	33	0,311	0,326	0,363
org.hibernate.test.schemaupdate	1	34	0,971	0,047	0,018
org.hibernate.test.annotations.onetomany	0	13	1,000	0,000	0,000
org.hibernate.test.bytecode.enhancement.lazy.proxy	0	16	1,000	0,125	0,125
org.hibernate.test.criteria	0	17	1,000	0,077	0,077
org.hibernate.internal	374	97	0,206	0,243	0,551
org.hibernate.jpa.test.metamodel	18	4	0,182	0,108	0,710
org.hibernate.type.descriptor.sql	128	6	0,045	0,167	0,789
org.hibernate.boot.model.naming	146	9	0,058	0,657	0,285

# Uso

C:\Program Files\cmdr

λ drtools-metric D:\JavaApps\Doutorado\repos\hibernate-orm-master\ -n --console --top 35

-----		
NAMESPACES	NOC	NAC
-----		
org.hibernate.test.legacy	126	9
org.hibernate.annotations	110	0
org.hibernate.type	110	19
org.hibernate.dialect	102	6
org.hibernate.test.hql	91	5
org.hibernate.boot.model.source.spi	89	78
org.hibernate.boot.model.source.internal.hbm	86	16
org.hibernate	85	28
org.hibernate.hql.internal.ast.tree	72	24
org.hibernate.event.spi	71	40
org.hibernate.engine.spi	63	33
org.hibernate.cfg	59	11
org.hibernate.mapping	57	18
org.hibernate.type.descriptor.java	53	6
org.hibernate.criterion	51	10
org.hibernate.test.annotations.entity	44	2
org.hibernate.userguide.mapping.basic	43	1
org.hibernate.event.internal	43	7
org.hibernate.id	43	14
org.hibernate.test.schemaupdate	43	2
org.hibernate.test.annotations.onetomany	40	0
org.hibernate.test.bytecode.enhancement.lazy.proxy	40	5
org.hibernate.test.criteria	39	3
org.hibernate.internal	37	9
org.hibernate.jpa.test.metamodel	37	4
org.hibernate.type.descriptor.sql	36	6
org.hibernate.boot.model.naming	35	23
org.hibernate.test.ops	35	1
org.hibernate.cache.spi.support	32	11
org.hibernate.sql	31	4
org.hibernate.test.annotations	31	2
org.hibernate.test.annotations.lob	31	2
org.hibernate.test.annotations.embedded	30	0
rg.hibernate.envers.test.integration.modifiedflags	30	2
org.hibernate.userguide.mapping.identifier	29	0

Processing time: 15 seconds



# Uso

C:\Program Files\cmdr

λ drtools-metric D:\JavaApps\Doutorado\repos\hibernate-orm-master\ -t --console --top 35

TYPES	SLOC	NOM	NPM	WMC	DEP	I-DEP	NOA
org.hibernate.persister.entity.AbstractEntityPersister	4529	393	218	9	138	110	114
org.hibernate.test.legacy.FooBarTest	4490	110	109	217	79	44	0
org.hibernate.boot.model.source.internal.hbm.ModelBinder	3602	137	64	19	144	125	31
org.hibernate.test.hql.ASTParserLoadingTest	3318	125	116	8	82	50	4
org.hibernate.cfg.AnnotationBinder	3101	60	13	504	160	37	2
org.hibernate.internal.SessionImpl	3031	302	233	20	170	121	36
org.hibernate.cfg.annotations.reflection.JPAOverriddenAnnotationReader	2709	115	9	552	124	7	15
org.hibernate.test.querycache.AbstractQueryCacheResultTransformerTest	2256	269	178	5	35	21	7
org.hibernate.loader.Loader	2136	104	15	4	86	69	10
org.hibernate.userguide.hql.HQLTest	1868	147	143	148	46	17	0
org.hibernate.boot.internal.InFlightMetadataCollectorImpl	1861	164	126	24	99	79	65
org.hibernate.persister.collection.AbstractCollectionPersister	1855	165	113	5	90	78	93
org.hibernate.test.criteria.CriteriaQueryTest	1750	34	32	52	54	29	1
org.hibernate.engine.internal.StatefulPersistenceContext	1708	131	112	324	64	42	27
org.hibernate.cfg.annotations.CollectionBinder	1567	65	43	289	105	41	44
org.hibernate.test.readonly.ReadOnlyProxyTest	1507	44	42	45	19	9	0
org.hibernate.internal.CoreMessageLogger	1421	392	392	392	41	15	0
org.hibernate.internal.util.collections.BoundedConcurrentHashMap	1364	150	111	17	20	0	58
org.hibernate.query.internal.AbstractProducedQuery	1357	150	103	255	94	40	22
org.hibernate.jpa.test.query.QueryTest	1356	60	59	103	39	13	0
org.hibernate.test.immutable.ImmutableTest	1353	35	31	3	23	13	1
org.hibernate.dialect.AbstractHANADialect	1316	147	121	14	98	65	30
org.hibernate.test.hql.BulkManipulationTest	1301	58	54	3	34	14	6
org.hibernate.test.hql.HQLTest	1295	188	181	222	62	40	0
org.hibernate.test.readonly.ReadOnlySessionLazyNonLazyTest	1280	19	16	26	18	5	0
org.hibernate.internal.SessionFactoryImpl	1269	142	120	24	126	94	50
org.hibernate.dialect.Dialect	1198	215	199	251	111	85	25
org.hibernate.test.readonly.ReadOnlyCriteriaQueryTest	1191	33	30	12	24	9	9
org.hibernate.test.readonly.ReadOnlyCriteriaQueryTest	1133	19	18	26	25	16	0
org.hibernate.cfg.annotations.EntityBinder	1131	63	49	7	87	31	41
org.hibernate.test.legacy.ParentChildTest	1126	25	25	36	43	20	0
org.hibernate.hql.internal.ast.HqlSqlWalker	1122	91	34	14	88	66	26
org.hibernate.boot.internal.SessionFactoryOptionsBuilder	1102	155	149	184	122	37	80
org.hibernate.test.readonly.ReadOnlySessionTest	1102	25	25	79	17	6	0
org.hibernate.query.criteria.internal.CriteriaBuilderImpl	1098	170	169	226	75	45	1

Processing time: 15 seconds

# Uso

C:\Program Files\cmdr

λ drtools-metric D:\JavaApps\Doutorado\repos\hibernate-orm-master\ -m --console --top 35

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
zzToProcess, Map<XClass,InheritanceState> inheritanceStatePerClass, MetadataBuildingContext context)	320	44	151	8	3
org.hibernate.hql.internal.classic.SelectParser.token(String token, QueryTranslatorImpl q)	152	40	67	4	2
l NaturalIdDataAccess naturalIdRegionAccessStrategy, final PersisterCreationContext creationContext)	450	36	253	6	4
org.hibernate.dialect.TeradataDialect.getSelectClauseNullString(int sqlType)	44	31	0	0	1
org.hibernate.cfg.annotations.CollectionBinder.bind()	167	30	106	7	0
org.hibernate.test.legacy.FooBarTest.testQuery()	441	30	522	2	0
org.hibernate.spatial.dialect.hana.HANASpatialDialect.supports(SpatialFunction function)	61	29	0	1	1
query.procedure.internal.ProcedureParameterImpl.prepare(CallableStatement statement, int startIndex)	130	28	56	5	2
l NaturalIdDataAccess naturalIdRegionAccessStrategy, final PersisterCreationContext creationContext)	318	28	182	4	4
xt, Ejb3Column[] mapKeyColumns, Ejb3JoinColumn[] mapKeyManyToManyColumns, String targetPropertyName)	215	27	85	4	9
e.internal.StatefulPersistenceContext.deserialize(ObjectInputStream ois, SessionImplementor session)	124	26	60	5	2
ession, final PreLoadEvent preLoadEvent, final Iterable<PreLoadEventListener> preLoadEventListeners)	188	26	82	4	6
g.hibernate.hql.internal.ast.tree.SelectClause.initializeExplicitSelectClause(FromClause fromClause)	148	25	59	6	1
ctionProviderInitiator.initiateService(Map configurationValues, ServiceRegistryImplementor registry)	132	25	33	4	2
ate.context.internal.TransactionProtectionWrapper.invoke(Object proxy, Method method, Object[] args)	70	25	41	4	3
org.hibernate.engine.jdbc.internal.FormatProcess.perform()	97	25	45	3	0
org.hibernate.engine.jdbc.ClobProxy.invoke(Object proxy, Method method, Object[] args)	72	25	25	3	3
Enabled, XProperty property, PropertyHolder propertyHolder, MetadataBuildingContext buildingContext)	116	24	103	7	6
org.hibernate.query.internal.AbstractProducedQuery.setHint(String hintName, Object value)	96	24	63	4	2
org.hibernate.cfg.AnnotationBinder.bindDefaults(MetadataBuildingContext context)	106	23	33	8	1
lveEntityCallbacks(XClass beanClass, CallbackType callbackType, ReflectionManager reflectionManager)	128	23	57	7	3
nformation existingDatabase, ExecutionOptions options, Dialect dialect, GenerationTarget... targets)	114	23	42	6	5
rnate.event.internal.DefaultRefreshEventListener.onRefresh(RefreshEvent event, Map refreshedAlready)	182	23	83	3	2
tiveQueryNonScalarRootReturn rtnSource, PersistentClass pc, HbmLocalMetadataBuildingContext context)	172	22	64	6	4
org.hibernate.hql.internal.classic.QueryTranslatorImpl.renderSQL()	115	22	63	6	0
org.hibernate.hql.internal.classic.QueryTranslatorImpl.renderScalarSelect()	93	22	40	6	0
org.hibernate.engine.internal.StatefulPersistenceContext.serialize(ObjectOutputStream oos)	111	22	84	5	1
org.hibernate.metamodel.internal.MetadataContext.wrapUp()	100	22	54	5	0
org.hibernate.test.cut.CompositeDateTime.equals(Object obj)	61	22	6	2	1
r, XProperty property, List<PropertyData> inFlightPropertyDataList, MetadataBuildingContext context)	107	21	57	8	4

Processing time: 14 seconds

# Uso

C:\Program Files\cmdr

λ drtools-metric D:\JavaApps\Doutorado\Pathfinder\softwarepathfinder\src\ -d --console --top 2

-----  
Type: com.softwarepathfinder.visualization.prefuse.AggregateDemo      SLOC: 300      Number of Dependencies: 40

DEPENDENCIES:

com.softwarepathfinder.model.Invocation  
com.softwarepathfinder.model.Method  
com.softwarepathfinder.model.Path  
com.softwarepathfinder.model.Project  
com.softwarepathfinder.model.Type  
java.awt.Cursor  
java.awt.event.MouseEvent  
java.awt.geom.Point2D  
java.awt.geom.Rectangle2D  
java.util.HashMap  
java.util.Iterator  
java.util.List  
javax.swing.JFrame  
javax.swing.SwingUtilities  
prefuse.Constants  
prefuse.Display  
prefuse.Visualization  
prefuse.action.ActionList  
prefuse.action.RepaintAction  
prefuse.action.assignment.ColorAction  
prefuse.action.assignment.DataColorAction  
prefuse.action.layout.CircleLayout  
prefuse.action.layout.Layout  
prefuse.action.layout.graph.NodeLinkTreeLayout  
prefuse.activity.Activity  
prefuse.controls.ControlAdapter  
prefuse.controls.PanControl  
prefuse.controls.ZoomControl  
prefuse.data.Graph  
prefuse.data.Node  
prefuse.render.DefaultRendererFactory  
prefuse.render.LabelRenderer  
prefuse.render.PolygonRenderer  
prefuse.render.Renderer  
prefuse.util.ColorLib  
prefuse.util.GraphicsLib  
prefuse.visual.AggregateItem  
prefuse.visual.AggregateTable  
prefuse.visual.VisualGraph  
prefuse.visual.VisualItem



# Uso

```
λ drtools-metric D:\JavaApps\Doutorado\Pathfinder\softwarepathfinder\src\ -id --console --top 5
```

```
-----  
Type: com.softwarepathfinder.visualization.prefuse.AggregateDemo      SLOC: 300      Number of Internal Dependencies: 5
```

```
INTERNAL DEPENDENCIES:
```

```
com.softwarepathfinder.model.Invocation  
com.softwarepathfinder.model.Method  
com.softwarepathfinder.model.Path  
com.softwarepathfinder.model.Project  
com.softwarepathfinder.model.Type
```

```
-----  
Type: com.softwarepathfinder.model.Type SLOC: 246      Number of Internal Dependencies: 2
```

```
INTERNAL DEPENDENCIES:
```

```
com.softwarepathfinder.persistence.jpa.PersistenceManager  
com.softwarepathfinder.utils.Log
```

```
-----  
Type: com.softwarepathfinder.parsing.java.InvocationVisitor      SLOC: 220      Number of Internal Dependencies: 5
```

```
INTERNAL DEPENDENCIES:
```

```
com.softwarepathfinder.model.Field  
com.softwarepathfinder.model.Method  
com.softwarepathfinder.model.Project  
com.softwarepathfinder.model.Type  
com.softwarepathfinder.utils.Log
```

```
-----  
Type: com.softwarepathfinder.visualization.yEd.TypeGraphML      SLOC: 199      Number of Internal Dependencies: 6
```

```
INTERNAL DEPENDENCIES:
```

```
com.softwarepathfinder.model.Field  
com.softwarepathfinder.model.Invocation  
com.softwarepathfinder.model.Locus  
com.softwarepathfinder.model.Method  
com.softwarepathfinder.model.Type  
com.softwarepathfinder.utils.exceptions.ConfException
```

```
-----  
Type: com.softwarepathfinder.parsing.php.InvocationVisitor      SLOC: 181      Number of Internal Dependencies: 3
```

```
INTERNAL DEPENDENCIES:
```

```
com.softwarepathfinder.model.Method  
com.softwarepathfinder.model.Project  
com.softwarepathfinder.model.Type
```

```
Processing time: 708 milliseconds
```



**metric  
visualization**  
**DR-Tools**

# Uso



**DR-Tools**

## Metric Visualization

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

PROJECT SUMMARY

### Software Pathfinder

[View Thresholds](#)

**13**

Namespaces

**4665**

SLOC

**65**

Number of Types

**402**

Number of Methods



**DR-Tools**



# Uso



## Thermometer

Summary Visualization



## Namespaces

Using NOC and NAC



## Types

Using NOM, SLOC, and WMC



## Methods

Using CYCLO, MLOC, and CALLS



## Internal Dependencies

Internal dependencies of types/classes



## Type Coupling

Coupling between types/classes (input/output)



## Namespace Coupling

Using CA and CE



## Instability/Abstractness/Distance

Using LA and D



## Instability and Abstractness

Using LA and A



**DR-Tools**



## Metric Thresholds Information

Project: Software Pathfinder

[Back](#)

### PROJECT

Acronym	Name	Description
<b>SMALL</b>	Small Project	small project with < 50 KLOC or 200 < classes
<b>MEDIUM</b>	Medium Project	medium project with (50 KLOC <= project <= 250 KLOC) or (200 <= classes <= 1000)
<b>LARGE</b>	Large Project	large project with > 250 KLOC or > 1000 classes

### NAMESPACE

Acronym	Name	Description
<b>NOC</b>	Number of Types/Classes	Good: <= 11; Regular: between 11 and 28; Bad: > 28
<b>NAC</b>	Number of Abstract Types/Classes	without references

### TYPE

Acronym	Name	Description
<b>SLOC</b>	Type/Class Line of Code	Bad: > 500



**DR-Tools**

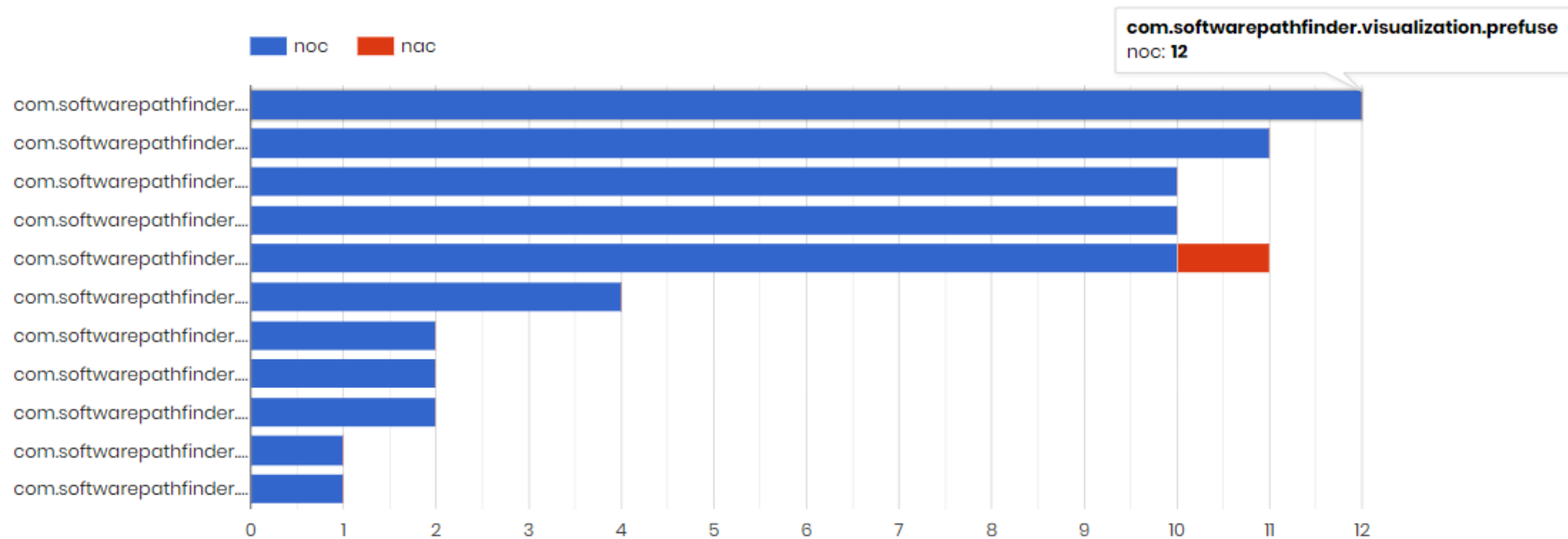


## Namespace Visualization

NOC (Number of Classes/Types) and NAC (Number of Abstract Classes/Types)

Project: Software Pathfinder

[Back](#)



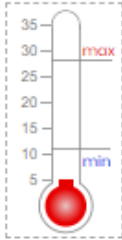
DR-Tools

## Thermometer Visualization

Project: Software Pathfinder

[Back](#)

### Types (types/namespaces)



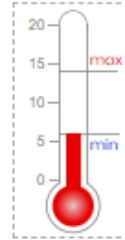
Total of Namespaces: 13

Total of Types: 65

Types/namespaces: 5

Total of SLOC: 4665

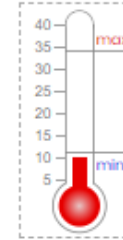
### Methods (methods/types)



Total of Methods: 402

Methods/Types: 6

### Complexity (WMC/types)



Total of Complexity: 673

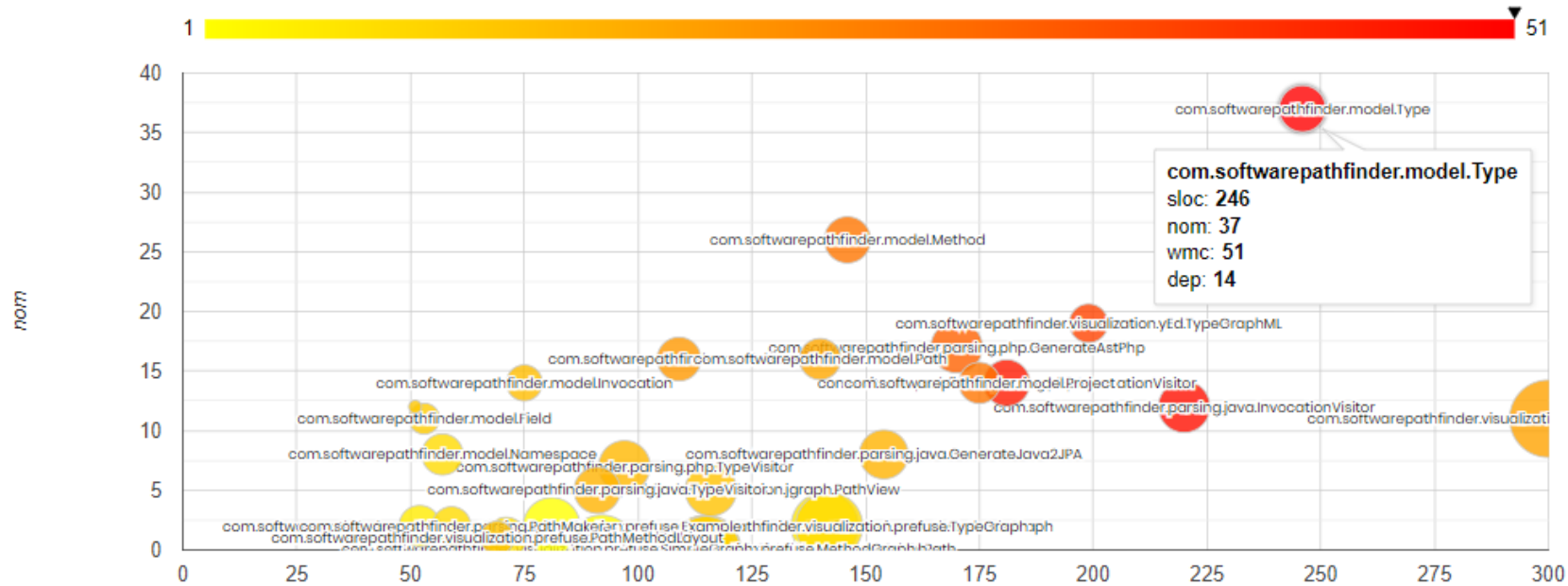
Complexity/Types: 10



## 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: Software Pathfinder

[Back](#)

## DR-Tools

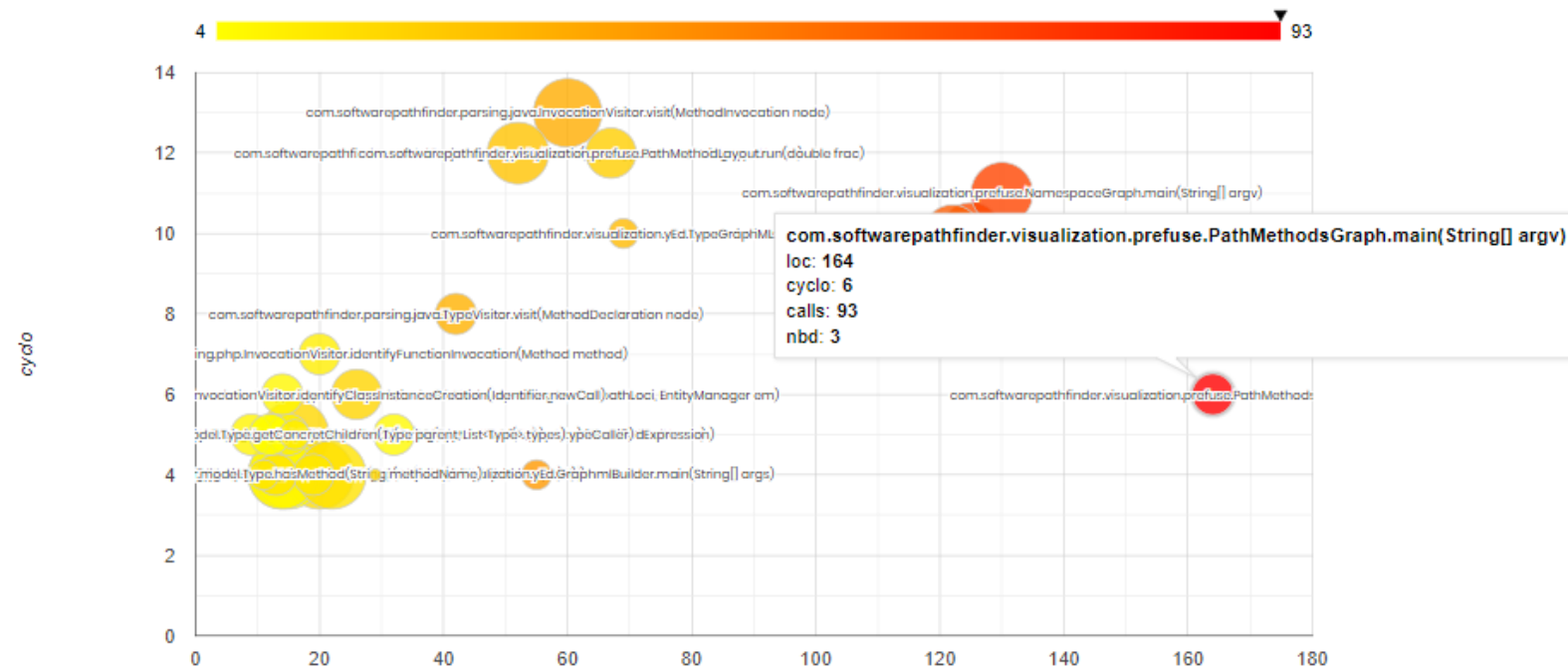


## Method Visualization

Methods with Complexity (CYCLO - y axis), Lines of Code (MLOC - x axis), Number of Invocations (CALLS - bubble color), and Nested Block Depth (NBD - bubble size)

Project: Software Pathfinder

[Back](#)



DR-Tools



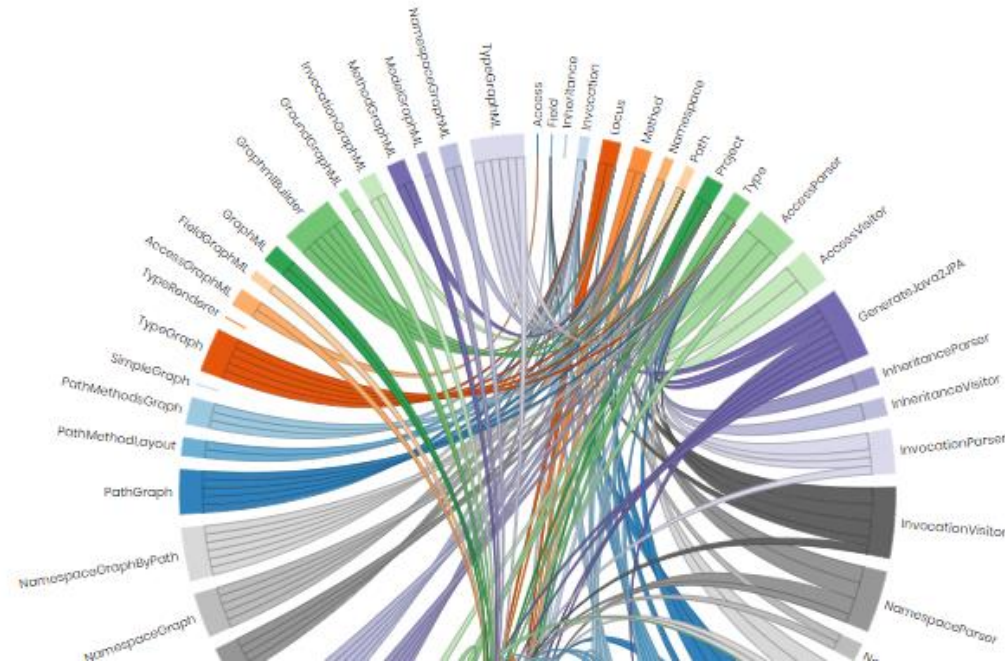


## Internal Dependencies Visualization

Internal dependencies between type/classes

Project: Software Pathfinder

[Back](#)



**DR-Tools**

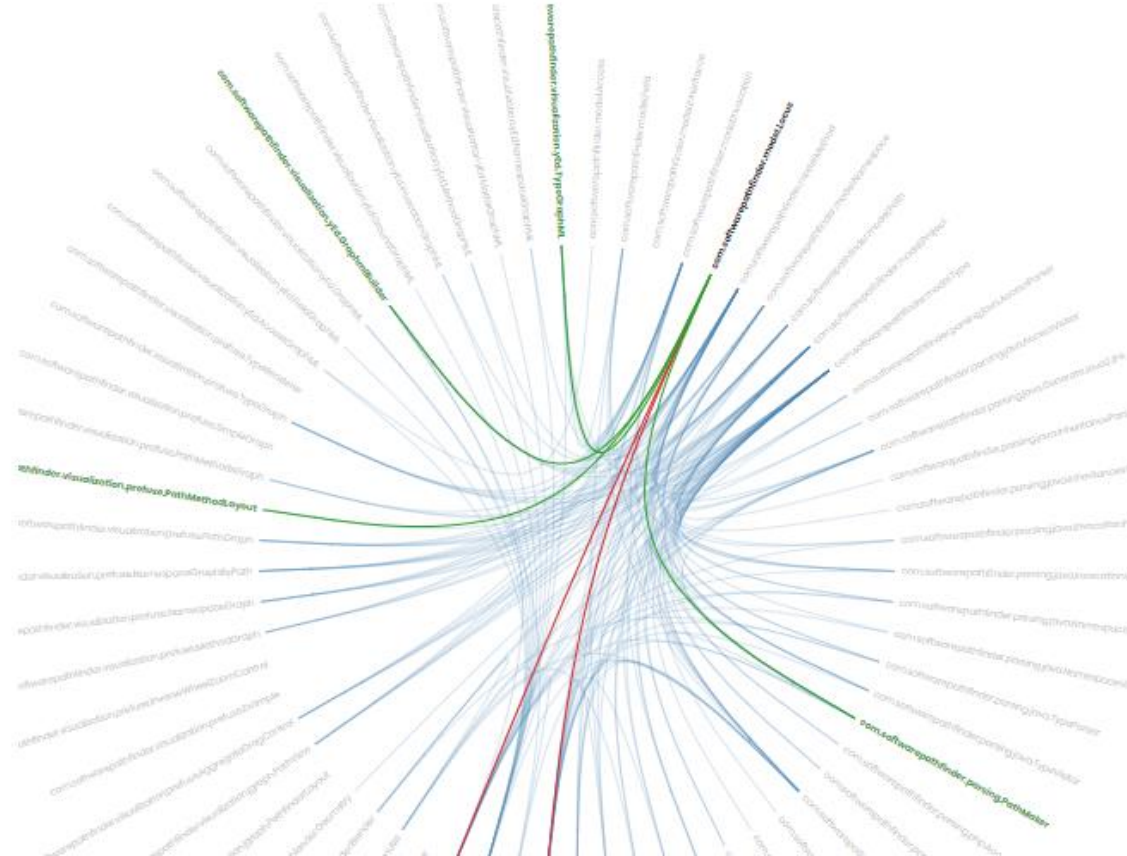
# Uso

## Coupling (Input and Output) Visualization

Red lines (output coupling) and green lines (input coupling)

Project: Software Pathfinder

[Back](#)

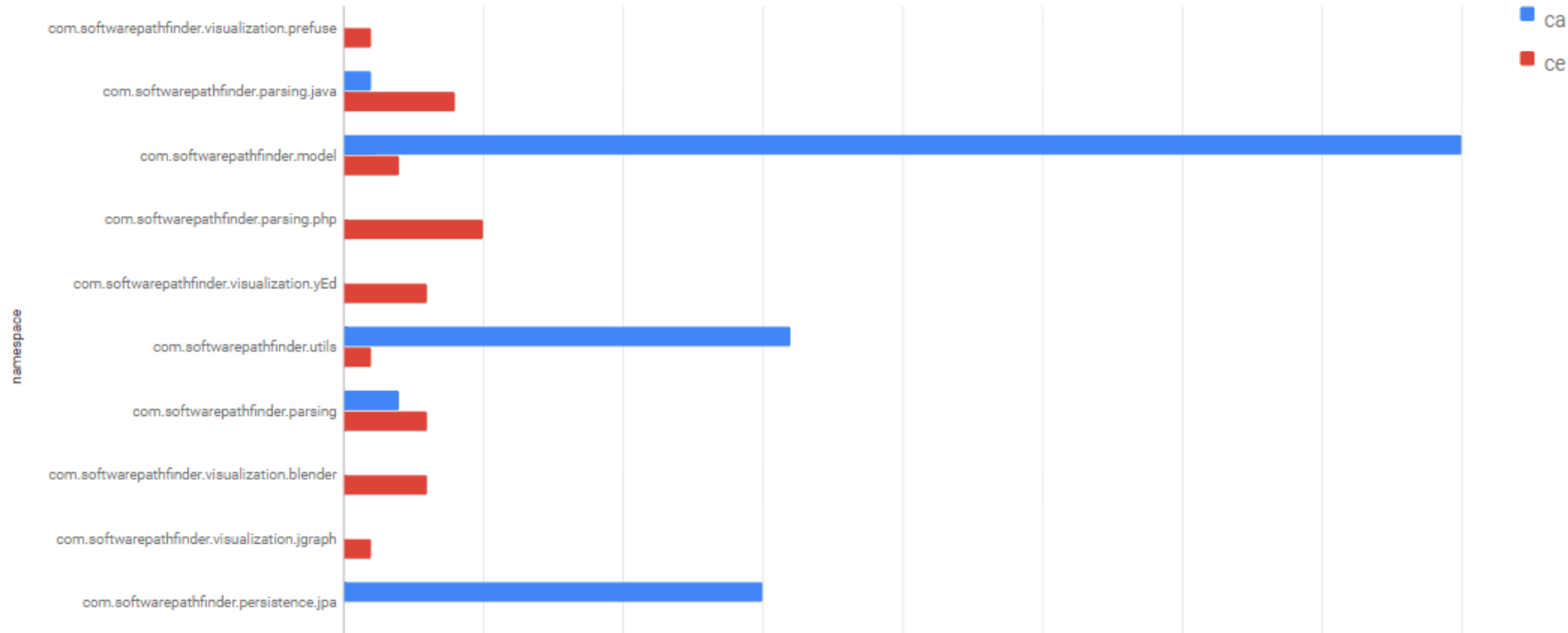


DR-Tools

## Namespace Coupling Visualization

CA (Afferent Coupling) and CE (Efferent Coupling)  
Project: Software Pathfinder

[Back](#)





# Instability/Abstractness/Distance Visualization

I (Instability), A (Abstractness Degree), and D (Normalized Distance)

Project: Software Pathfinder

[Back](#)



DR-Tools

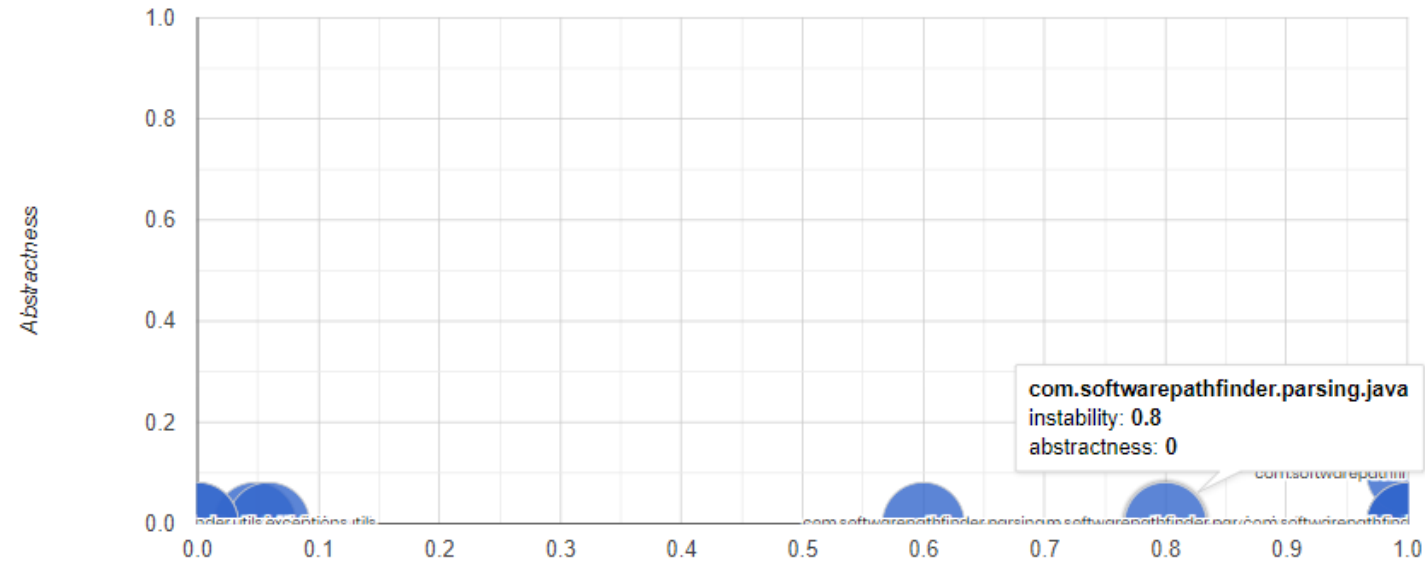


## Instability and Abstractness Visualization

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

Project: Software Pathfinder

[Back](#)



DR-Tools



# DR-Tools

---

32 Heurísticas de Análise

**[drtools.dev](https://drtools.dev)**



# Heurísticas - Summary

- 1. Contexto que fornece informações gerais sobre dimensões do Projeto**  
*Indicativo de filtrar melhor as informações usando a opção '--top X' para ajudar no entendimento (Classificação: SMALL, MEDIUM e LARGE)*
- 2. Considere o número médio de classes por namespace**  
*Indicativo de que as classes não estão distribuídas uniformemente*
- 3. Avalie o número médio de SLOC por classes**  
*Indicativo de classes muito grandes*
- 4. Observe a distribuição média de métodos por classes**  
*Indicativo de muitos comportamentos por classe*
- 5. Considere a complexidade média por classes**  
*Indicativo de como está a complexidade das classes em geral*



# Heurísticas - Namespaces

## 6. Observe a distribuição de classes por namespace

*Se um namespace tem muitas classes (NOC alto), pode ser um indicativo de 'promiscuous package'*

## 7. Avalie a distribuição de tipos abstratos (classes abstratas, interfaces) por namespaces

*Indicativos para extensão e reuso*

## 8. Avalie a relação das métricas NOC e NAC do namespace

*Uma diferença muito grande entre eles pode indicar uma má distribuição entre tipos abstratos e tipos concretos*



# Heurísticas – Types (1)

## 9. Avalie as métricas além do SLOC

*WMC, DEPS (DEP e I-DEP) e NOM/NPM são bons indicativos de como está a classe*

## 10. Classe com NOA alto, mas baixo WMC e NOM alto

*Pode ser um indicativo de POJO (Plain Old Java Object)*

## 11. SLOC alto, mas sem muitos métodos (NOM/NPM baixo)

*Pode ser um indicativo de 'long methods'*

## 12. SLOC e WMC alto, mas sem muitos métodos (NOM/NPM baixo)

*Pode ser um indicativo de 'complex class'*

## 13. NOM/NPM alto pode ser indicativo de classe com muitas responsabilidades

*Indica baixa coesão e possivelmente 'god class'*



# Heurísticas – Types (2)

- 14.** **NOM/NPM alto e NOA baixo pode ser indicativo de classe com muitas responsabilidades**  
*Pode ser um indicativo de uma classe 'controller'*
- 15.** **NOM alto e NPM baixo pode indicar que o comportamento foi dividido**  
*Indicativo de métodos private/protected/default*
- 16.** **NOA alto pode ser indicativo de classe com muitas responsabilidades**  
*Pode ser um indicativo de baixa coesão, dificultando a manutenção*
- 17.** **DEP alto e I-DEP baixo pode indicar uma classe com muitas dependências externas**  
*Dependências de APIs externas (frameworks, libs)*
- 18.** **I-DEP alto (e por consequência DEP alto), pode indicar uma classe com muitas dependências de classe do projeto**  
*Incidência de alto acoplamento*



# Heurísticas – Methods

- 19.** PARAM alto pode ser indicativo de método com baixa coesão  
*Possivelmente é um 'long method'*
- 20.** CYCLO alto e MLOC baixo pode ser um 'complex method'  
*Indicativo de problema de complexidade, legibilidade e entendimento*
- 21.** NBD alto pode ser um 'complex/long method'  
*Indicativo de problema de complexidade, legibilidade e entendimento*
- 22.** CALLS alto pode indicar alto acoplamento  
*Indicativo de problema de várias dependências*
- 23.** MLOC alto, CYCLO alto, CALLS alto e NBD alto é forte indicativo de mais de um problema  
*Pode ser um indicativo de um 'complex/long method'*



# Heurísticas – Coupling (1)

**24. Evite dependência cíclicas**

*Tornam as mudanças complexas e gera a 'síndrome da compilação total'*

**25. CA alto pode indicar que o namespace é estável**

*Se ele mudar, vai fazer com que quem dependa dele seja alterado*

**26. CE alto pode indicar que o namespace é instável**

*A incidência de mudança em outros namespaces que ele depende vai fazer com que ele mude*

**27. I indica como está a instabilidade do namespace**

*I=0, namespace estável ao máximo; I=1, namespace instável ao máximo*



# Heurísticas – Coupling (2)

- 28.** Se  $I=0$ , indica que  $CA > 0$  e  $CE=0$ , indica uma estabilidade total  
*Ele é responsável e independente. Os dependentes tornam difícil alterá-lo e não tem dependência de outros que pode forçar a mudança*
- 29.** A indica como está o grau de abstração do namespace  
*A=0, namespace não tem tipos abstratos; A=1, namespace somente possui tipos abstratos*



# Heurísticas – Coupling (3)

- 30.** Considere namespaces que estão nas zonas de exclusão

*Zone of Pain (namespaces com I e A próximos a 0) e Zone of Uselessness (namespaces com I e A próximos a 1)*

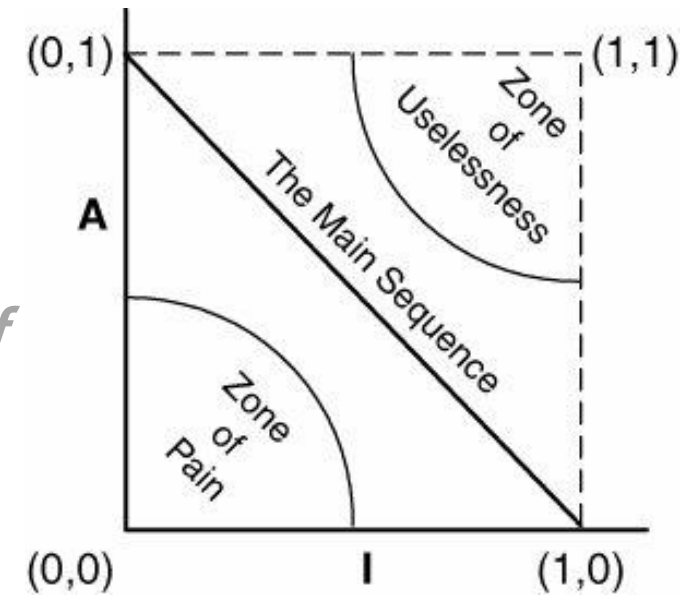
- 31.** Namespace situado próximo a sequência principal indica que não é abstrato nem instável demais

*Valor de D (entre 0 e 1) que vai indicar a posição na sequência principal*

- 32.** D indica o quão longe um namespace está da sequência principal

*D próximo a 0 indica proximidade da sequência principal; D próximo a 1, indica distância da sequência principal*

*Estes valores (mais próximo a 1) podem indicar quando um namespace está passível de manutenção e menos sensível a mudanças*





# Live demo!

**Questões??**



# DR-Tools

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

**[drtools.site](https://drtools.site)**