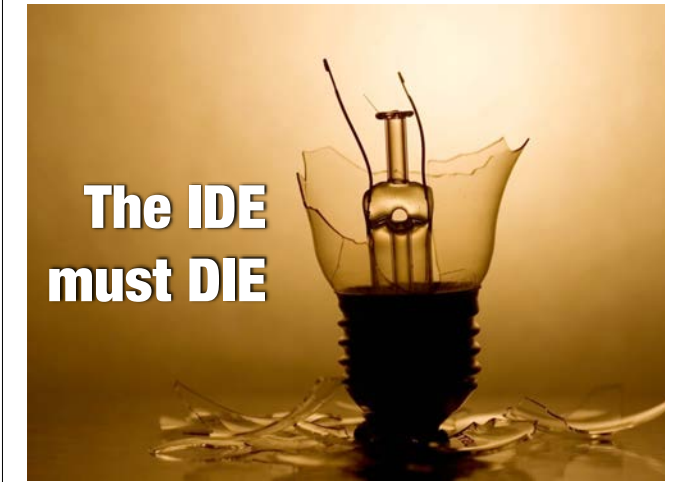


**It's time
for new
IDEas**



What is Software?



Programming is a kind of writing.

Gerald Weinberg

Psychology of Computer Programming

Dorset House, 1971

**PSYCHOLOGY
OF
COMPUTER
PROGRAMMING**
SILVER ANNIVERSARY
EDITION
GERALD M. WEINBERG



programming
is writing



programming
is writing

software
is text

```

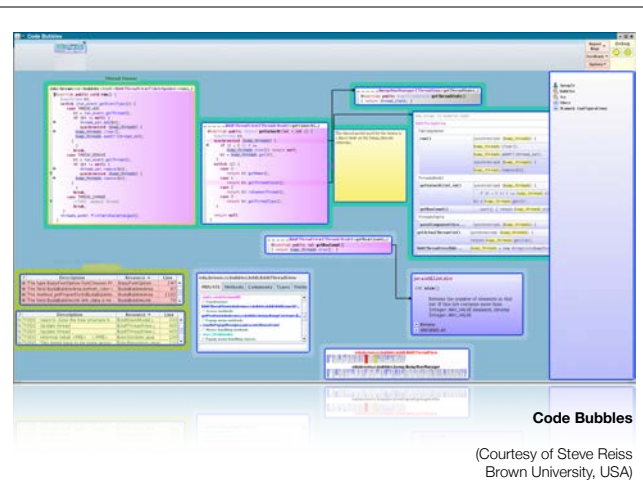
.....
/* A chess program similar to the IBM 704 (if non-Blank source), by R.D. Miller
.....
/* version 3.2 (2000 characters) features:
/* - recursive lookahead search
/* - evaluation search with responses
/* - response extraction
/* - iterative lookahead search
/* - best-move first sorting
/* - a hash table storing moves and best move
/* - full FIDE rules (except minor promotion) and move-legality checking
.....
#define W 16, H 8, F 10, I 10, O 10
#define W 16, H 8, F 10, I 10, O 10
.....
main()
{
    .....
}

```

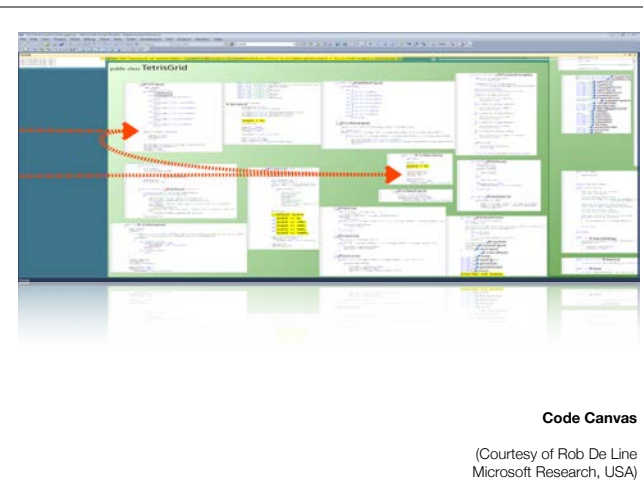
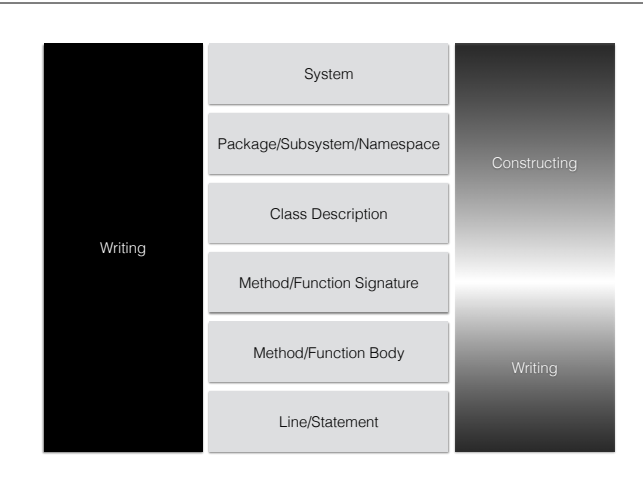
```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
.....
int main()
{
    .....
}

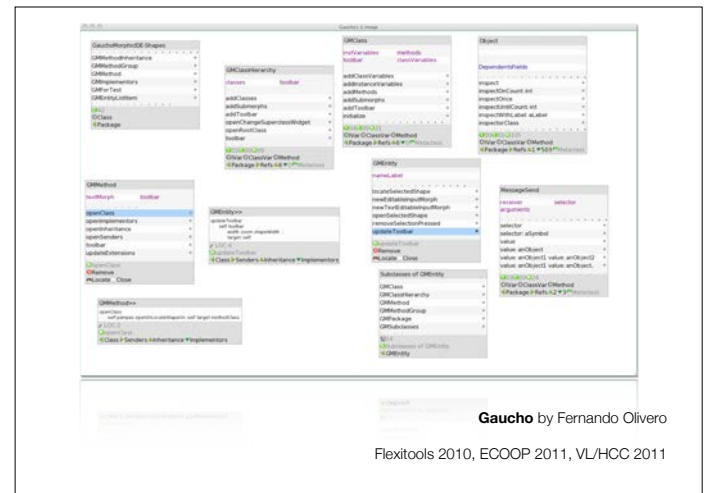
```



Code Bubbles
(Courtesy of Steve Reiss
Brown University, USA)



Code Canvas
(Courtesy of Rob De Line
Microsoft Research, USA)



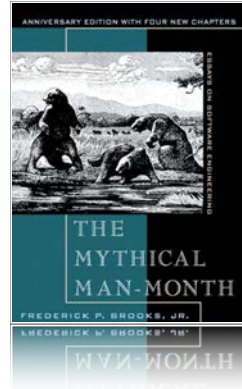
GauchO by Fernando Olivero
Flexitools 2010, ECOOP 2011, VL/HCC 2011

The programmer, like the poet, works only slightly removed from thought-stuff. He builds his castles in the air, from air, creating by exertion of the imagination.

Frederick P. Brooks, Jr.

The Mythical Man-Month

Addison-Wesley, 1975



ThrustSSC



110,000 BHP

1,227 km/h

PART II ASCENSION

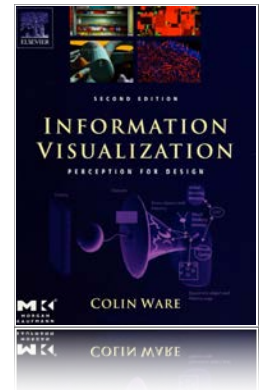


We acquire more information through vision than through all the other senses combined.

Colin Ware

Information Visualization Perception for Design

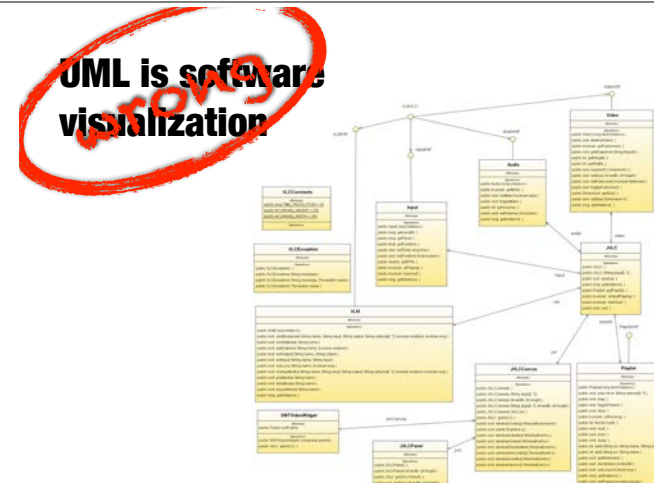
Morgan Kaufmann, 2004



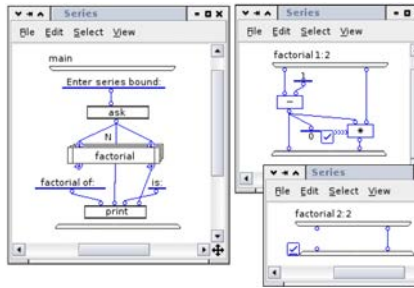
(software)
visualization myths



a picture is worth
a thousand words



visual programming is software visualization



static visualization

dynamic visualization

wrong

wrong

wrong

software visualization



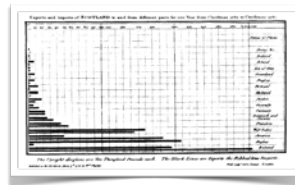
about software visualization

software visualization

information of software visualization

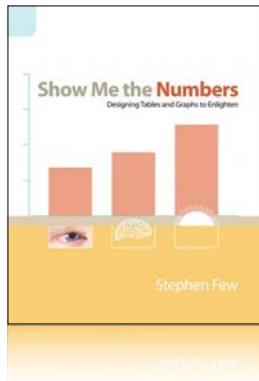
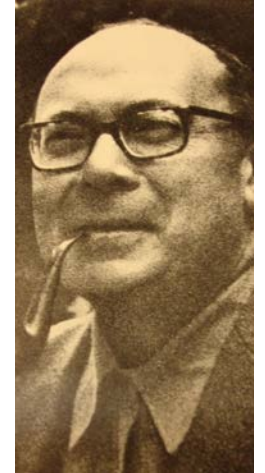
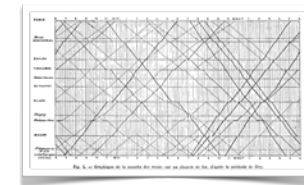
William Playfair

1759 - 1823



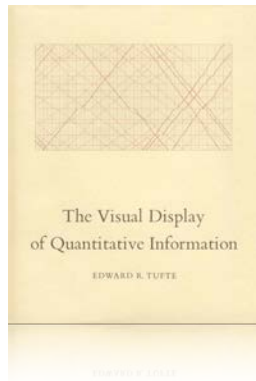
Jacques Bertin

Semiology of Graphics, 1967



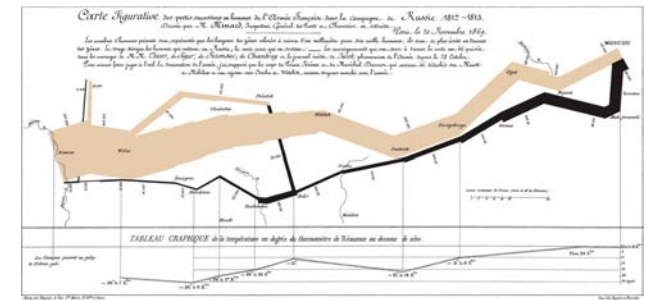
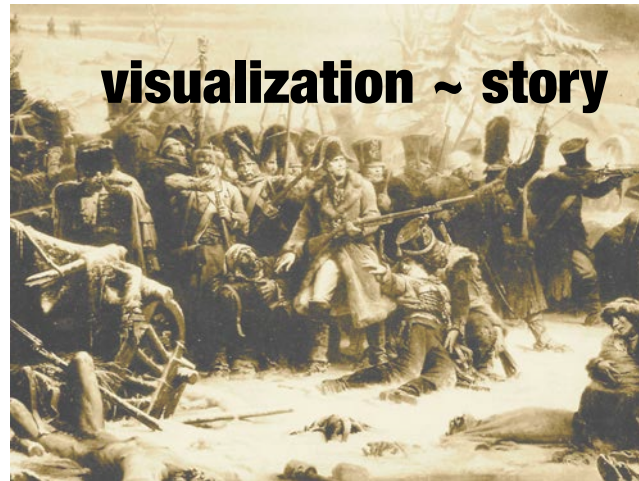
Stephen Few
Show Me the Numbers

Analytics Press, 2004



Edward R. Tufte
The Visual Display of Quantitative Information

Morgan Kaufmann, 2004



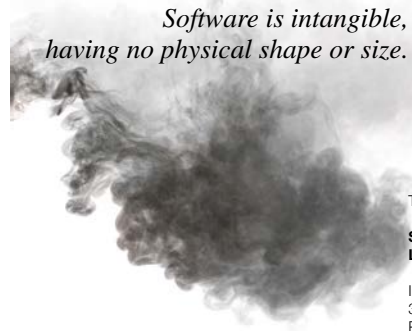
It was the twenty-eight of November. An immense confused mass of men, horses, vehicles besieged the narrow entrances to the bridges and began to flow over them.



Those in front, pushed by the weight of those behind were crushed, trampled on, or forced into the ice-filled water of the Berezina.

The confusion was so great that when Napoleon himself wished to cross, it was necessary to use force to clear a passage. Some there were who, determined to pass at all costs, cut a horrible way for themselves with their swords. Others opened an even crueller road for their carriages, driving them pitilessly through the helpless crowd, crushing men and women, in their odious greed sacrificing their companions in misery.

*Software is intangible,
having no physical shape or size.*



Thomas Ball, Stephen Eick

Software Visualization in the Large

In *Computer*, vol. 29, no. 4, pp. 33-43, IEEE Computer Society Press, 1996

Metaphors

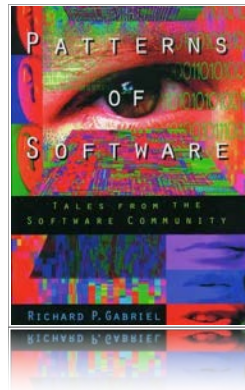


Habitability is the characteristic of source code that enables programmers [...] to understand its construction and intentions and to change it comfortably and confidently.

Richard P. Gabriel

Patterns of Software

Oxford University Press, 1998



Software Cities

Software systems as cities is a versatile metaphor which enables the creation of efficient software visualizations to support reverse engineering.

Richard Wetzel

Software Systems as Cities

PhD Thesis, University of Lugano, 2010

Software Systems as Cities

Richard Wetzel



The City Metaphor

package ~ district

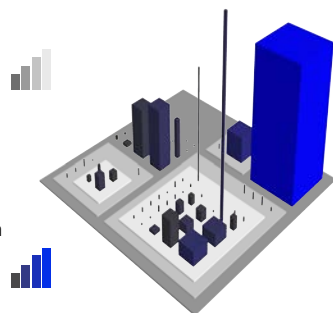
nesting level ~ color

class ~ building

methods (NOM) ~ height

attributes (NOA) ~ width, length

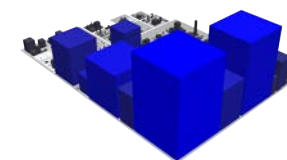
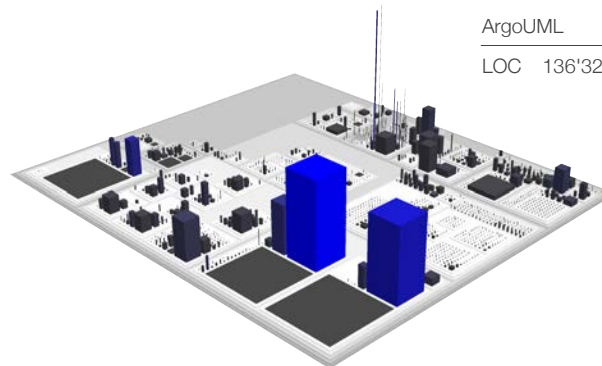
lines (LOC) ~ color



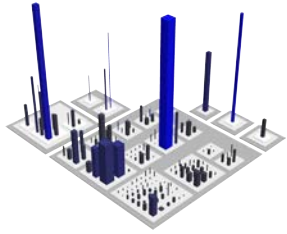
Program Comprehension

ArgoUML

LOC 136'325

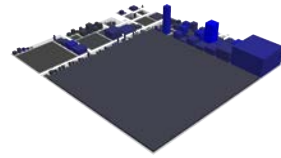


System	Language	NOP	NOC	kLOC
ScummVM	C++	141	3'117	305



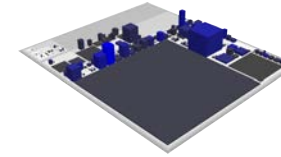
System	Language	NOP	NOC	kLOC
ScummVM	C++	141	3'117	305
CodeCity	Smalltalk	34	173	29

iTextSharp	C#	22	485	58
iText	Java	36	566	59



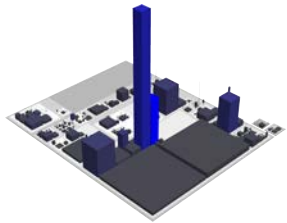
System	Language	NOP	NOC	kLOC
ScummVM	C++	141	3'117	305
CodeCity	Smalltalk	34	173	29

iTextSharp	C#	22	485	58
------------	----	----	-----	----



System	Language	NOP	NOC	kLOC
ScummVM	C++	141	3'117	305
CodeCity	Smalltalk	34	173	29

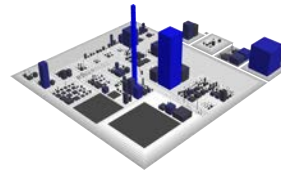
iTextSharp	C#	22	485	58
iText	Java	36	566	59



System	Language	NOP	NOC	kLOC
ScummVM	C++	141	3'117	305
CodeCity	Smalltalk	34	173	29

iTextSharp	C#	22	485	58
iText	Java	36	566	59

Jmol	Java	50	558	85
------	------	----	-----	----

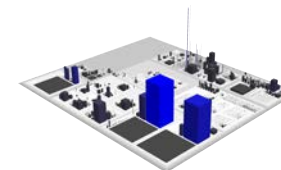


System	Language	NOP	NOC	kLOC
ScummVM	C++	141	3'117	305
CodeCity	Smalltalk	34	173	29

iTextSharp	C#	22	485	58
iText	Java	36	566	59

Jmol	Java	50	558	85
------	------	----	-----	----

jEdit	Java	59	966	98
-------	------	----	-----	----



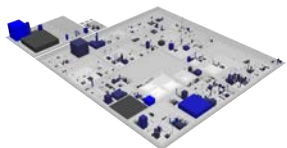
System	Language	NOP	NOC	kLOC
ScummVM	C++	141	3'117	305
CodeCity	Smalltalk	34	173	29

iTextSharp	C#	22	485	58
iText	Java	36	566	59

Jmol	Java	50	558	85
------	------	----	-----	----

jEdit	Java	59	966	98
-------	------	----	-----	----

ArgoUML	Java	88	1'817	144
---------	------	----	-------	-----



System	Language	NOP	NOC	kLOC
ScummVM	C++	141	3'117	305
CodeCity	Smalltalk	34	173	29

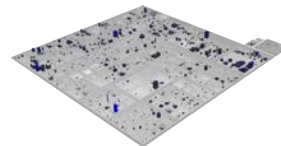
iTextSharp	C#	22	485	58
iText	Java	36	566	59

Jmol	Java	50	558	85
------	------	----	-----	----

jEdit	Java	59	966	98
-------	------	----	-----	----

ArgoUML	Java	88	1'817	144
---------	------	----	-------	-----

GWT	Java	302	4'372	212
-----	------	-----	-------	-----



System	Language	NOP	NOC	kLOC
ScummVM	C++	141	3'117	305
CodeCity	Smalltalk	34	173	29

iTextSharp	C#	22	485	58
iText	Java	36	566	59

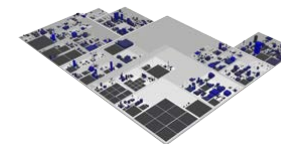
Jmol	Java	50	558	85
------	------	----	-----	----

jEdit	Java	59	966	98
-------	------	----	-----	----

ArgoUML	Java	88	1'817	144
---------	------	----	-------	-----

GWT	Java	302	4'372	212
-----	------	-----	-------	-----

JBoss	Java	1'507	7'881	435
-------	------	-------	-------	-----



System	Language	NOP	NOC	kLOC
ScummVM	C++	141	3'117	305
CodeCity	Smalltalk	34	173	29

iTextSharp	C#	22	485	58
iText	Java	36	566	59

Jmol	Java	50	558	85
------	------	----	-----	----

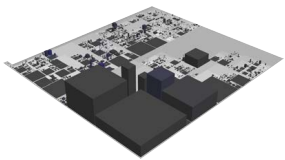
jEdit	Java	59	966	98
-------	------	----	-----	----

ArgoUML	Java	88	1'817	144
---------	------	----	-------	-----

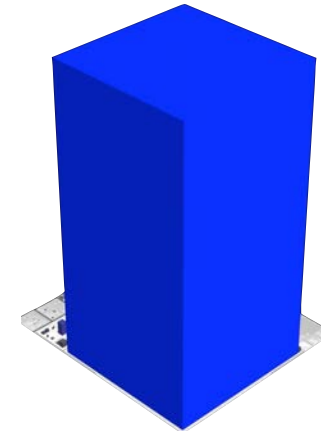
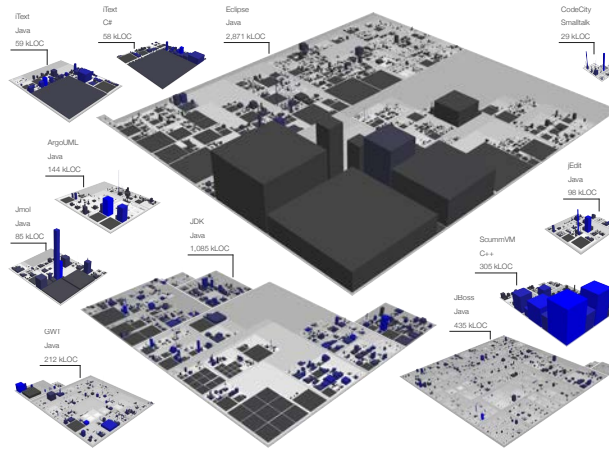
GWT	Java	302	4'372	212
-----	------	-----	-------	-----

JBoss	Java	1'507	7'881	435
-------	------	-------	-------	-----

JDK 1.5	Java	664	12'888	1'085
---------	------	-----	--------	-------

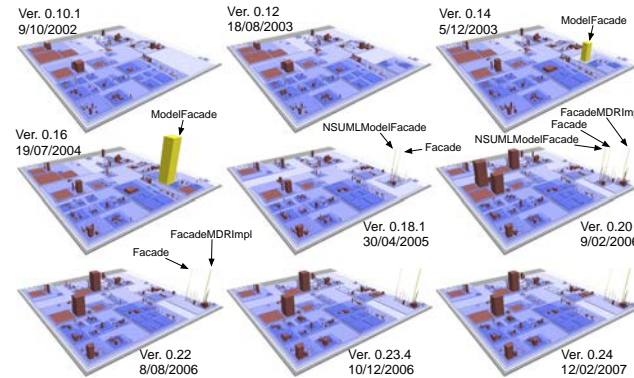


System	Language	NOP	NOC	kLOC
ScummVM	C++	141	3'117	305
CodeCity	Smalltalk	34	173	29
iTextSharp	C#	22	485	58
iText	Java	36	566	59
Jmol	Java	50	558	85
jEdit	Java	59	966	98
ArgoUML	Java	88	1'817	144
GWT	Java	302	4'372	212
JBoss	Java	1'507	7'881	435
JDK 1.5	Java	664	12'888	1'085
Eclipse	Java	1'800	27'900	2'871

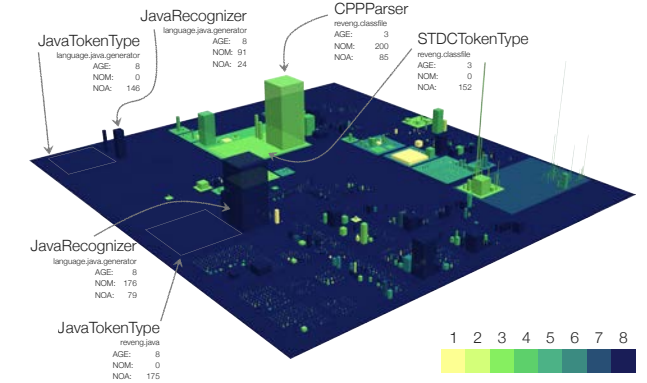


System	Language	NOP	NOC	kLOC
ScummVM	C++	141	3'117	305
CodeCity	Smalltalk	34	173	29
iTextSharp	C#	22	485	58
iText	Java	36	566	59
Jmol	Java	50	558	85
jEdit	Java	59	966	98
ArgoUML	Java	88	1'817	144
GWT	Java	302	4'372	212
JBoss	Java	1'507	7'881	435
JDK 1.5	Java	664	12'888	1'085
Eclipse	Java	1'800	27'900	2'871
OpenSwing Java		268	1'754	112

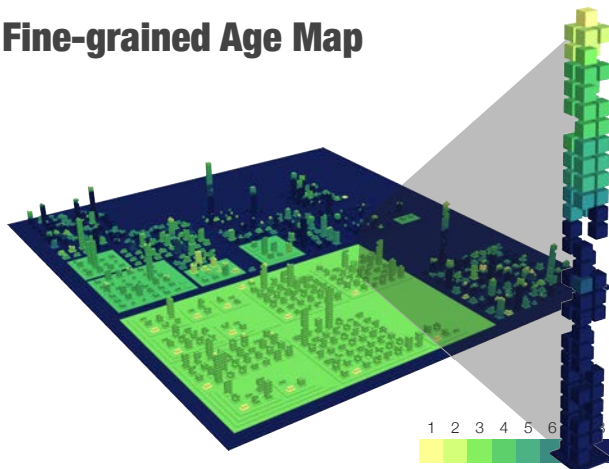
Software Evolution Analysis



Coarse-grained Age Map



Fine-grained Age Map



age: 1 2 3 4 5 6 7 8

stable



very old

rarely updated



old

highly unstable



young

updated often,
rather unstable

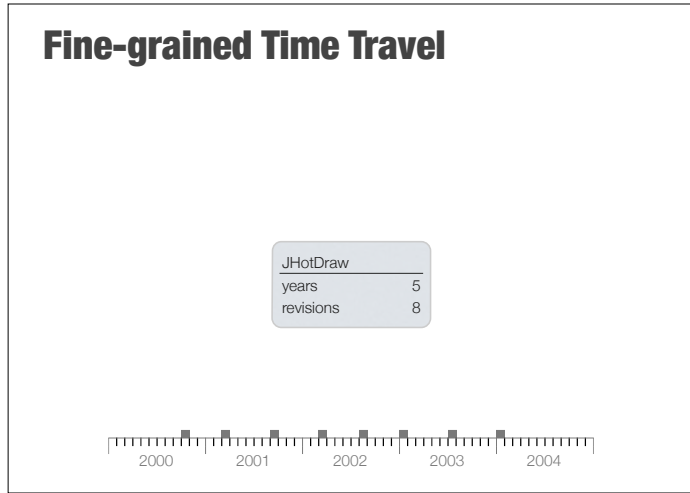
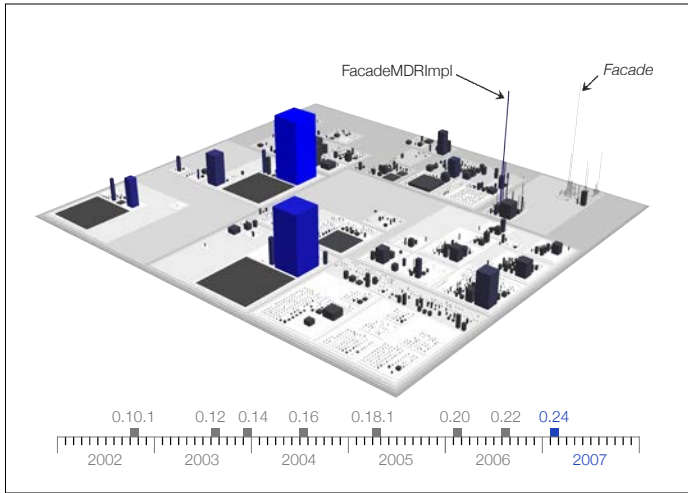
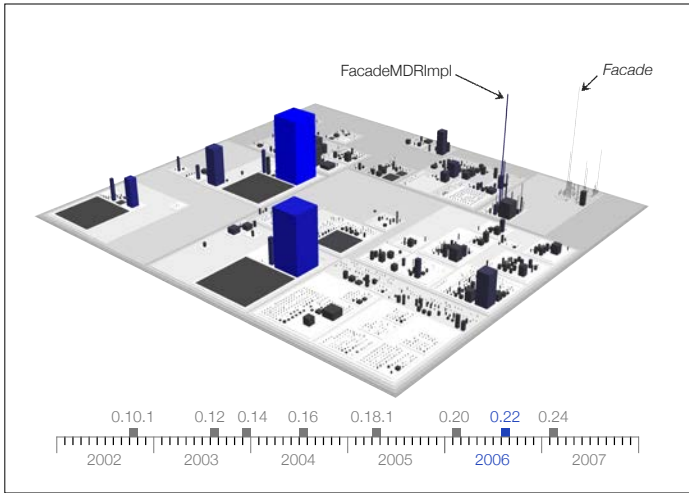
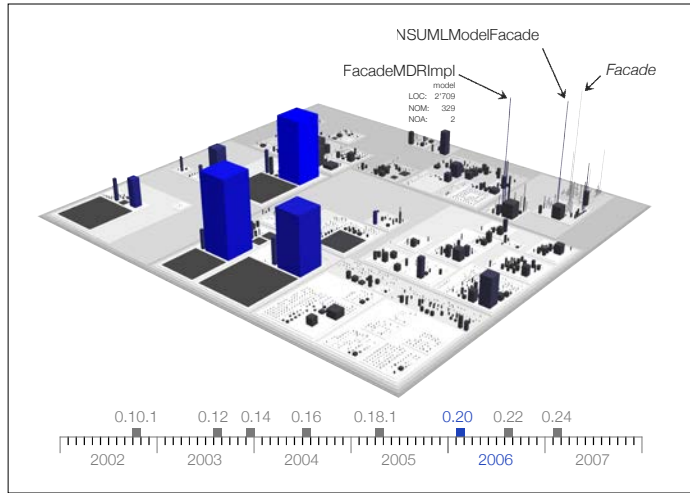
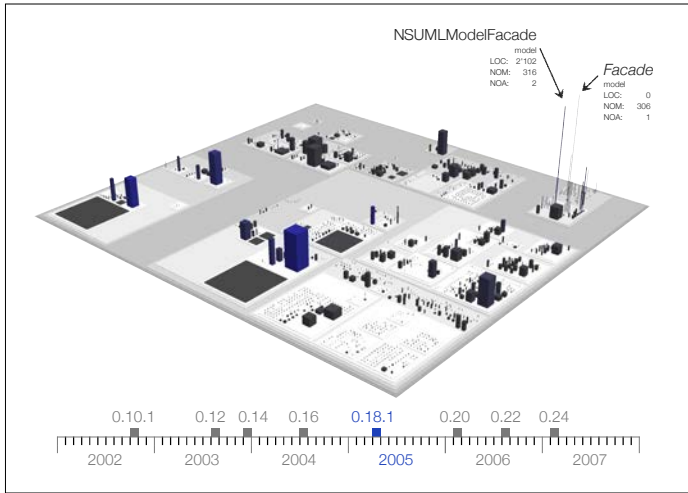
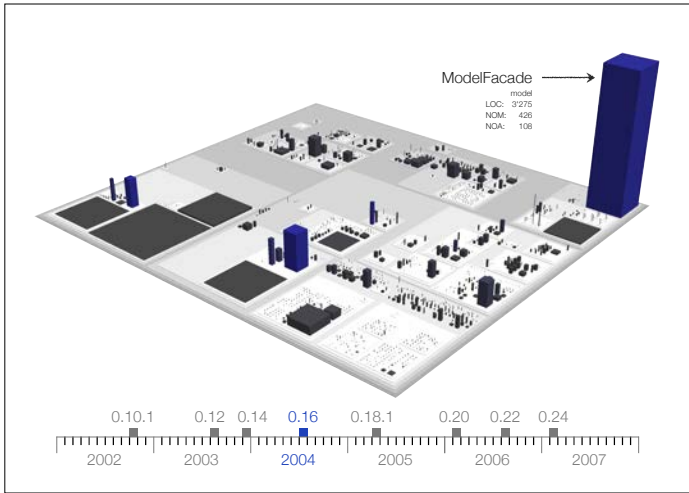
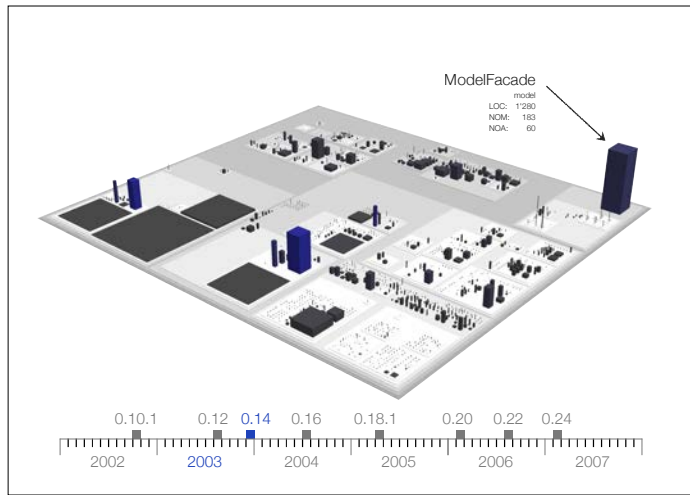
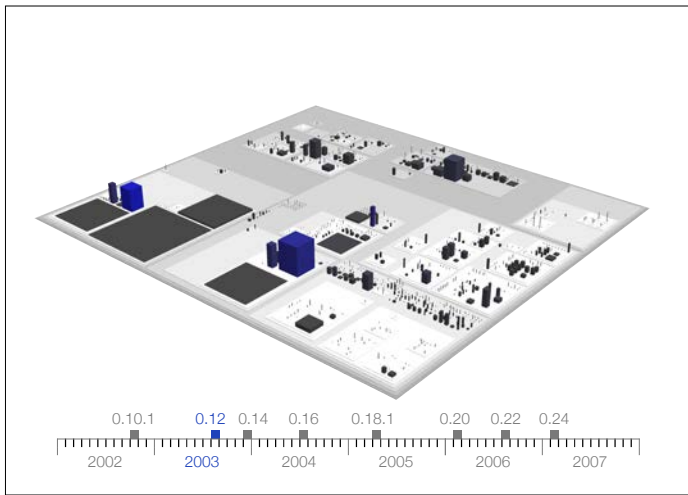
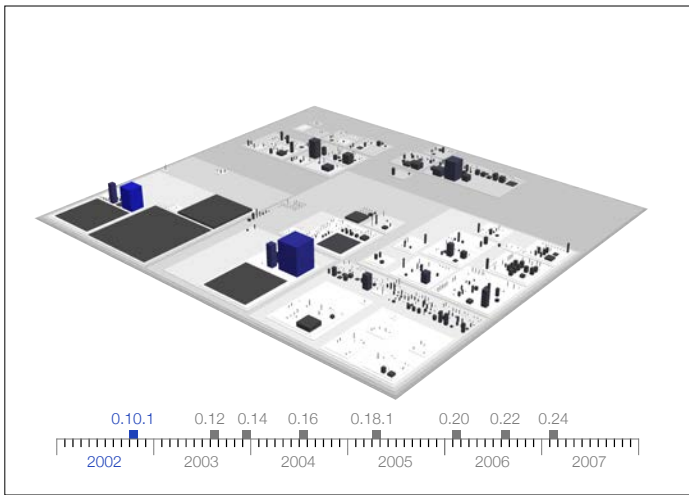


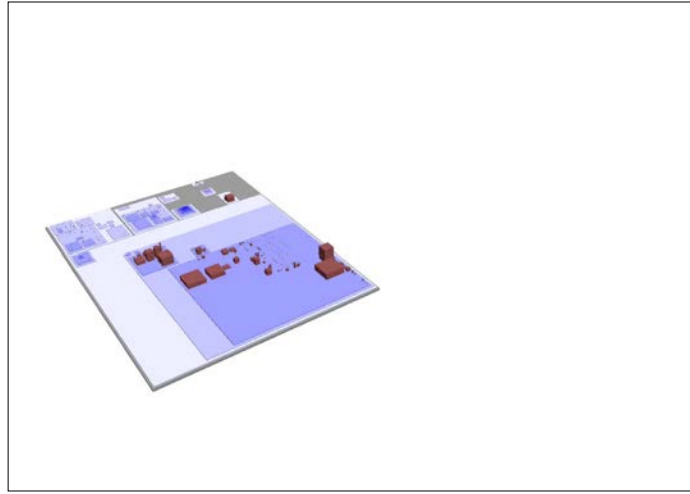
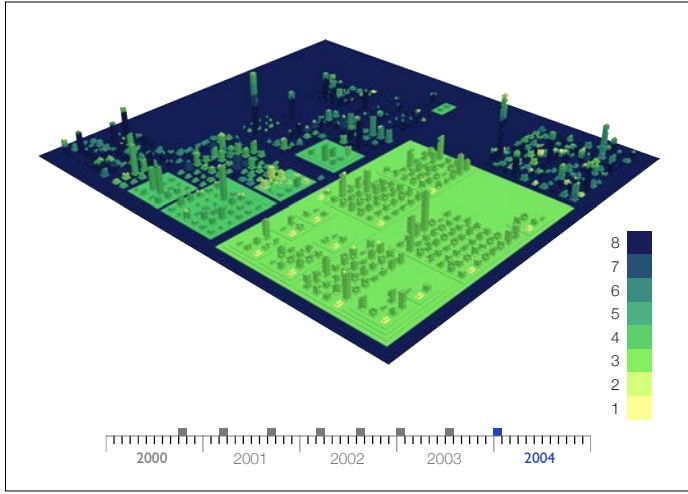
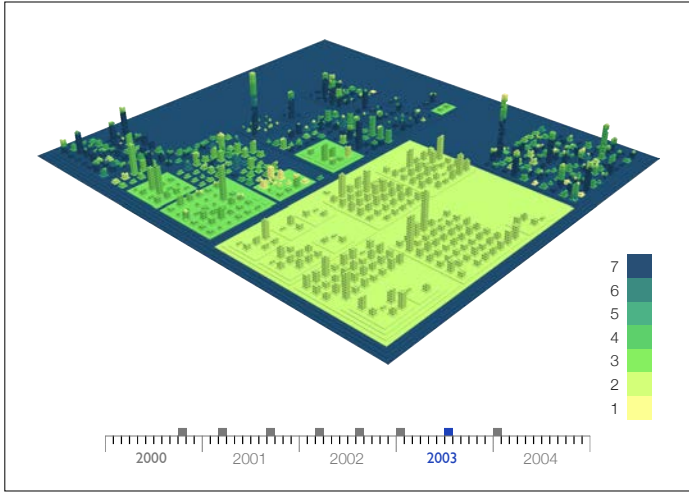
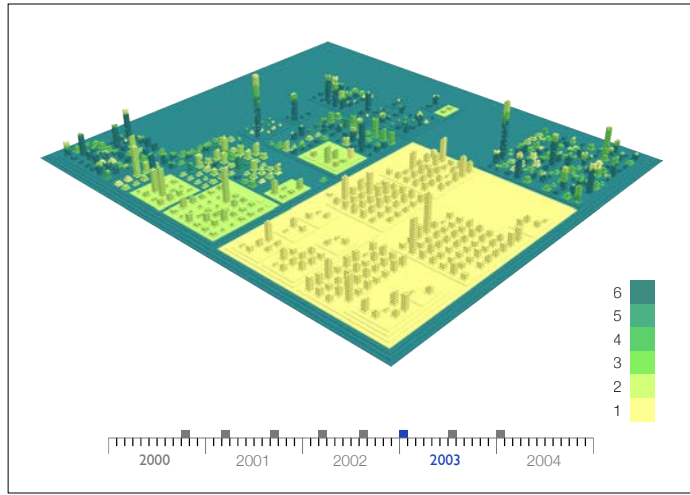
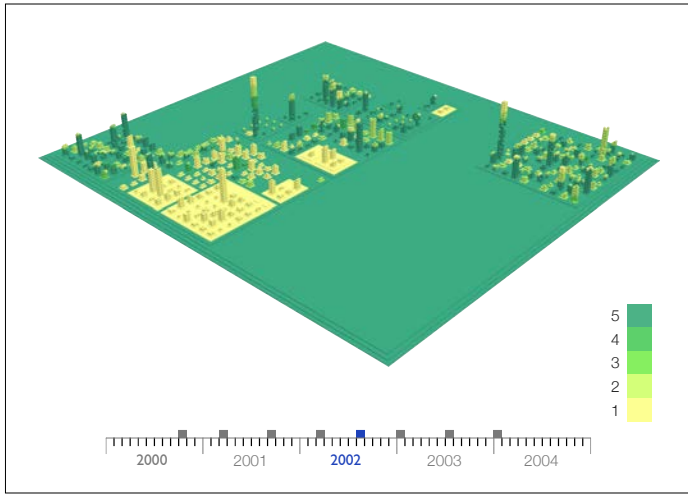
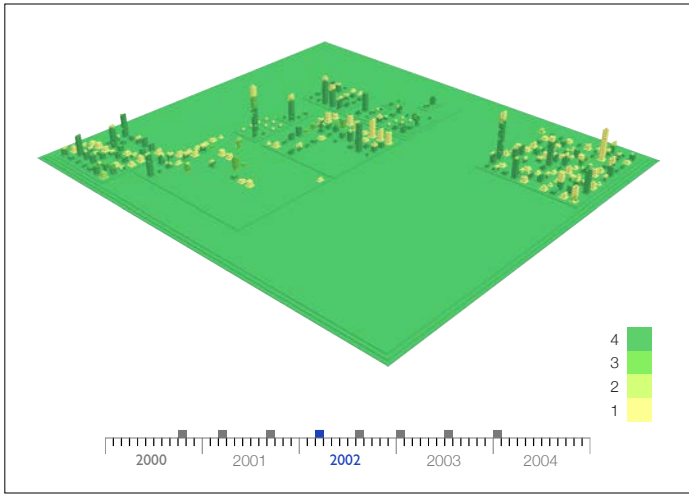
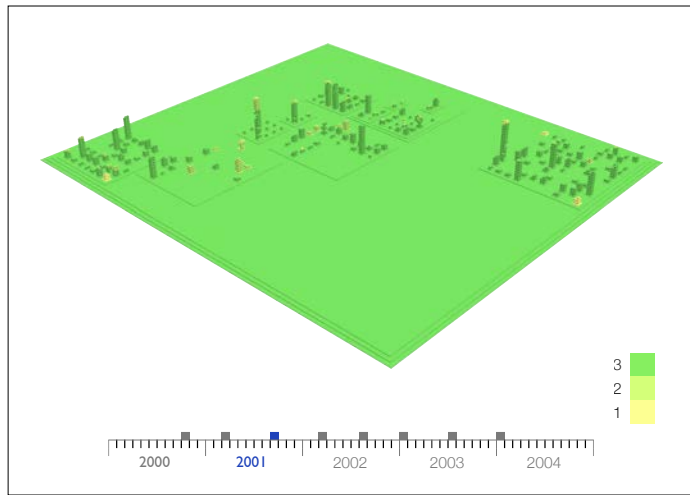
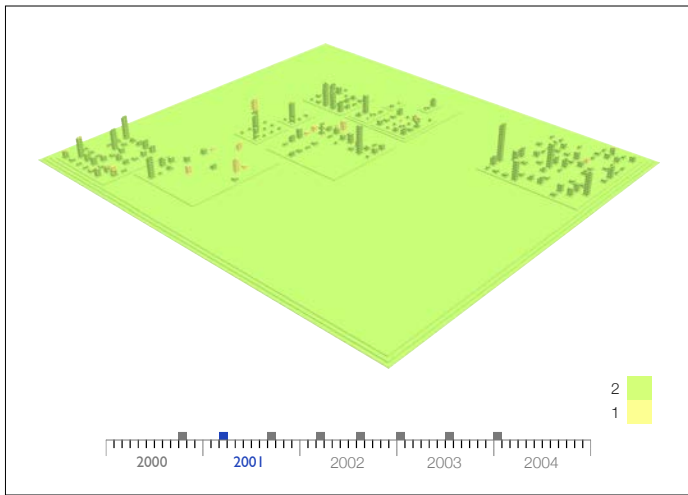
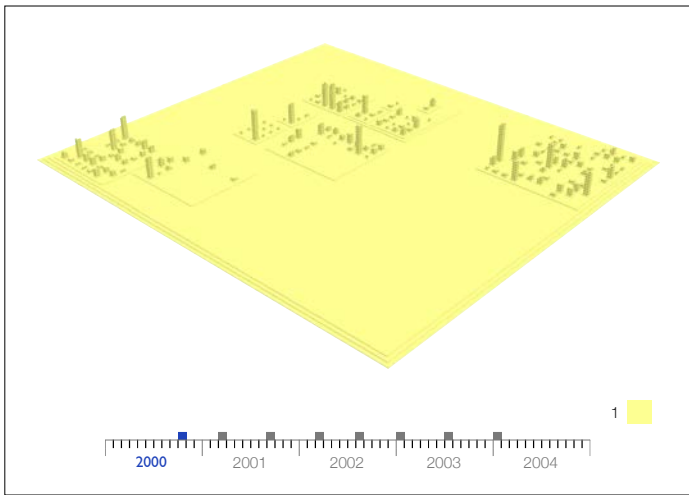
very old

Coarse-grained Time Travel

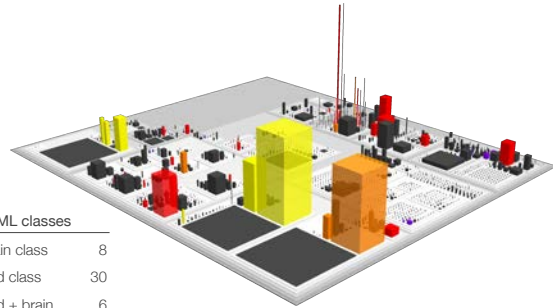
ArgoUML	years
years	6
major releases	8







Problem Detection & Analysis



ArgoUML classes

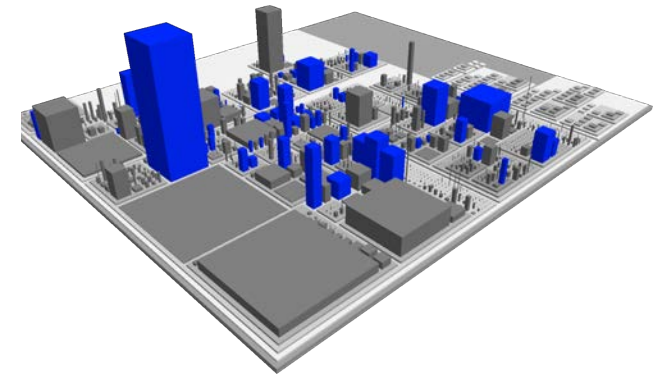
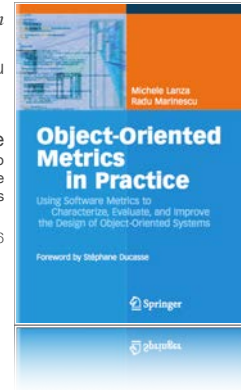
brain class	8
god class	30
god + brain	6
data class	17
unaffected	1'715

Metrics? Don't trust them

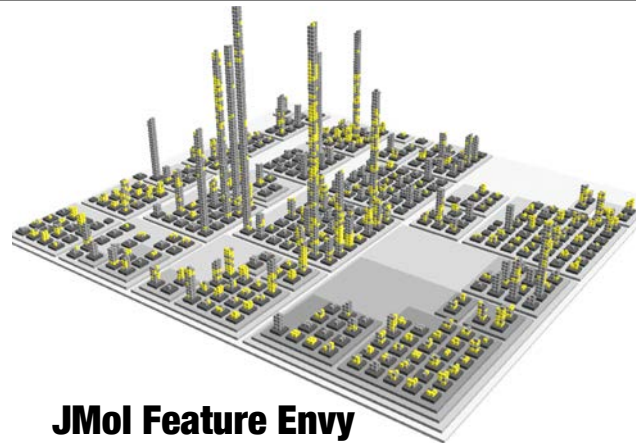
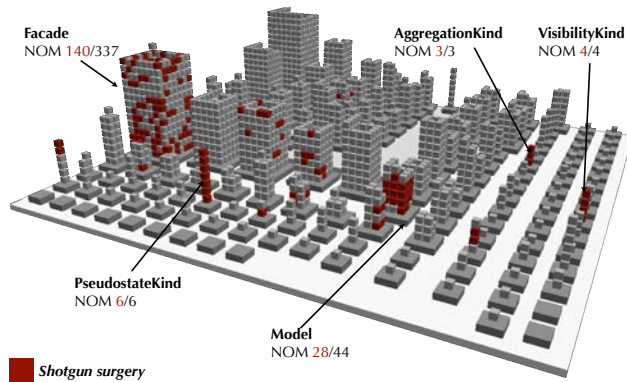
Michele Lanza & Radu Marinescu

Object-Oriented Metrics in Practice
Using Software Metrics to
Characterize, Evaluate, and Improve
the Design of Object-Oriented Systems

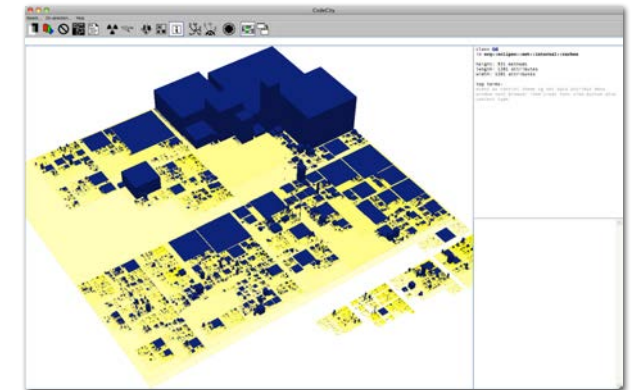
Springer, 2006



JDK 1.5 god classes

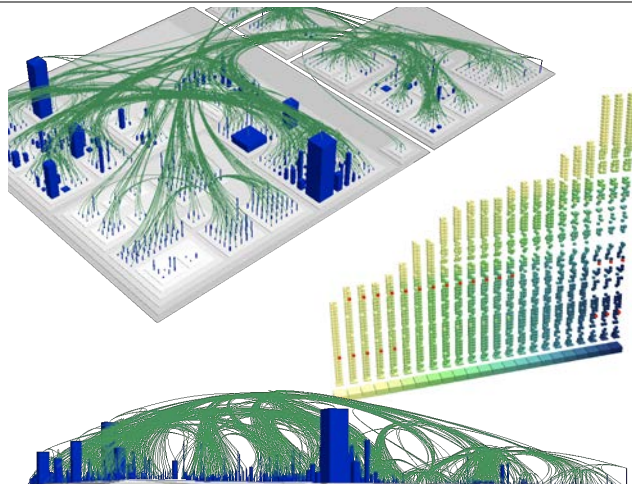


JMoI Feature Envy



CodeCity

codacity.inf.usi.ch



"I am not convinced I would learn anything from the visualizations"

**ICSE
REJECTED**



Knock me down, I'll just come back running

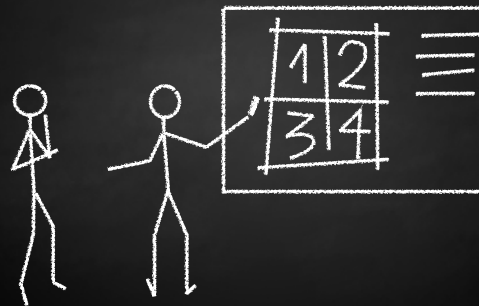
In the Name of Science



Experiment Design Desiderata

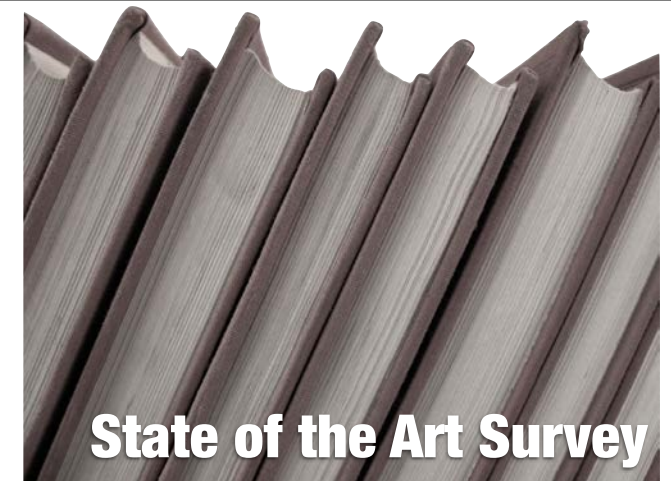
- 1 Avoid comparing using a technique against not using it.
- 2 Involve participants from the industry.
- 3 Provide a not-so-short tutorial of the experimental tool to the participants.
- 4 Avoid, whenever possible, giving the tutorial right before the experiment.
- 5 Use the tutorial to cover both the research behind the approach and the tool.
- 6 Find a set of relevant tasks.
- 7 Choose real object systems that are relevant for the tasks.
- 8 Include more than one object system in the design.
- 9 Provide the same data to all participants.
- 10 Limit the amount of time allowed for solving each task.
- 11 Provide all the details needed to make the experiment replicable.
- 12 Report results on individual tasks.
- 13 Include tasks whose expected result is not to the advantage of the tool being evaluated.
- 14 Take into account the possible wide range of experience level of the participants.

Design



Experiment Design Desiderata

- 1 Avoid comparing using a technique against not using it.
- 2 Involve participants from the industry.
- 3 Provide a not-so-short tutorial of the experimental tool to the participants.
- 4 Avoid, whenever possible, giving the tutorial right before the experiment.
- 5 Use the tutorial to cover both the research behind the approach and the tool.
- 6 Find a set of relevant tasks.
- 7 Choose real object systems that are relevant for the tasks.
- 8 Include more than one object system in the design.
- 9 Provide the same data to all participants.
- 10 Limit the amount of time allowed for solving each task.
- 11 Provide all the details needed to make the experiment replicable.
- 12 Report results on individual tasks.
- 13 Include tasks whose expected result is not to the advantage of the tool being evaluated.
- 14 Take into account the possible wide range of experience level of the participants.



State of the Art Survey

Finding a Baseline



1. program comprehension
2. design quality assessment
3. system evolution analysis

Research Questions

1

Does the use of CodeCity increase the correctness of the solutions to program comprehension tasks, compared to non-visual exploration tool, regardless of the object system size?

Research Questions

1

Does the use of CodeCity increase the correctness of the solutions to program comprehension tasks, compared to non-visual exploration tool, regardless of the object system size?

2

Does the use of CodeCity reduce the time needed to solve program comprehension tasks, compared to non-visual exploration tools, regardless of the object system size?

Tasks

- A1 Identify the convention used in the system to organize unit tests.
- A2.1& A2.2 What is the spread of term T in the name of the classes, their attributes and methods?
- A3 Evaluate the change impact of class C, in terms of intensity and dispersion.
- A4.1 Find the three classes with the highest number of methods.
- A4.2 Find the three classes with the highest average number of lines of code per method.
- B1.1 Identify the package with the highest percentage of god classes.
- B1.2 Identify the god class with the largest number of methods.
- B2.1 Identify the dominant (affecting the highest number of classes) class-level design problem.
- B2.2 Write an overview of the class-level design problems in the system.

Tasks

Program Comprehension

6

- A1 Identify the convention used in the system to organize unit tests.
- A2.1& A2.2 What is the spread of term T in the name of the classes, their attributes and methods?
- A3 Evaluate the change impact of class C, in terms of intensity and dispersion.
- A4.1 Find the three classes with the highest number of methods.
- A4.2 Find the three classes with the highest average number of lines of code per method.
- B1.1 Identify the package with the highest percentage of god classes.
- B1.2 Identify the god class with the largest number of methods.
- B2.1 Identify the dominant (affecting the highest number of classes) class-level design problem.
- B2.2 Write an overview of the class-level design problems in the system.

Design Quality Assessment

4

Tasks

Quantitative

9

- A1 Identify the convention used in the system to organize unit tests.
- A2.1& A2.2 What is the spread of term T in the name of the classes, their attributes and methods?
- A3 Evaluate the change impact of class C, in terms of intensity and dispersion.
- A4.1 Find the three classes with the highest number of methods.
- A4.2 Find the three classes with the highest average number of lines of code per method.
- B1.1 Identify the package with the highest percentage of god classes.
- B1.2 Identify the god class with the largest number of methods.
- B2.1 Identify the dominant (affecting the highest number of classes) class-level design problem.
- B2.2 Write an overview of the class-level design problems in the system.

Qualitative

1

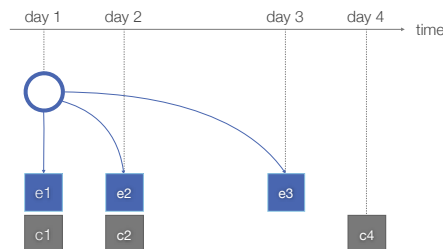
Execution



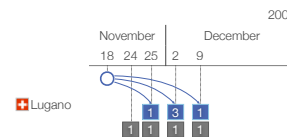
Experiment Runs

training session
(1 hour)

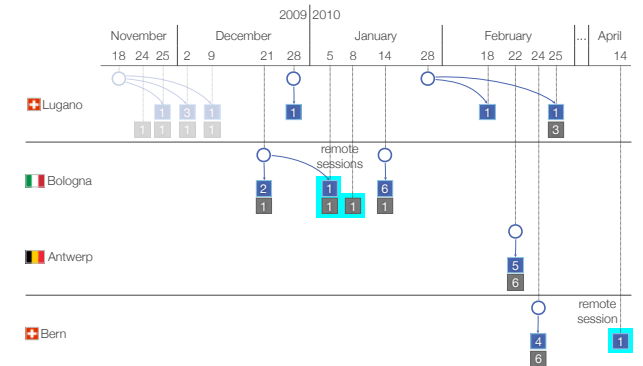
experiment session
(2 hours)



Testing the Waters



Experiment Timeline



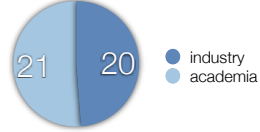


41 Subjects



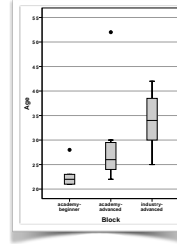
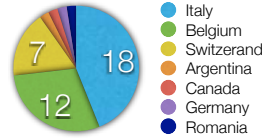
Subjects

Background



- developer
- software engineer
- system analyst
- software architect
- project manager
- IT lead
- CTO
- consultant
- professor
- Ph.D. student
- Master student

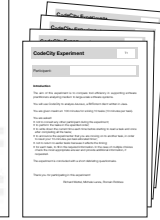
Country of origin



On Replicability



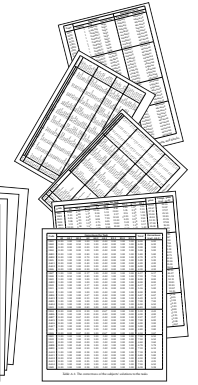
Pre-experiment questionnaire



Experiment handouts

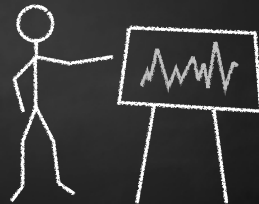


Task solutions & grading schemes



Data set

Results

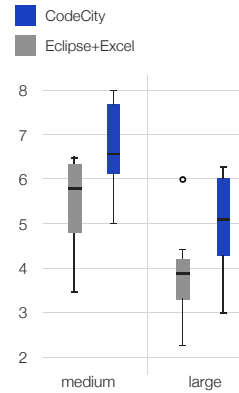


Correctness

24.26%

more correct with CodeCity

ANOVA (two-way analysis of variance)
 statistically significant
 95% confidence interval
 large effect size (d=0.89)

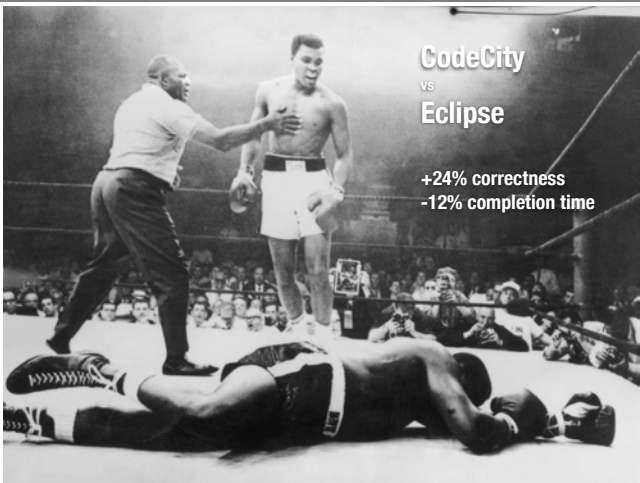
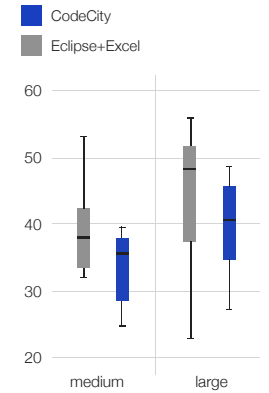


Completion Time

12.01%

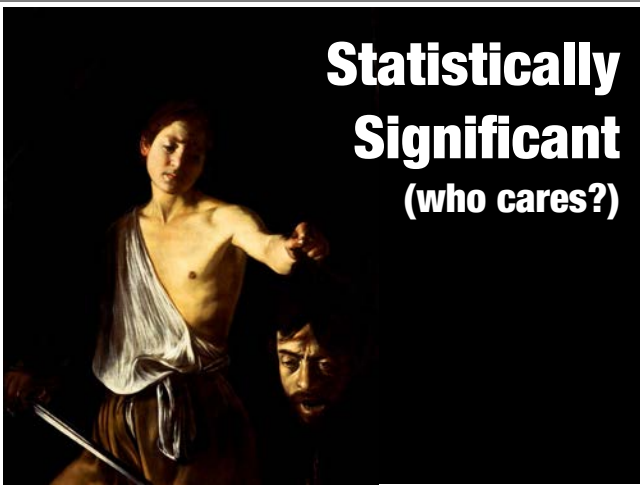
faster with CodeCity

ANOVA (two-way analysis of variance)
 statistically significant
 95% confidence interval
 moderate effect size (d=0.63)



Statistically Significant!!

Statistically Significant..



Richard Wetzel, Michele Lanza,
Romain Robbes

**Software Systems as Cities:
A Controlled Experiment**

In Proceedings of ICSE 2011
(33rd International Conference on Software Engineering)
pp. 551 - 560. IEEE CS Press, 2011

Software Systems as Cities: A Controlled Experiment
Richard Wetzel and Michele Lanza
Wetzel@cs.uni-saarland.de
lanza@cs.uni-saarland.de

Romain Robbes
Robbes@cs.uni-saarland.de

ABSTRACT
This paper reports on an experiment that aims at understanding the impact of architectural decisions on the development of software systems. The experiment is conducted in a controlled environment, where the participants are given a set of tasks to perform. The results of the experiment are analyzed and compared with the results of a previous experiment. The experiment shows that the participants who used a structured approach to software development were able to complete their tasks more efficiently and with fewer errors than the participants who used an unstructured approach. This suggests that structured software development is more effective than unstructured software development.

Categories and Subject Descriptors
D.2.2 [Software Engineering]: Design Tools and Techniques; D.2.4 [Software Engineering]: Programming Languages; D.2.7 [Software Engineering]: Testing and Debugging

General Terms
Design, Experimentation, Performance, Reliability, Testing

KEYWORDS
Software development, controlled experiment, architectural decisions, software systems, cities, structured vs. unstructured development





Impact

5,143 citations

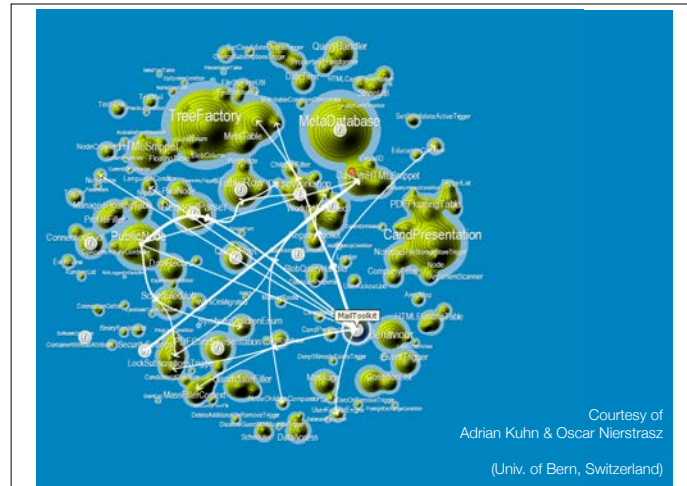
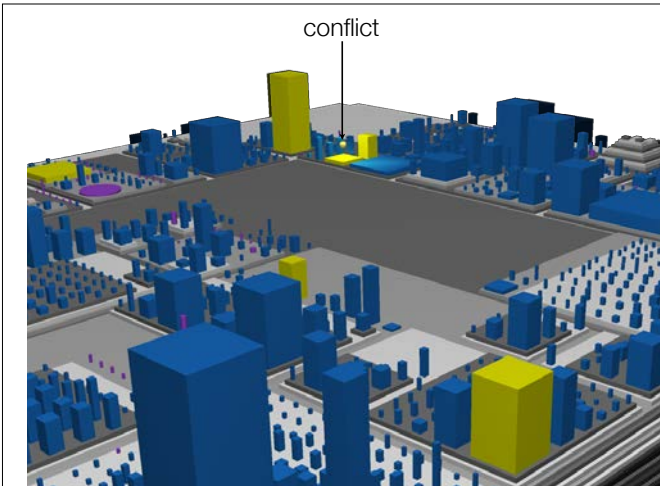
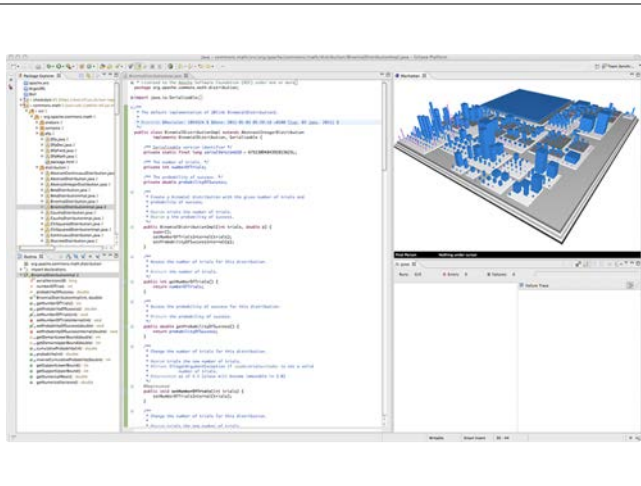
5,143 citations
BULLST**
n-index 37



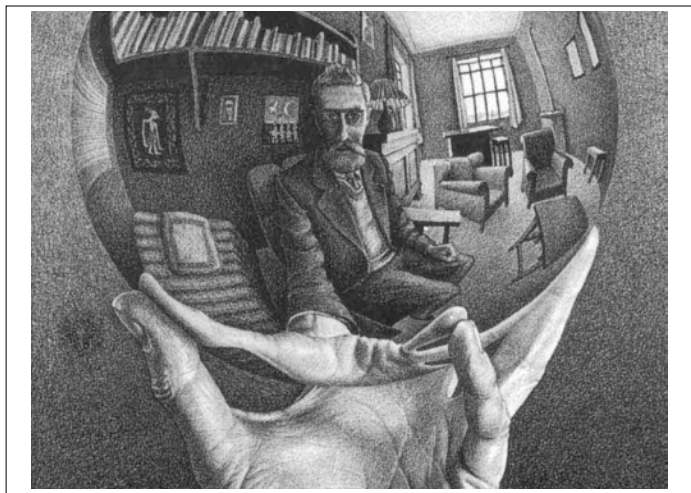
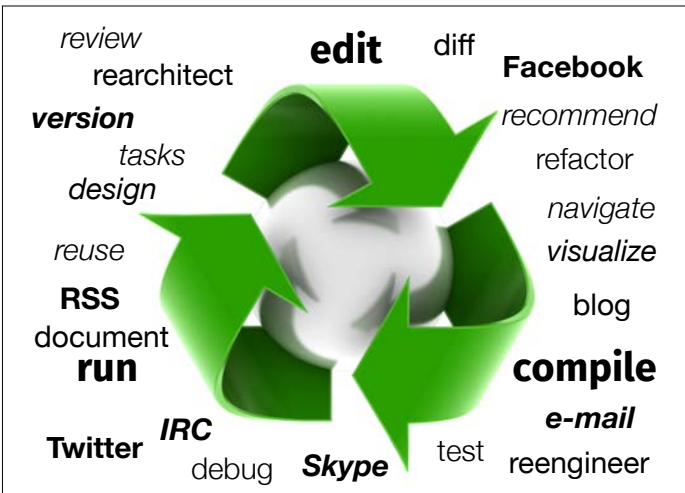
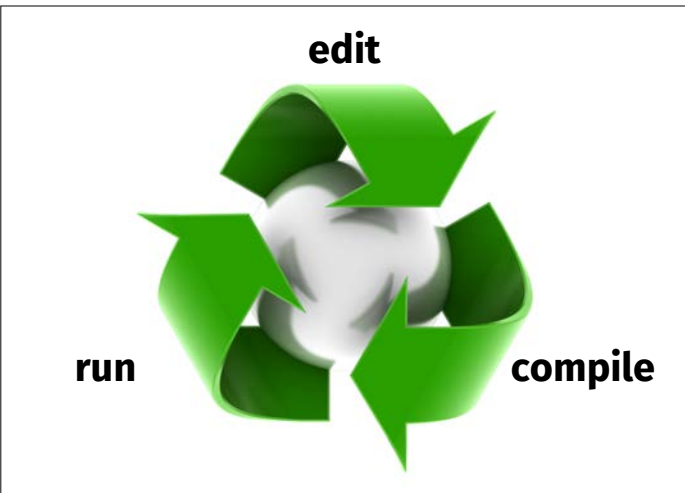
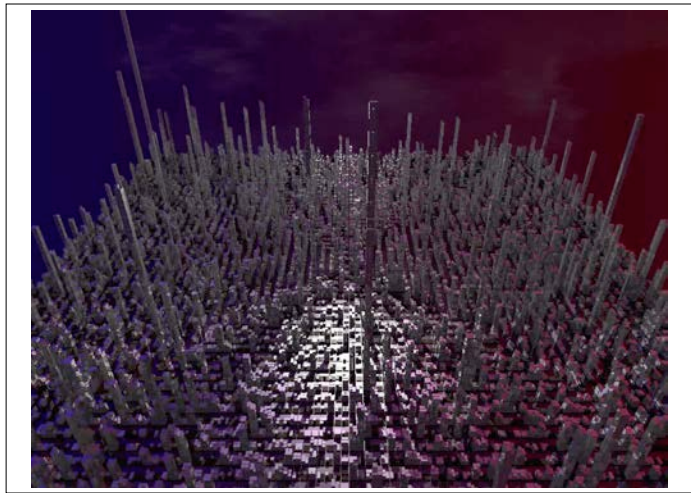
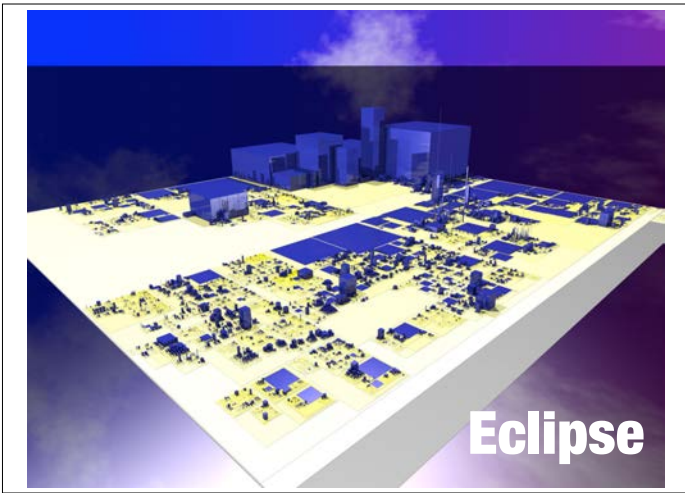
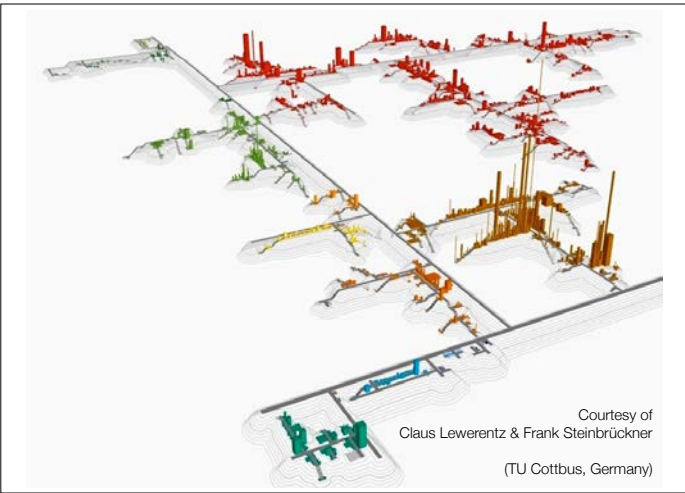
Academics @ Work

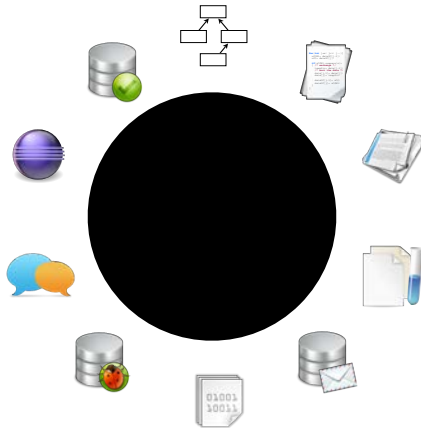
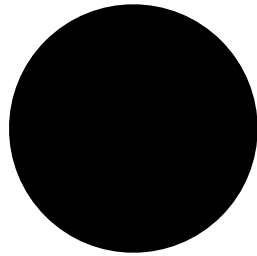


Manhattan
manhattan.inf.usi.ch



Courtesy of
Adrian Kuhn & Oscar Nierstrasz
(Univ. of Bern, Switzerland)





Issue Details

Key: LUCENE-2513
 Type: Bug
 Status: Open
 Priority: Major
 Assignee: Michael McCandless
 Reporter: Michael McCandless
 Votes: 0
 Watchers: 0

Lucene-Java
 IndexReader overwrites future commits when you open it on a past commit
 Created: Thursday, 10:27 AM. Updated: Thursday, 10:38 AM.
 Components: INDEX
 Affects Versions: None
 Fix Versions: 3.1-4.0

File Name	Date Attached	Attached By	Size
LUCENE-2513.pdf	2010-06-24 10:38 AM	Michael McCandless	6 KB

Lucene Fields: New

Description

Hi this is trying to build up a test index for perftesting...
 IndexReader (and Writer) accept an IndexCommit to open.
 This is quite powerful, because, if you use a deletion policy that keeps multiple commits around, you can open a not-current commit, make some changes, write a new commit, all without altering the "future" commits.
 I use this to first build up a big wikipedia index, including one commit at multiple segments, then another commit after optimization, and then I open an writer to perform deletions of both those commits. This gives me a single test index that has all four combinations (single vs multi segment, deletions vs no deletions).
 But IndexReader has a bug whereby it overwrites the segments_N file. (IndexWriter works correctly).

from Tudor Cirba <tudor.girba@gmail.com> 4/20/10 8:55 AM
 subject: [Moose-dev] Re: glamour tree children by level
 to: Related to the development of Moose and other related tools <moose-dev@lam.unlbe.ch>

Hi Lukas,
 On 19 April 2010, at 10:52, Lukas Renggli wrote:
 Very cool indeed, I can use that.
 What do I do if I have different kinds of objects per level?
 You have to take care of that somehow in the children block. The level option is one possibility to distinguish between objects. Another one is a "skindiff", but that you can do yourself. It would be interesting to offer other means of mapping (maybe something that builds a map behind the scenes), but I would not know how to do that. If you have ideas, I would be happy to hear them :)

Cheers,
 Doru

Lukas
 On 19 April 2010 10:49, Stéphane Ducasse <stephane.ducasse@inria.fr> wrote:
 this is really cool.

Stef
 On April 18, 2010, at 11:46 PM, Tudor Cirba wrote:
 Hi,
 I added the possibility to specify the children of a tree also based on the level at which the item resides.
 An example can be found here:
 GLMBaseExample>>treeWithChildrenByLevel

Alice wrote:
 > On Mon 23, Bob wrote:
 > > Dear list,
 > > When starting up ArgouML on my MacOS X system (Java 2)
 > > it throws a NullPointerException very soon. You'll find the
 > > trace below. I hope someone knows a solution. Thanks a lot!

```
> > Exception in thread "main" java.lang.NullPointerException
> > at
> >   javax.swing.event.SwingSupport.fireChange(SwingChange.java)
> >   at javax.swing.AbstractAction.setEnabled(AbstractAction.java)
> >   [...]
> >   at uci.uml.Main.main(Main.java:148)
```

> I'm sorry I can't help you Bob but thanks for sharing the stack...
 > Alice.
 > "Beware of programmers who carry screwdrivers." --L. Brandwein

Alice, I believe the flawed Explorer.java class generates Bob's issue:

```
public void setEnclosingFig(Fig each) {
  super.setEnclosingFig(each);
  if (each != null || (each.getOwner() instanceof MPackage)) {
    m = (MPackage) each.getOwner();
  }
}
```

 The problem is in the condition, I attach the diff with this version:

```
--- src/org/argouml/ui/explorer/Explorer.java (revision 14338)
+++ src/org/argouml/ui/explorer/Explorer.java (working copy)
@@ -147,1 +147,1 @@
[...]
super.setEnclosingFig(each);
- if (each != null || (each.getOwner() instanceof MPackage)) {
+ if (each != null && (each.getOwner() instanceof MPackage)) {
  m = (MPackage) each.getOwner();
}
```

 Probably ModelTree is also affected, if so, please change it =)
 Cheers, Carl.
 -- I used to have a sig, but it took up much space so I got rid of it!

To unsubscribe, e-mail: dev-...@argouml.tigris.org
 For additional commands, e-mail: dev-...@argouml.tigris.org

(1) Alice wrote:
 (2) > On Mon 23, Bob wrote:
 (3) > > Dear list,
 (4) > > When starting up ArgouML on my MacOS X system (Java 2)
 (5) > > it throws a NullPointerException very soon. You'll find the
 (6) > > trace below. I hope someone knows a solution. Thanks a lot!

```
(7) > > Exception in thread "main" java.lang.NullPointerException
(8) > > at
(9) > >   javax.swing.event.SwingSupport.fireChange(SwingChange.java)
(10) > >   at javax.swing.AbstractAction.setEnabled(AbstractAction.java)
(11) > >   [...]
(12) > >   at uci.uml.Main.main(Main.java:148)
```

(12) > I'm sorry I can't help you Bob but thanks for sharing the stack...
 (13) > Alice.
 (14) > "Beware of programmers who carry screwdrivers." --L. Brandwein
 (15) >

```
(16) Alice, I believe the flawed Explorer.java class generates Bob's issue:
(17) public void setEnclosingFig(Fig each) {
(18)   super.setEnclosingFig(each);
(19)   if (each != null || (each.getOwner() instanceof MPackage)) {
(20)     m = (MPackage) each.getOwner();
(21)   }
(22) }
(23) The problem is in the condition, I attach the diff with this version:
(24) --- src/org/argouml/ui/explorer/Explorer.java (revision 14338)
(25) +++ src/org/argouml/ui/explorer/Explorer.java (working copy)
(26) @@ -147,1 +147,1 @@
(27) [...]
(28) super.setEnclosingFig(each);
(29) - if (each != null || (each.getOwner() instanceof MPackage)) {
(30) + if (each != null && (each.getOwner() instanceof MPackage)) {
(31)   m = (MPackage) each.getOwner();
(32) }
```

(29) Probably ModelTree is also affected, if so, please change it =)
 (30) Cheers, Carl.
 (31) -- I used to have a sig, but it took up much space so I got rid of it!
 (32) --
 (33) To unsubscribe, e-mail: dev-...@argouml.tigris.org
 (34) For additional commands, e-mail: dev-...@argouml.tigris.org

- Non-relevant
- Natural Language
- Stack trace
- Source code
- Patch

The content of unstructured data produced during the evolution of a software system is a valuable information source to support software understanding and evolution and complements data mined from structured sources.

Alberto Bacchelli

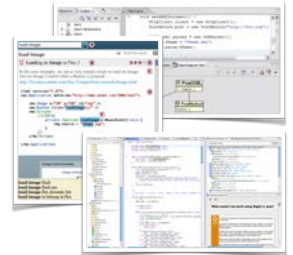
Mining Unstructured Software Data

PhD Thesis, University of Lugano, 2013

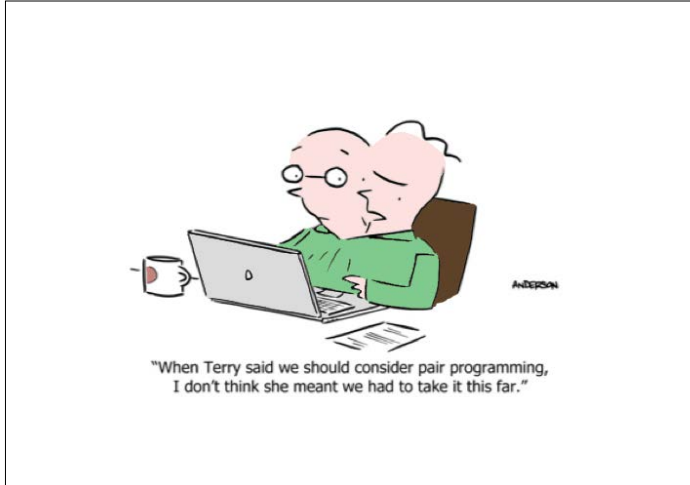


Recommender Systems for Software Engineering

"RSSEs are software applications that provide information estimated to be valuable for a software engineering task in a given context"



M. P. Robillard, R. J. Walker, and T. Zimmerman
 Recommender systems for software engineering
 IEEE Software, 2010



REmail

No Spontaneous Recommendation

No Self-Confidence

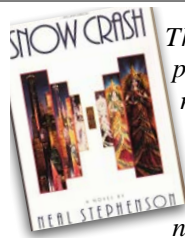
Next-gen Development Environments + Recommender Systems

Intelligent/Immersive Development Environments

Make Weapons out of Imperfections

Prompter

prompter.inf.usi.ch



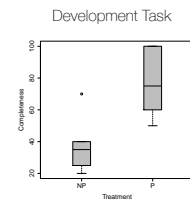
The Librarian daemon looks like a pleasant, fiftyish, silver-haired, bearded man.

Even though he's just a piece of software, the librarian has reason to be cheerful; he can move through the nearly infinite stacks of information in the Library with the agility of a spider dancing across a vast web of cross-references. The only thing he can't do is think.

"Yes, sir," the Librarian says. He is eager without being obnoxious, he clasps his hands behind his back, rocks forward slightly on the balls of his feet, raises his eyebrows expectantly over his half-glasses.

Prompter

is effective in development tasks



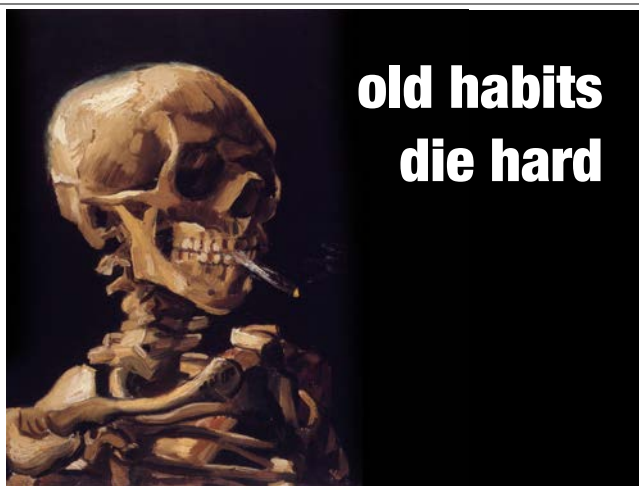
NP = Without Prompter
P = With Prompter

Librarians are there to: help; aid; assist; teach; collate; enthuse; catalogue; index; arrange; organize; find; discover; promote; display; interest; intrigue; amuse; amaze; help children, adults, old people, the underprivileged, the rich, the poor, those with voices and those without; protect resources, archive them and save them for the future; provide differing viewpoints; engender thought, conversation, fun; provide the best answers possible and match the answer to the enquirer; provide just enough information without overwhelming the user.

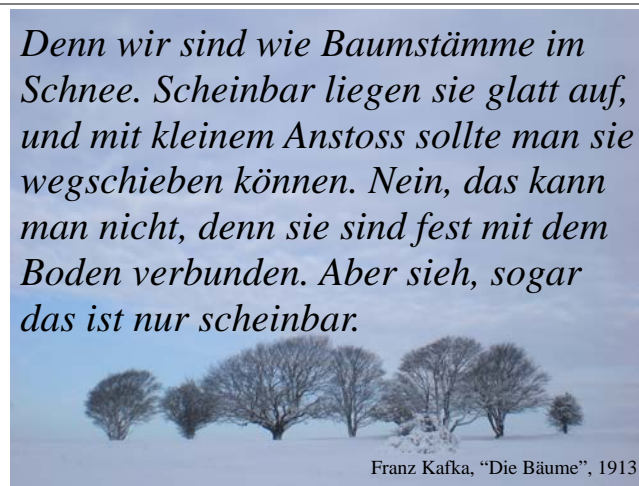
Google is there to: make money.



The End
...of the Beginning

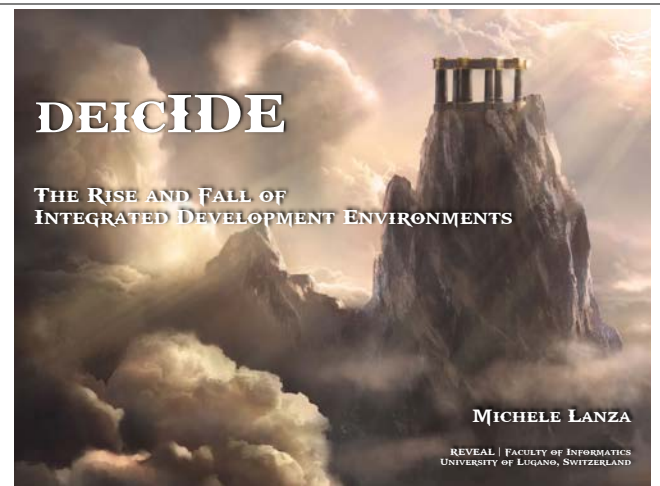


old habits die hard



Denn wir sind wie Baumstämme im Schnee. Scheinbar liegen sie glatt auf, und mit kleinem Anstoss sollte man sie wegschieben können. Nein, das kann man nicht, denn sie sind fest mit dem Boden verbunden. Aber sieh, sogar das ist nur scheinbar.

Franz Kafka, "Die Bäume", 1913



DEICIDE

THE RISE AND FALL OF INTEGRATED DEVELOPMENT ENVIRONMENTS

MICHELE LANZA

REVEAL | FACULTY OF INFORMATICS
UNIVERSITY OF LUIGANO, SWITZERLAND