Understanding your code assets with Moose

Stéphane Ducasse, Tudor Gîrba, Adrian Kuhn

Software Composition Group

University of Bern, Switzerland
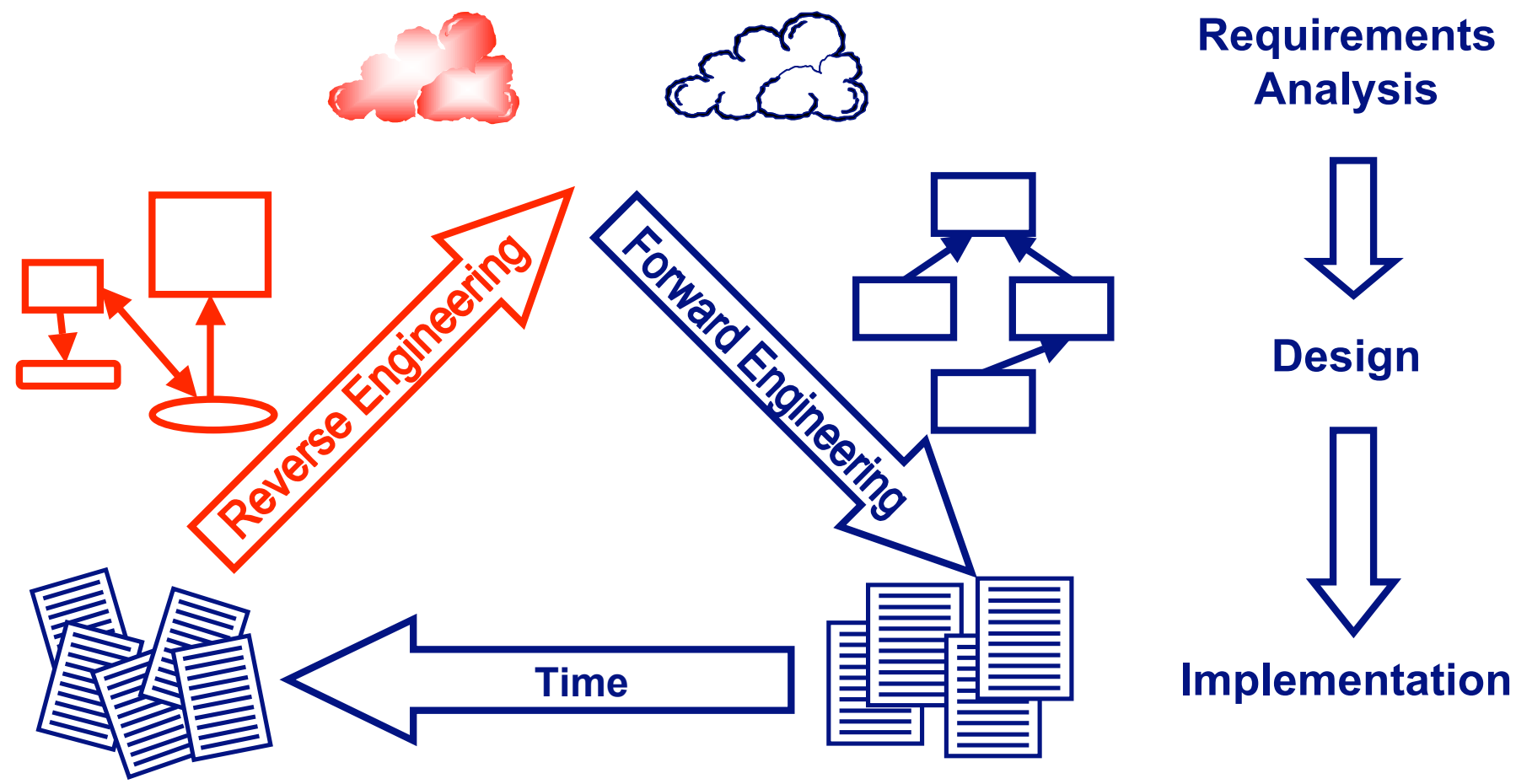
Michele Lanza

Faculty of Informatics

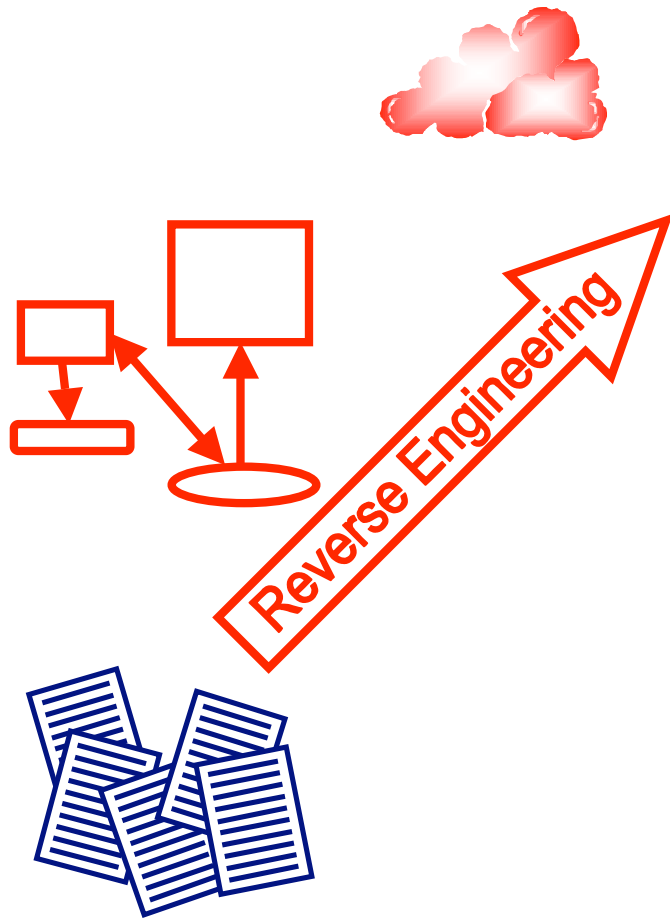**University of Lugano, Switzerland**

…

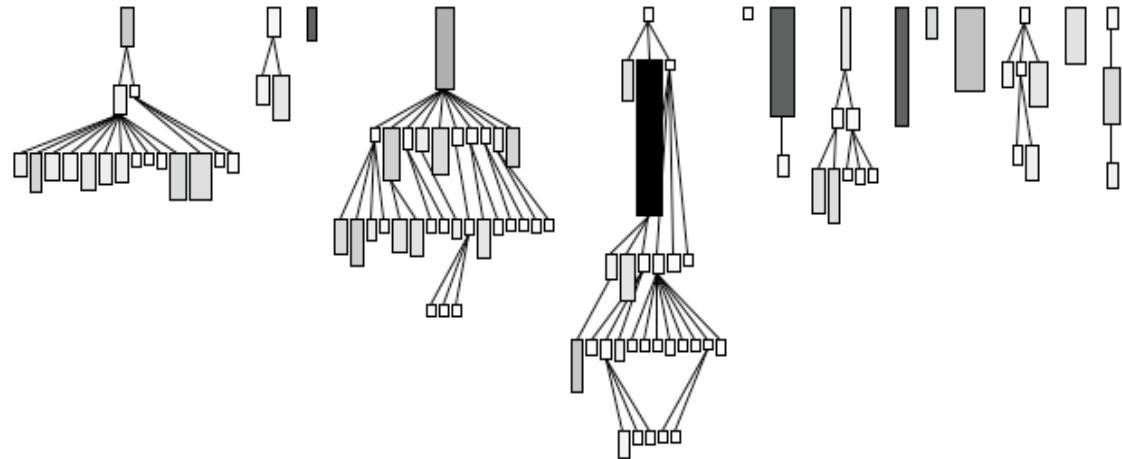# Context: Software development is more than forward engineering

Reverse Engineering

Forward Engineering

Time

Requirements Analysis

Design

Implementation

Reverse engineering is creating high level views of the system …
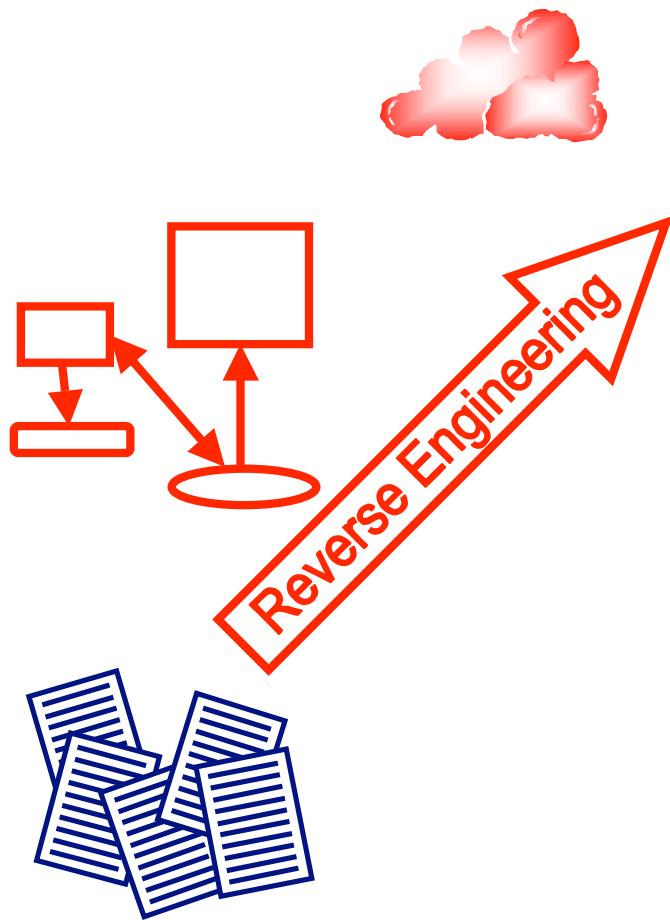
For example:

Reverse Engineering

It provides an overview, by striping away details

Reverse engineering is creating high level views of the system …

Multiple techniques:
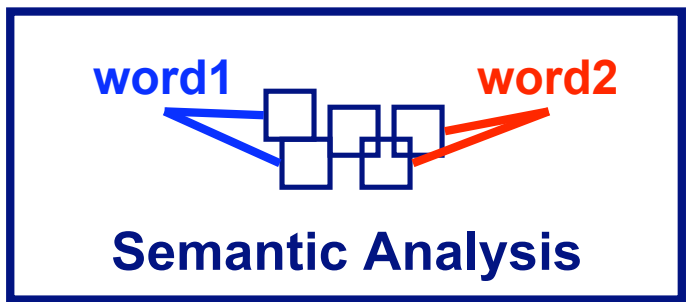 **UML diagrams**
 Metrics
 Visualization
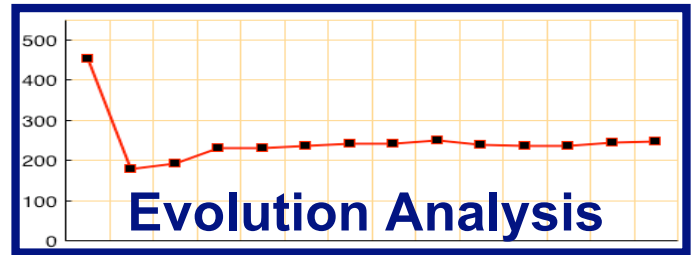
 …

Reverse Engineering

Not an algorithm

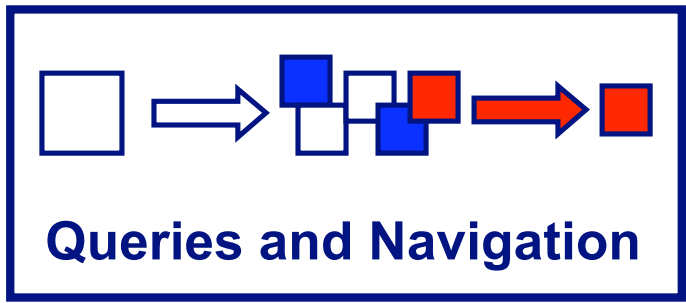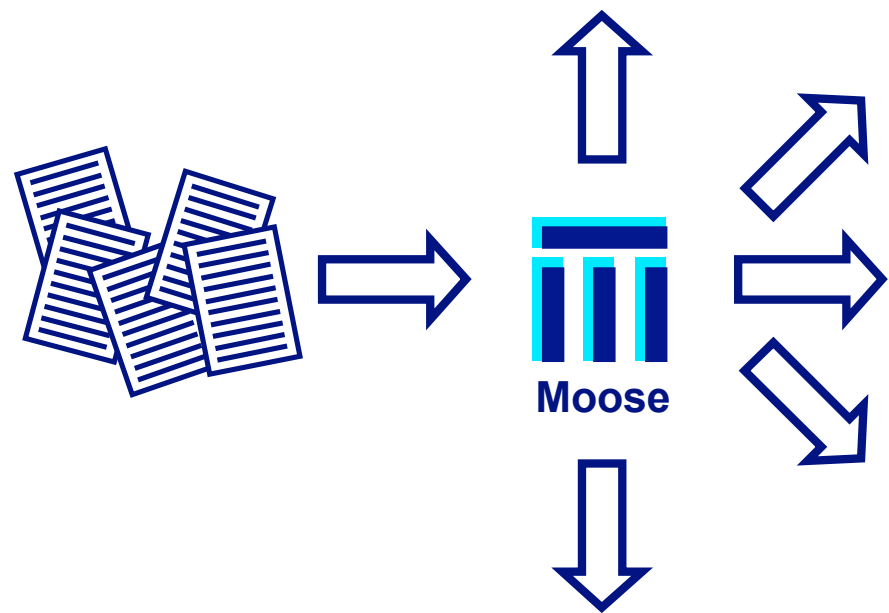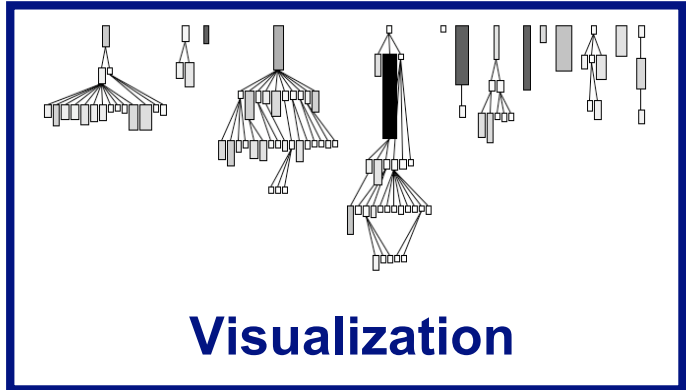**Large systems**

Moose is a reengineering tool which integrates multiple techniques

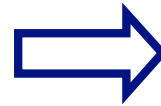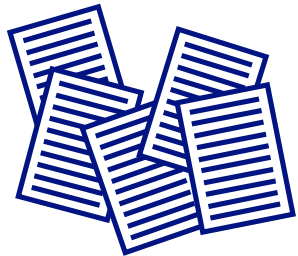Number of classes   = 382
Number of methods = 4268
...
Metrics

Visualization

Moose

Queries and Navigation

Evolution Analysis

word1   word2

Semantic Analysis

...

Metrics compress the system into numbers and enable comparisons

Number of classes	= 382
Number of methods	= 4268
Average number of methods per class	= 7

…

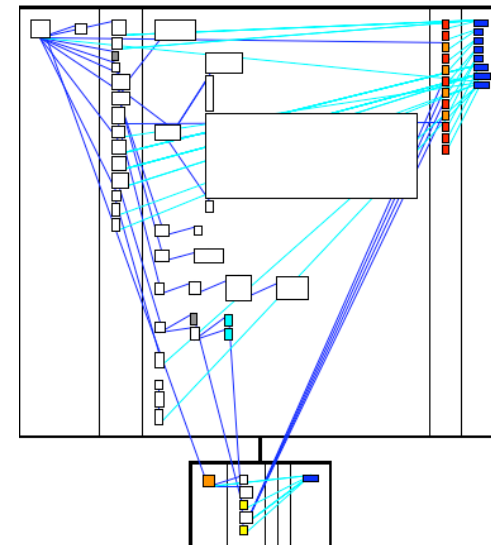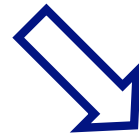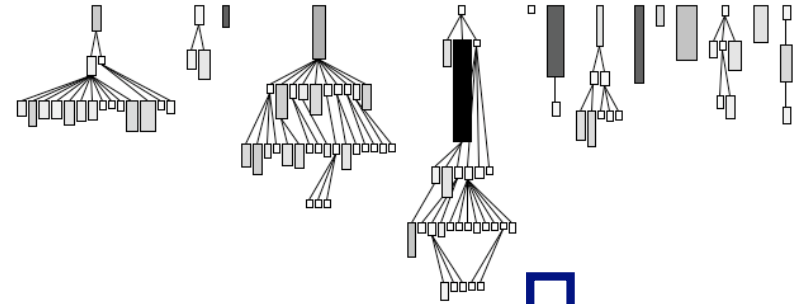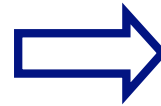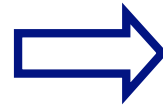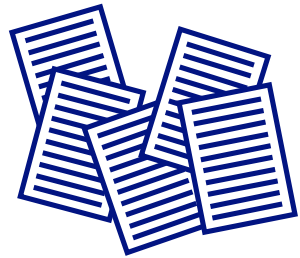## Metrics can answer questions:

How big is the system?

How is the complexity spread over the system?

…

Visualization compresses the system into pictures
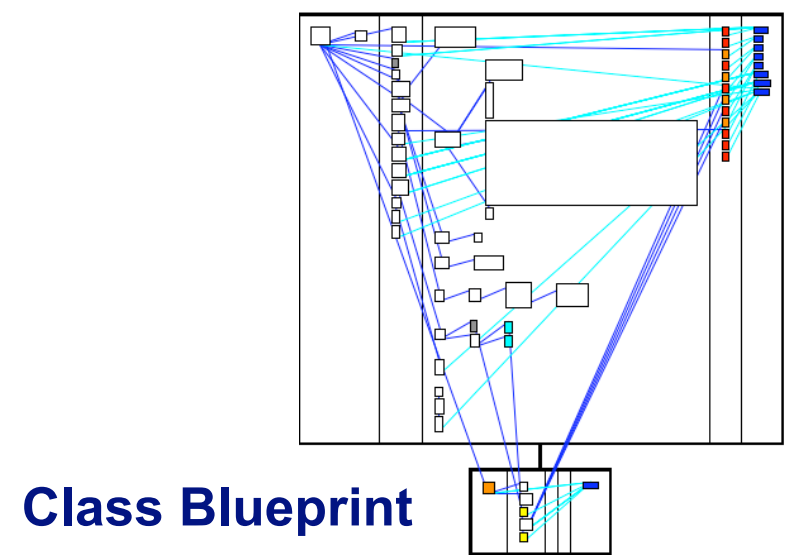

Visualization can provide:

Quick overview

Spatial orientation

…

Visualization: Polymetric views can display up to 5 metrics on a 2D figure

Width Metric

Height Metric

Position Metrics

Color Metric

System Complexity View

Class Blueprint
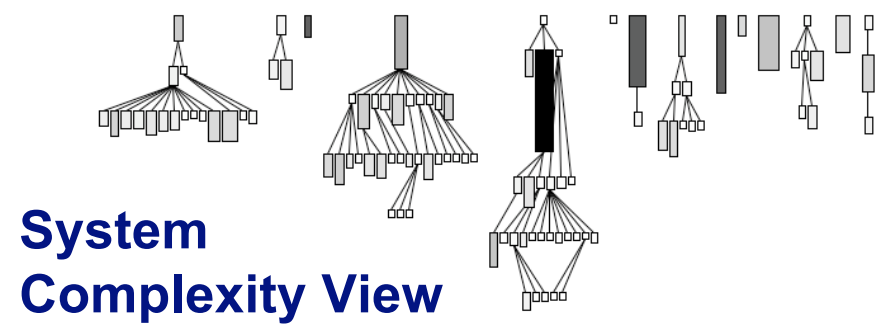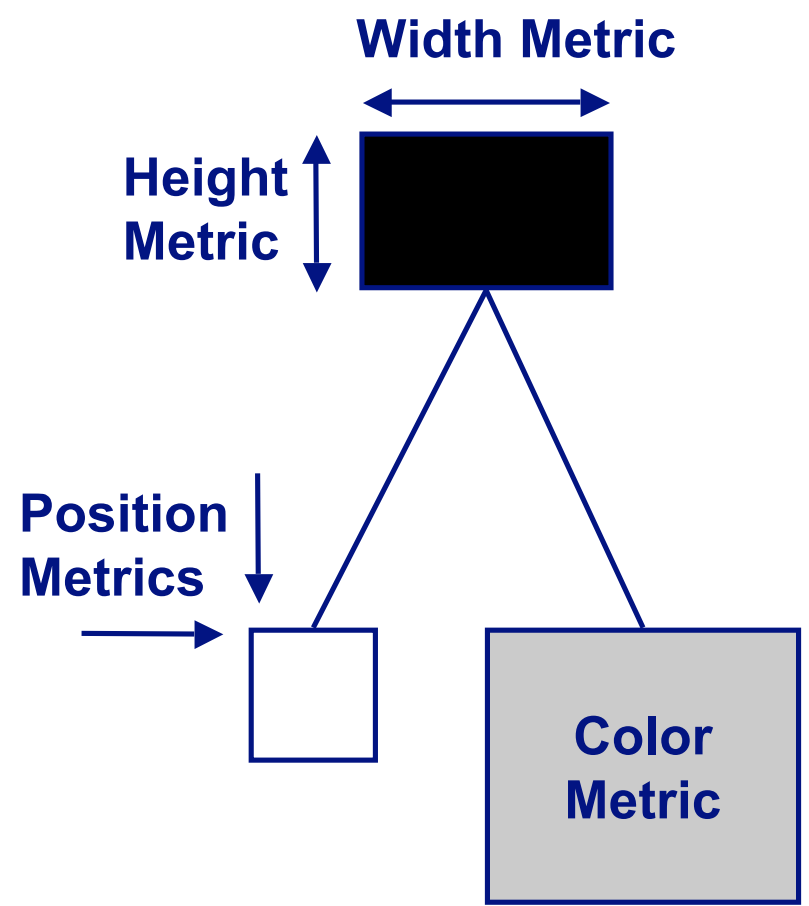
Queries reduce the analysis space
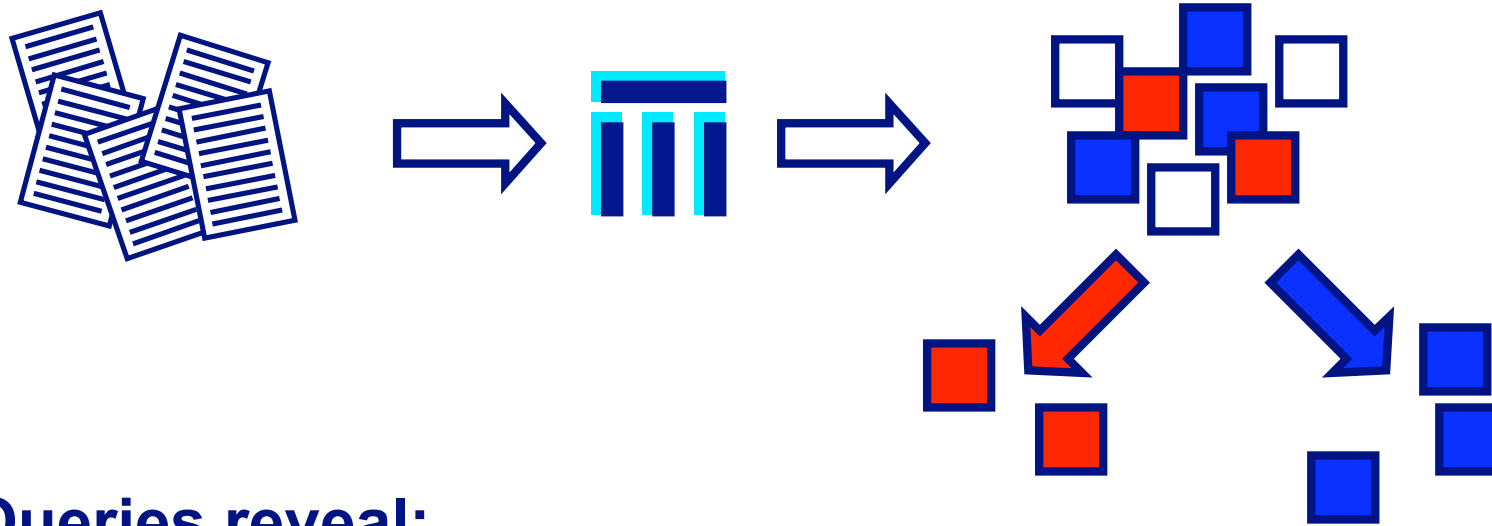
Queries reveal:

Similar entities

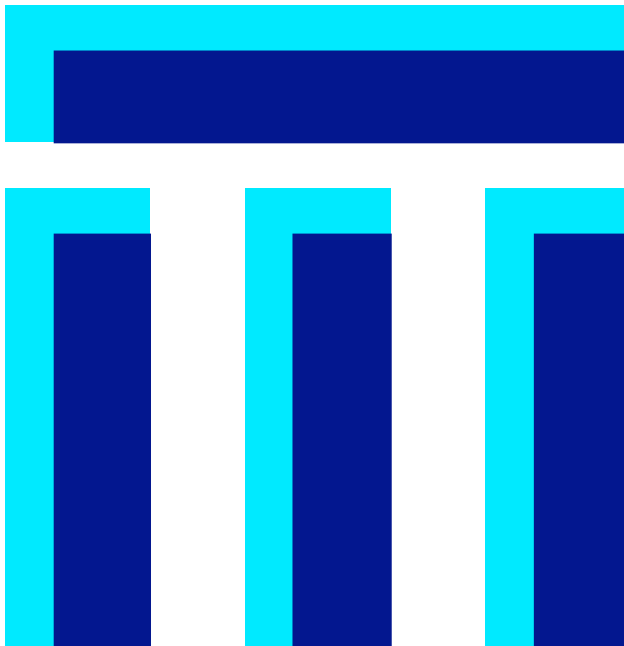Entities which are out of ordinary

Queries make use of properties (e.g., metrics)

Moose at work: analyzing the structure of software systems

Number of classes = 382
Number of methods = 4268
...

Metrics

Visualization

Queries and Navigation
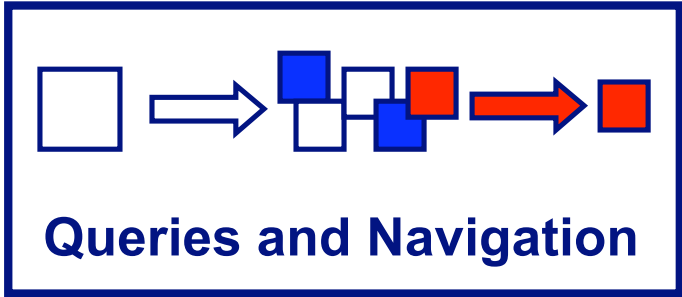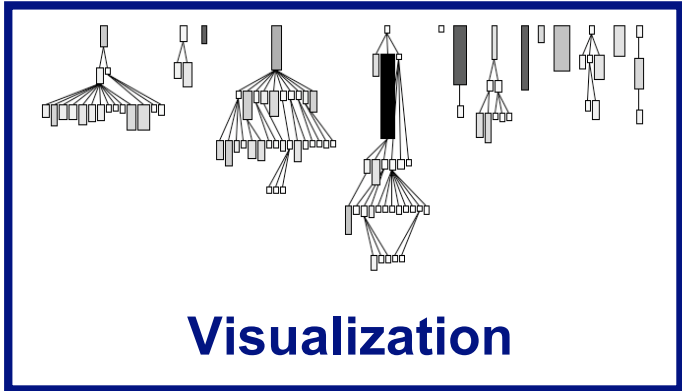
Semantic analysis attaches semantics to structure

word1
word2
word3

…
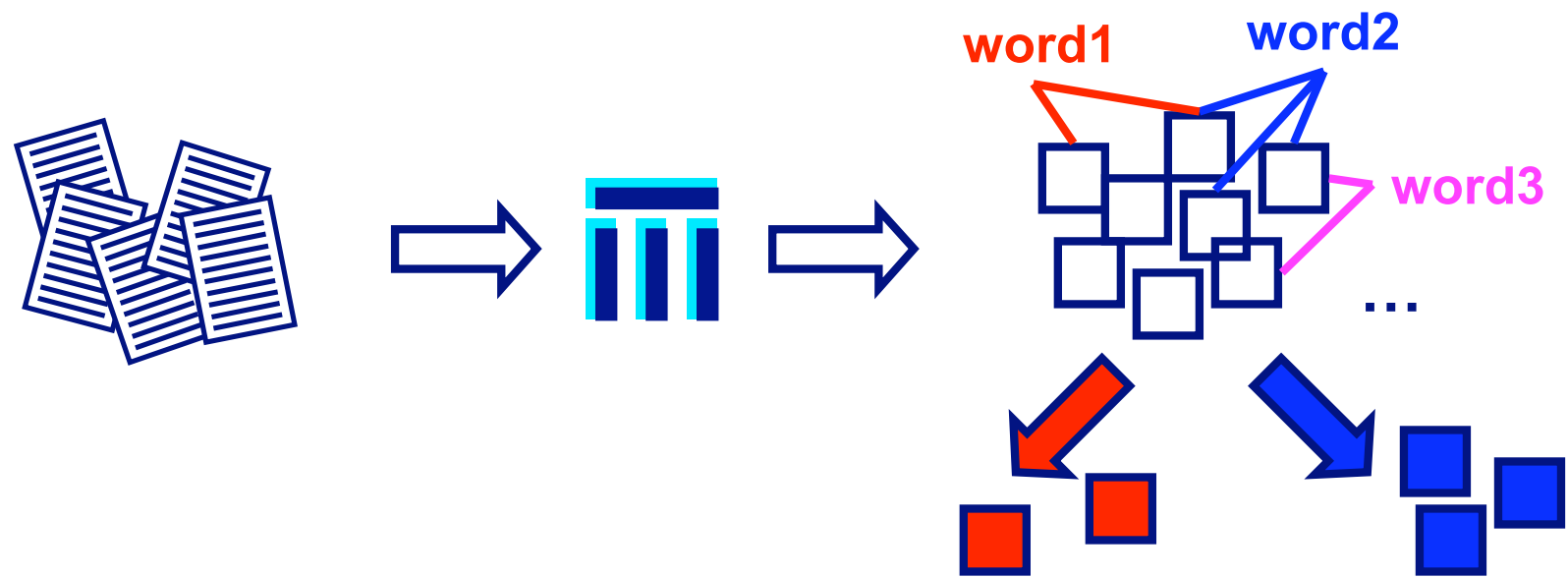
Applicability:

Text Search

Semantic Clustering

…

Moose at work: analyzing the semantics of software systems

word1   word2

Semantic Analysis
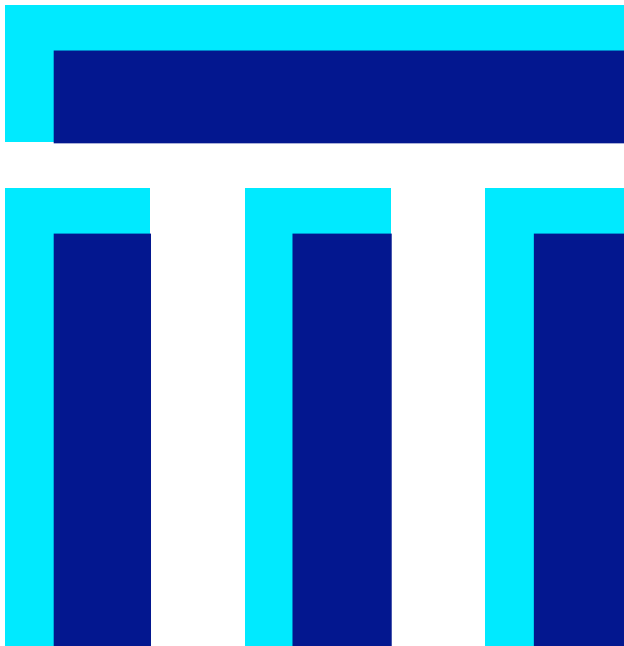
Time contains useful information for reverse engineering

Reverse Engineering

Forward Engineering

Time

Requirements Analysis

Design

Implementation
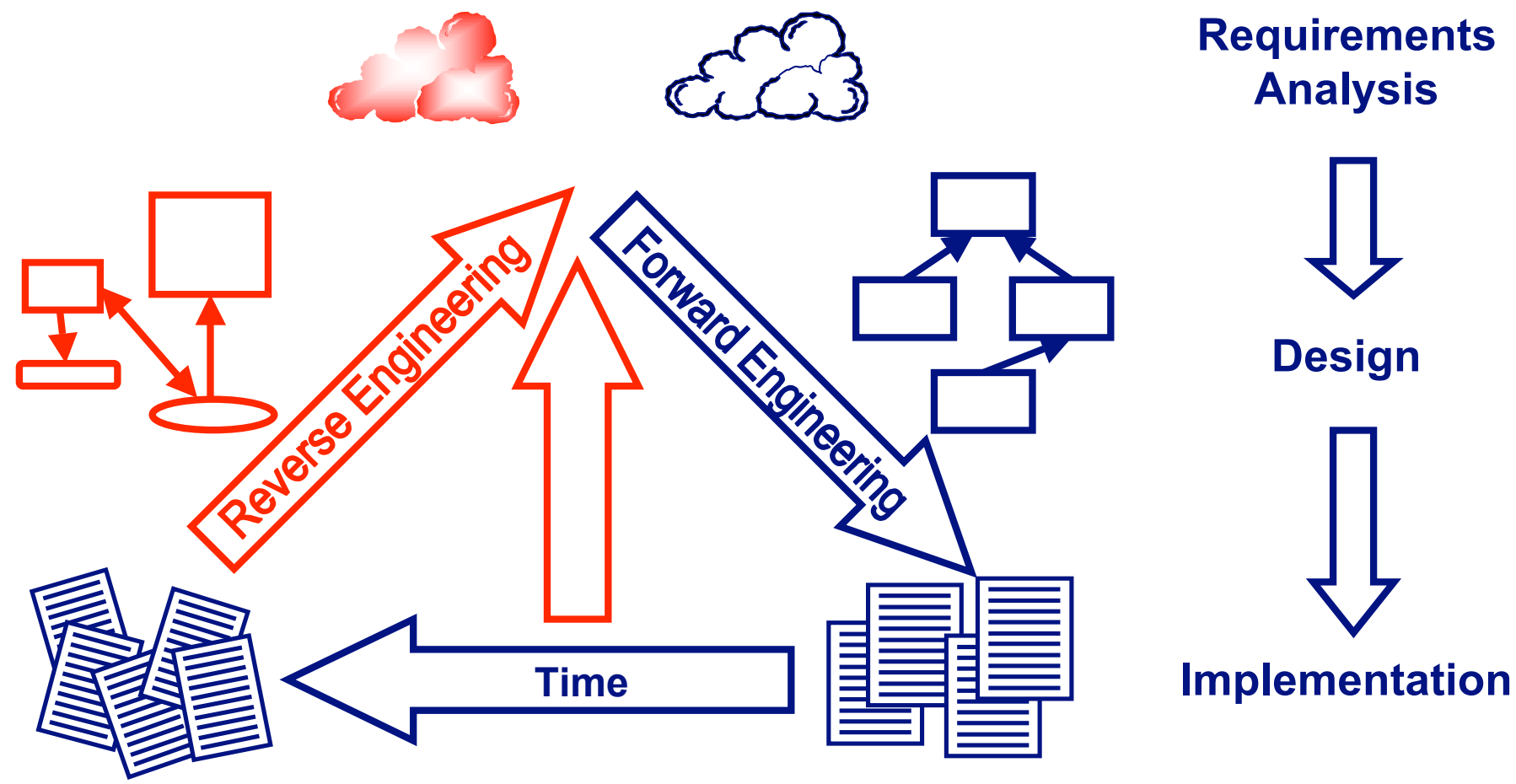
The evolution analysis compresses time through metrics, visualization …
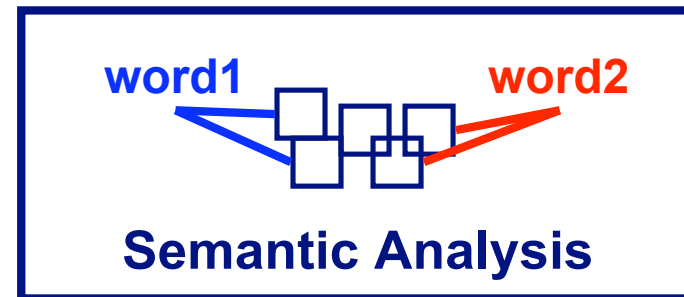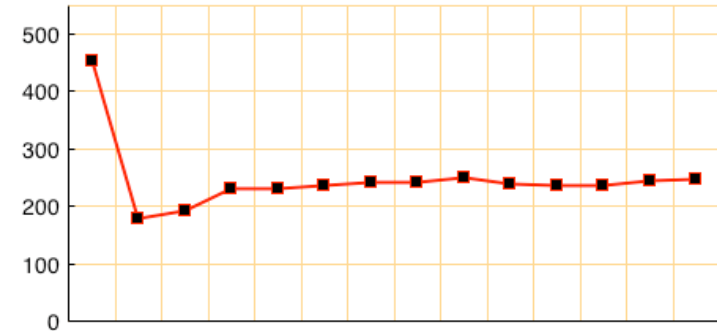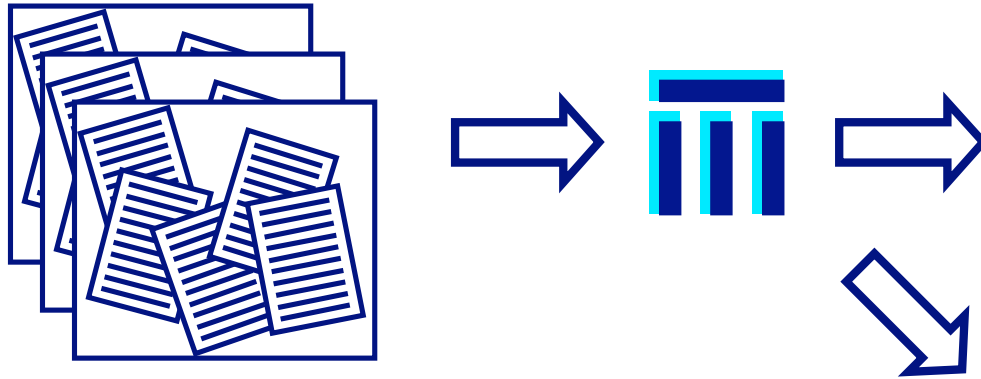

History can answer questions:

Which parts are change-prone?

How did the system get in the current shape?

…
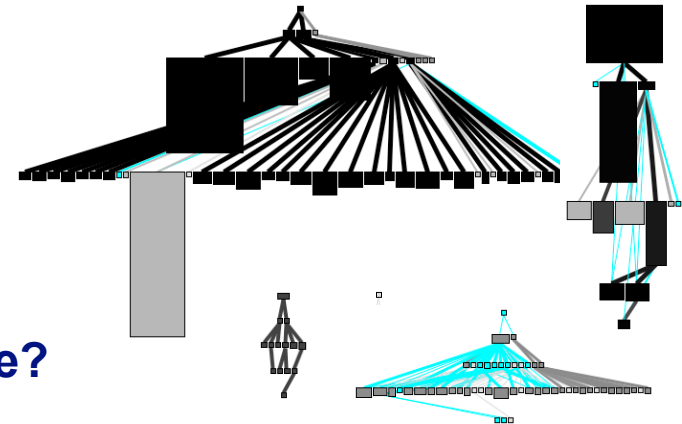
Problem: Large amounts of data

Our approach: History as a first class entity encapsulates the evolution

History can be measured: How much was a class changed in its history?

Evolution of Number of Methods (**E_NOM**) $= 3 + 2 + 1 + 0 = 6$

Stability of Number of Methods (**S_NOM**) $= \dfrac{0 + 0 + 0 + 1}{4} = 0.25$

Moose at work: analyzing the history of software systems

Evolution Analysis
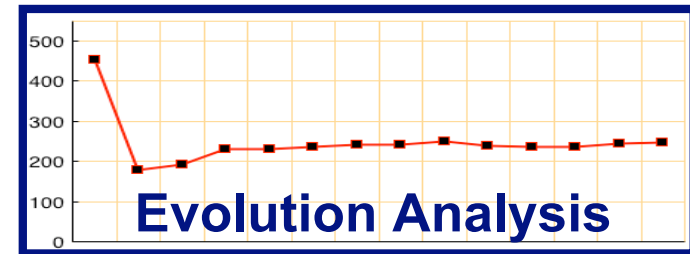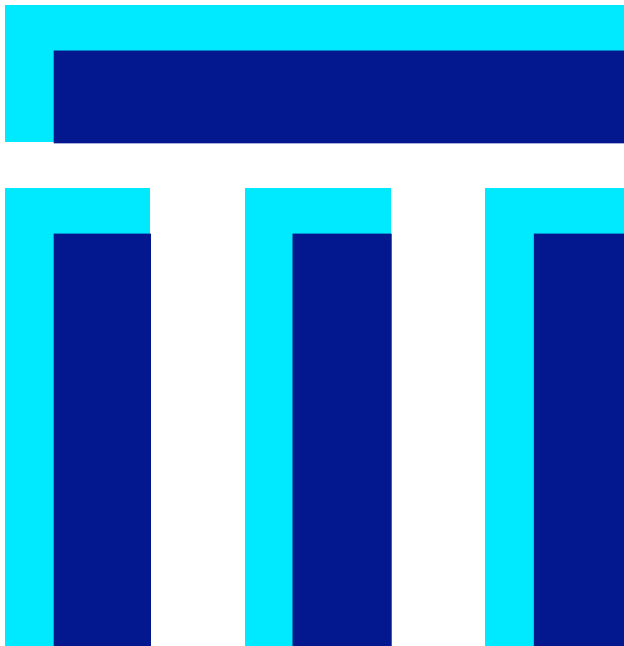
Actually, Moose is an environment for reengineering
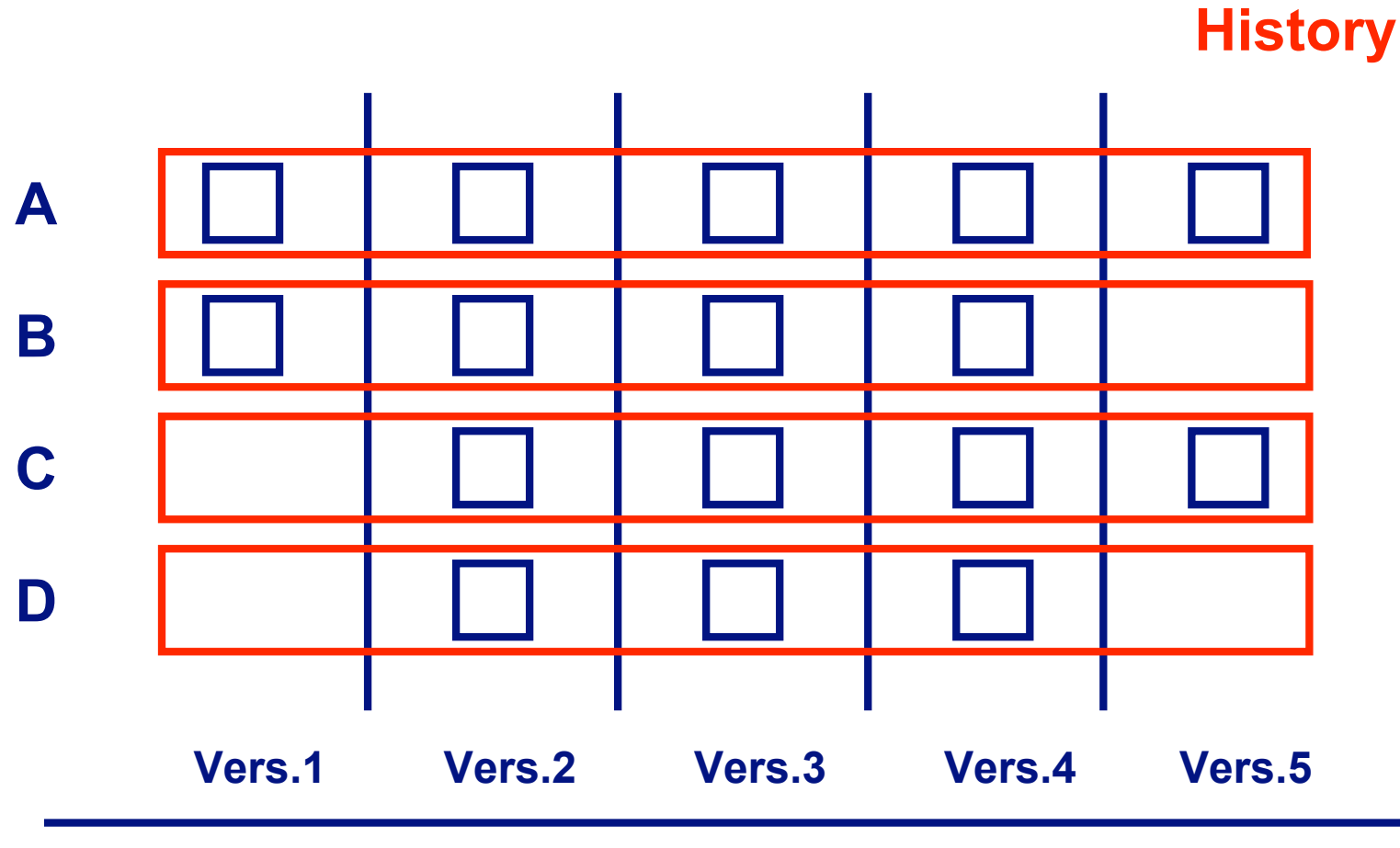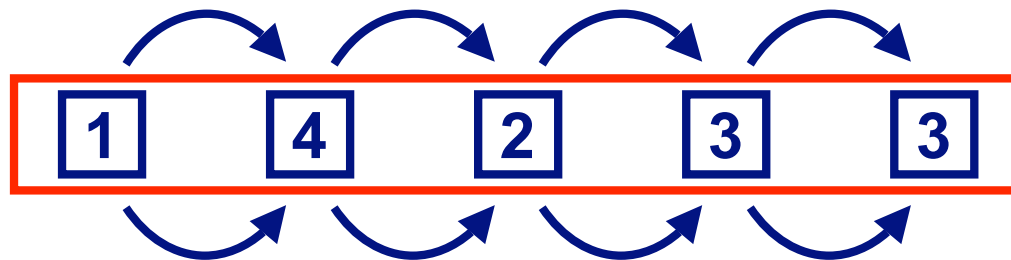
Semantic analysis
Concept analysis
Dynamic analysis
Clustering
Duplication detection

. . .

CodeCrawler

Visualization

Van

Evolution analysis

VW Smalltalk

Java

C/C++

Parser

Cobol

Tools integration mechanisms

Metrics

Language independent model repository
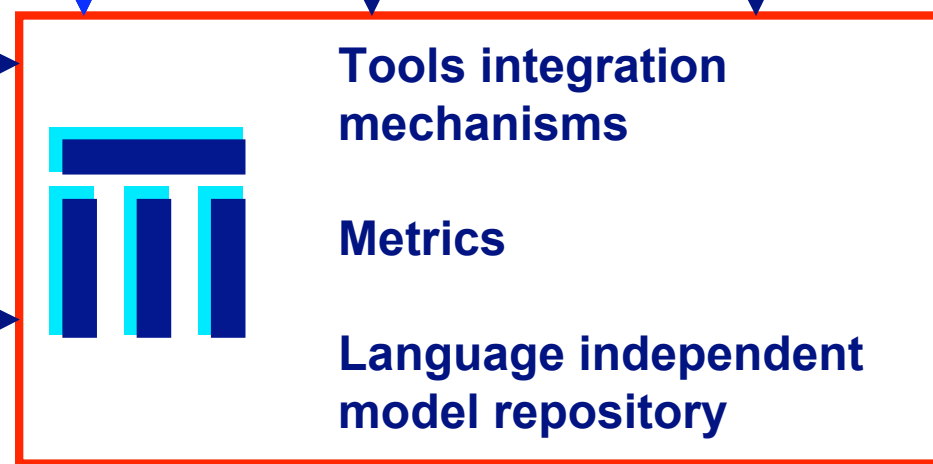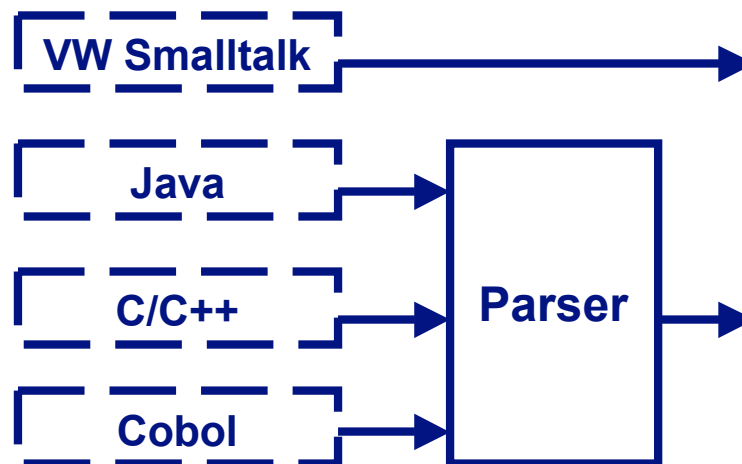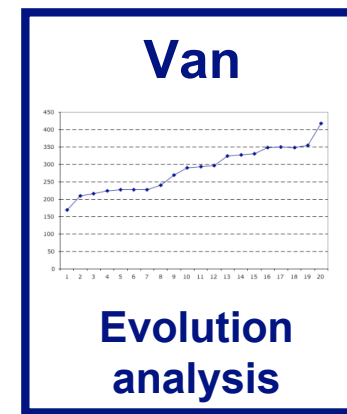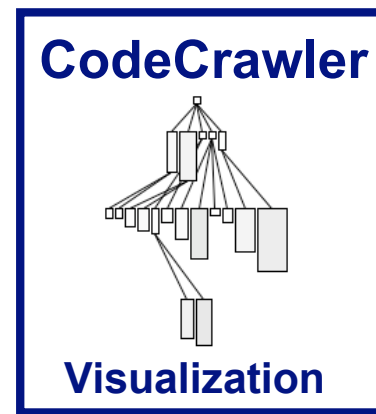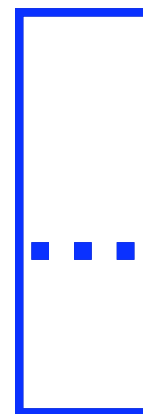
Moose is language independent through the FAMIX meta-model
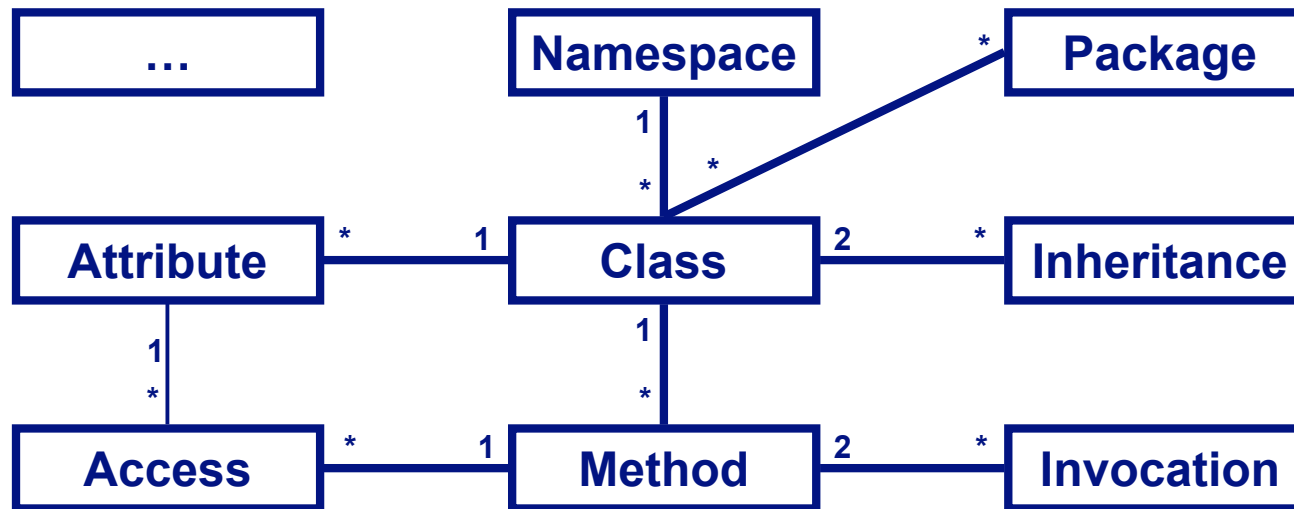
Java/C++ Parser: iPlasma (Loose Research Group)

Java Parser: jFamix Eclipse Plugin (http://jfamix.sourceforge.net/)

C++ Parser: Columbus (http://www.frontendart.com/products.html#Columbus)

Moose has been validated on real life systems written in different languages

System	Language	Lines of Code	Classes
Z	C++	1'200'000	~2300
Y	C++/Java	120'000	~400
X	C/C++	1'000'000	~11000
Jun	Smalltalk	135'000	~700
JBoss	Java	300'000	~4900
Squeak	Smalltalk	260'000	~1800
…	…	…	…

Summary: Moose is an extensible and collaborative reengineering environment
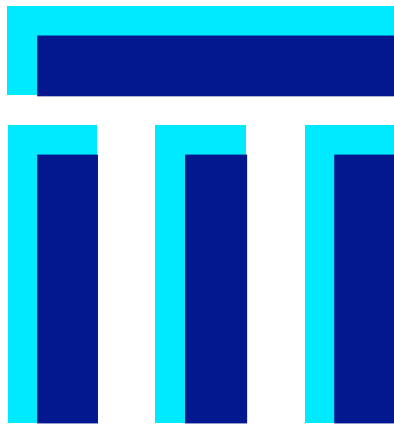
is research driven: metrics, visualization, evolution analysis …

is industrially validated

is free under BSD license

Accessibility:

www.iam.unibe.ch/~scg/Research/Moose/

VisualWorks distribution CD

Questions?

Ownership Map reveals how developers drive software evolution

file present from the first import

commit by the green author followed by the ownership

file removed by the blue author

File A

File B

file created by the green author

small commit by the blue author. the file is still ownedby the green author

Time