Dedicated and Cost Effective Software Analysis

synectique
Inventive | Analysis

http://www.synectique.eu

Synectique Team

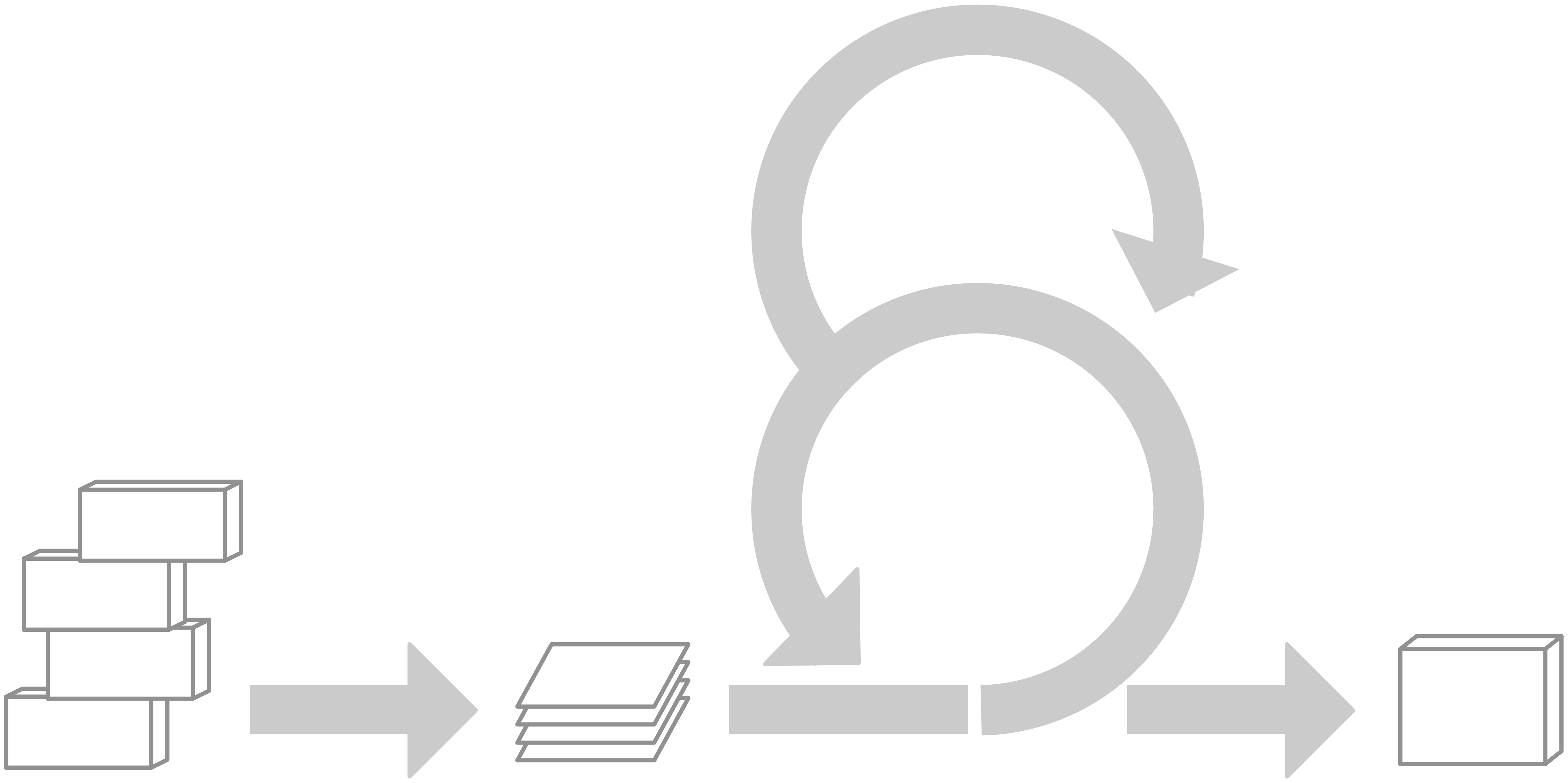
In Software Evolution and Maintenance since 1996

Author of Object-Oriented Reengineering Patterns

A team with over 35 years of combined experience in reengineering

Serge Demeyer, Stéphane Ducasse, Oscar Nierstrasz
Object-Oriented Reengineering
Patterns

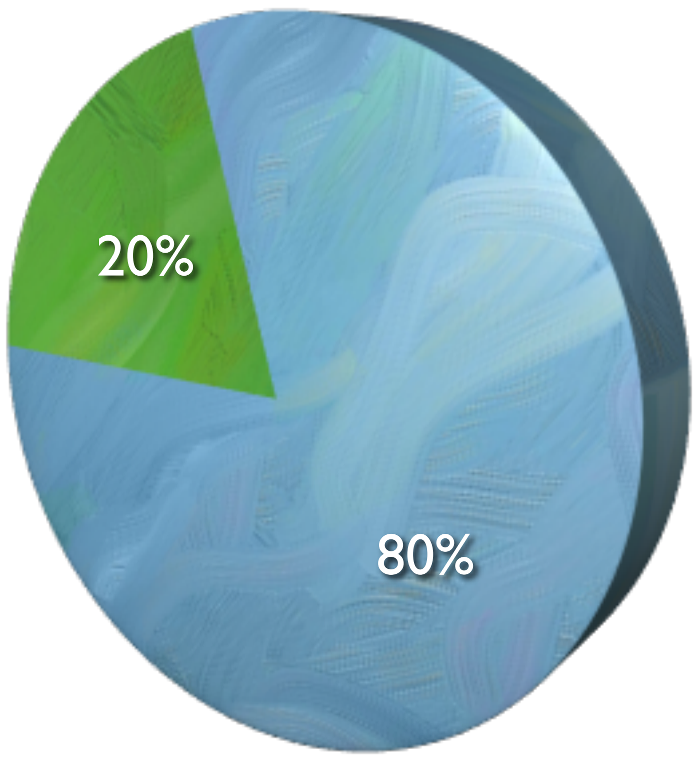
Controlling industrial processes



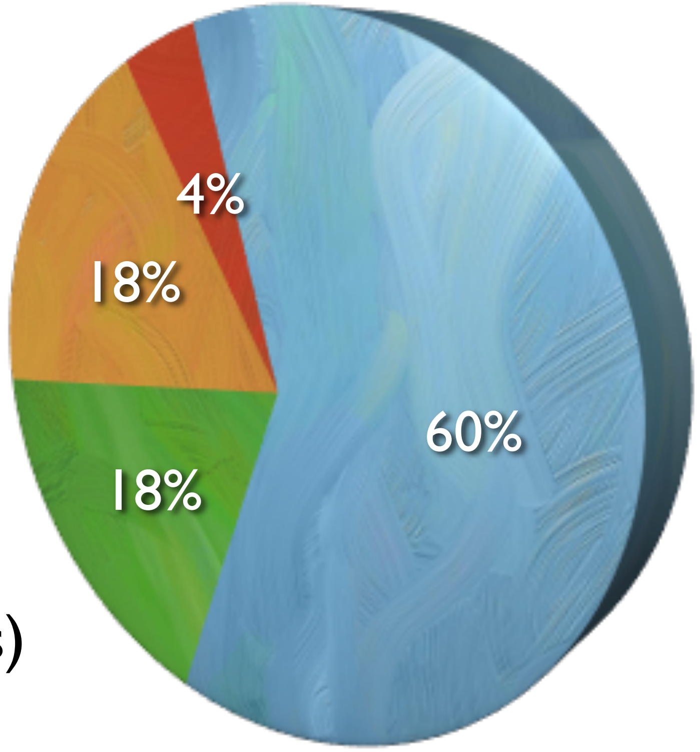
Getting feedback **is key**

$$\begin{tikzcd}
 & \text{contextual} \\
\text{But good feedback should be} \arrow[ur] \arrow[r] \arrow[dr] & \text{dedicated} \\
 & \text{continuous}
\end{tikzcd}$$

Maintenance is *continuous* development



Between **50%** and **80%** of ***global*** effort is spent on "maintenance" !



4% Other

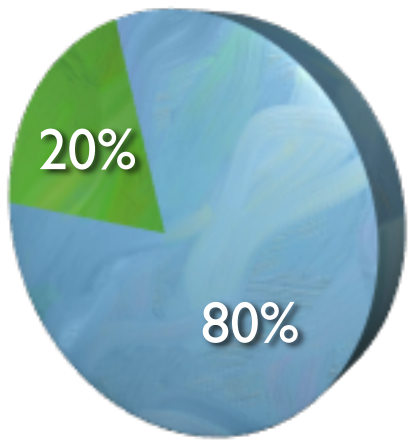
18% Adaptive
(new platforms)

18% Corrective
(fixing reported errors)

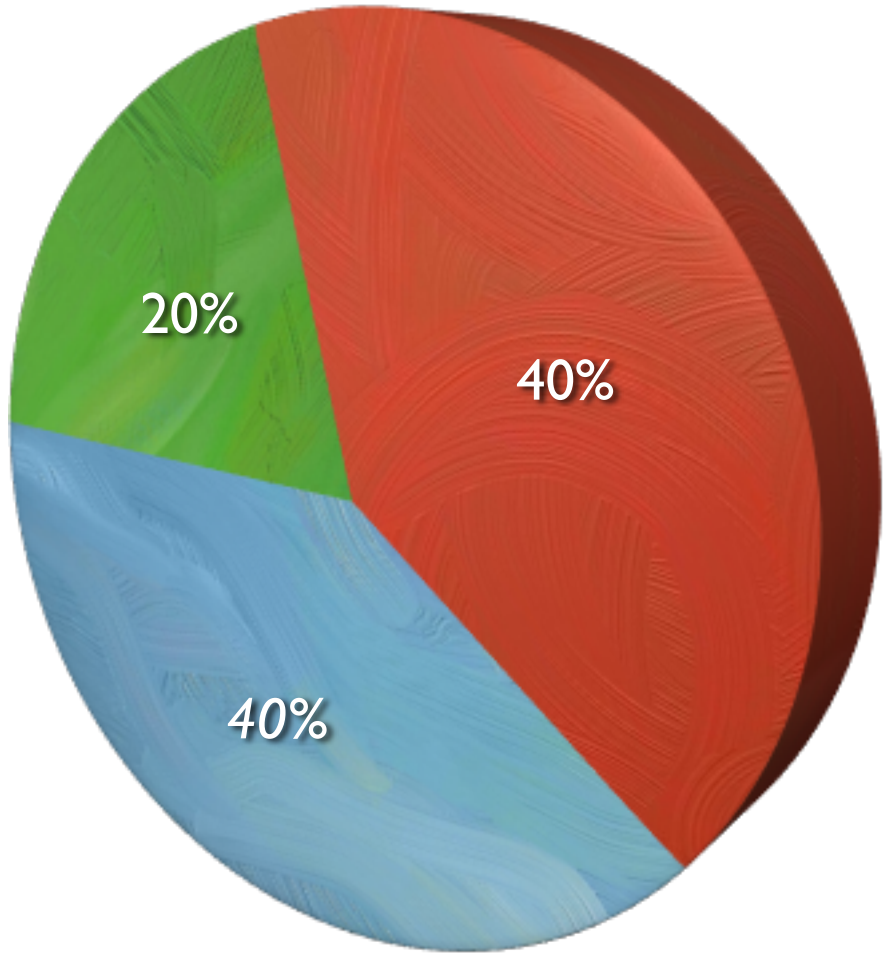
60% Perfective
(new functionality)

"Maintenance"

50% of development time is lost trying to understand code !



Between **50%** and **80%** of the ***overall cost is spent in the evolution***

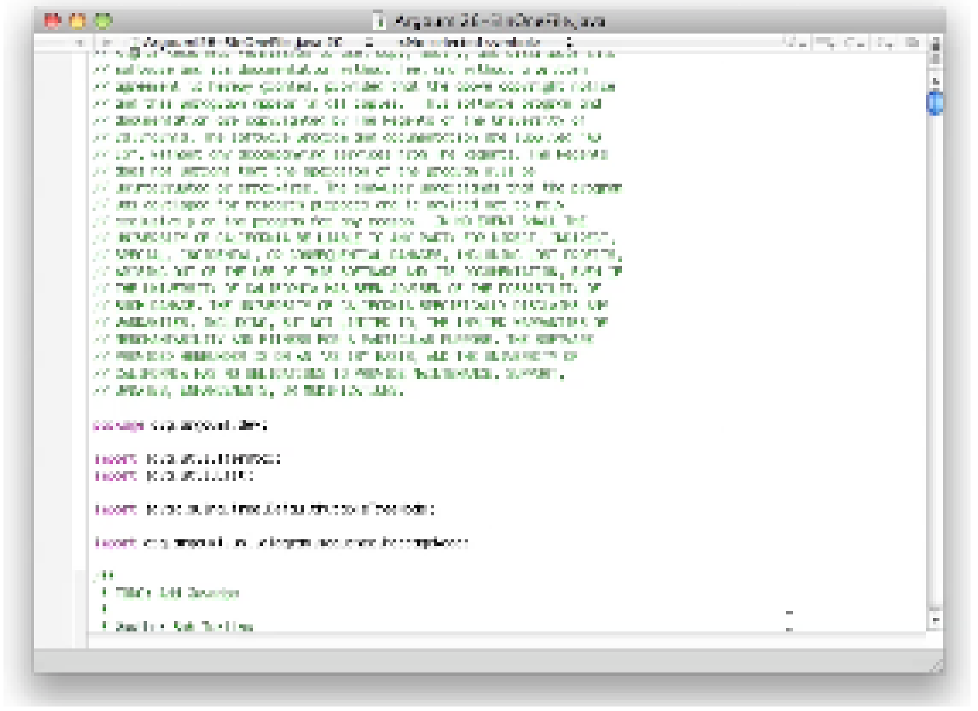


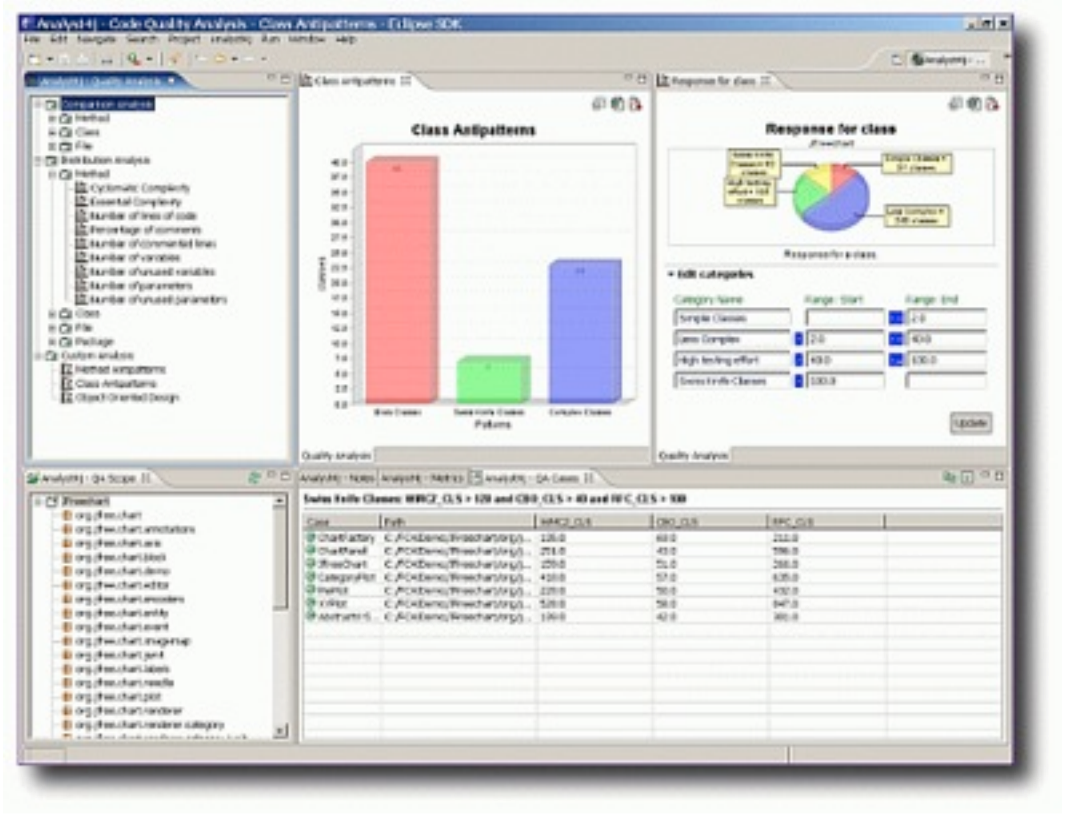
We lose a lot of time with inappropriate and ineffective practices

When did you take a *real* decision based on software metrics?

~~manual~~
dedicated

automatic
~~generic~~




Analyst4j - Code Quality Analysis - Class Antipatterns - Eclipse SDK
Class Antipatterns
Response for class
Edit categories
Update

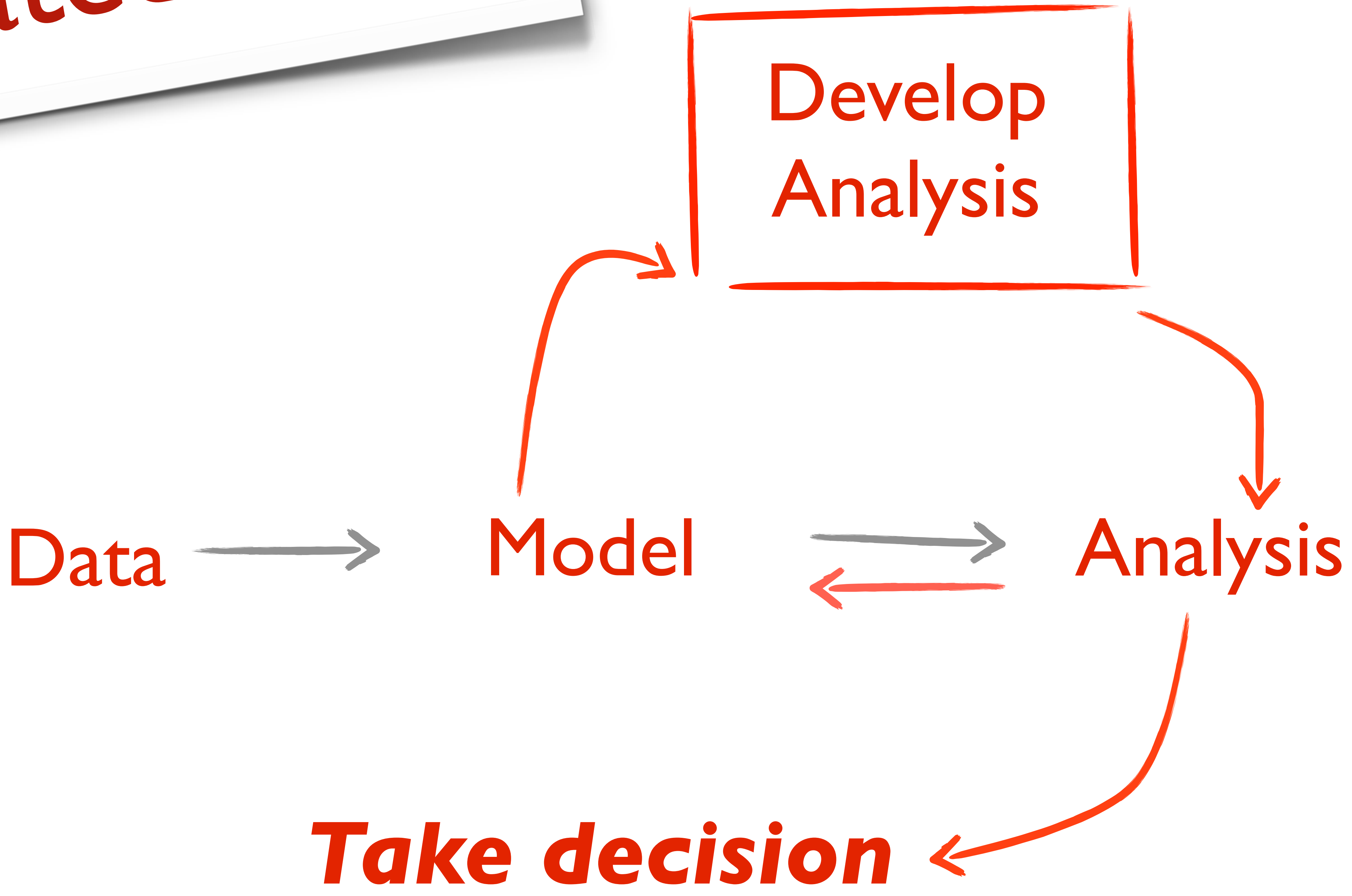
synectique
Inventive Analysis

Our solutions

Dedicated tools tailored to your problems

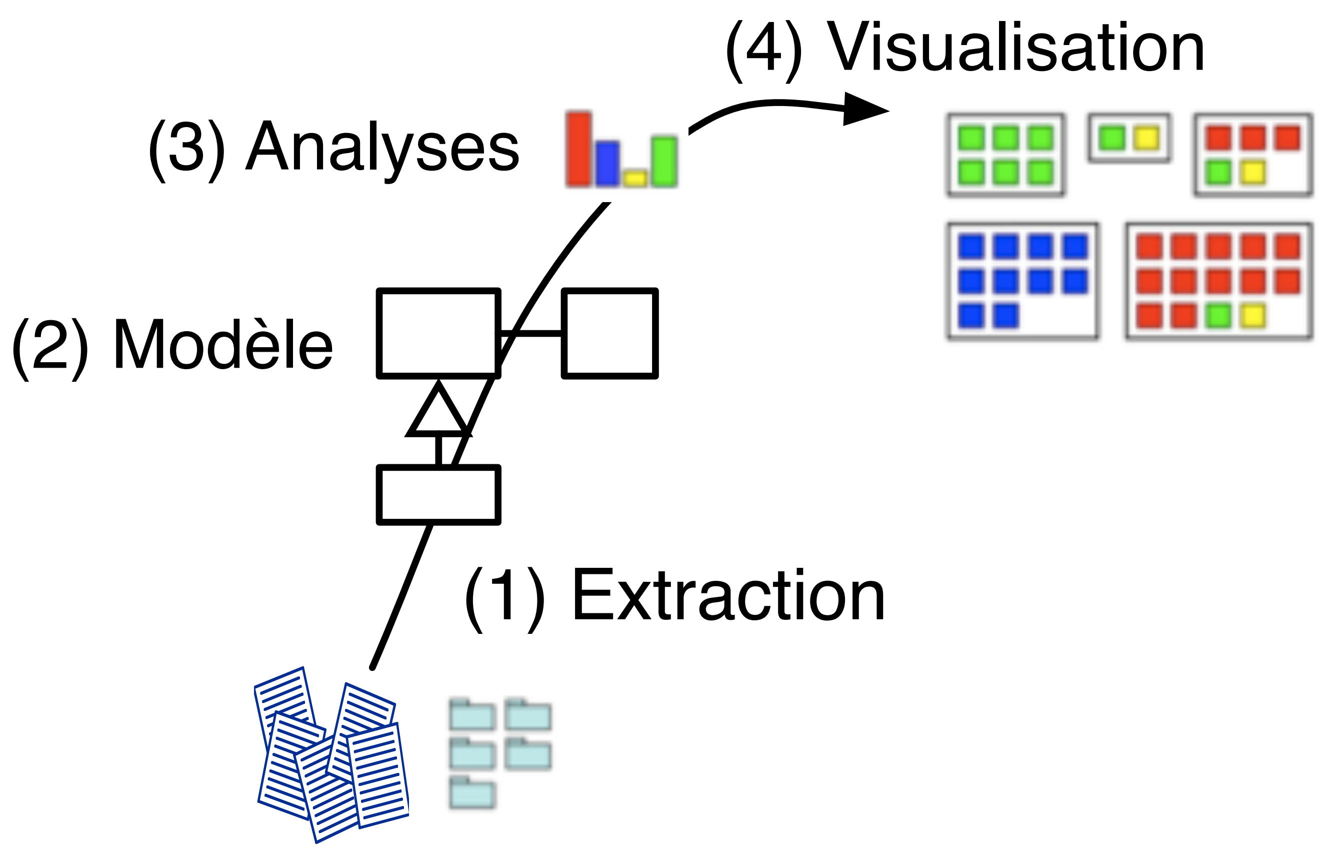
Profitable in terms of cost

Dedicated Tools

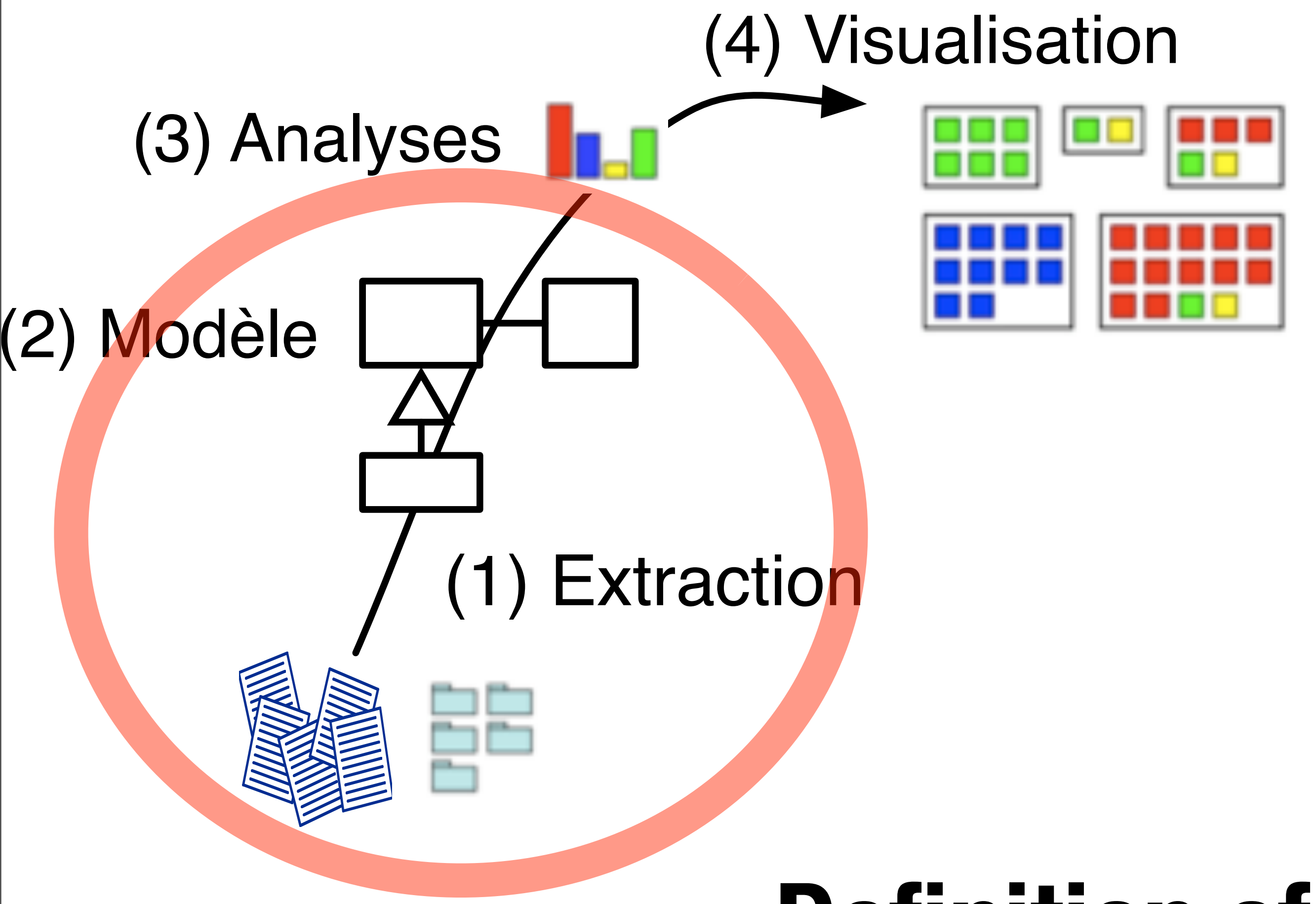


Analysis should lead to a decision

Example : Who is behind package X ?



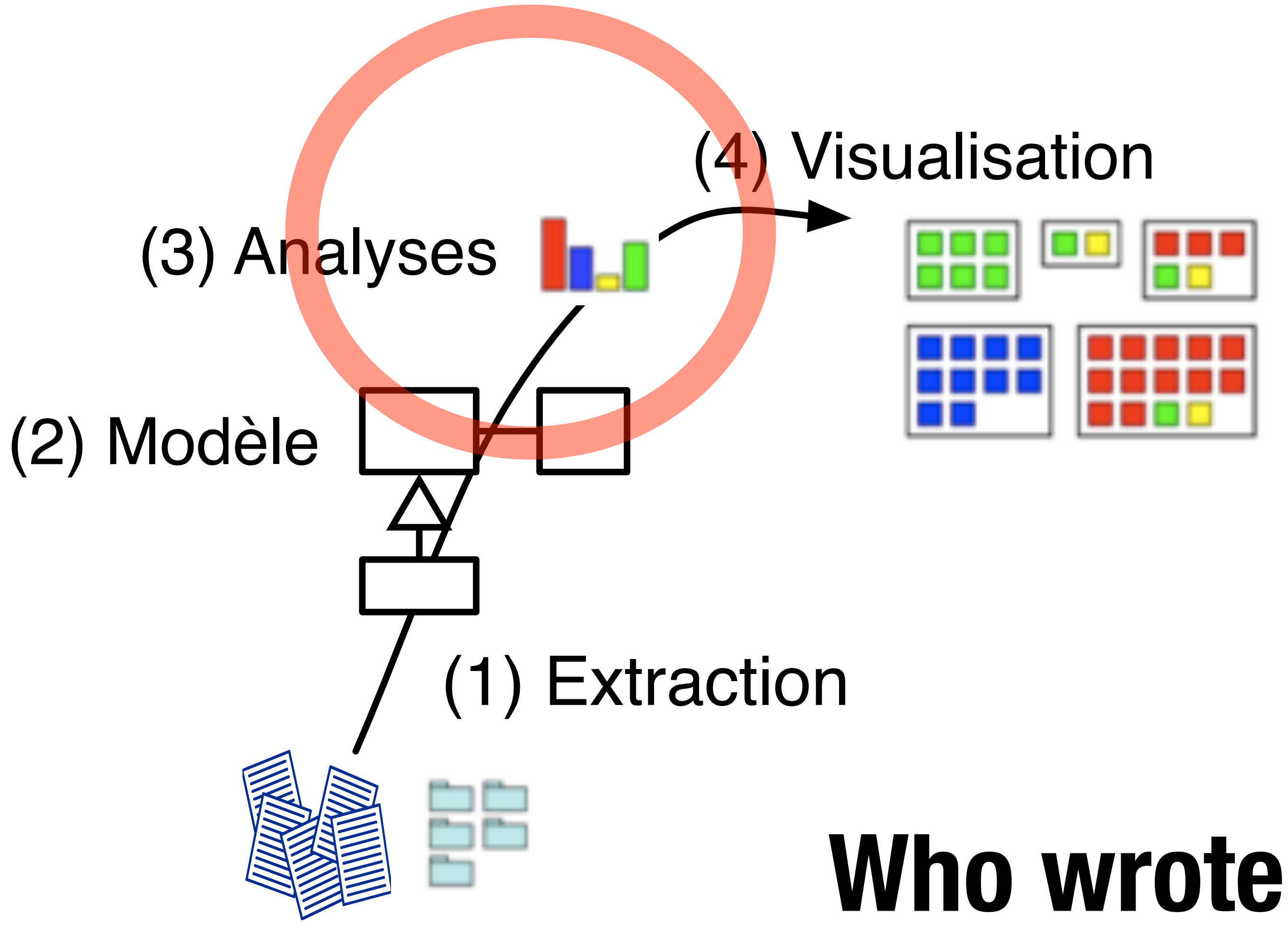
Step 1 - Model Creation/Import



Definition of a model to represent entities

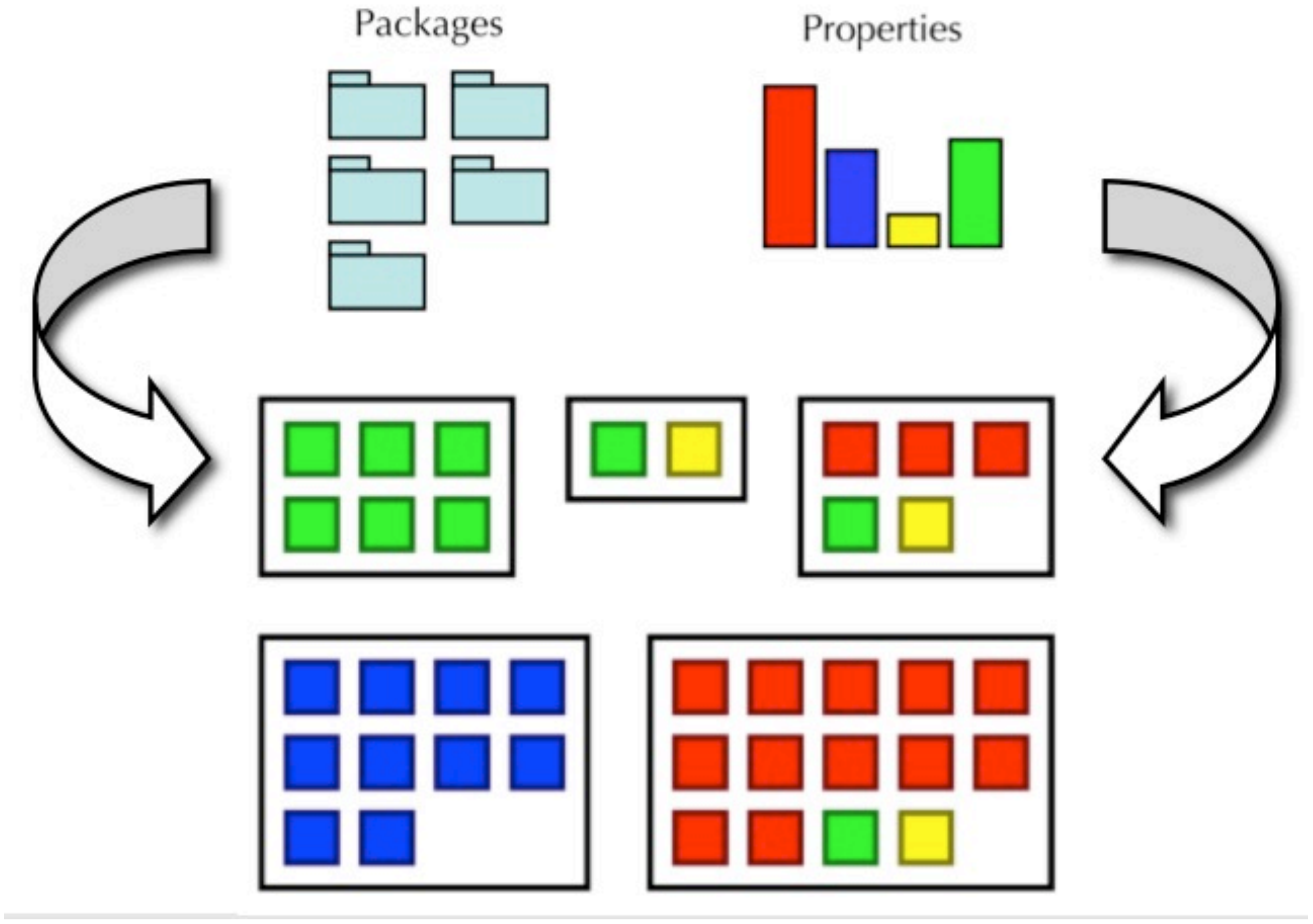
Data Extraction (CVS...)

Step 2 - Analyses



Who wrote how many lines of code?

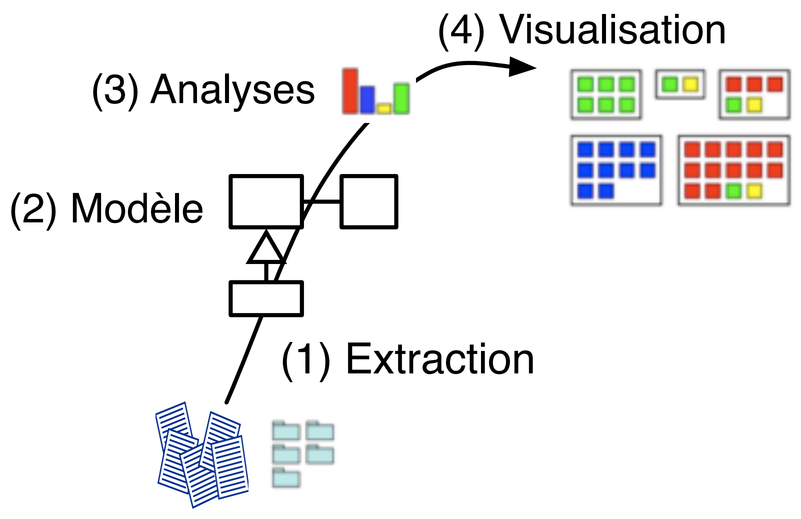
Step : 3 - Creating the Map

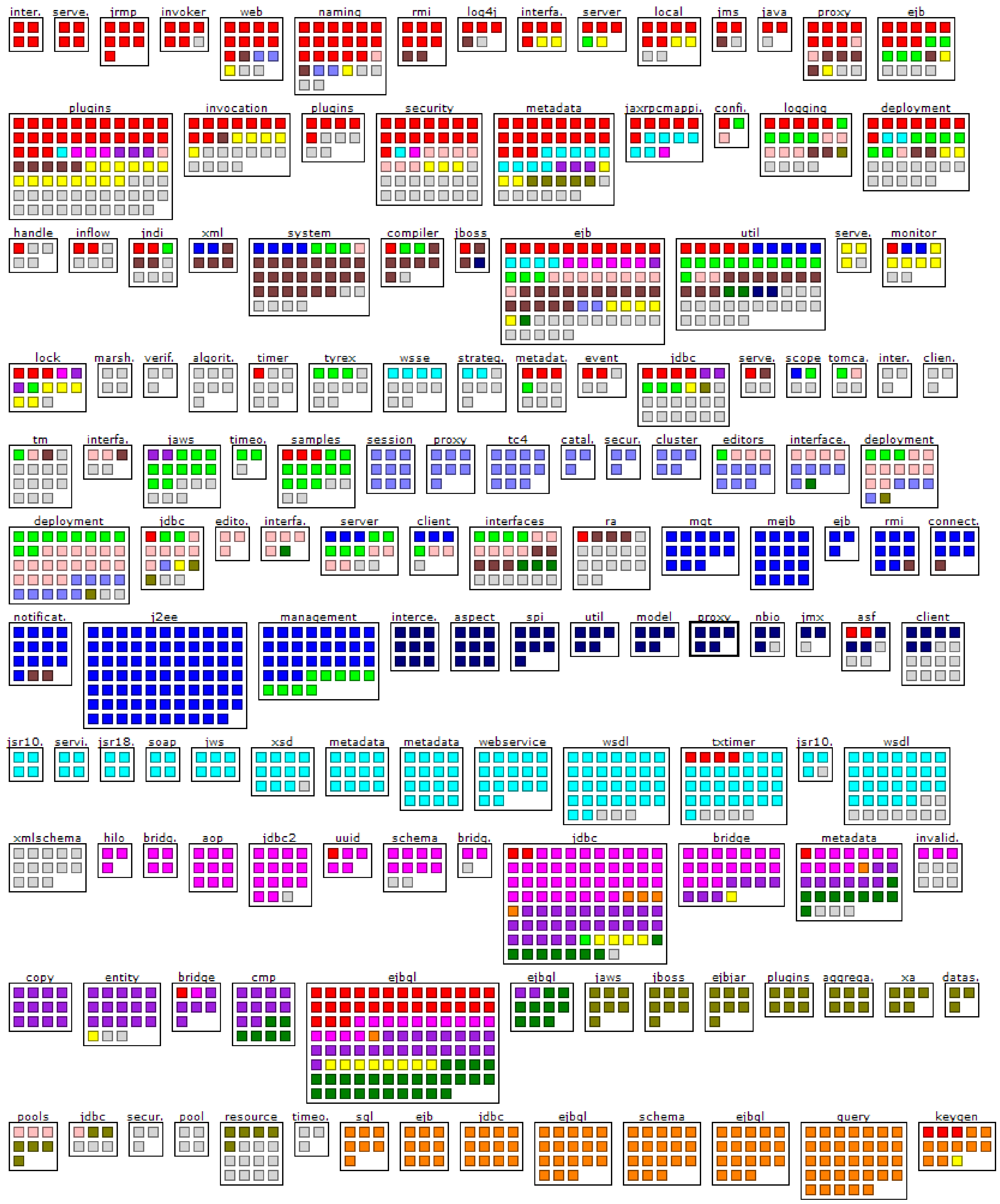


JBoss at a glance

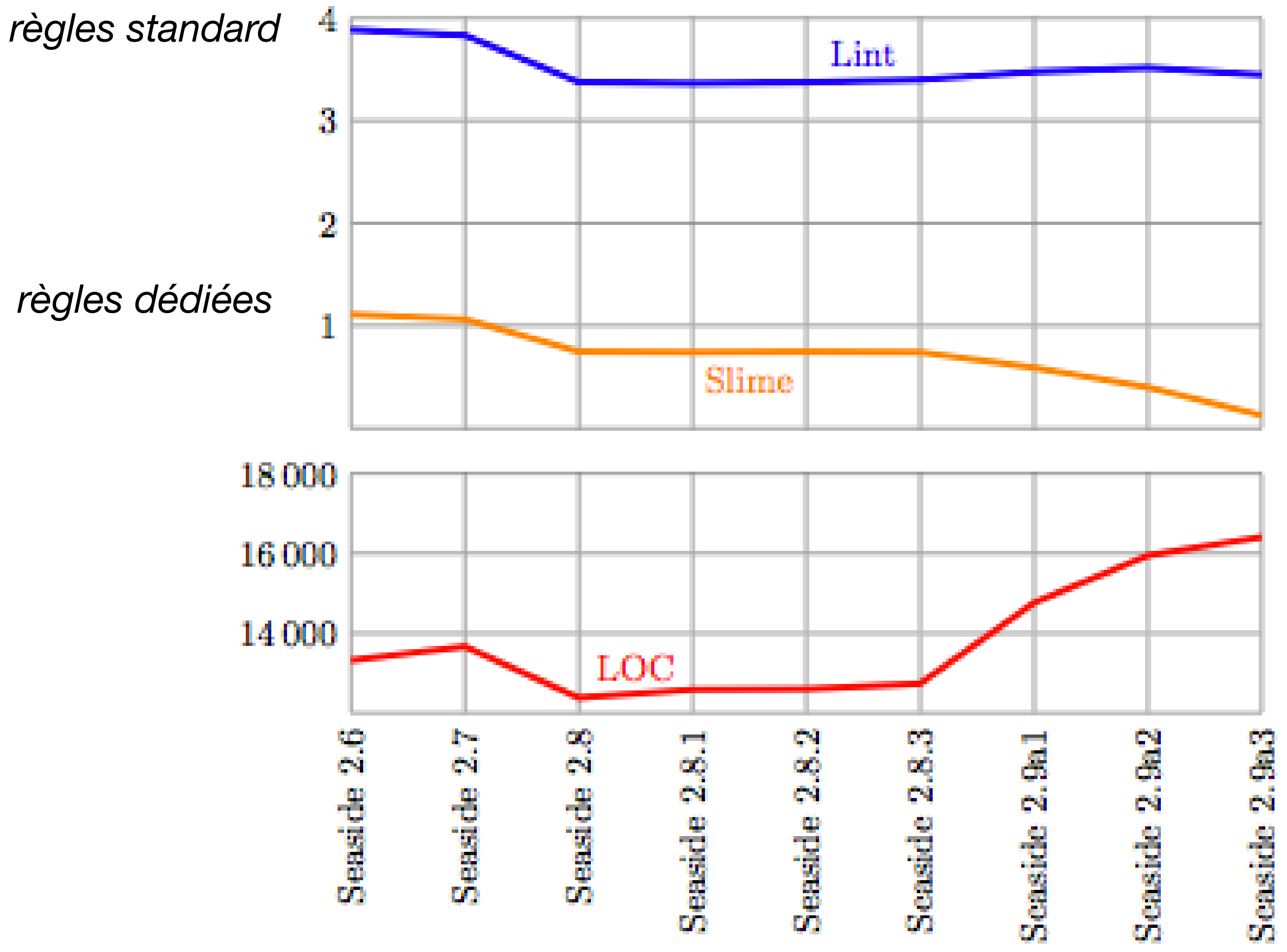
Interactive tool

Data in perspective



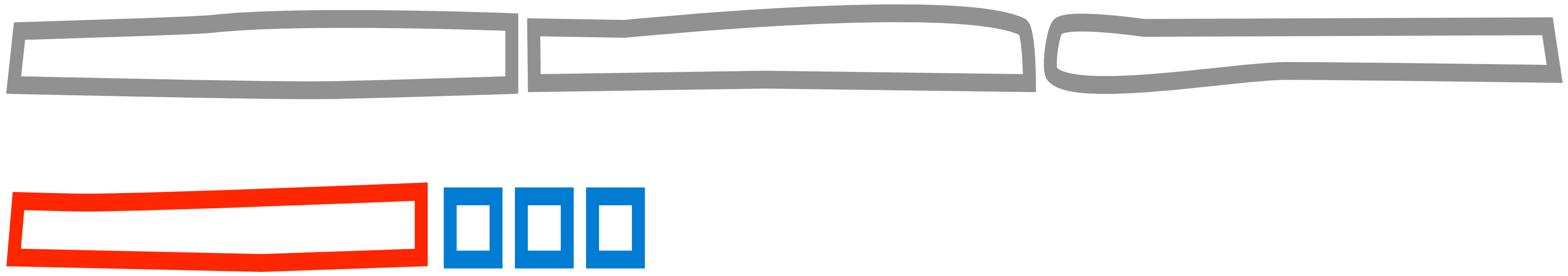


It is advantageous to carry out dedicated analysis



What about the cost of dedicated tools?

You are already paying the cost (50% of the maintenance activity that can be done efficiently with better tools)



Analysis and Migration Support

Problem: Since 30 years company X develops insurance solution. The old compiler costs more and more.

Which part to migrate first?

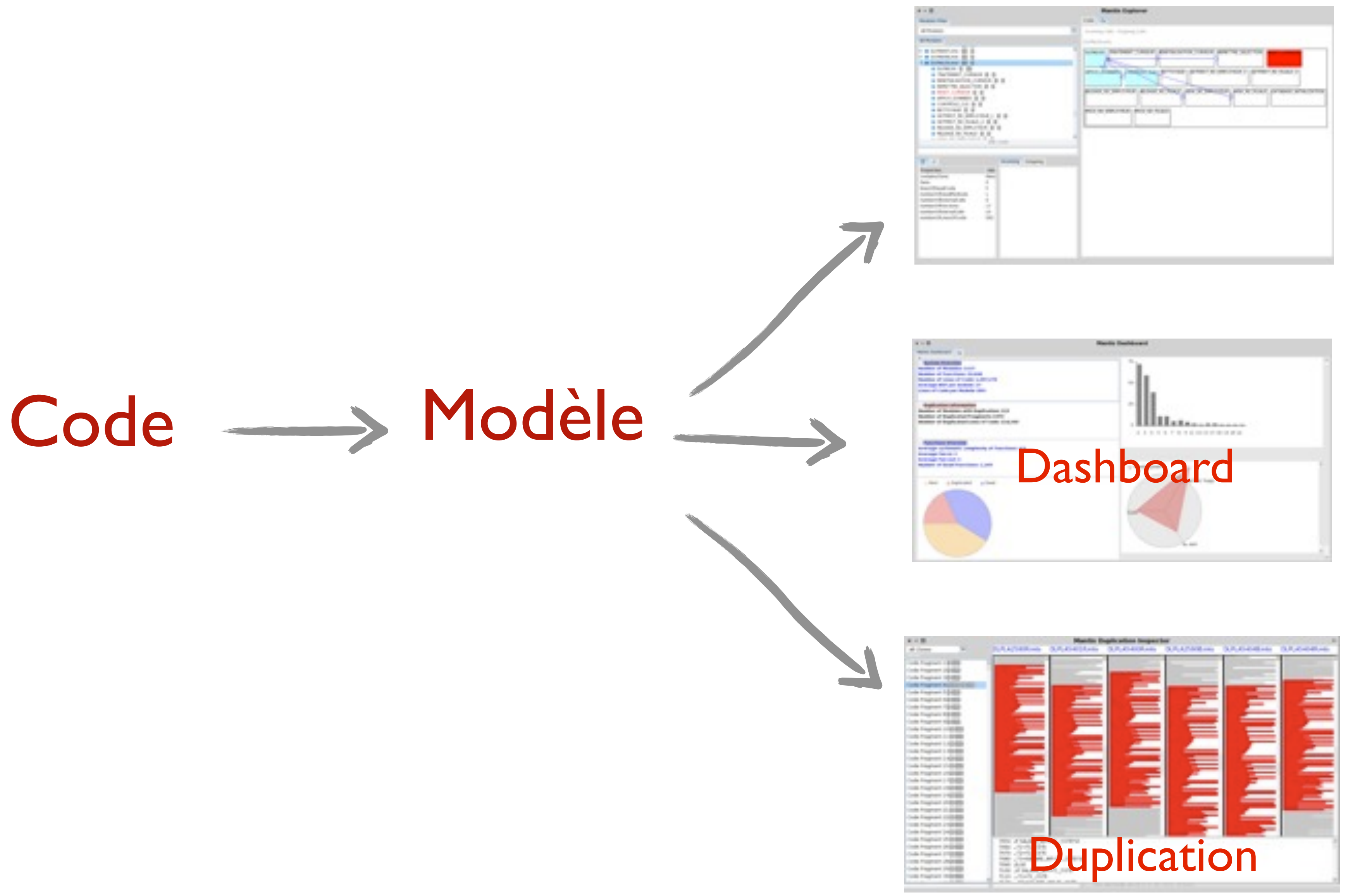
How to reduce the migration cost (duplicated code, screen numbers)?

How to control the migration?

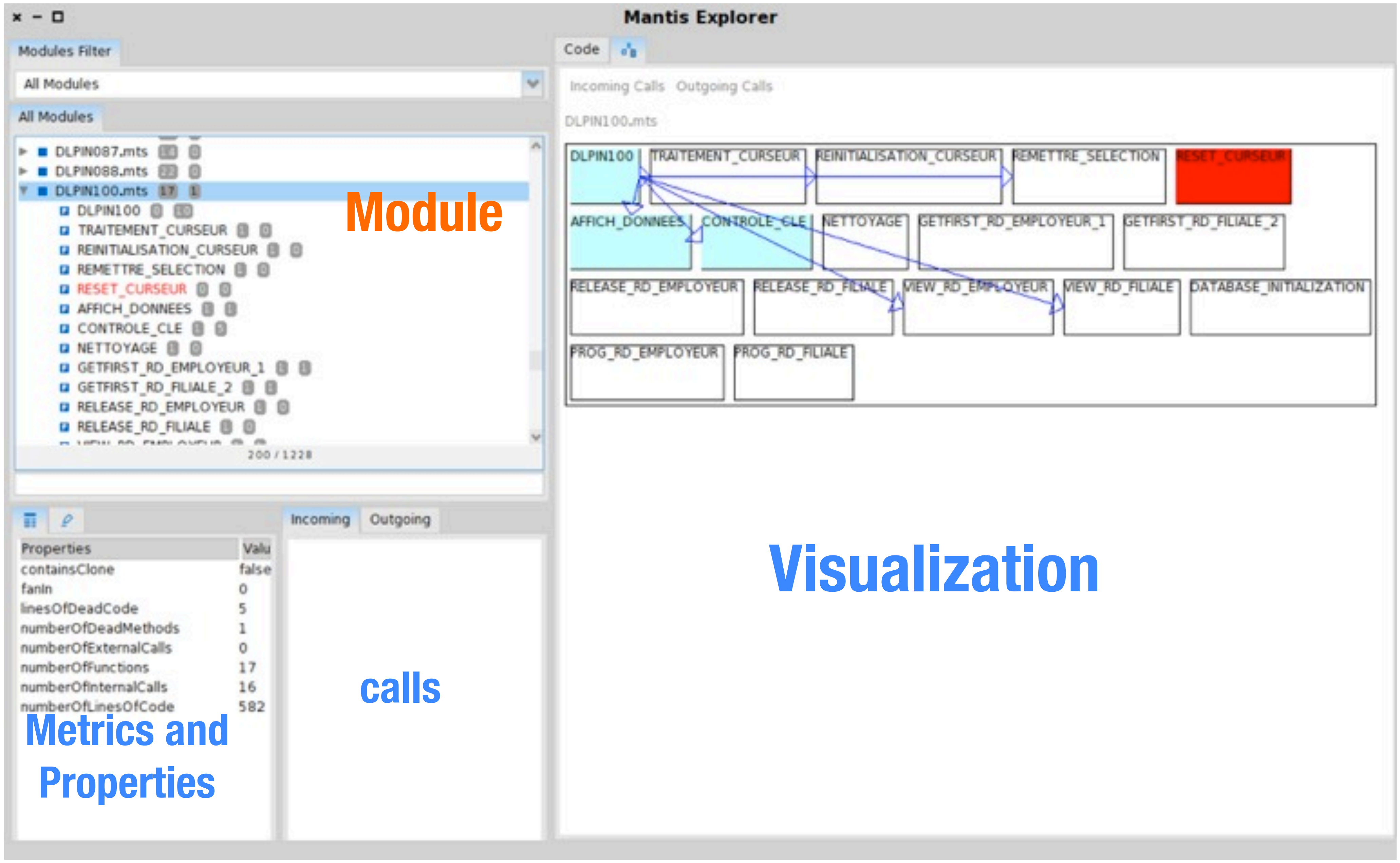
Solution :

- Build a specific analysis tools (**2 cycles of 6 weeks**)
- Domain and problem analysis
- Engineer formation

Three Levels: Three Tools



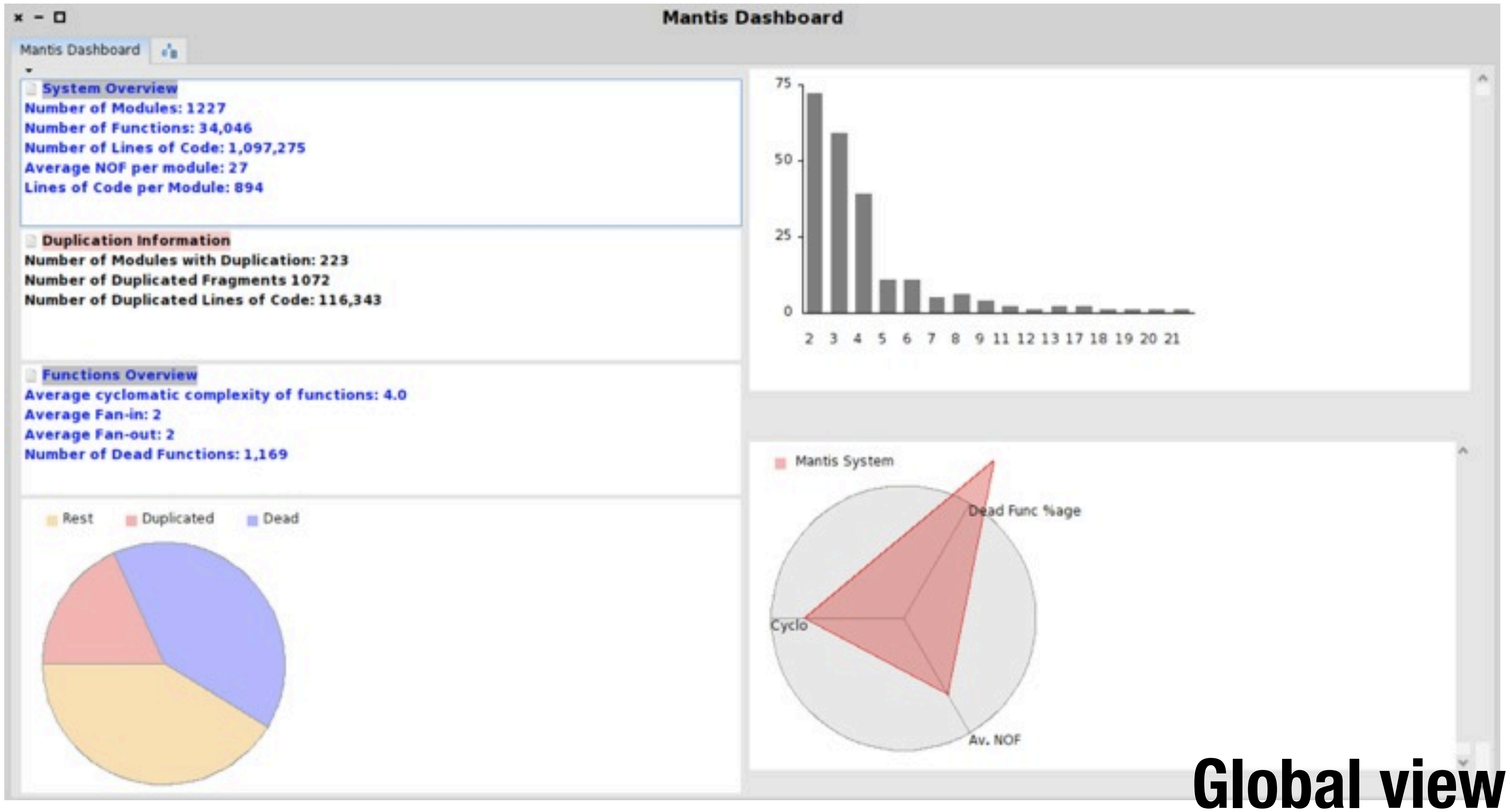
Specific Programmer IDE



```
DLPIN087.mts
DLPIN088.mts
DLPIN100.mts
    DLPIN100
    TRAITEMENT_CURSEUR
    REINITIALISATION_CURSEUR
    REMETTRE_SELECTION
    RESET_CURSEUR
    AFFICH_DONNEES
    CONTROLE_CLE
    NETTOYAGE
    GETFIRST_RD_EMPLOYEUR_1
    GETFIRST_RD_FILIALE_2
    RELEASE_RD_EMPLOYEUR
    RELEASE_RD_FILIALE
200 / 1228
```

Properties	Valu
containsClone	false
fanIn	0
linesOfDeadCode	5
numberOfDeadMethods	1
numberOfExternalCalls	0
numberOfFunctions	17
numberOfInternalCalls	16
numberOfLinesOfCode	582

Executive Dashboard



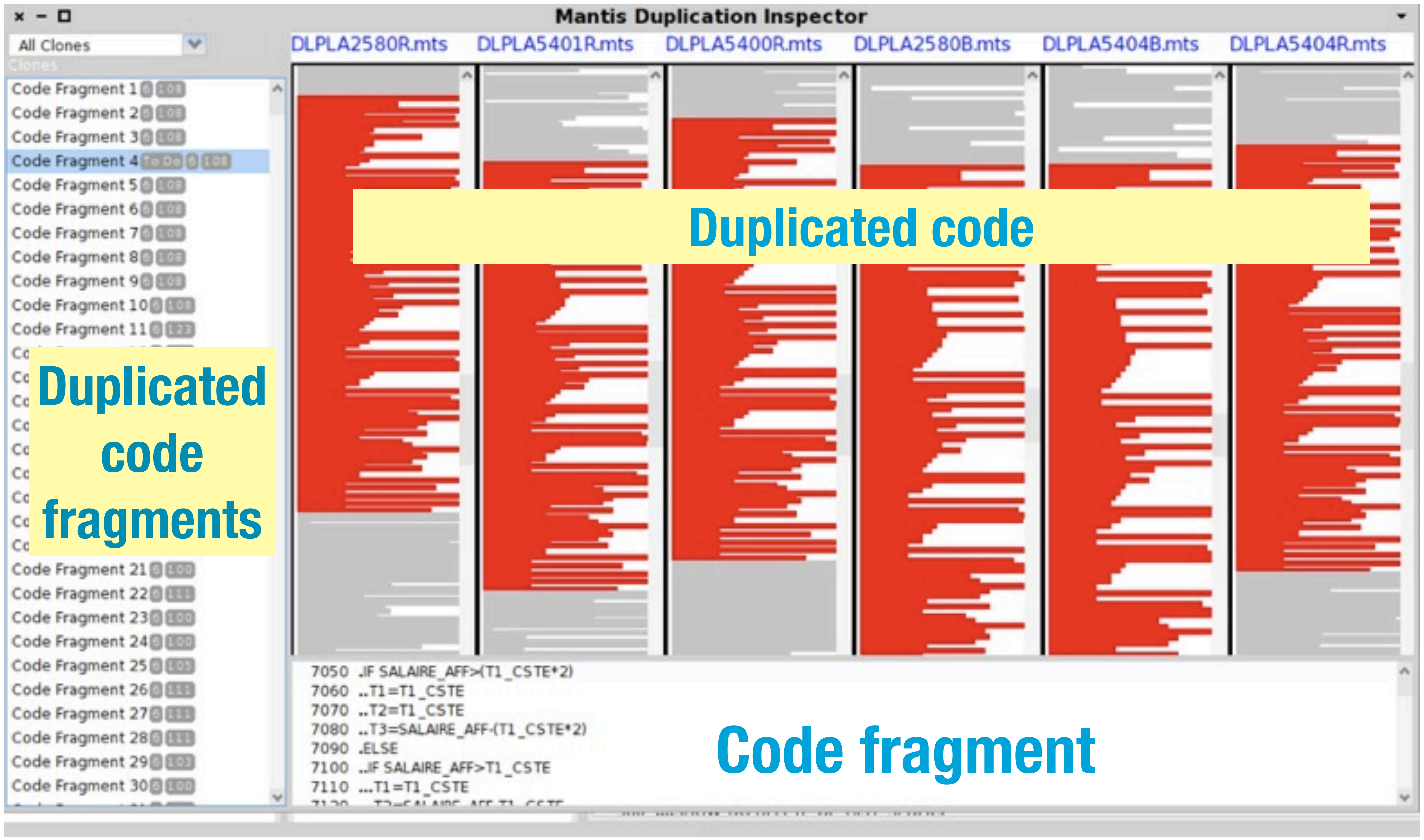
System Overview
Number of Modules: 1227
Number of Functions: 34,046
Number of Lines of Code: 1,097,275
Average NOF per module: 27
Lines of Code per Module: 894

Duplication Information
Number of Modules with Duplication: 223
Number of Duplicated Fragments 1072
Number of Duplicated Lines of Code: 116,343

Functions Overview
Average cyclomatic complexity of functions: 4.0
Average Fan-in: 2
Average Fan-out: 2
Number of Dead Functions: 1,169

Global view

Duplication Browser



Code Fragment 1 0 108
Code Fragment 2 0 108
Code Fragment 3 0 108
Code Fragment 4 To Do 0 108
Code Fragment 5 0 108
Code Fragment 6 0 108
Code Fragment 7 0 108
Code Fragment 8 0 108
Code Fragment 9 0 108
Code Fragment 10 0 108
Code Fragment 11 0 123
Code Fragment 21 0 100
Code Fragment 22 0 111
Code Fragment 23 0 100
Code Fragment 24 0 100
Code Fragment 25 0 105
Code Fragment 26 0 111
Code Fragment 27 0 111
Code Fragment 28 0 111
Code Fragment 29 0 103
Code Fragment 30 0 100

```
7050 .IF SALAIRE_AFF>(T1_CSTE*2)
7060 ..T1=T1_CSTE
7070 ..T2=T1_CSTE
7080 ..T3=SALAIRE_AFF-(T1_CSTE*2)
7090 .ELSE
7100 ..IF SALAIRE_AFF>T1_CSTE
7110 ...T1=T1_CSTE
```

Problem : Papyrus (Atos, CEA, INRIA) 800 Java packages.   For 2 years, the software suffers from problems of architecture.

Driving software architect crazy. Meetings, meetings, meetings ...

Solution :

Construct a tool for architecture extraction

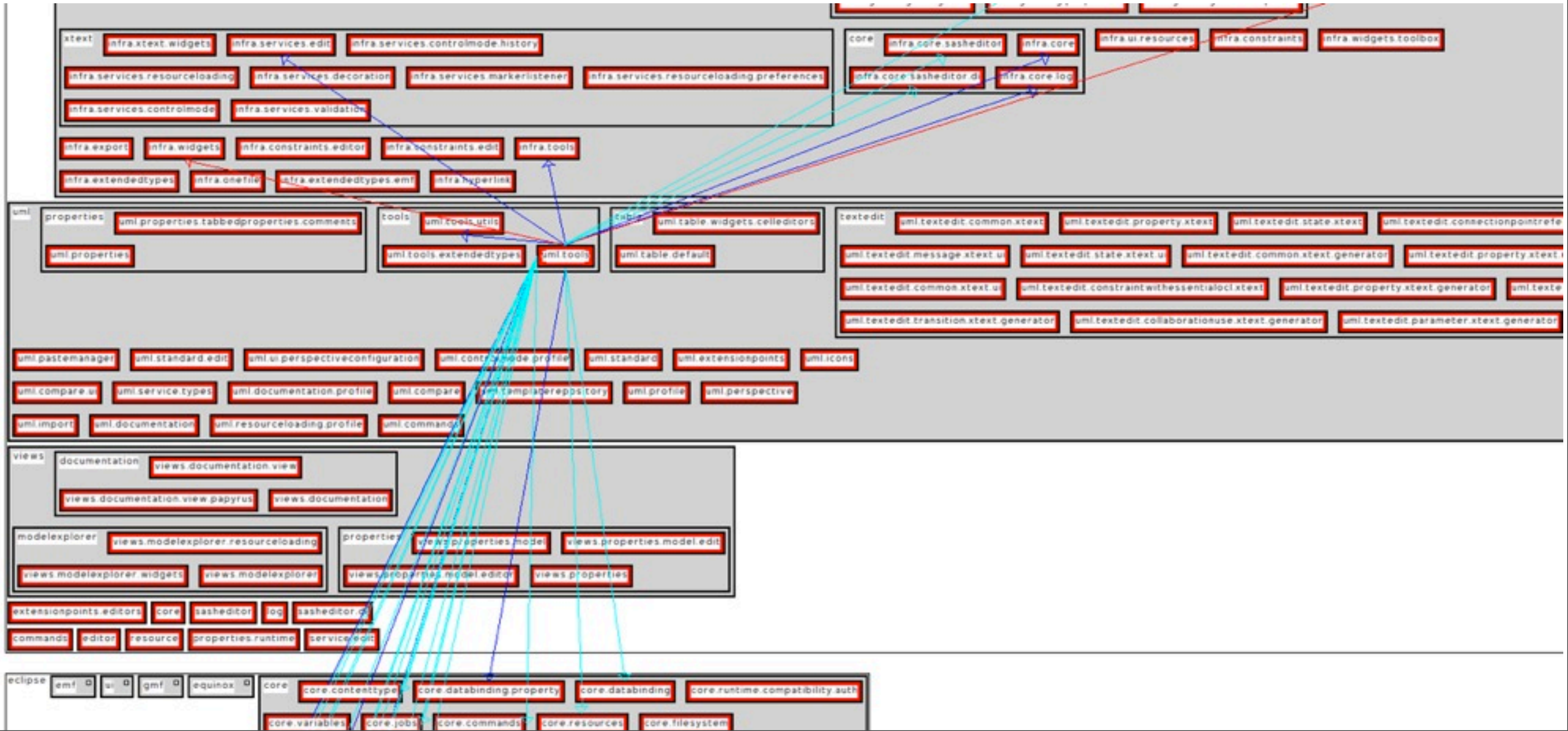
(6-8 weeks)

Construct a rule checker

Architecture Extraction

Papyrus UML: 800 java packages

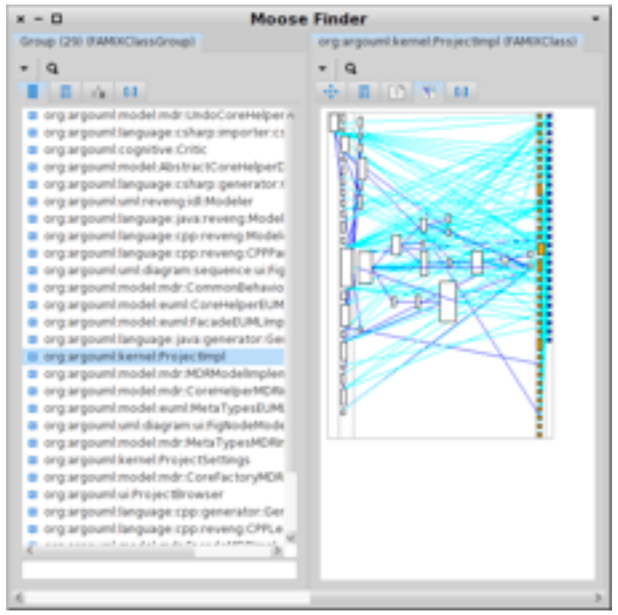
Identification of architecture and layers



classes select: #isGod

McCabe = 21

LOC = 753,000

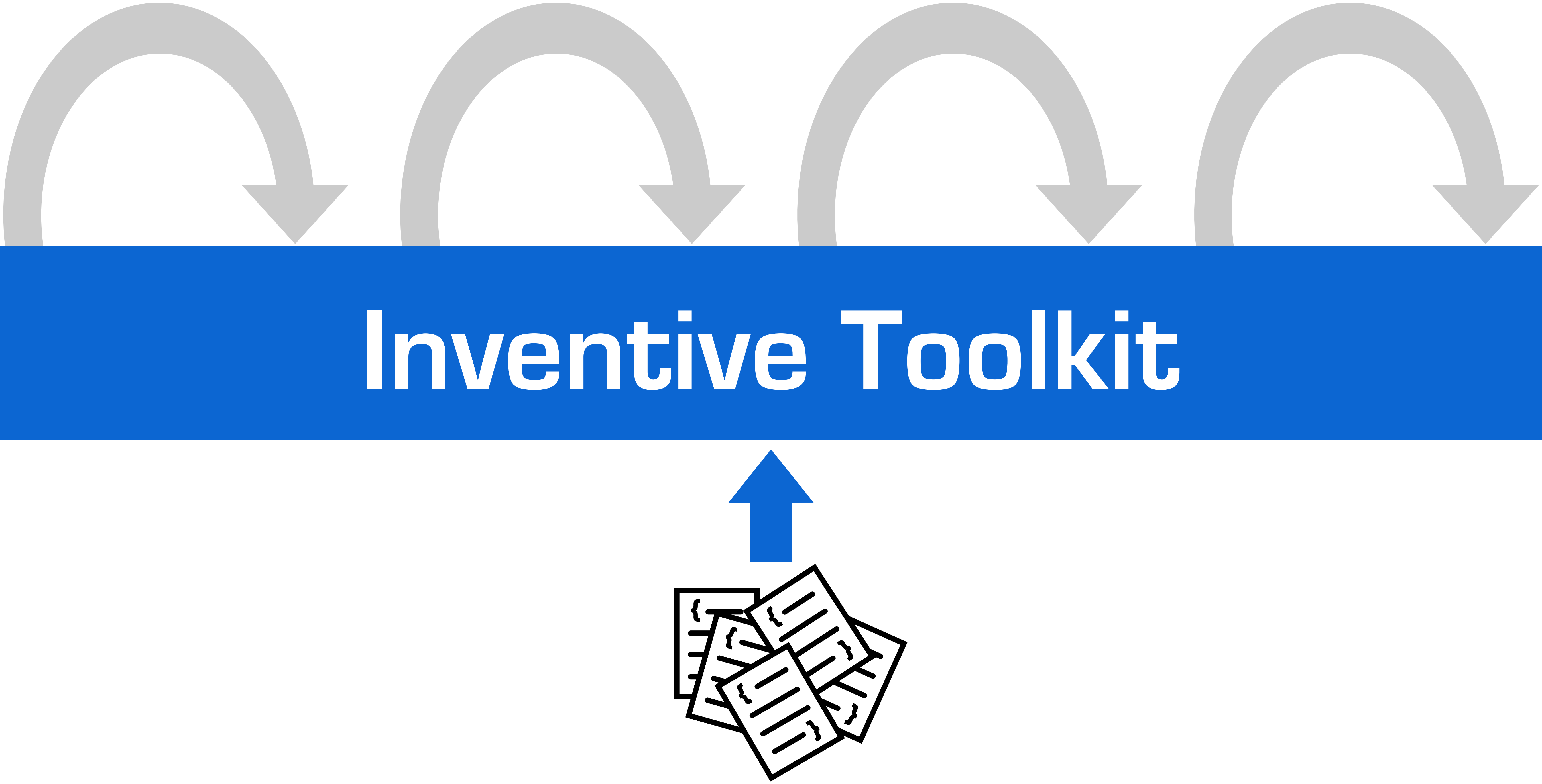

Moose Finder

Group (29) (FAMIXClassGroup)

org::argouml::kernel::ProjectImpl (FAMIXClass)

- org::argouml::model::mdr::UndoCoreHelper
- org::argouml::language::csharp::importer::
- org::argouml::cognitive::Critic
- org::argouml::model::AbstractCoreHelper
- org::argouml::language::csharp::generator::
- org::argouml::uml::reveng::idl::Modeler
- org::argouml::language::java::reveng::Model
- org::argouml::language::cpp::reveng::Model
- org::argouml::language::cpp::reveng::CPPFa
- org::argouml::uml::diagram::sequence::ui::Fig
- org::argouml::model::mdr::CommonBehavio
- org::argouml::model::euml::CoreHelperEUM
- org::argouml::model::euml::FacadeEUMLImp
- org::argouml::language::java::generator::Ge
- org::argouml::kernel::ProjectImpl
- org::argouml::model::mdr::MDRModelImplen
- org::argouml::model::mdr::CoreHelperMDR
- org::argouml::model::euml::MetaTypesEUML
- org::argouml::uml::diagram::ui::FigNodeMode
- org::argouml::model::mdr::MetaTypesMDRI
- org::argouml::kernel::ProjectSettings
- org::argouml::model::mdr::CoreFactoryMDR
- org::argouml::ui::ProjectBrowser
- org::argouml::language::cpp::generator::Ger
- org::argouml::language::cpp::reveng::CPPLe

...



Inventive Toolkit

Software Metrics (best of)

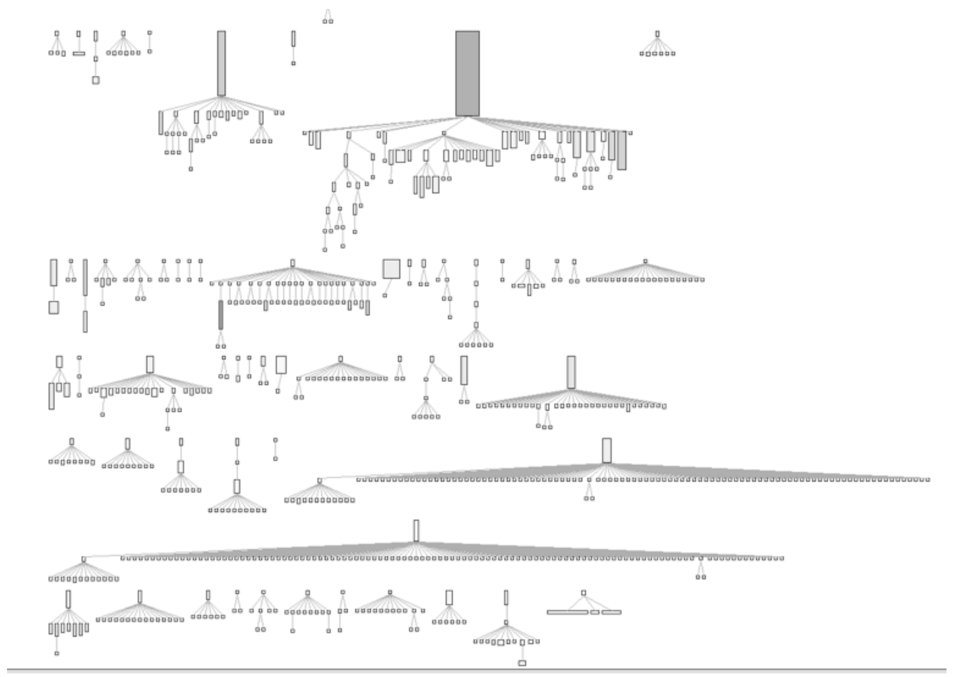
Quality Models

ISO 9126, Squale (PSA-AirFrance)

Rapid Adaptation

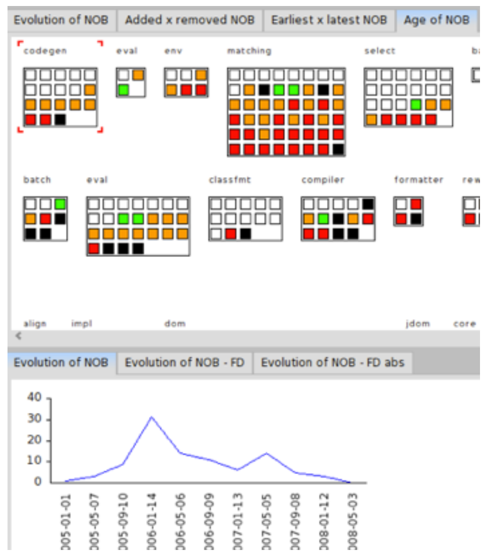
Specific to your business

Dedicated Visualizations for *Software Business Intelligence*



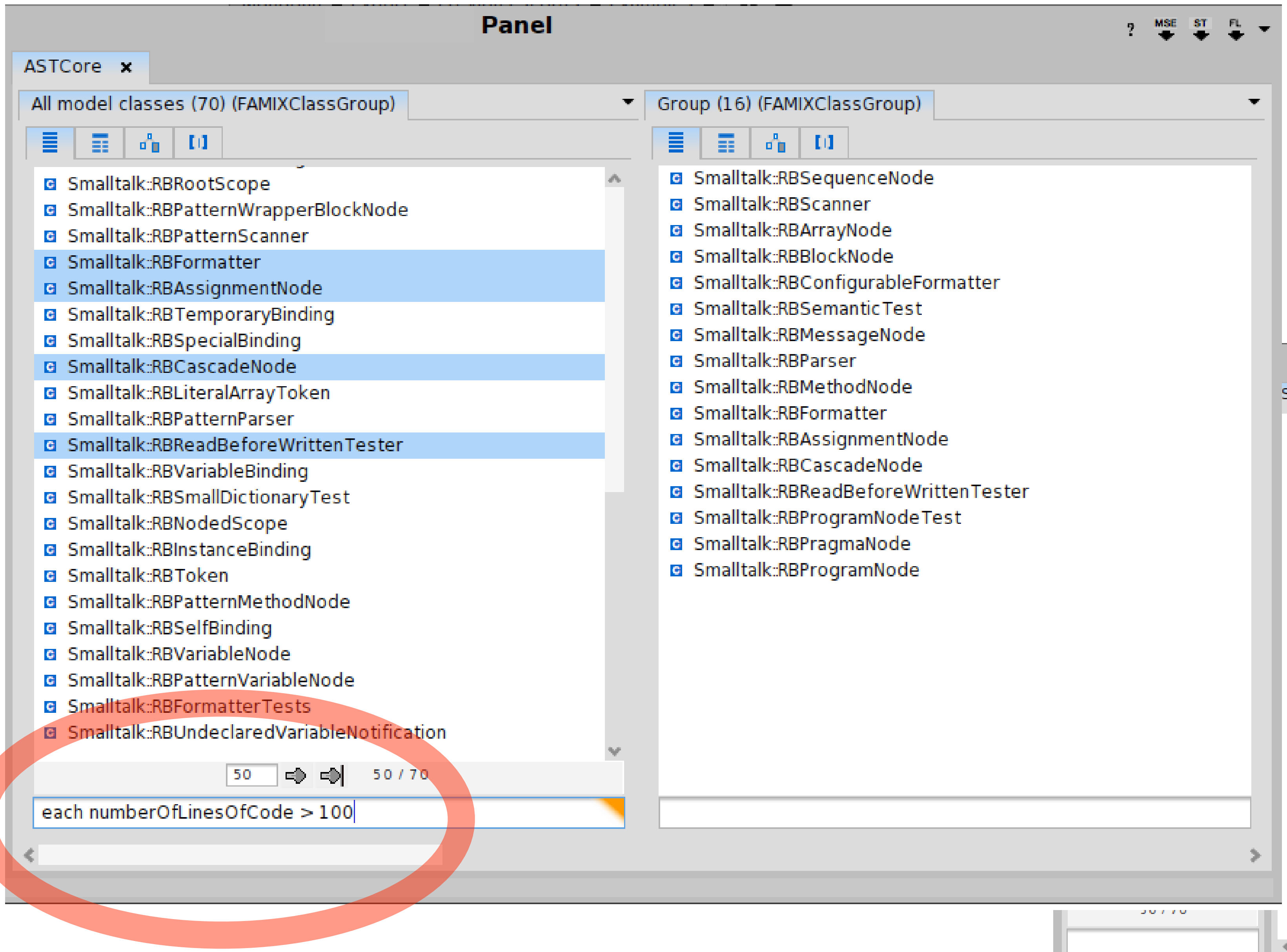
System Complexity

Distribution Map



...

Queries for a Contextual Feedback



Panel

ASTCore

All model classes (70) (FAMIXClassGroup)

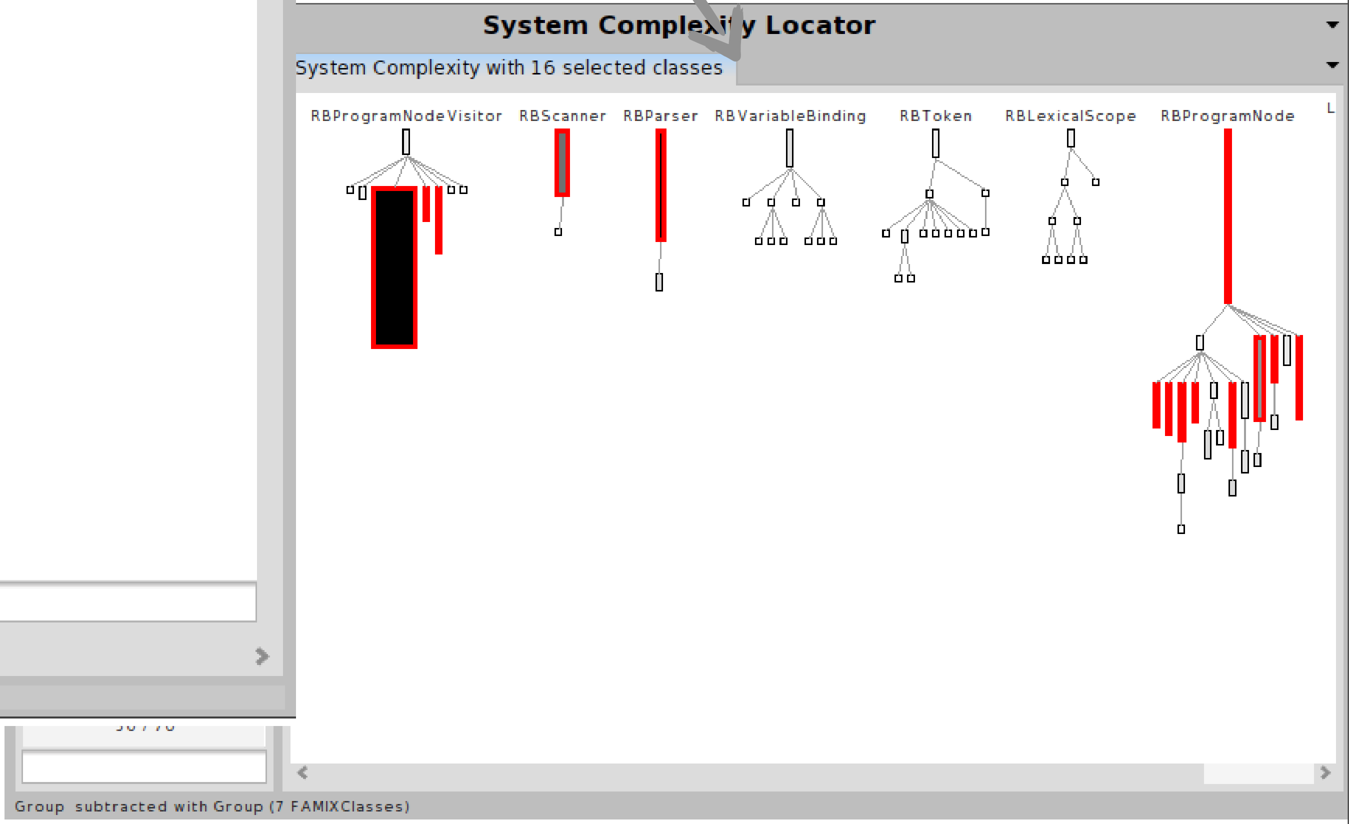
- Smalltalk::RBRootScope
- Smalltalk::RBPatternWrapperBlockNode
- Smalltalk::RBPatternScanner
- Smalltalk::RBFormatter
- Smalltalk::RBAssignmentNode
- Smalltalk::RBTemporaryBinding
- Smalltalk::RBSpecialBinding
- Smalltalk::RBCascadeNode
- Smalltalk::RBLiteralArrayToken
- Smalltalk::RBPatternParser
- Smalltalk::RBReadBeforeWrittenTester
- Smalltalk::RBVariableBinding
- Smalltalk::RBSmallDictionaryTest
- Smalltalk::RBNodedScope
- Smalltalk::RBInstanceBinding
- Smalltalk::RBToken
- Smalltalk::RBPatternMethodNode
- Smalltalk::RBSelfBinding
- Smalltalk::RBVariableNode
- Smalltalk::RBPatternVariableNode
- Smalltalk::RBFormatterTests
- Smalltalk::RBUndeclaredVariableNotification

50 50 / 70

```
each numberOfLinesOfCode > 100
```

Group (16) (FAMIXClassGroup)

- Smalltalk::RBSequenceNode
- Smalltalk::RBScanner
- Smalltalk::RBArrayNode
- Smalltalk::RBBlockNode
- Smalltalk::RBConfigurableFormatter
- Smalltalk::RBSemanticTest
- Smalltalk::RBMessageNode
- Smalltalk::RBParser
- Smalltalk::RBMethodNode
- Smalltalk::RBFormatter
- Smalltalk::RBAssignmentNode
- Smalltalk::RBCascadeNode
- Smalltalk::RBReadBeforeWrittenTester
- Smalltalk::RBProgramNodeTest
- Smalltalk::RBPragmaNode
- Smalltalk::RBProgramNode



System Complexity Locator

System Complexity with 16 selected classes

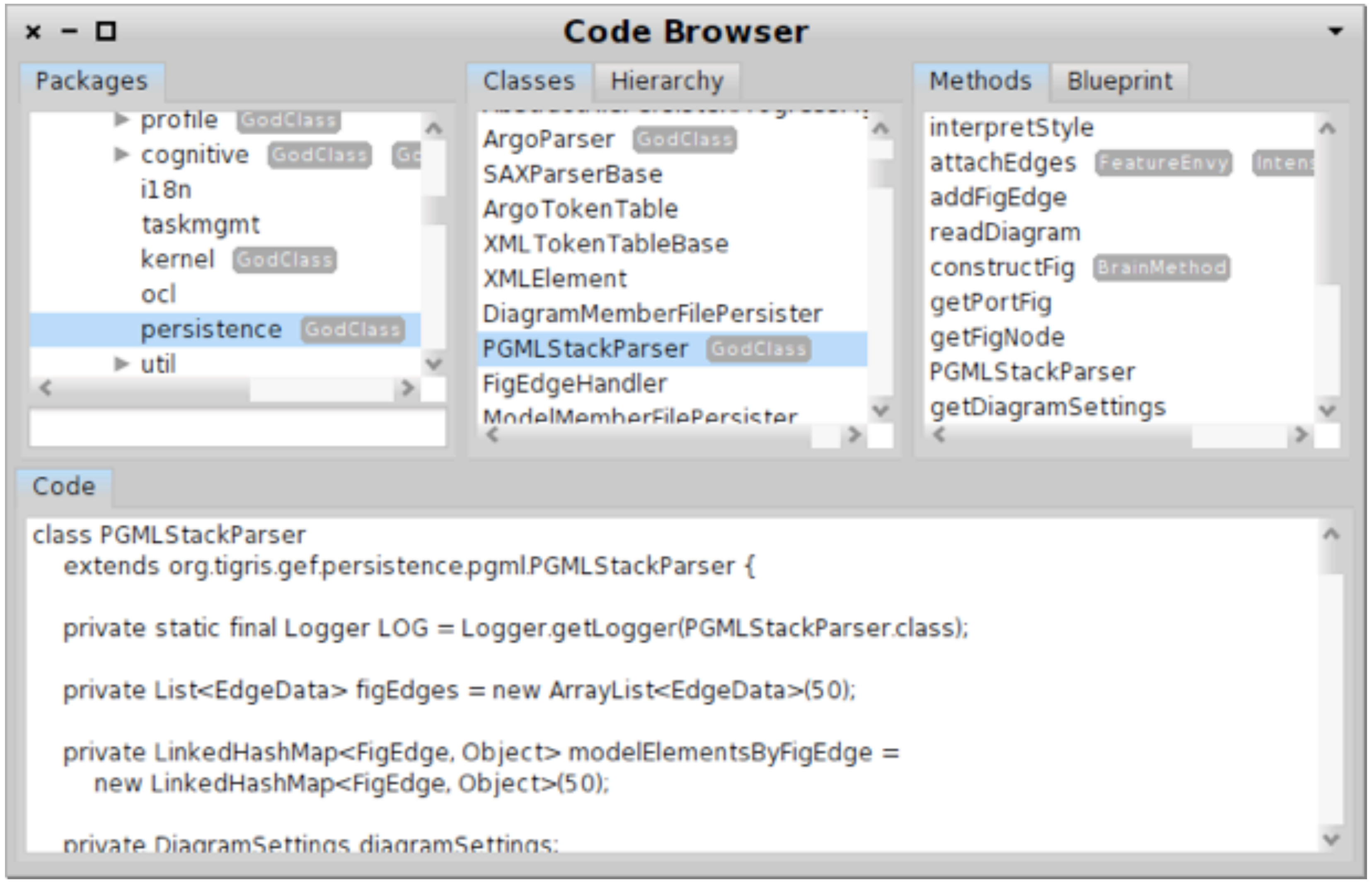
Group subtracted with Group (7 FAMIXClasses)

Dedicated Tools

Rich

Compact

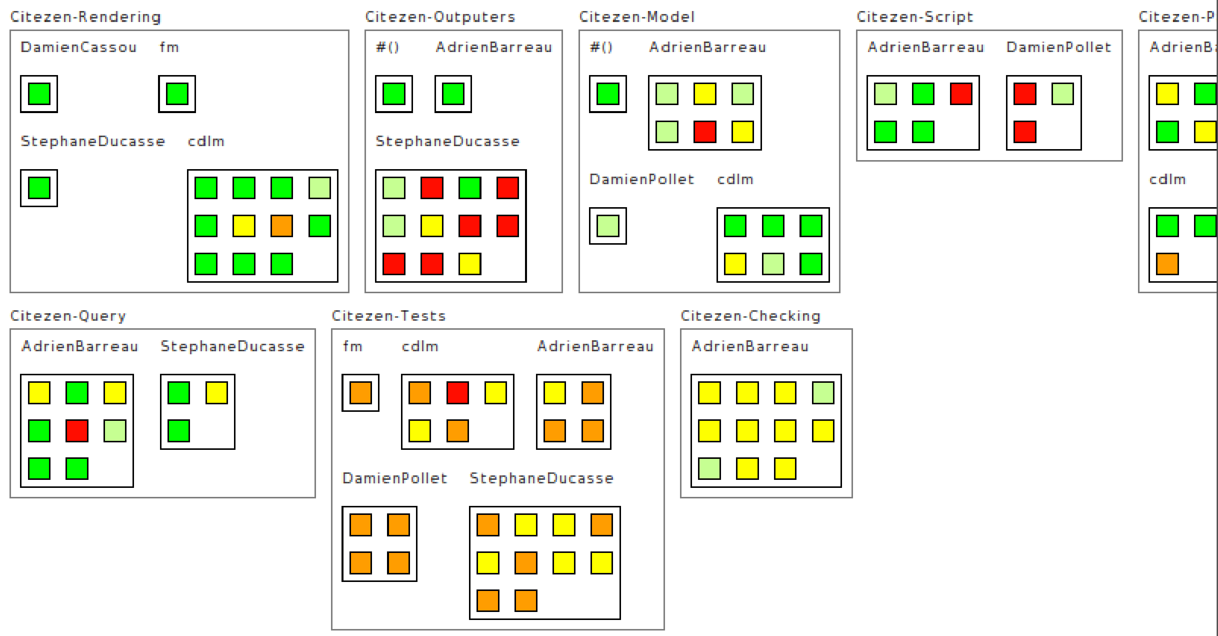
Best Focus



```
class PGMLStackParser
    extends org.tigris.gef.persistence.pgml.PGMLStackParser {

    private static final Logger LOG = Logger.getLogger(PGMLStackParser.class);

    private List<EdgeData> figEdges = new ArrayList<EdgeData>(50);

    private LinkedHashMap<FigEdge, Object> modelElementsByFigEdge =
        new LinkedHashMap<FigEdge, Object>(50);

    private DiagramSettings diagramSettings;
```

Data Aggregation / Bridges between tools

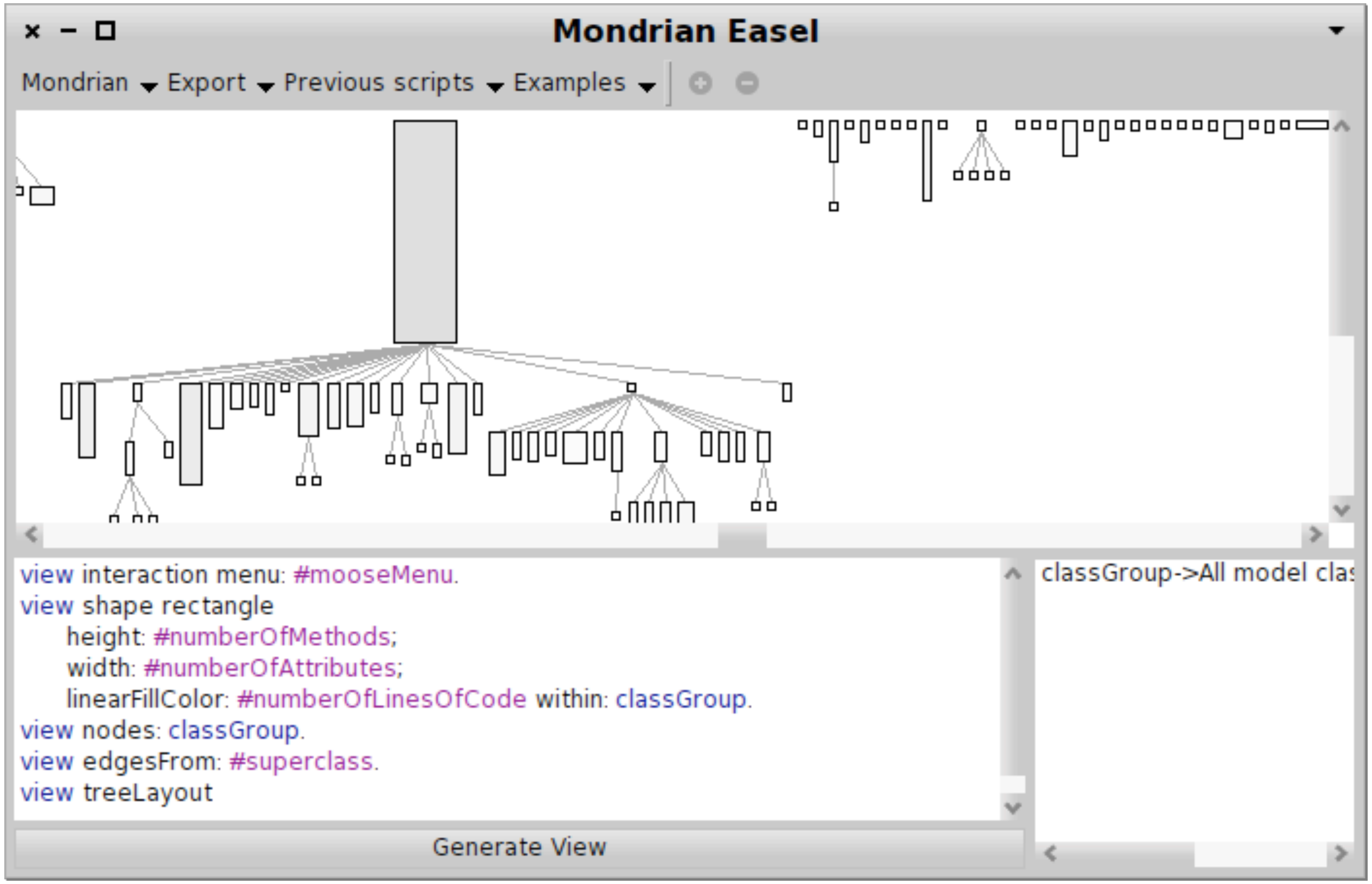
Combinator Parsers
Modular



Example :
Correlate bugs and test coverage

Dedicated Analyses

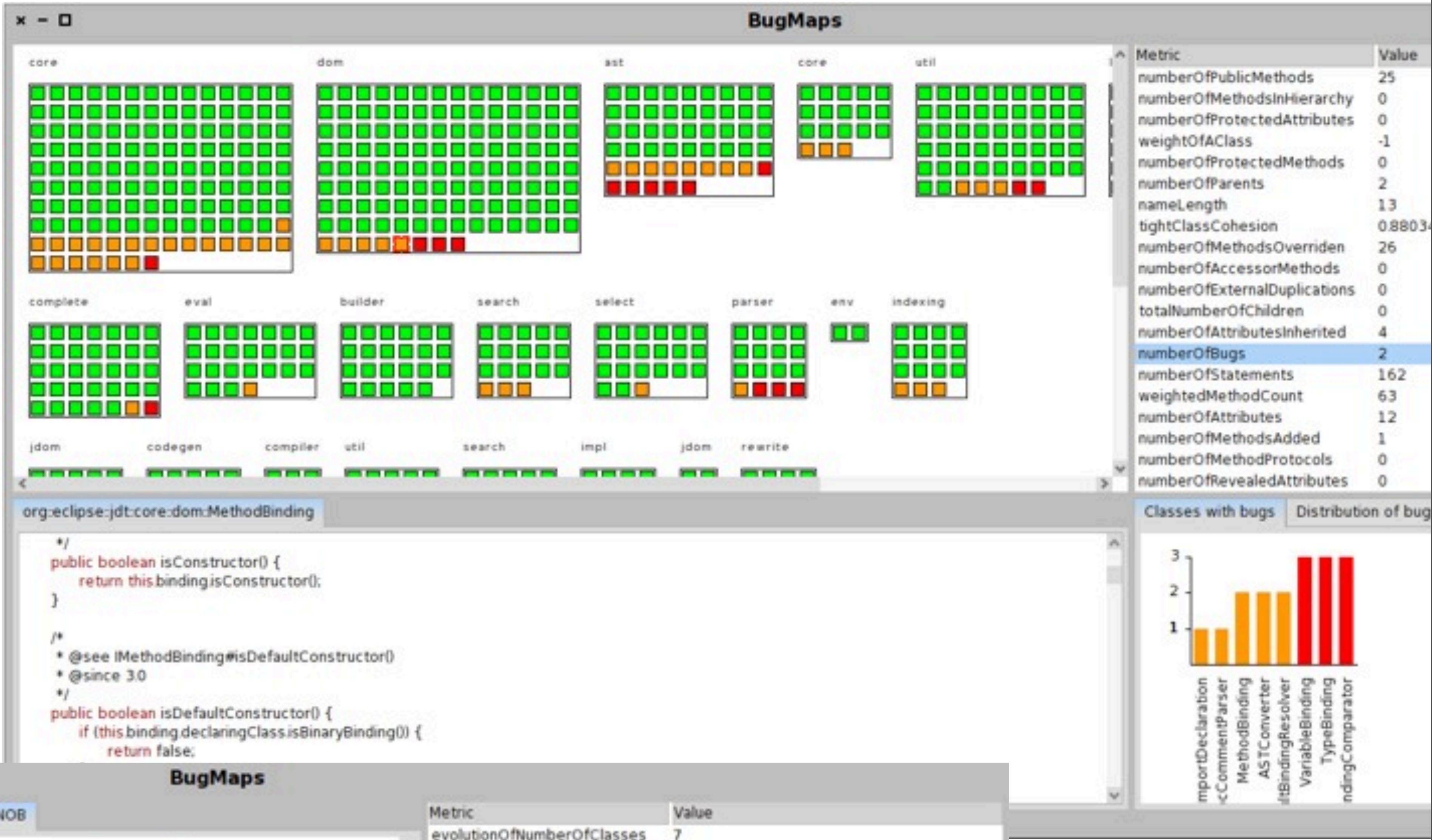
...



```
view interaction menu: #mooseMenu.
view shape rectangle
    height: #numberOfMethods;
    width: #numberOfAttributes;
    linearFillColor: #numberOfLinesOfCode within: classGroup.
view nodes: classGroup.
view edgesFrom: #superclass.
view treeLayout
```

classGroup->All model clas

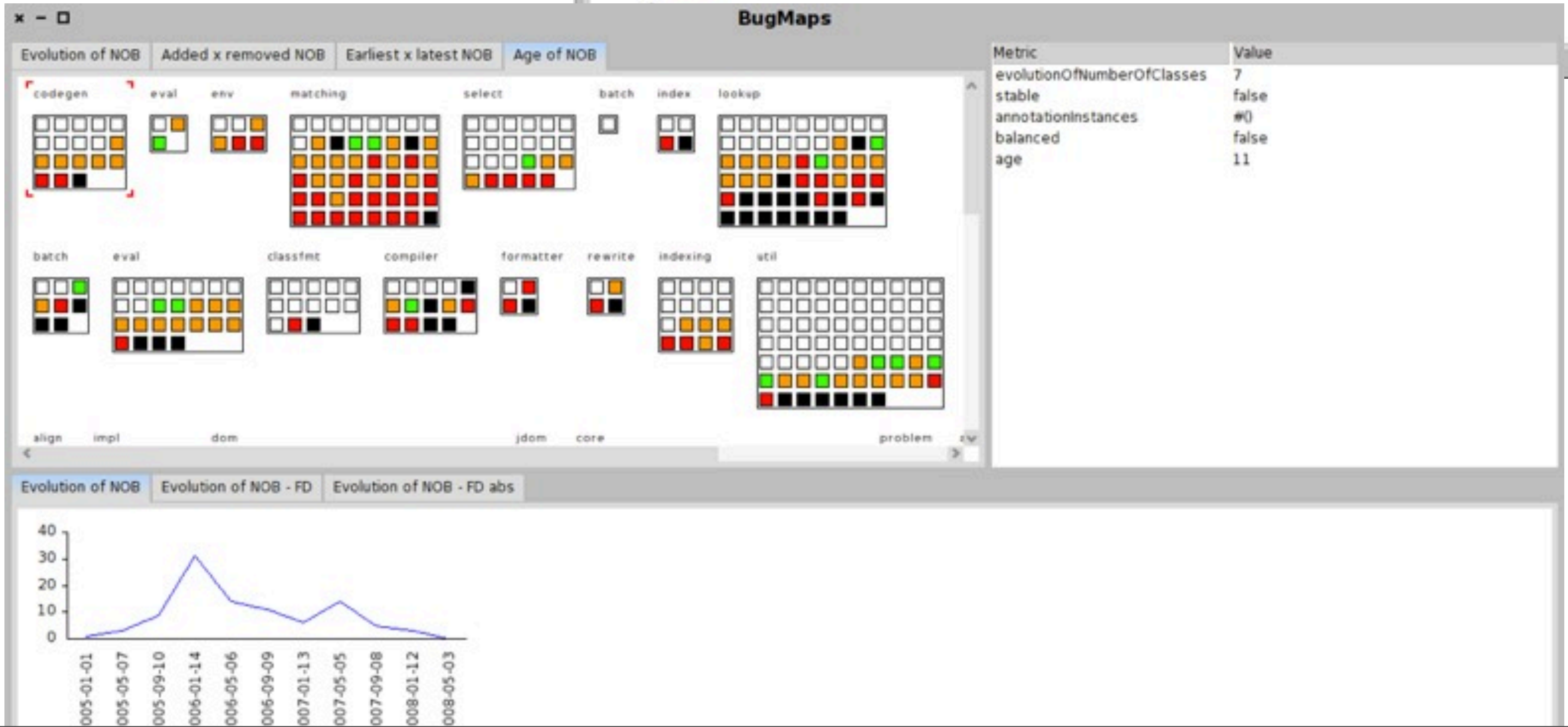
Analysis of bugs



org:eclipse:jdt:core:dom:MethodBinding

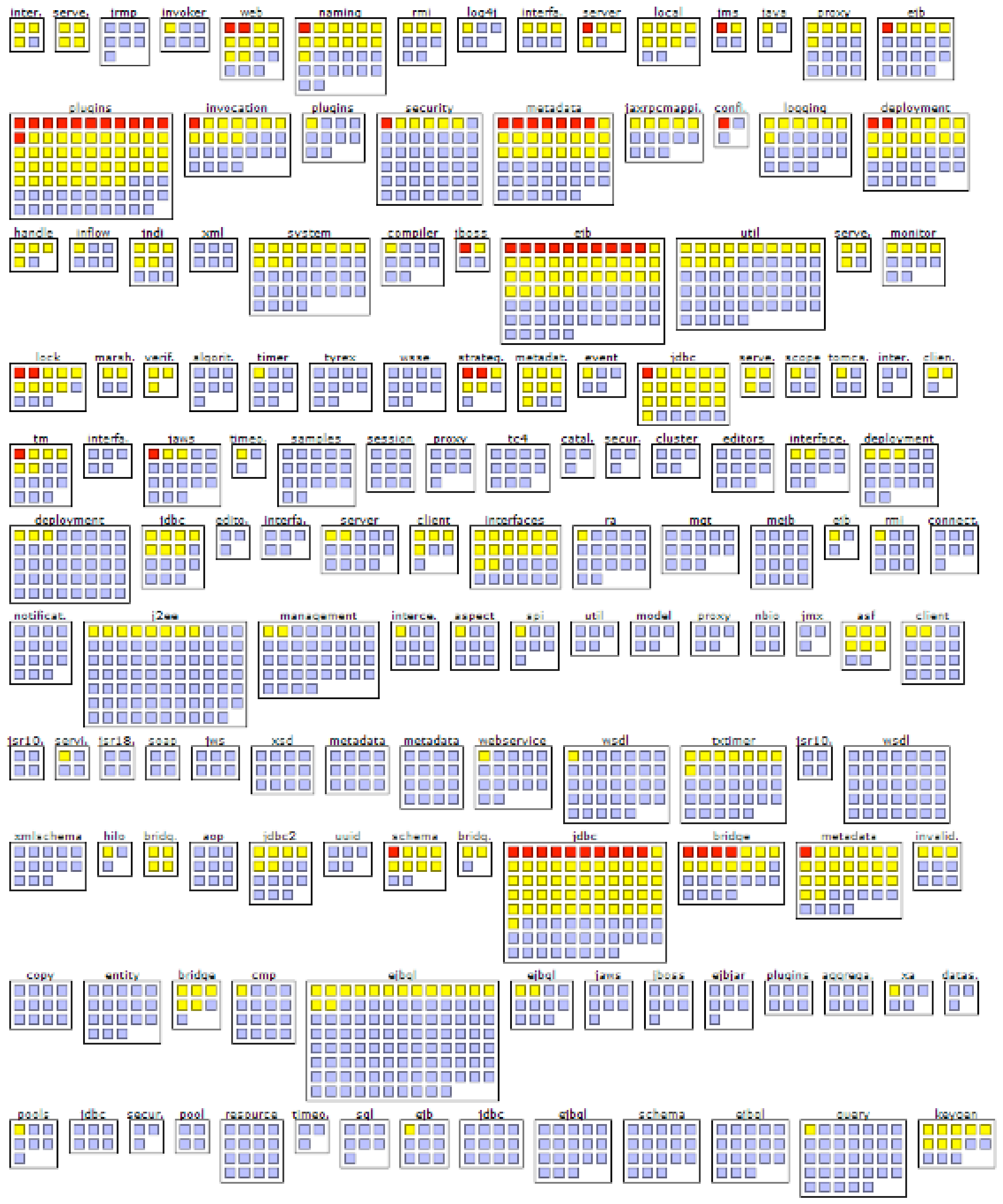
```
 */
public boolean isConstructor() {
    return this.binding.isConstructor();
}

/*
 * @see IMethodBinding#isDefaultConstructor()
 * @since 3.0
 */
public boolean isDefaultConstructor() {
    if (this.binding.declaringClass.isBinaryBinding()) {
        return false;
```

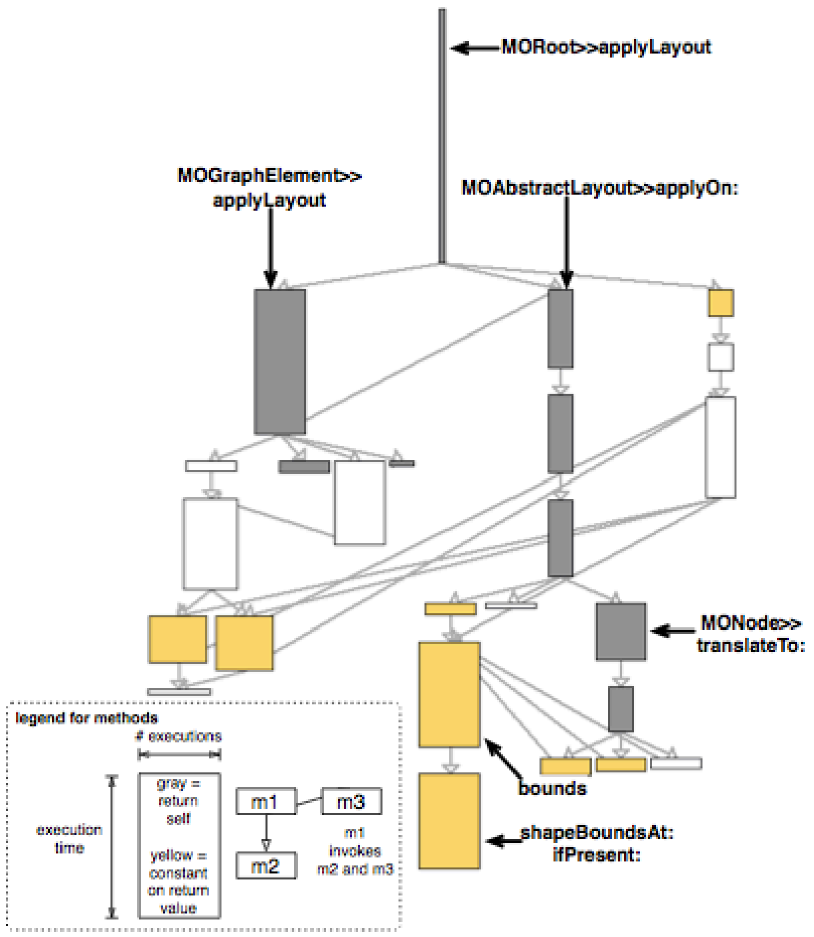
Metric	Value
numberOfPublicMethods	25
numberOfMethodsInHierarchy	0
numberOfProtectedAttributes	0
weightOfAClass	-1
numberOfProtectedMethods	0
numberOfParents	2
nameLength	13
tightClassCohesion	0.88034
numberOfMethodsOverriden	26
numberOfAccessorMethods	0
numberOfExternalDuplications	0
totalNumberOfChildren	0
numberOfAttributesInherited	4
numberOfBugs	2
numberOfStatements	162
weightedMethodCount	63
numberOfAttributes	12
numberOfMethodsAdded	1
numberOfMethodProtocols	0
numberOfRevealedAttributes	0



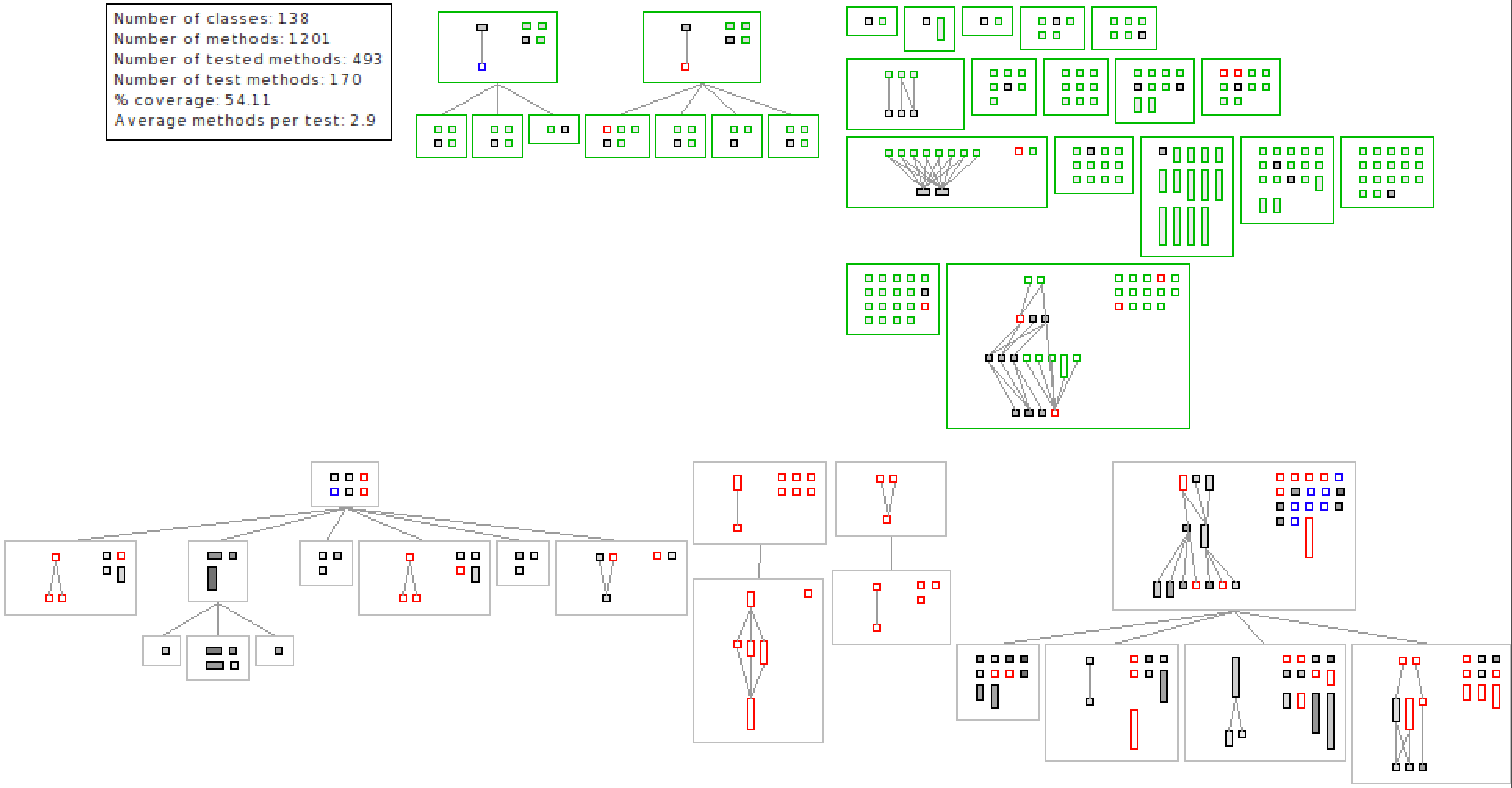
Metric	Value
evolutionOfNumberOfClasses	7
stable	false
annotationInstances	#()
balanced	false
age	11

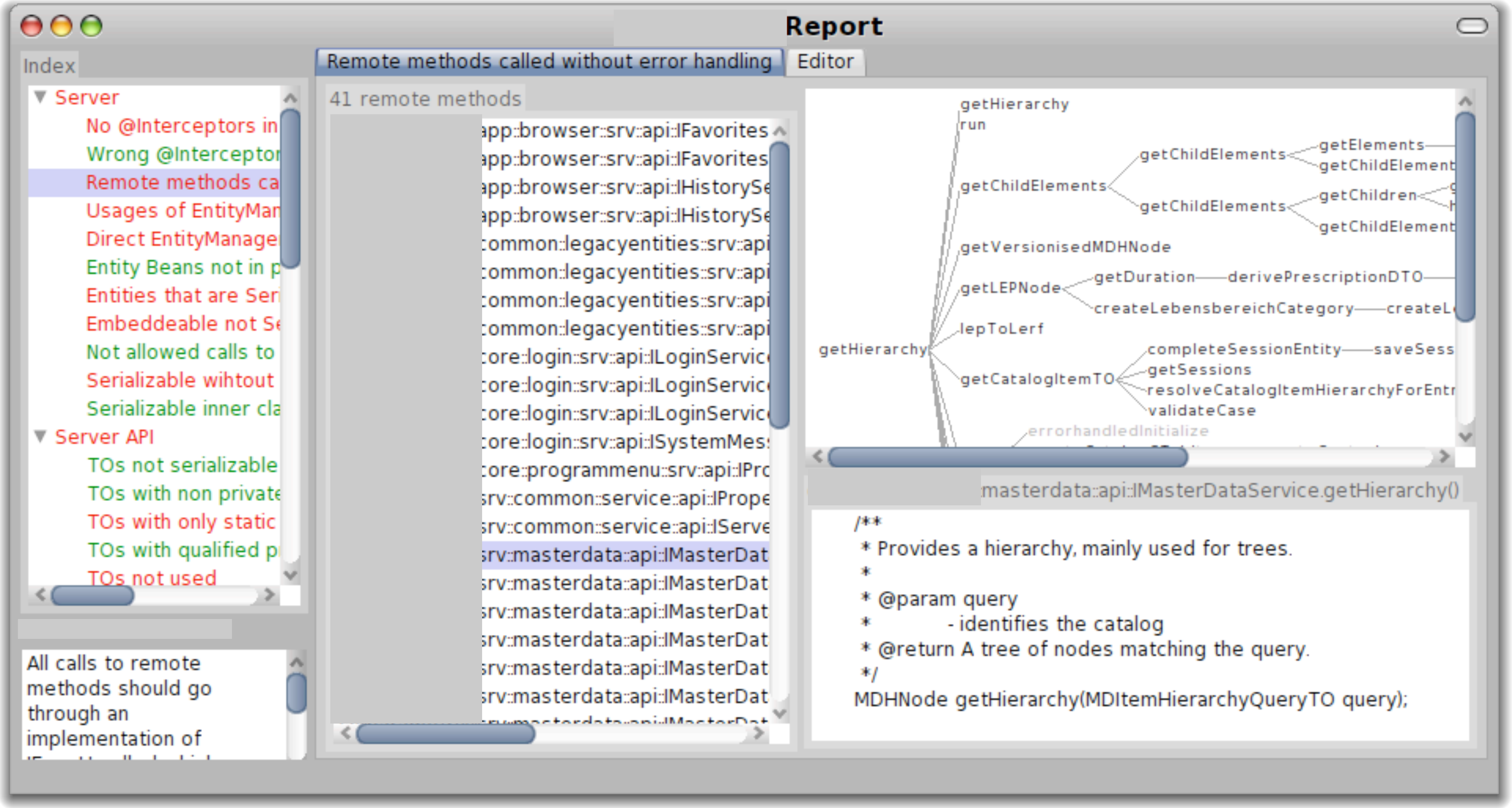
Logs and Performance





Putting in Perspective Test Coverage


Number of classes: 138
Number of methods: 1201
Number of tested methods: 493
Number of test methods: 170
% coverage: 54.11
Average methods per test: 2.9



- Server
  - No @Interceptors in
  - Wrong @Interceptor
  - Remote methods ca
  - Usages of EntityMan
  - Direct EntityManage
  - Entity Beans not in p
  - Entities that are Ser
  - Embeddeable not Se
  - Not allowed calls to
  - Serializable wihtout
  - Serializable inner cla
- Server API
  - TOs not serializable
  - TOs with non private
  - TOs with only static
  - TOs with qualified p
  - TOs not used

All calls to remote methods should go through an implementation of

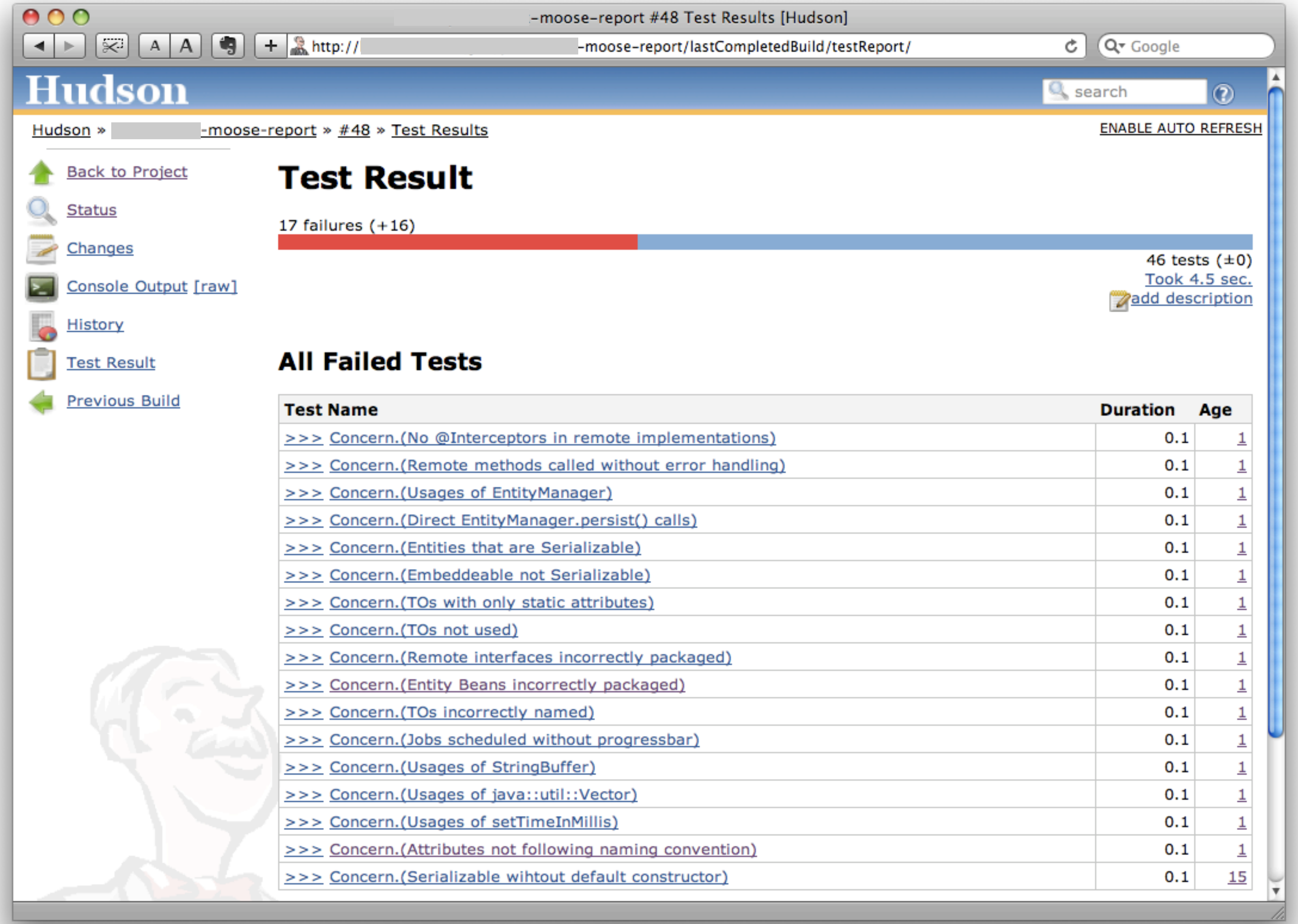
```
app::browser::srv::api::IFavorites
app::browser::srv::api::IFavorites
app::browser::srv::api::IHistorySe
app::browser::srv::api::IHistorySe
common::legacyentities::srv::api
common::legacyentities::srv::api
common::legacyentities::srv::api
common::legacyentities::srv::api
core::login::srv::api::ILoginServic
core::login::srv::api::ILoginServic
core::login::srv::api::ILoginServic
core::login::srv::api::ISystemMes
core::programmenu::srv::api::IPro
srv::common::service::api::IPrope
srv::common::service::api::IServe
srv::masterdata::api::IMasterDat
srv::masterdata::api::IMasterDat
srv::masterdata::api::IMasterDat
srv::masterdata::api::IMasterDat
srv::masterdata::api::IMasterDat
srv::masterdata::api::IMasterDat
```

masterdata::api::IMasterDataService.getHierarchy()

```
/**
 * Provides a hierarchy, mainly used for trees.
 *
 * @param query
 *            - identifies the catalog
 * @return A tree of nodes matching the query.
 */
MDHNode getHierarchy(MDItemHierarchyQueryTO query);
```

A dedicated report

Continuous Integration



[redacted]-moose-report #48 Test Results [Hudson]

http://[redacted]-moose-report/lastCompletedBuild/testReport/

Hudson

Hudson » [redacted]-moose-report » #48 » Test Results

ENABLE AUTO REFRESH

- Back to Project
- Status
- Changes
- Console Output [raw]
- History
- Test Result
- Previous Build

Test Result

17 failures (+16)

46 tests (±0)
Took 4.5 sec.
add description

All Failed Tests

Test Name	Duration	Age
>>> Concern.(No @Interceptors in remote implementations)	0.1	1
>>> Concern.(Remote methods called without error handling)	0.1	1
>>> Concern.(Usages of EntityManager)	0.1	1
>>> Concern.(Direct EntityManager.persist() calls)	0.1	1
>>> Concern.(Entities that are Serializable)	0.1	1
>>> Concern.(Embeddeable not Serializable)	0.1	1
>>> Concern.(TOs with only static attributes)	0.1	1
>>> Concern.(TOs not used)	0.1	1
>>> Concern.(Remote interfaces incorrectly packaged)	0.1	1
>>> Concern.(Entity Beans incorrectly packaged)	0.1	1
>>> Concern.(TOs incorrectly named)	0.1	1
>>> Concern.(Jobs scheduled without progressbar)	0.1	1
>>> Concern.(Usages of StringBuffer)	0.1	1
>>> Concern.(Usages of java::util::Vector)	0.1	1
>>> Concern.(Usages of setTimeInMillis)	0.1	1
>>> Concern.(Attributes not following naming convention)	0.1	1
>>> Concern.(Serializable wihtout default constructor)	0.1	15

Interested by your problems

- Migration support
- Decision making support
- Extraction and Definition of Rules
- Software Architecture Verification
- Visualizations
- Cost Prediction
- Impact change
- Service-oriented architecture
- Software Analysis

synectique
Inventive Analysis

Dedicated tools tailored to your problems

Profitable in terms of cost

http://www.synectique.eu