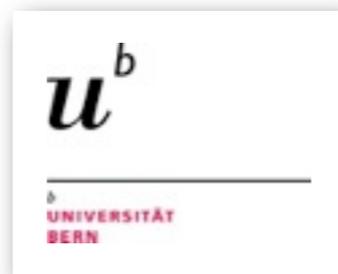


Agile Software Assessment

Oscar Nierstrasz
Software Composition Group
scg.unibe.ch



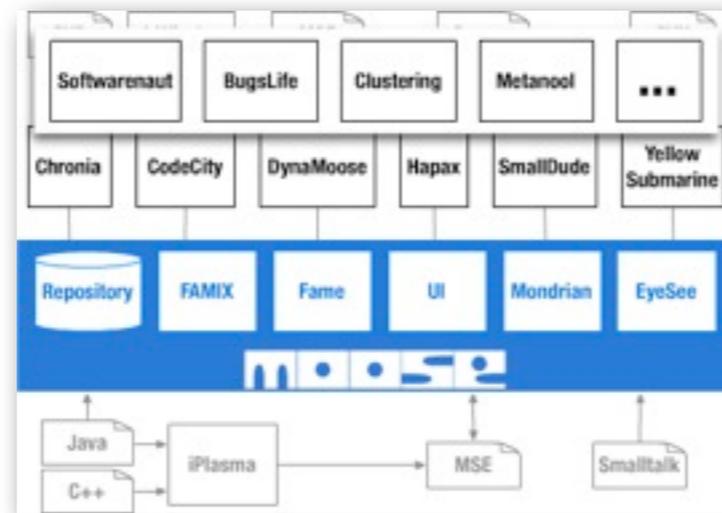
SCG Present and Past



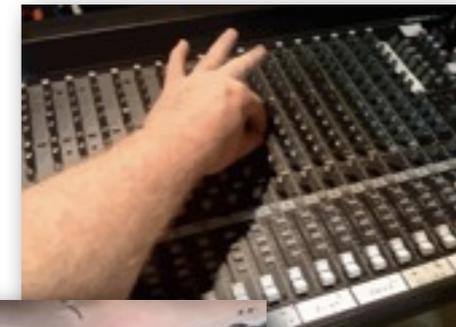
Agile Software Assessment



Motivation



Agility in Moose



Challenges

The need for Agile Software Assessment



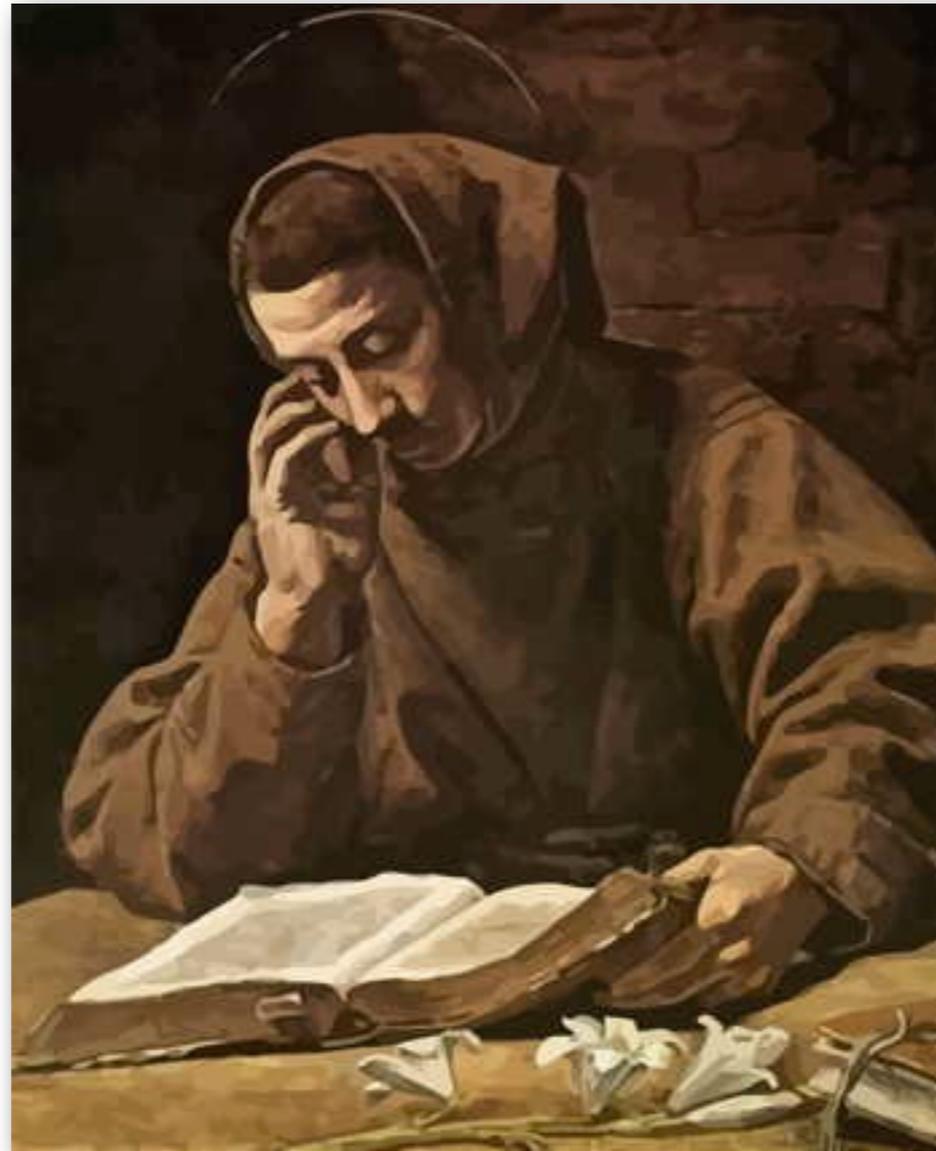
Legacy code is hard to understand



The architecture



... is not in the code



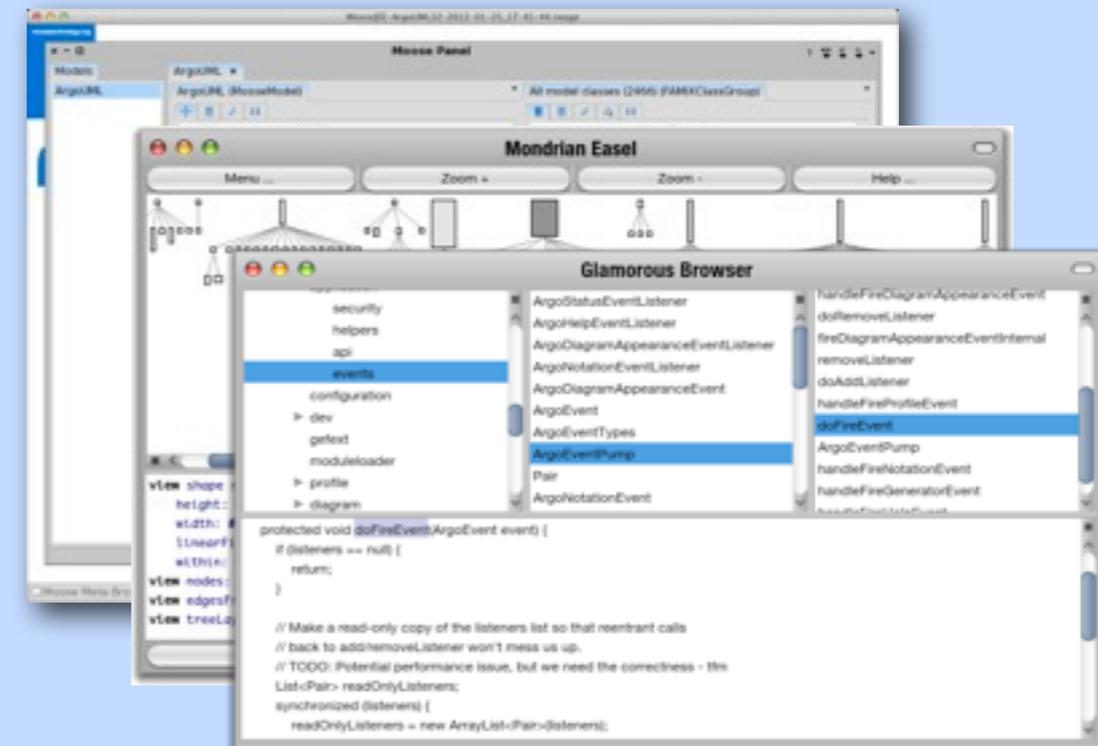
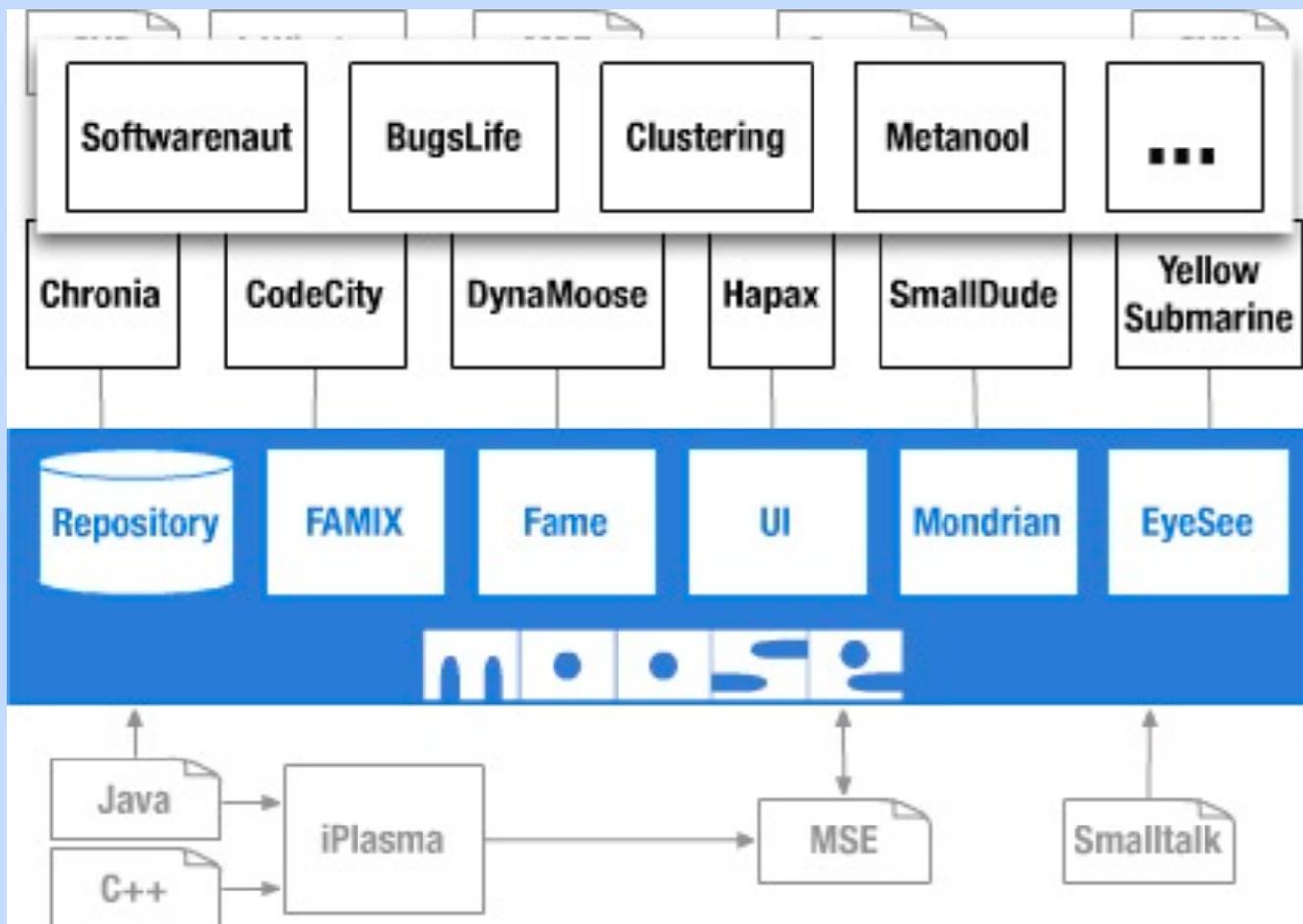
**Developers spend more time
reading than writing code**



**Specialized analyses
require custom tools**



Agility in Moose



Moose is a platform for software and data analysis

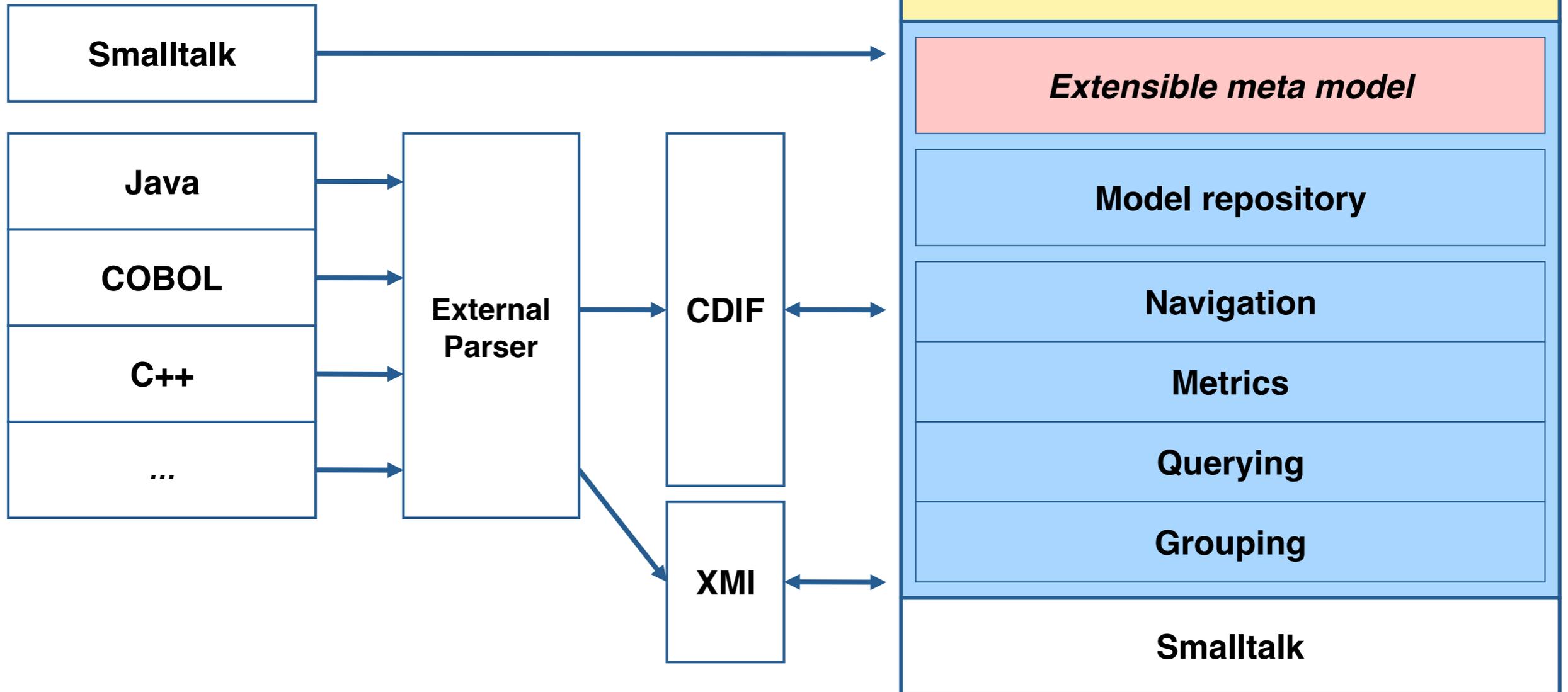
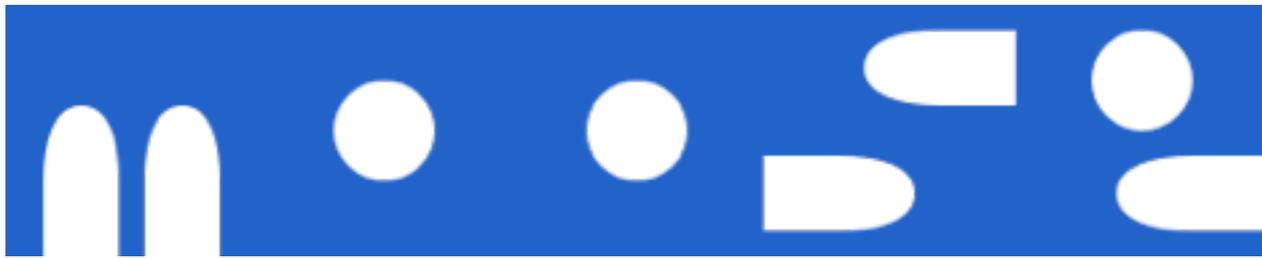
The screenshot displays the Moose Finder application window titled "Moose Finder - igeEnt86-2009-05-25 (MooseModel)". The interface is divided into two main panes. The left pane, titled "igeEnt86-2009-05-25 - MooseModel", contains a list of FAMIX objects with their counts. The right pane, titled "ClassGroup - 1814 items", shows a complex class hierarchy diagram with nodes and connecting lines. The "Complexity" tab is selected in the right pane.

Left Pane: igeEnt86-2009-05-25 - MooseModel

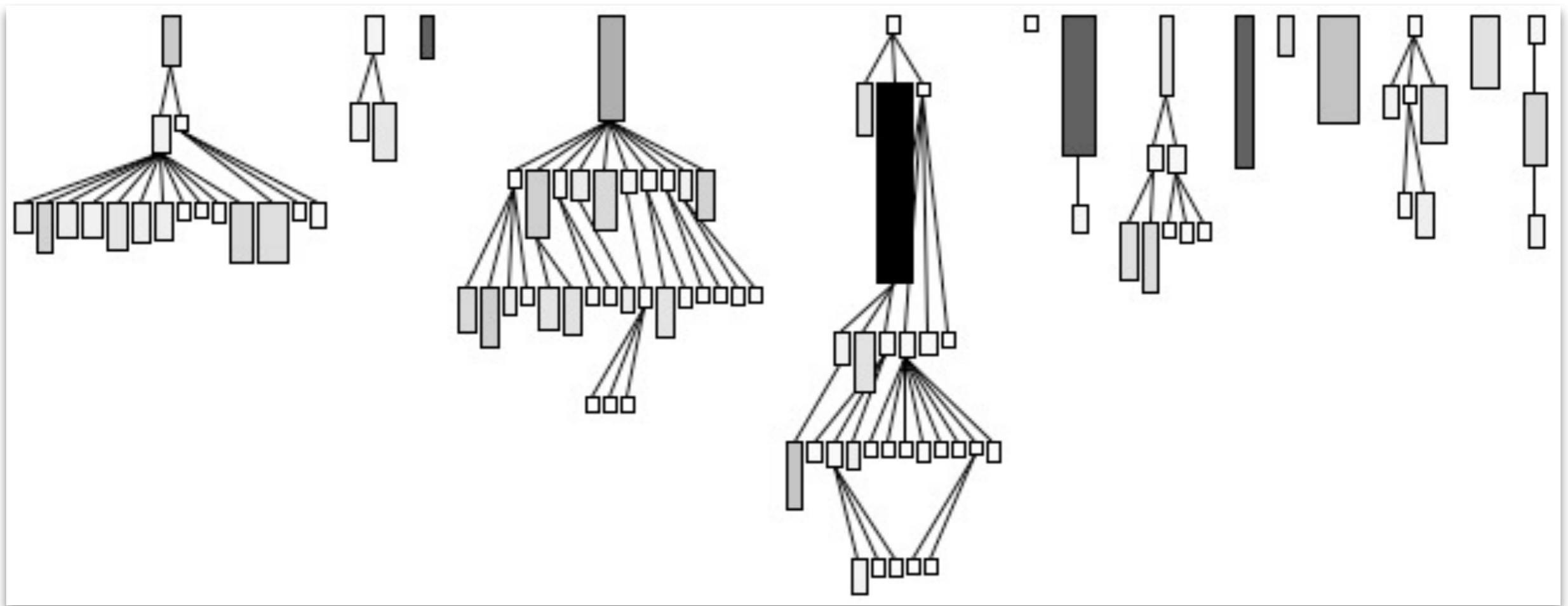
- All famixaccess (32789 FAMIXAccesses)
- All famixannotationinstance (3351 FAMIXAnnotationInstances)
- All famixannotationtype (11 FAMIXAnnotationTypes)
- All famixattribute (7036 FAMIXAttributes)
- All famixcaughtexception (2279 FAMIXCaughtExceptions)
- All famixclass (2447 FAMIXClasses)
- All famixdeclaredexception (5209 FAMIXDeclaredExceptions)
- All famixfunction (2 FAMIXFunctions)
- All famixinheritance (3338 FAMIXInheritances)
- All famixinvocation (35864 FAMIXInvocations)
- All famixlocalvariable (14303 FAMIXLocalVariables)
- All famixmethod (13827 FAMIXMethods)
- All famixnamespace (307 FAMIXNamespaces)
- All famixparameter (11958 FAMIXParameters)
- All famixprimitivetype (9 FAMIXPrimitiveTypes)
- All famixsessionbean (39 FAMIXSessionBeans)
- All famixthrownexception (869 FAMIXThrownExceptions)
- All model classes (1814 FAMIXClasses)
- All model namespaces (238 FAMIXNamespaces)
- Group (515 FAMIXMethods)

Right Pane: ClassGroup - 1814 items

Complexity

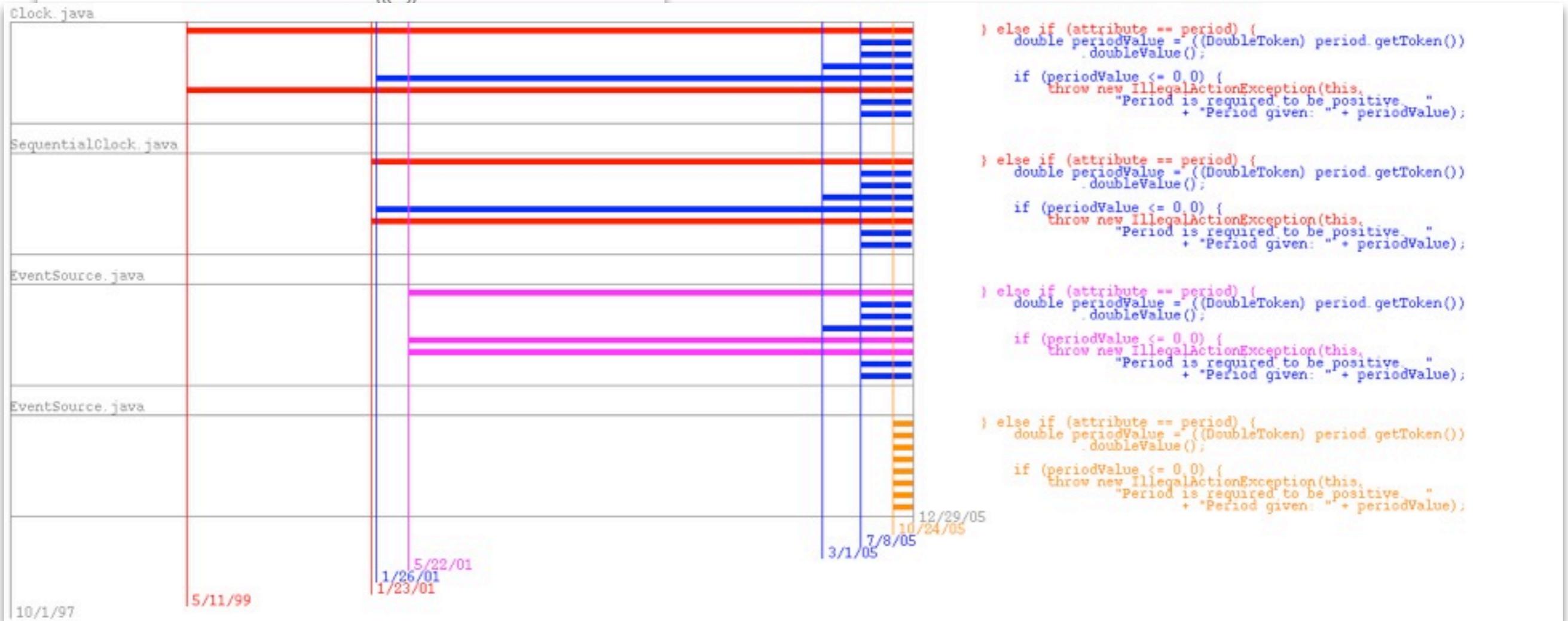
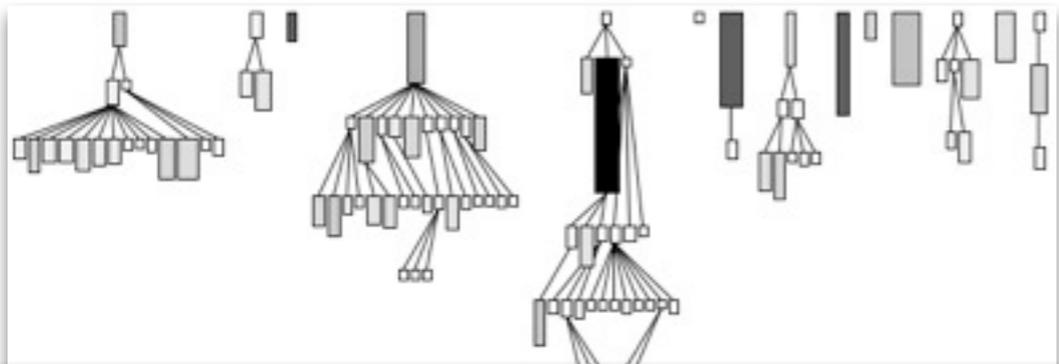


Nierstrasz et al. The Story of Moose. ESEC/FSE 2005

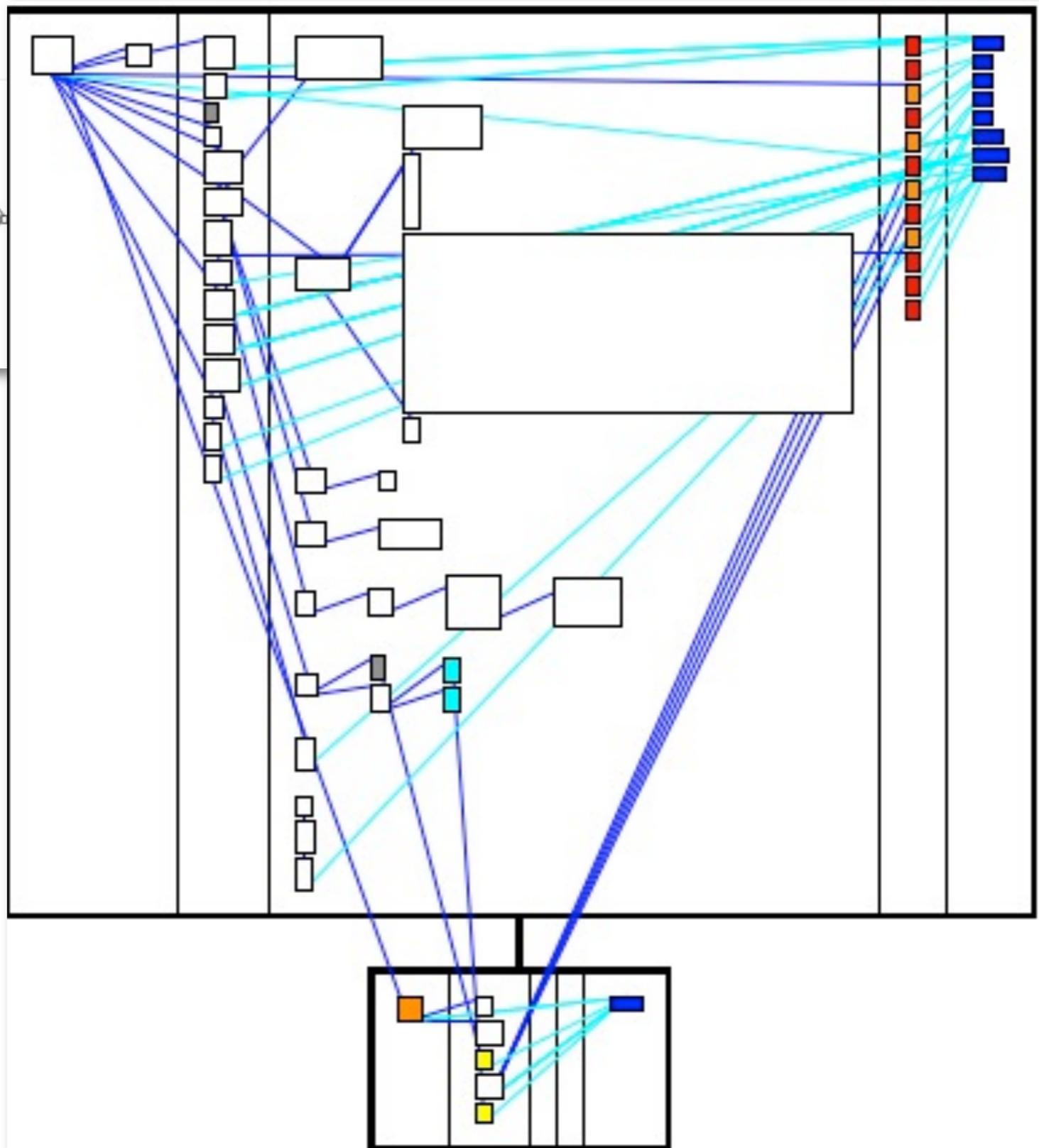
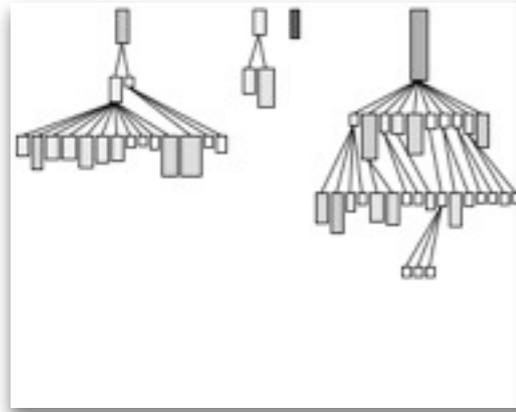


System complexity

Lanza et al. Polymetric Views. TSE 2003



Clone evolution



```

+ edge of (attribute == period) {
  double periodValue = (double) period.getValue();
  double value = 0.0;
  if (periodValue > 0.0) {
    value += (1/periodValue) * period.getValue();
    period is required to be positive ...
  }
}

+ edge of (attribute == period) {
  double periodValue = (double) period.getValue();
  double value = 0.0;
  if (periodValue > 0.0) {
    value += (1/periodValue) * period.getValue();
    period is required to be positive ...
  }
}

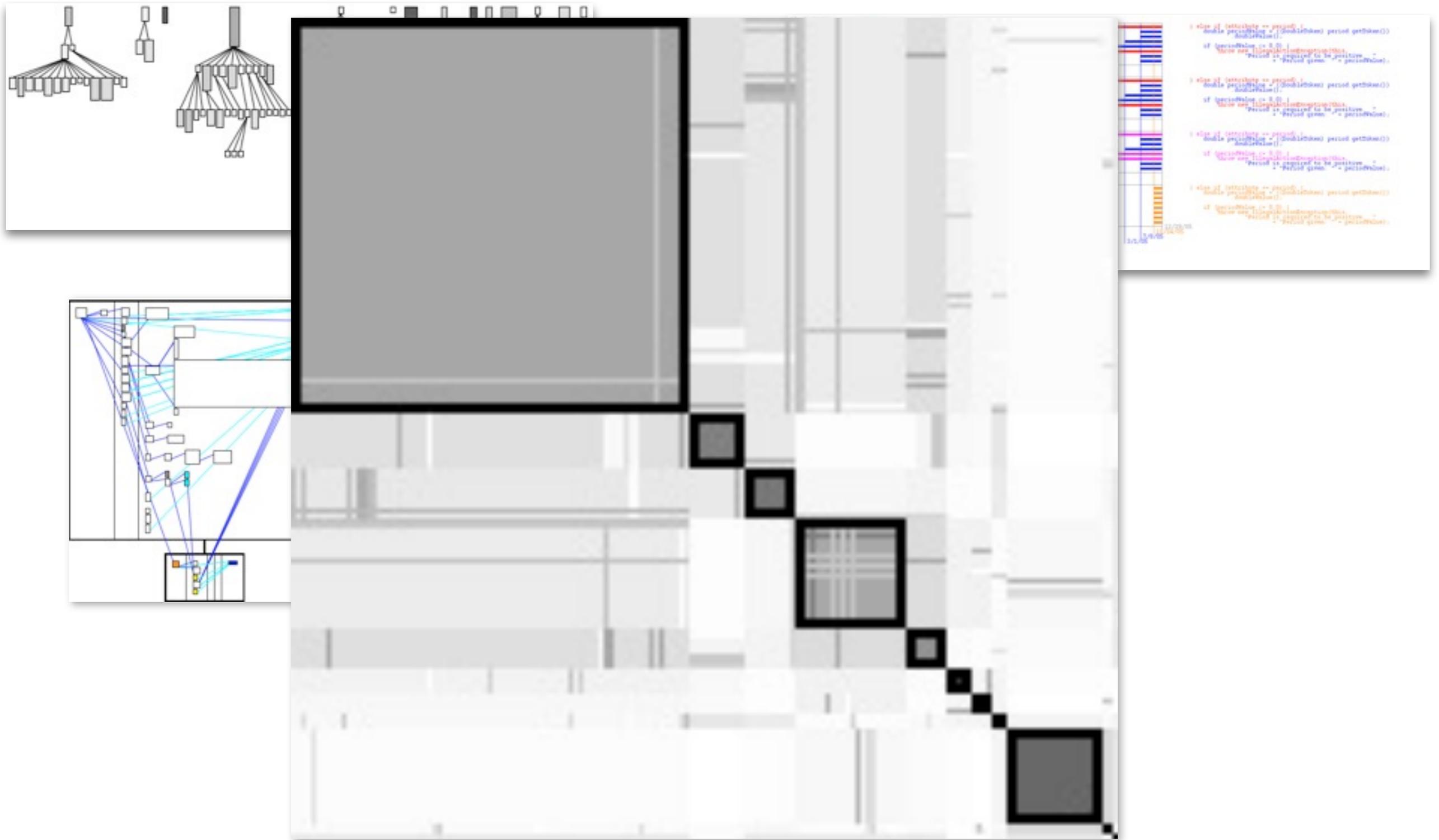
+ edge of (attribute == period) {
  double periodValue = (double) period.getValue();
  double value = 0.0;
  if (periodValue > 0.0) {
    value += (1/periodValue) * period.getValue();
    period is required to be positive ...
  }
}

+ edge of (attribute == period) {
  double periodValue = (double) period.getValue();
  double value = 0.0;
  if (periodValue > 0.0) {
    value += (1/periodValue) * period.getValue();
    period is required to be positive ...
  }
}

```

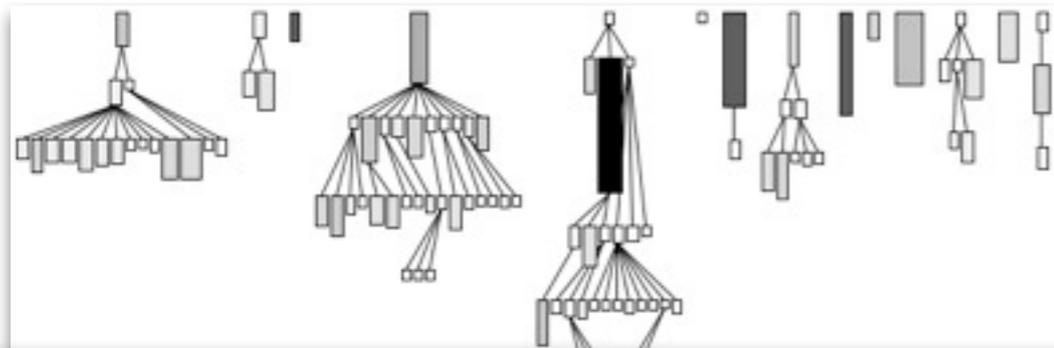
Class blueprint

Lanza et al. A Categorization of Classes based on the Visualization of their Internal Structure: the Class Blueprint. OOPSLA 2001

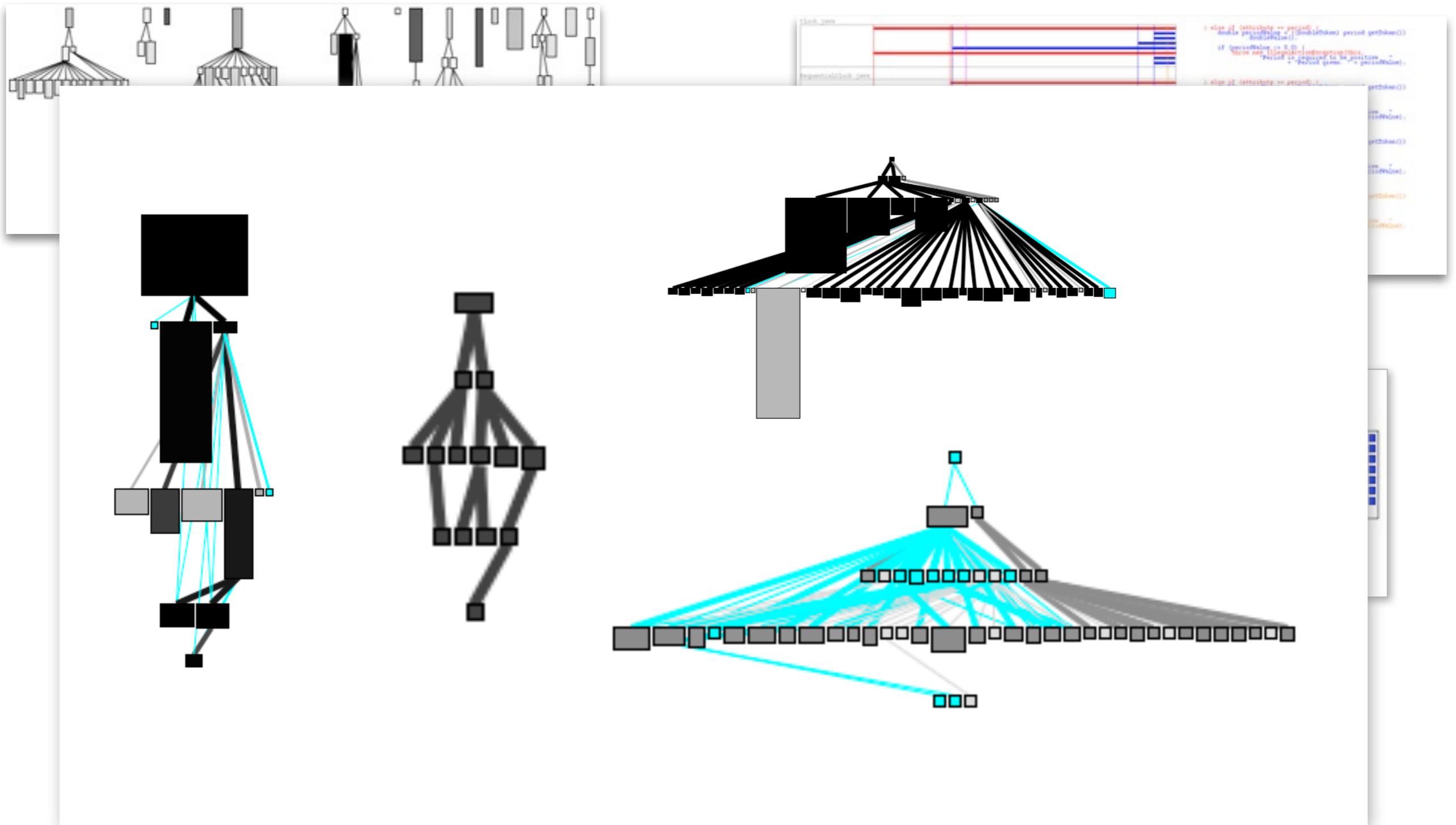


Topic correlation matrix

Kuhn et al. Semantic Clustering: Identifying Topics in Source Code. IST 2007

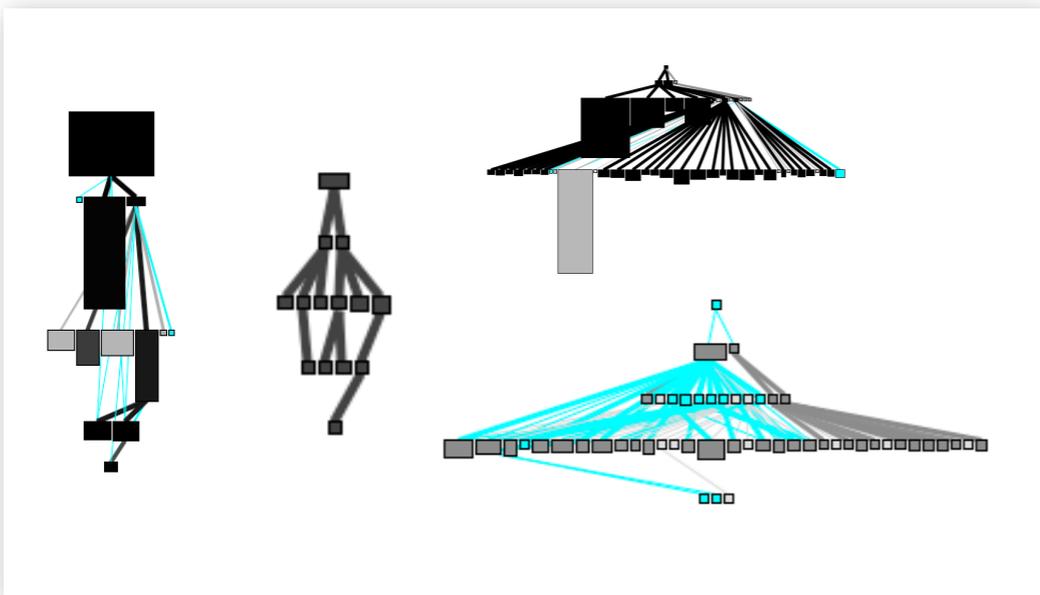
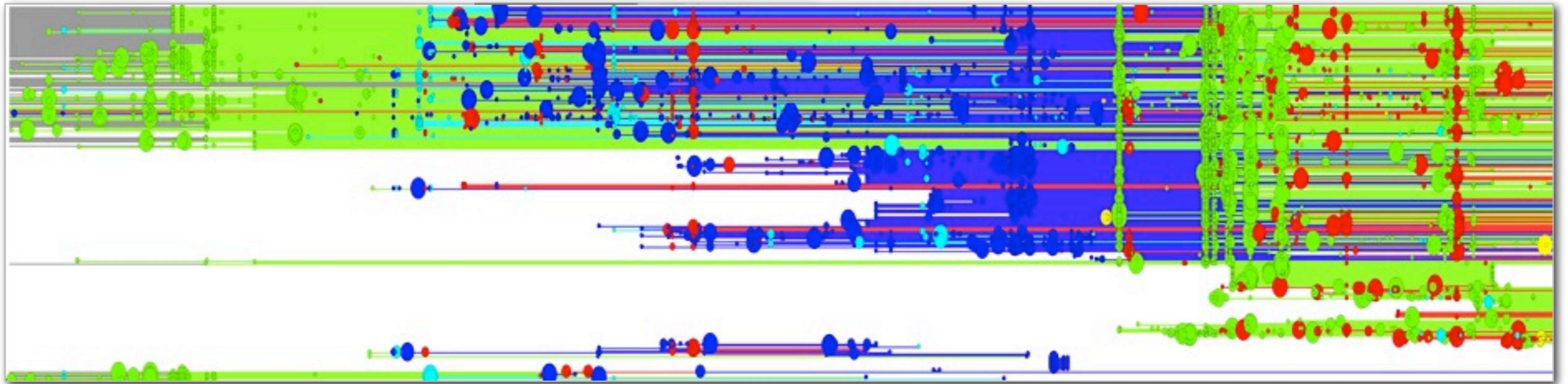
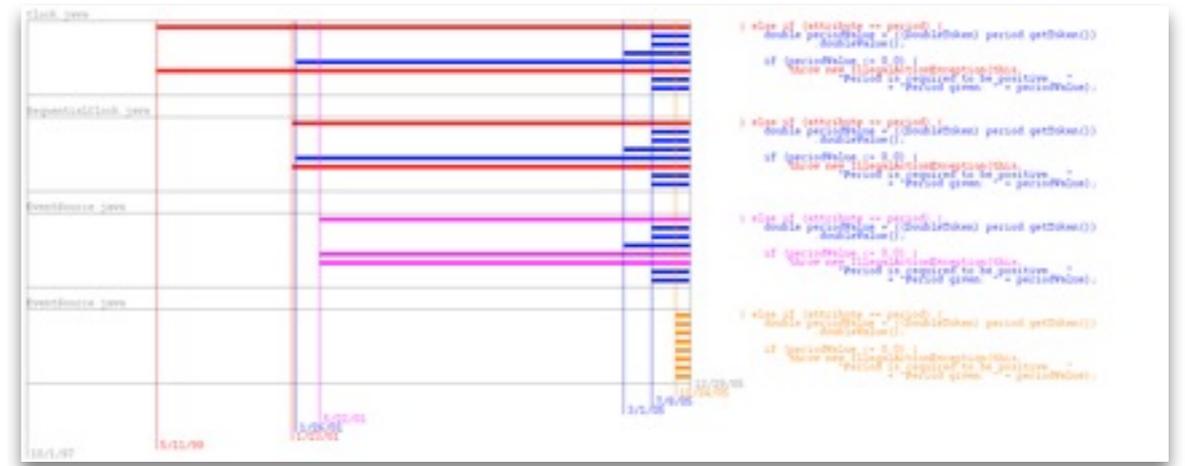
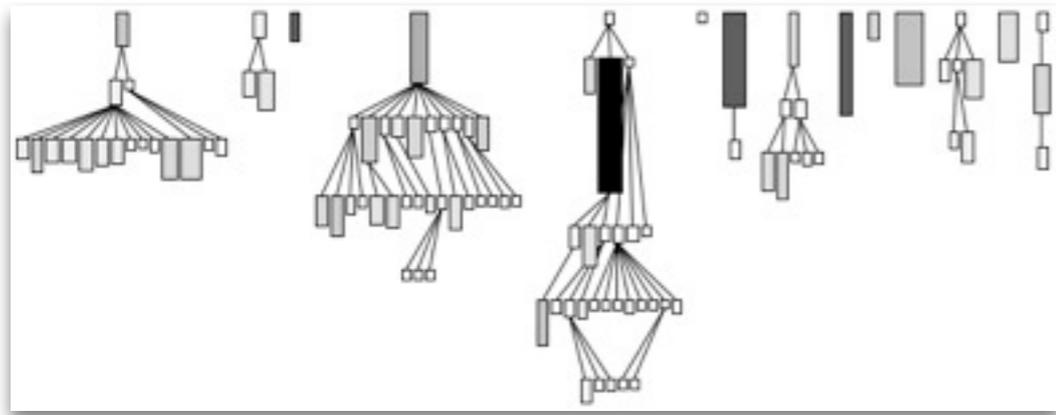


Distribution map
(topics spread over classes in packages)



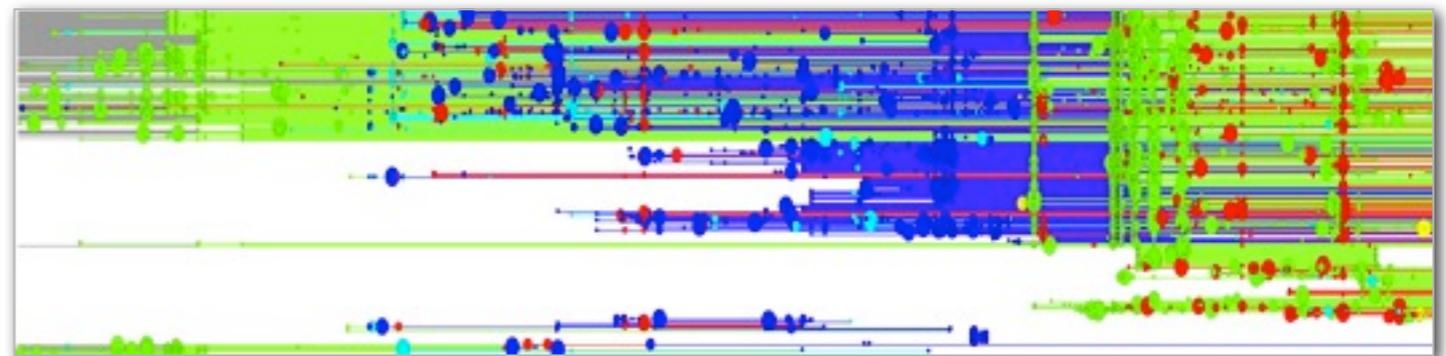
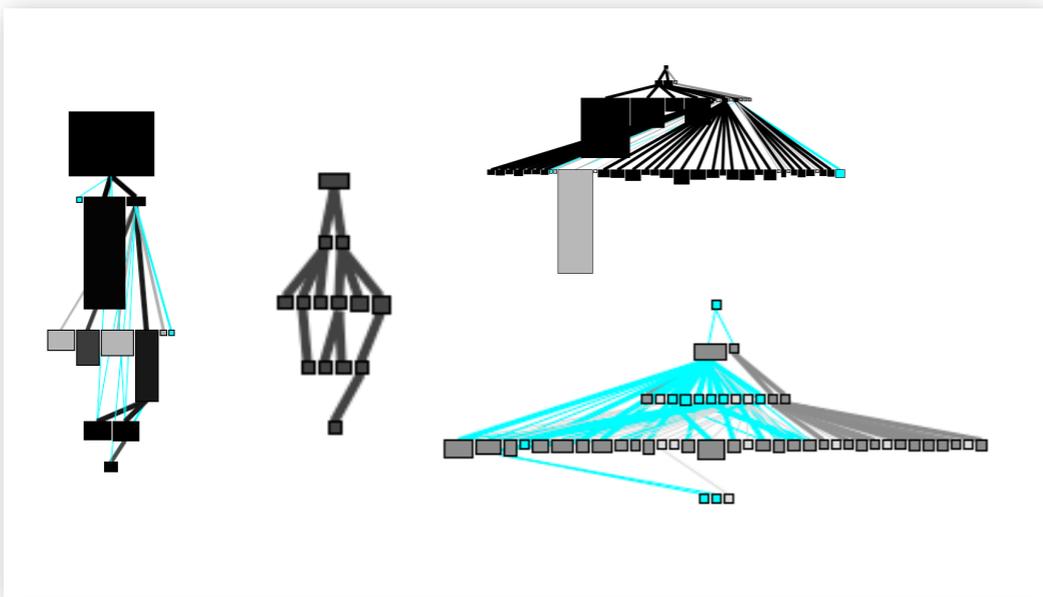
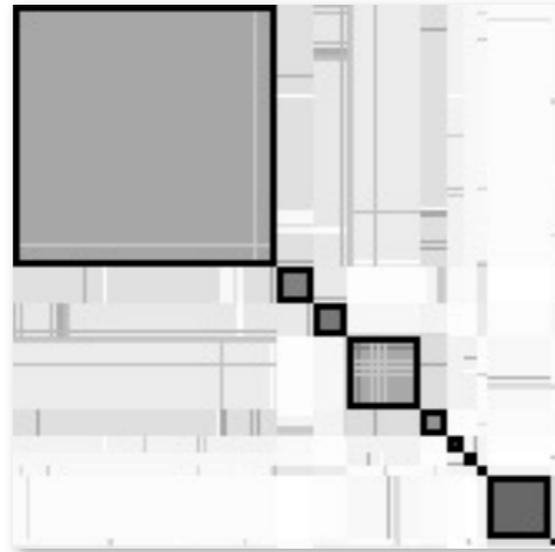
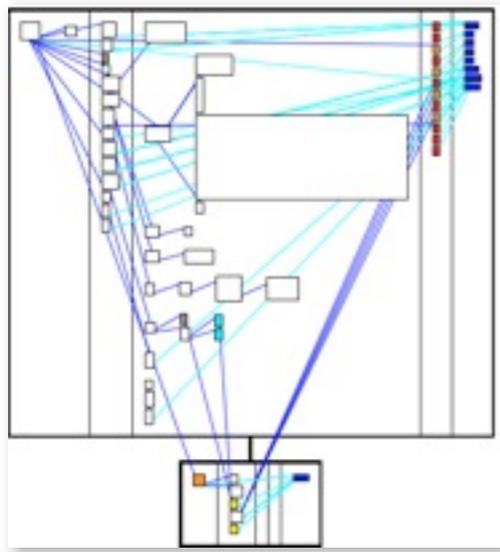
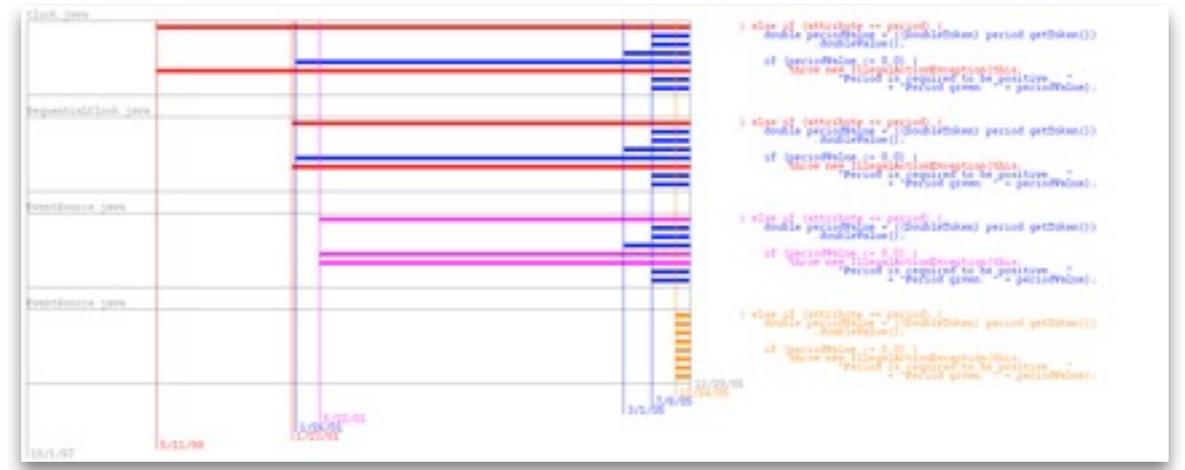
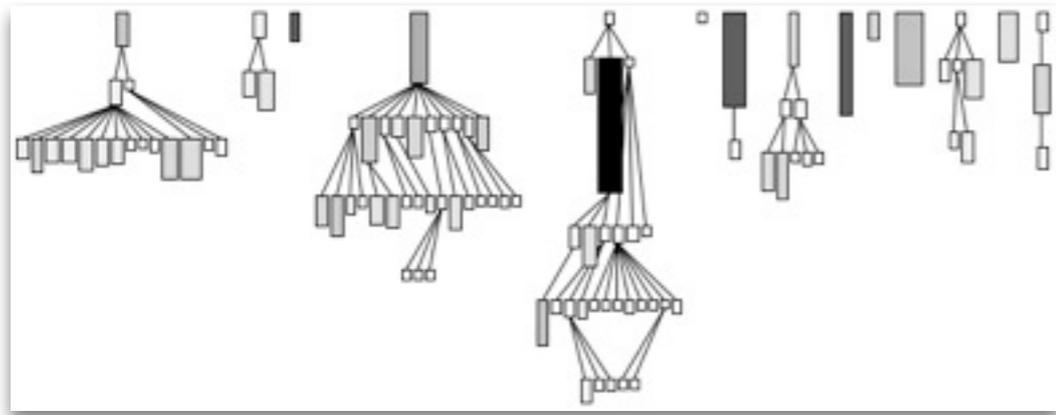
Hierarchy evolution view

Gîrba et al. Modeling History to Analyze Software Evolution. JSME 2006

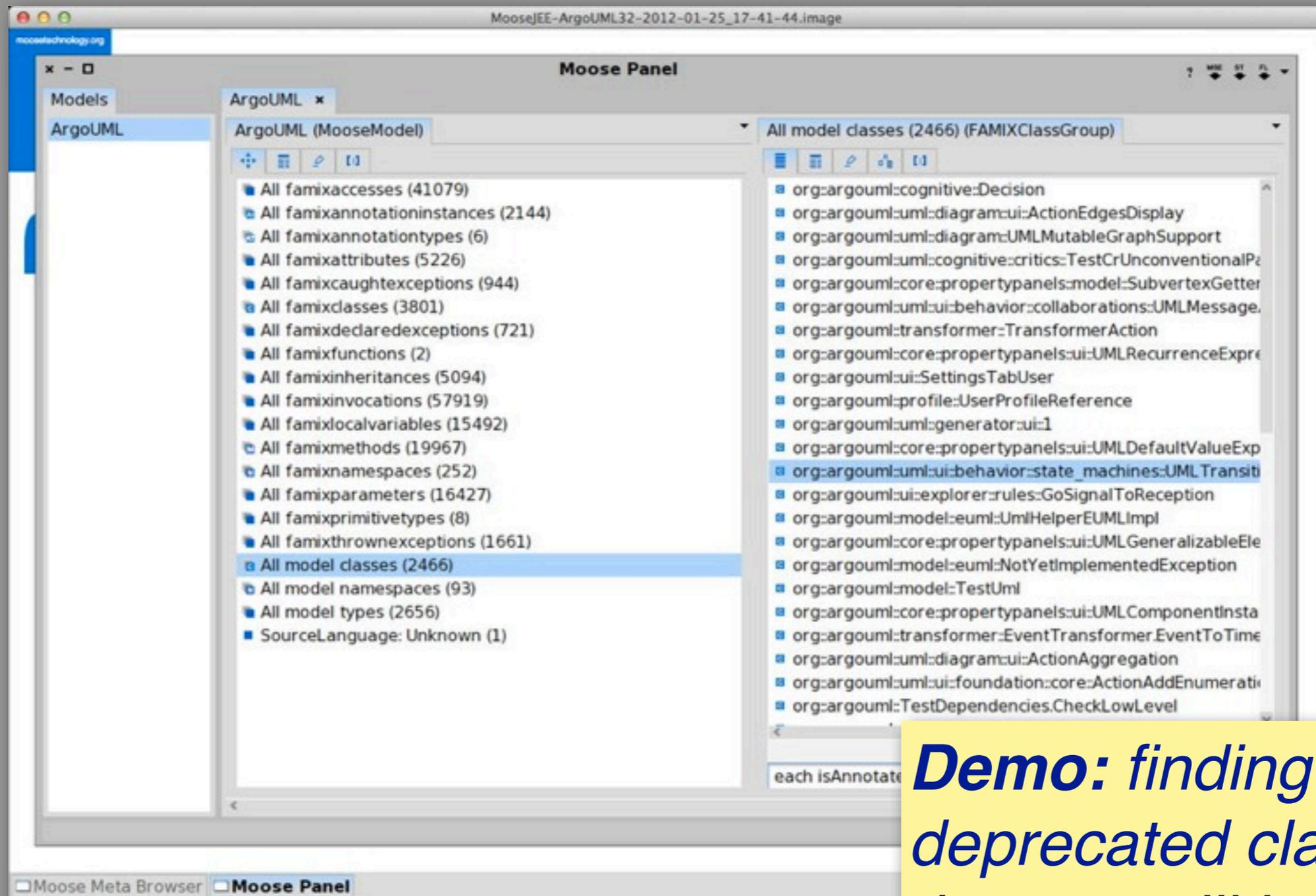


Ownership map

Gîrba et al. How Developers Drive Software Evolution. IWPSE 2005



Moose Demo



Demo: finding deprecated classes that are still in use ...

Mondrian Demo

Painting

Script

```
view shape rectangle
  size: #nameLength;
  identityFillColorOf: #namespaceScope.
view nodes: classGroup.
view edgesToAll: [ :class |
  classGroup select: [ :eachTarget |
    (eachTarget name pairsDistanceFrom: class name) > 0.5]].
view graphvizLayout neato.
```

Variable	Value
classGroup	Group (26...Classes)

Demo: visualizing name cohesion within packages

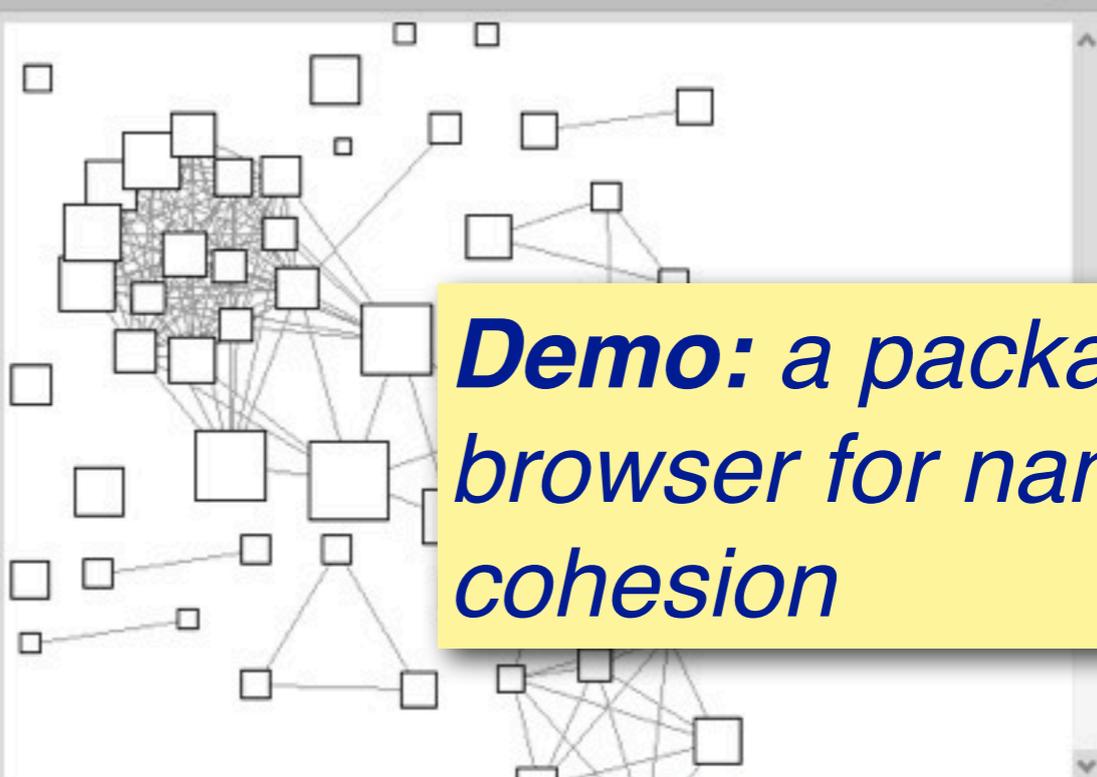
Meyer et al. Mondrian: An Agile Visualization Framework. SoftVis 2006

Glamour Demo

Glamorous Browser Editor

Preview

- _unknown_package_
- _anonymous
- <unknownPackage>
- ▶ jdepend
- ▼ javax
 - accessibility
 - ▶ swing
 - management
 - imageio
 - ▶ xml
 - ▶ jmi
- ▼ org
 - ▼ argouml
 - persistence**
 - ▶ profile



Script

```
composer tabulator with: [:t |  
  t column: #namespaces; column: #classes.  
  t transmit to: #namespaces; andShow: [:a |  
    a tree  
      display: [:mooseModel | mooseModel allNamespaces select:  
#isRoot ];  
      format: #name;  
      children: #childScopes ].  
  t transmit from: #namespaces; to: #classes; andShow: [:a |  
    a mondrian
```

Variable	Value
model	a MooseMo...(194511)

Demo: a package browser for name cohesion

Bunge et al. Scripting Browsers with Glamour. ESUG 2009

Challenges for Agile Software Assessment



Customization



Context



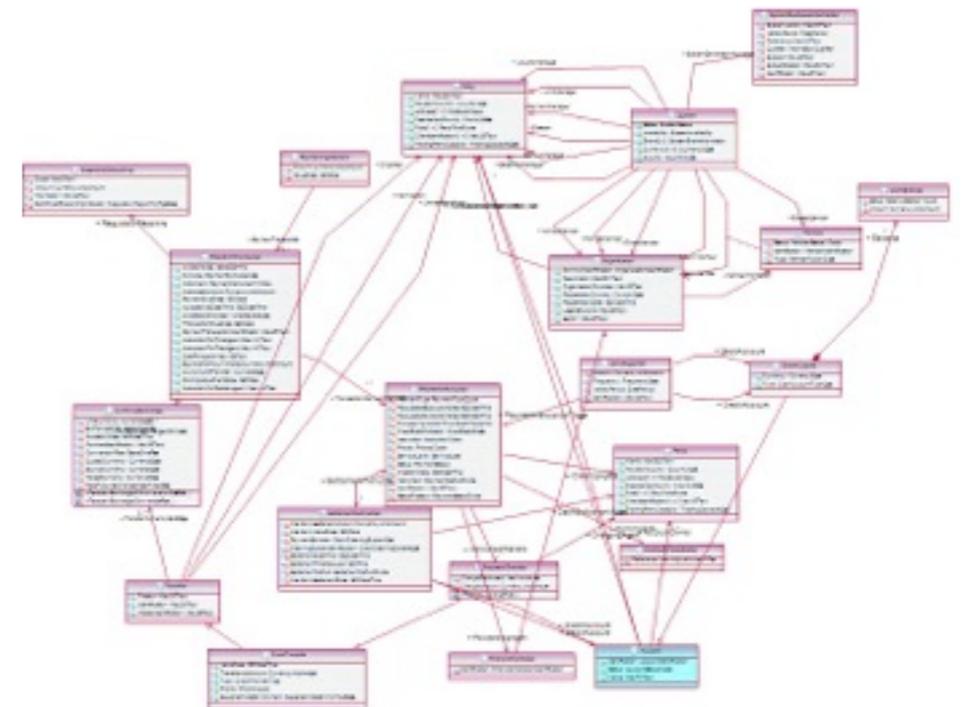
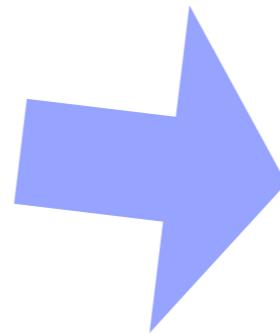
Continuous Assessment

Customization



Challenge

**Load the model in the morning,
analyze it in the afternoon**



Problems



Unknown languages



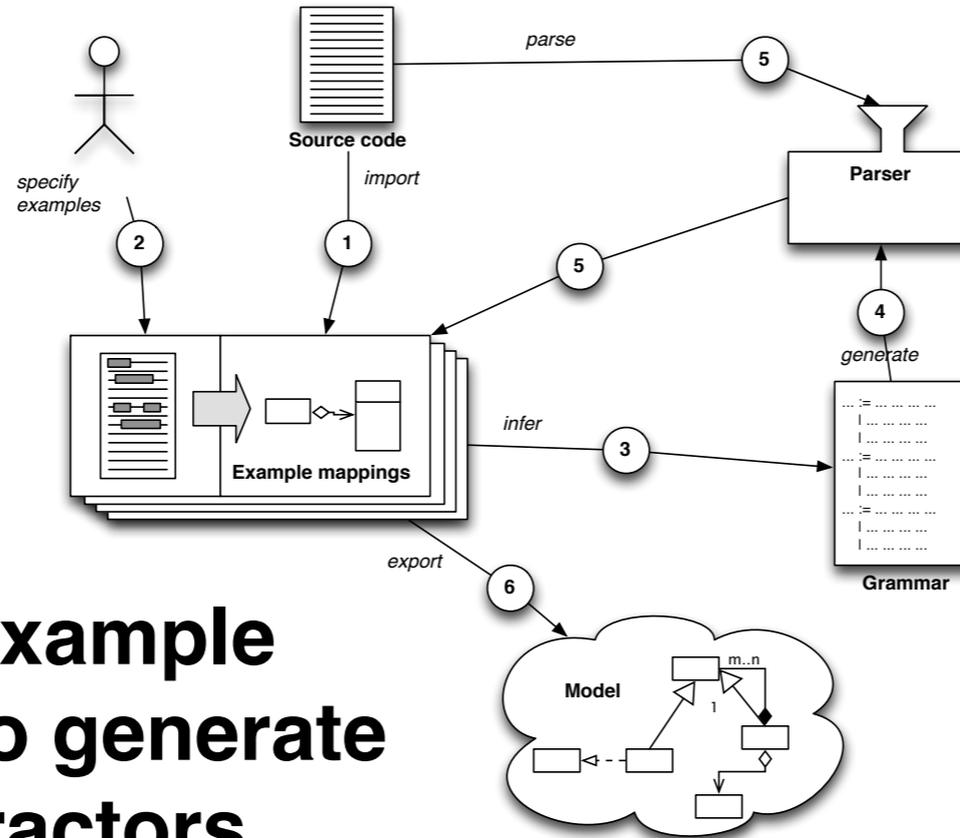
Unstructured text



Heterogeneous projects

Perin et al. Recovery and Analysis of Transaction Scope from Scattered Information in Java Enterprise Applications. ICSM 2010

Ideas

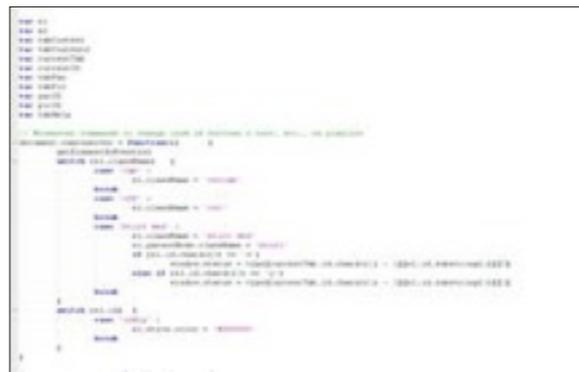


Nierstrasz et al. Example-Driven Reconstruction of Software Models. CSMR 2007

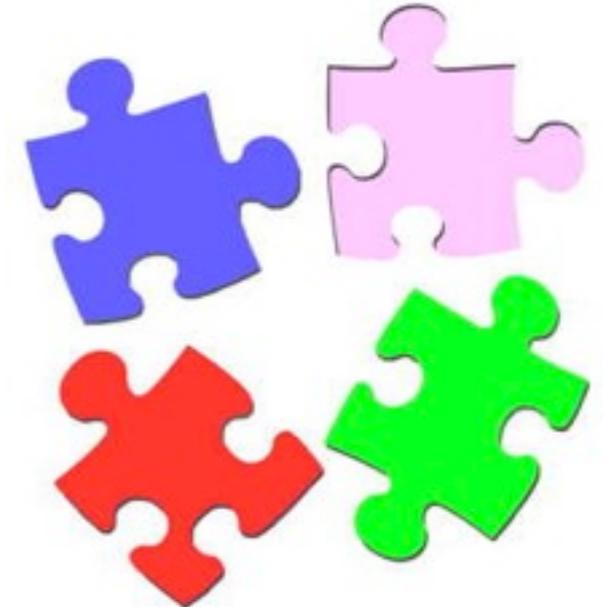
Exploit example mappings to generate fact extractors



Incrementally refine island grammars



Exploit eg indentation as a proxy for structure



Exploit similarities between languages (adapt and compose)

Challenge

Build a new assessment tool in ten minutes



Problems

What tools do developers really need?

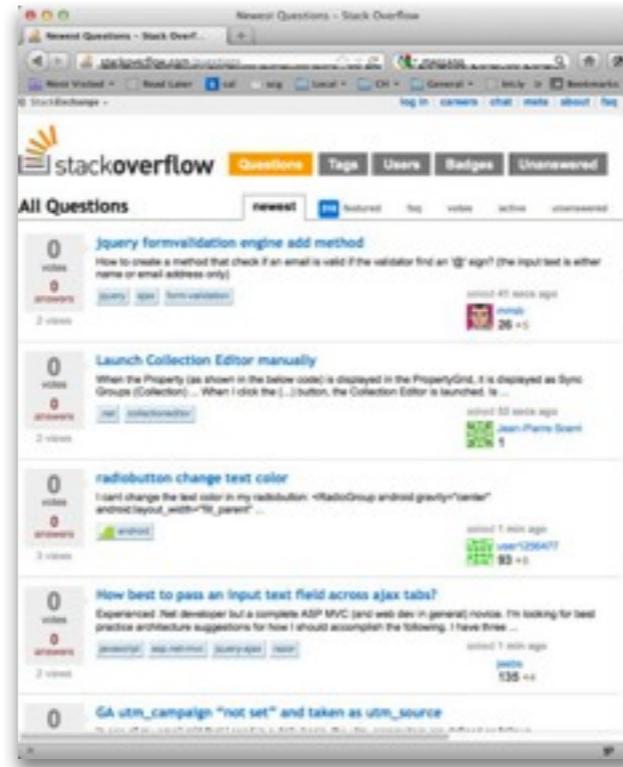


What is a unifying meta-model for tool construction?



What are appropriate meta-tools?

Ideas



**Analyze
developer
needs (!)**



**From meta-models to
interactive DSLs**

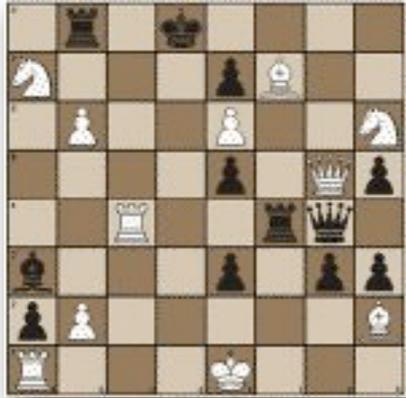


**“Malleable” IDE
(not just plug-ins)**

Context



Challenge



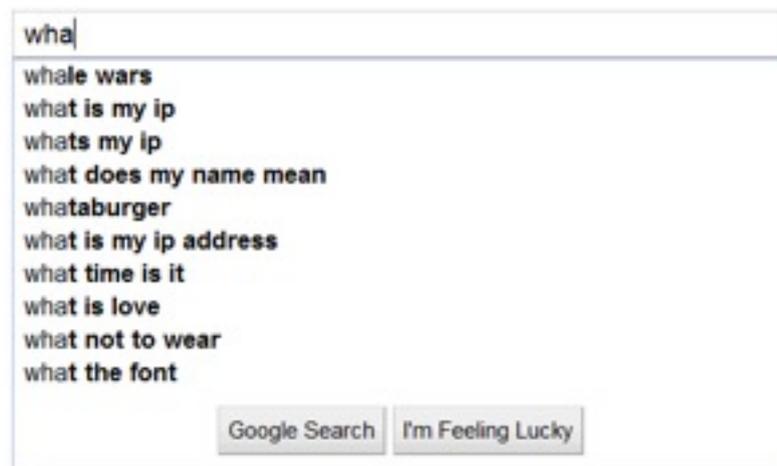
“Who had this problem before, and how did they solve it?”



Problems



Is the answer out there?

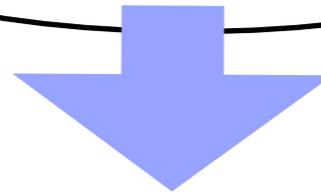
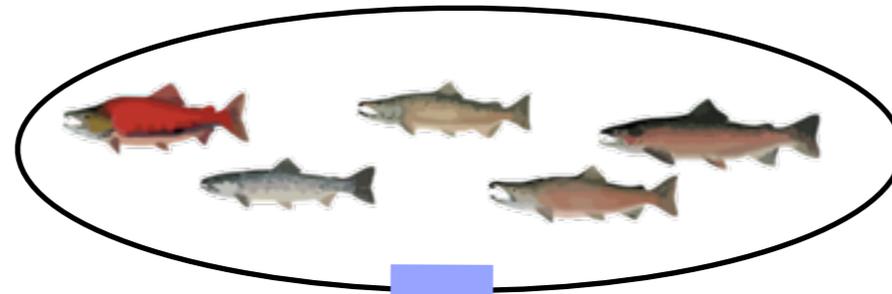


How to express the query?



Is intent captured?

Ideas



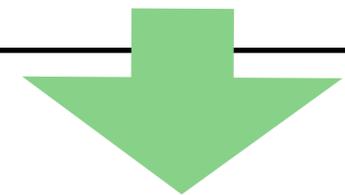
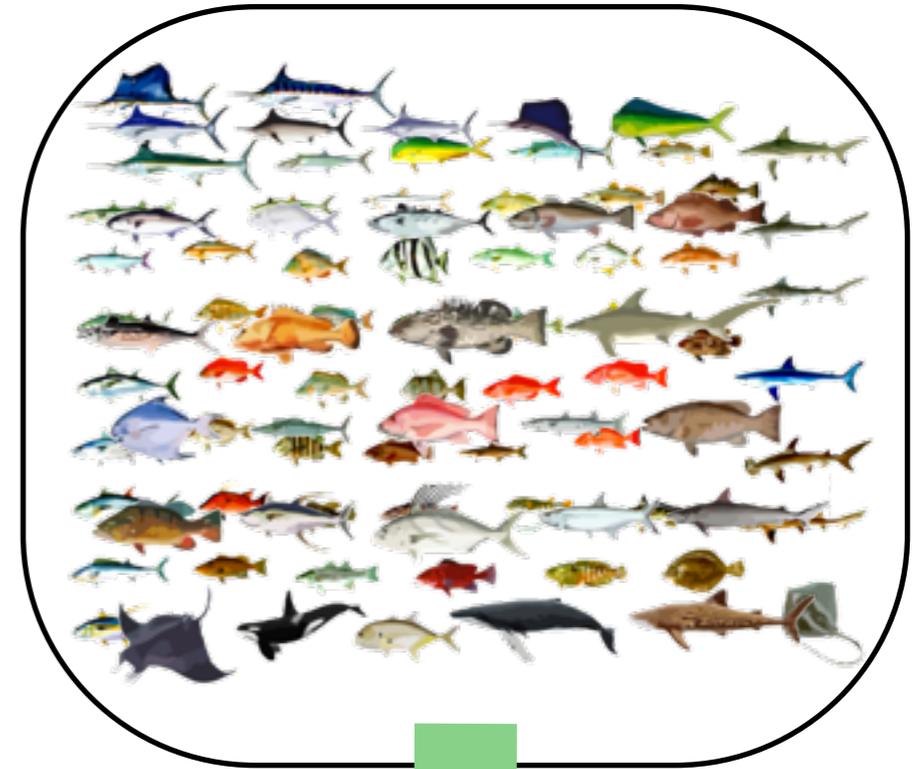
**Ecosystem
analysis**



**Clone analysis
for querying**



**Embedding
intelligence in
the IDE**



**Exploit big
software data**

*Lungu et al. Big Software Data
Analysis. ERCIM News 2012*

Continuous Assessment



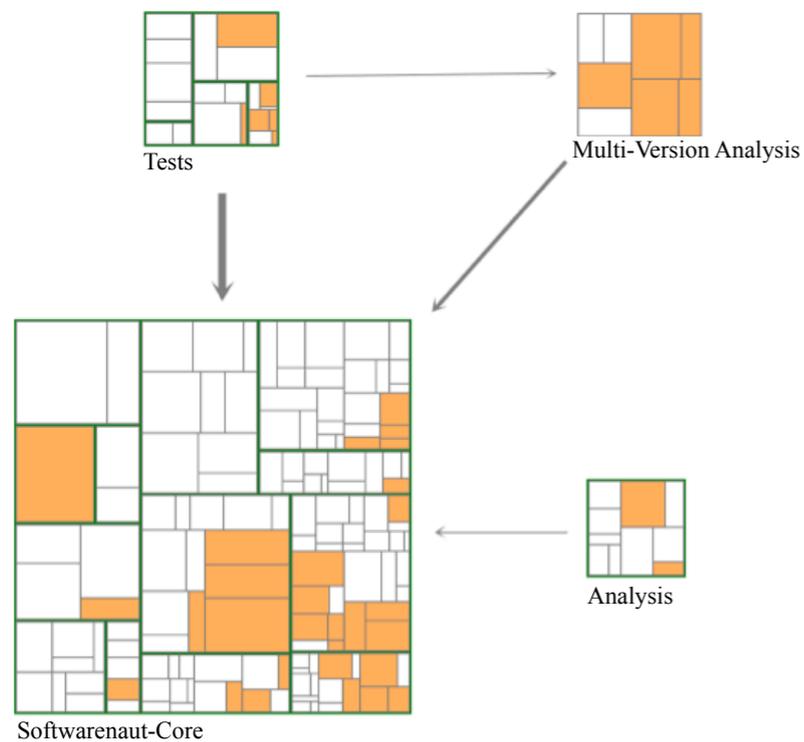
Challenge



***“What will my code
change impact?”***



Problems



Understanding architectural constraints

Lungu et al. Evolutionary and Collaborative Software Architecture Recovery with Softwareaut. SCP 2012



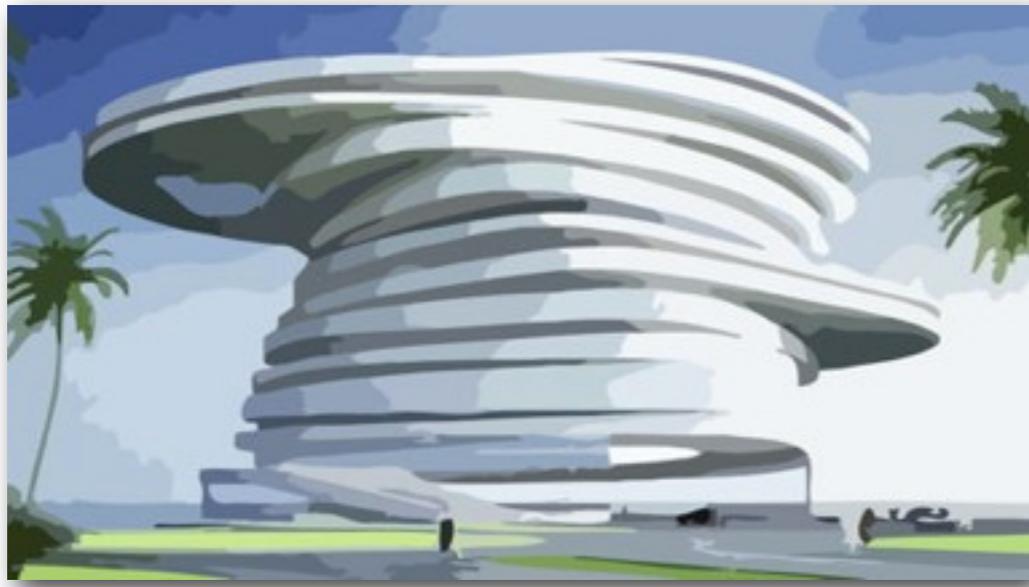
Implicit dependencies



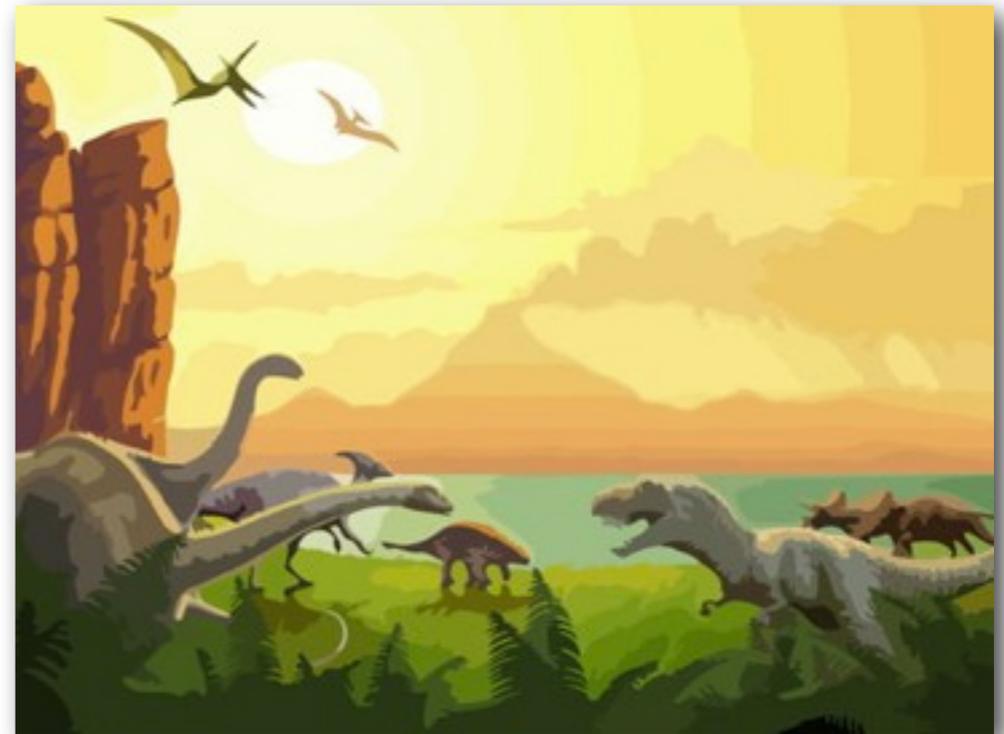
Ripple effects

Robbes et al. A Study of Ripple Effects in Software Ecosystems. ICSE-NIER 2011

Ideas



**Architecture
monitoring
(beyond layers)**



**Ecosystem
monitoring**

Conclusion

**Current IDEs offer
developers poor
support for software
assessment**



***Developers need support
for customization, context,
continuous assessment***