

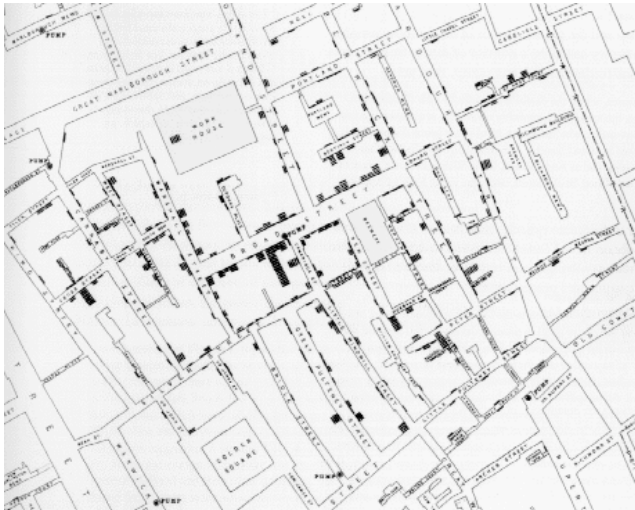
Some points on visualization

S. Ducasse

A picture is worth a thousand words

Proverb

1854, London cholera epidemic



How memory works?

Iconic memory (visual sensors)

Short-term memory (working memory)

Long-term memory

Short-term memory

where information resides during conscious processing

it is temporary

partly used for visual info

limited storage (three to nine)

This is why it is better to see everything in one glance

This is why seen everything in one glance is better

Preattentive processing

Early stage of visual perception below the level of consciousness

How many 5?

3332123466509000096766689877835367
7866750910919818971746453039821768
34567865860880221167687687789762
345678915116718199101081876616161
61819010180980808097767674333

How many 5?

3332123466509000096766689877835367
7866750910919818971746453039821768
34567865860880221167687687789762
345678915116718199101081876616161
61819010180980808097767674333

Preattentive attributes

Color (hue/intensity)

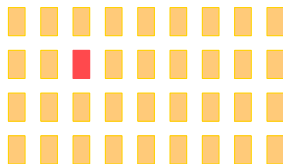
Form

orientation, line length, line width, size, shape, added marks, enclosure

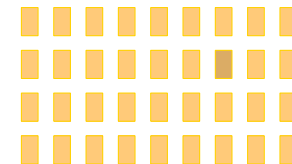
Spatial position (2D location)

Motion (flicker)

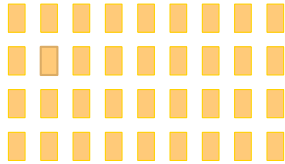
Color / hue



Color / Intensity



Color / Intensity



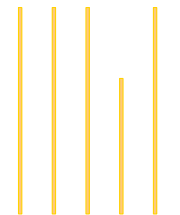
Position



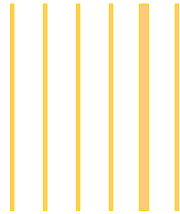
Form / Orientation



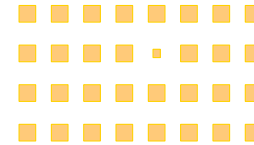
Form / Line length



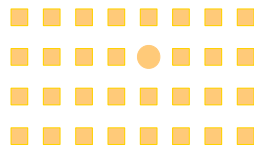
Form / Line width



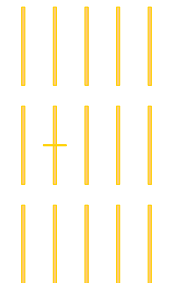
Form / Size



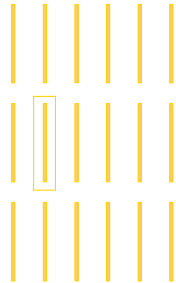
Form / Shapes



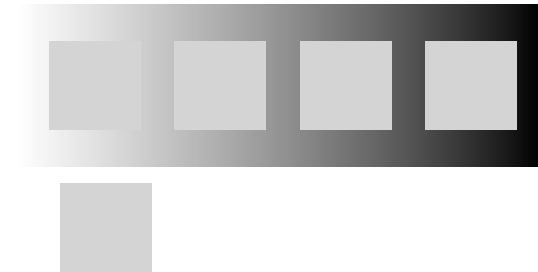
Form / Added marks



Form / Enclosure



Context



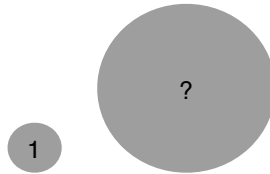
Context



Encoding Quantitative

Category	Attribute	Quantitative
Color	hue	No
	Intensity	Yes but limited
Position	2D	Yes
Form	Orientation	No
	Line length	Yes
	Line width	Yes but limited
	Size	Yes but limited
	Shape, Added marks, Enclosure	No

Size....



Gestalt Principles of Visual Perception

Back in 1912, from the Gestalt School of psychology

Still stand today

Gestalt means patterns

How do we perceive pattern, form, and organization?

Gestalt Principles of Visual Perception

Proximity

Similarity

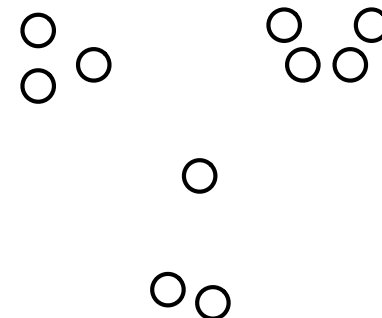
Enclosure

Closure

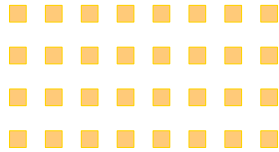
Continuity

Connection

Principle of Proximity



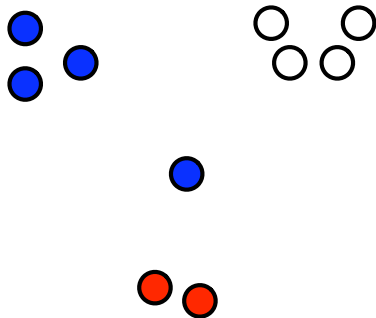
Principle of Proximity



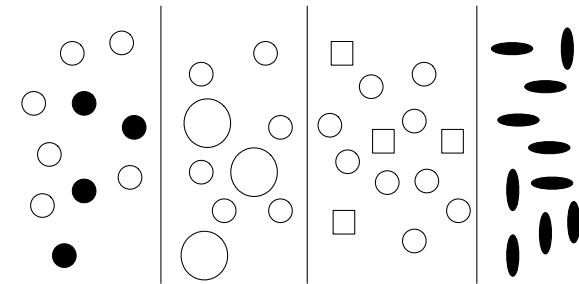
Principle of Proximity



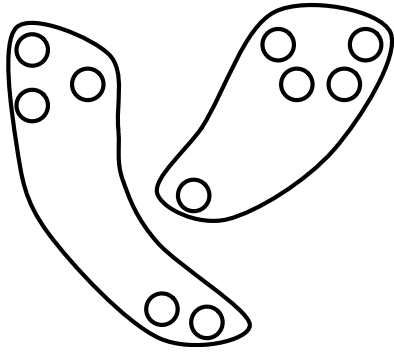
Principle of Similarity



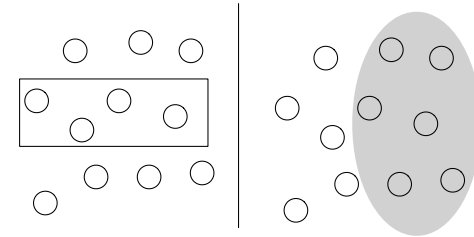
Principle of Similarity



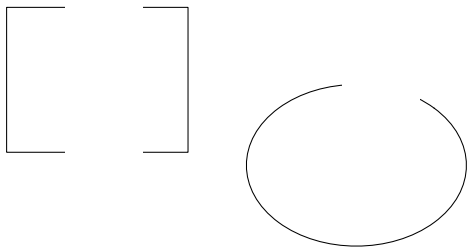
Principle of Enclosure



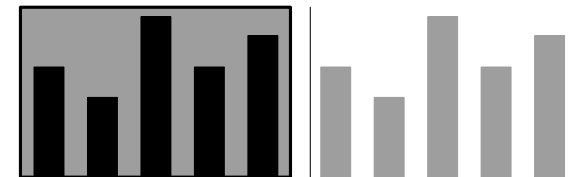
Principle of Enclosure



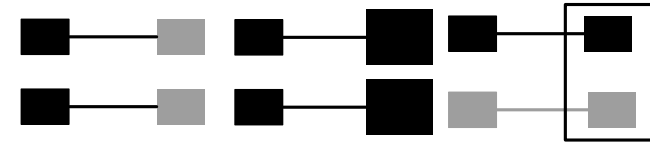
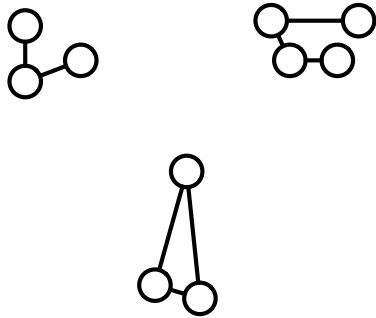
Principle of Closure



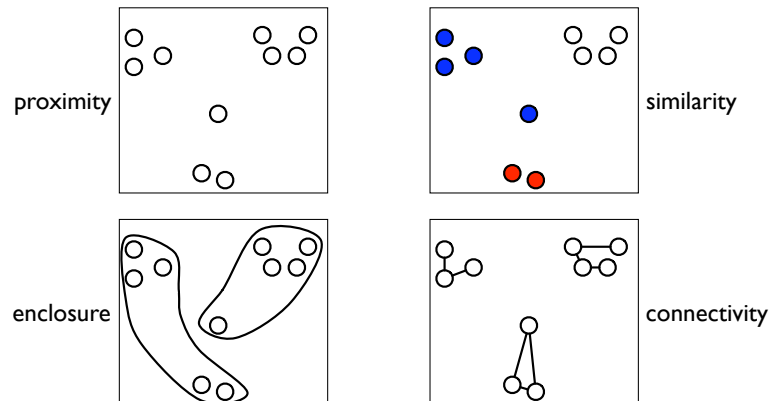
Principle of Closure



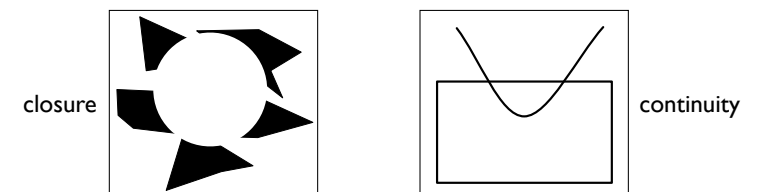
Principle of connectivity



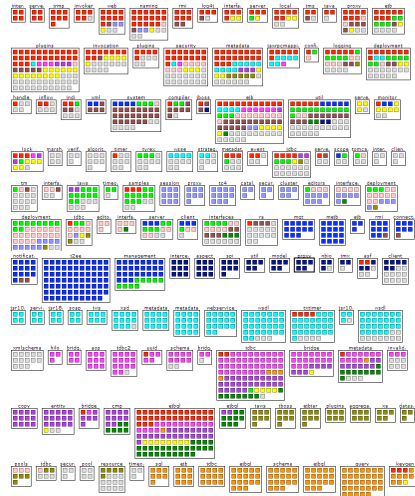
Gestalt principles



More Gestalt principles

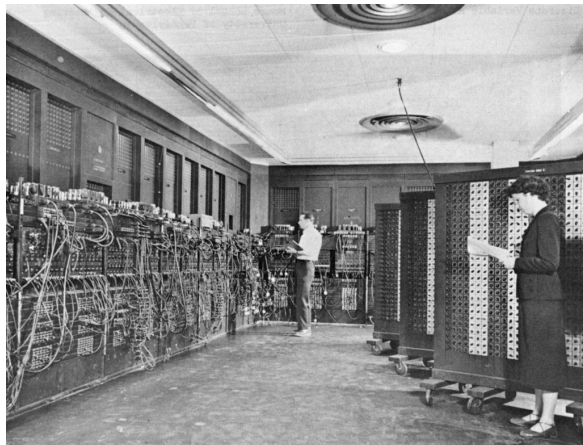


70% of our sensors are dedicated to vision



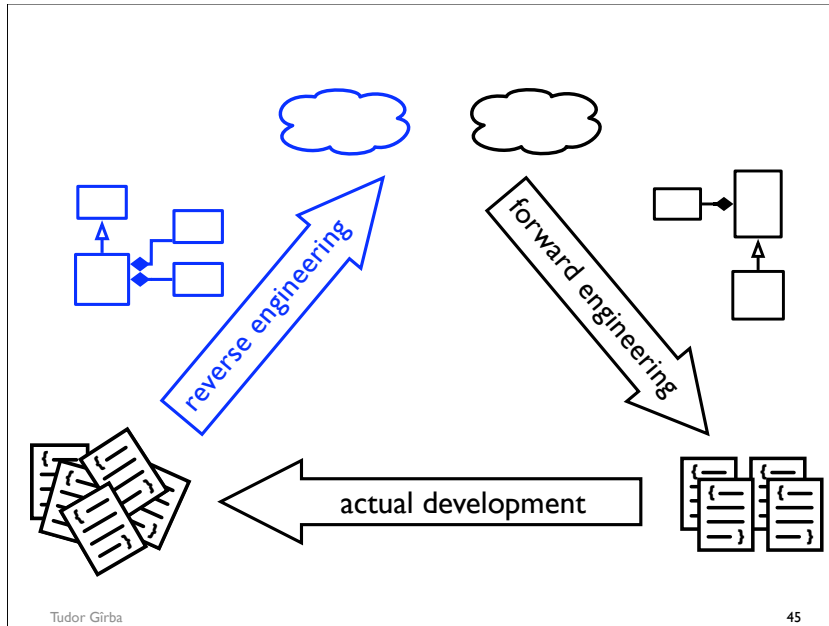
Why software visualization?

In 1946, programs had a physical shape



How large is your code?

100'000 lines of code
* 2 = 200'000 seconds
/ 3600 = 56 hours
/ 8 = 7 days



What to visualize?
How to visualize?

Tudor Girba 46

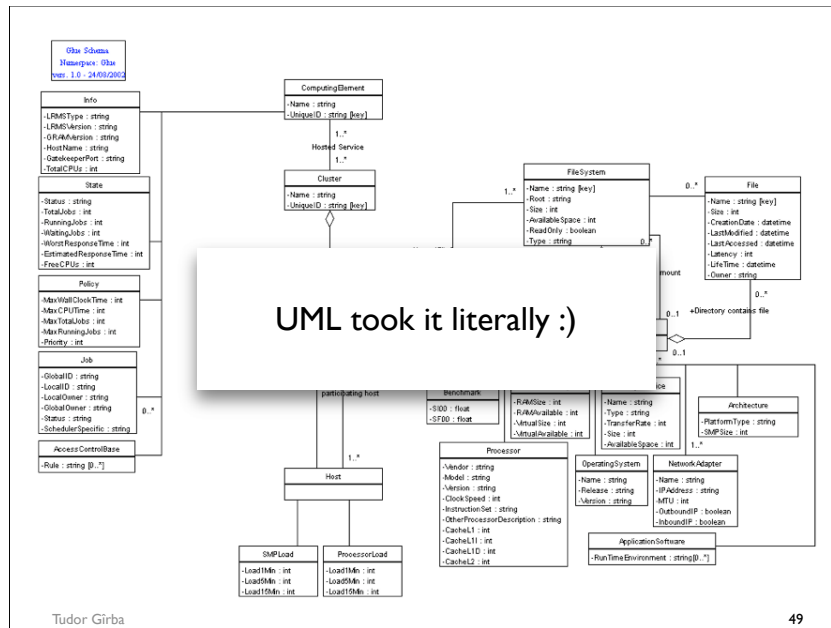
What to visualize?
How to visualize?

Tudor Girba 47

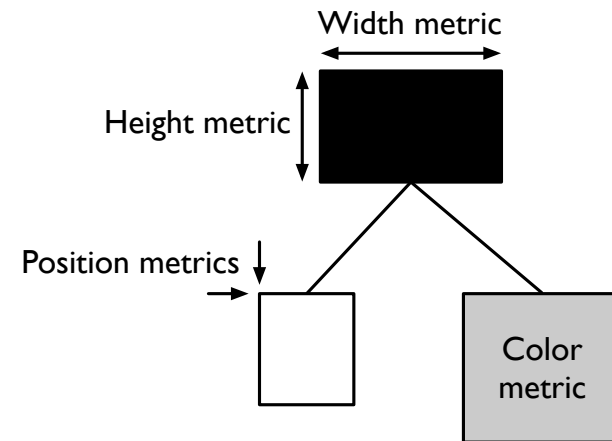
A picture is worth a thousand words

Proverb

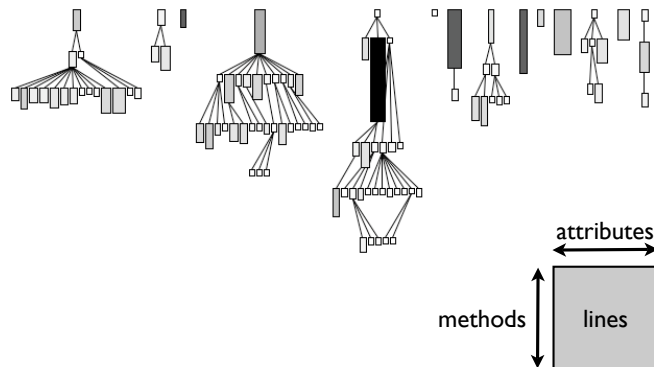
Tudor Girba 48



Polymetric views show up to 5 metrics

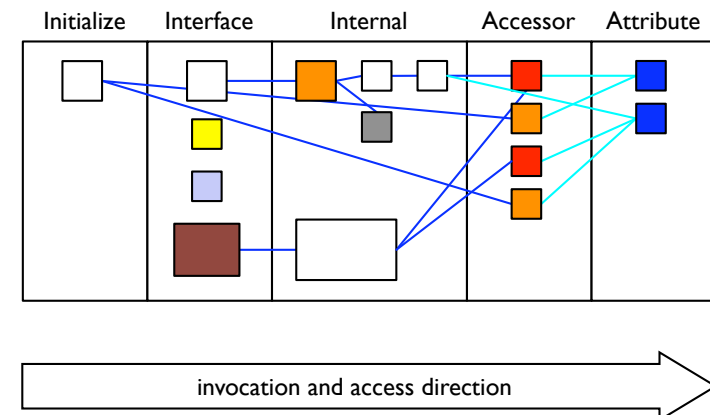


System Complexity shows class hierarchies

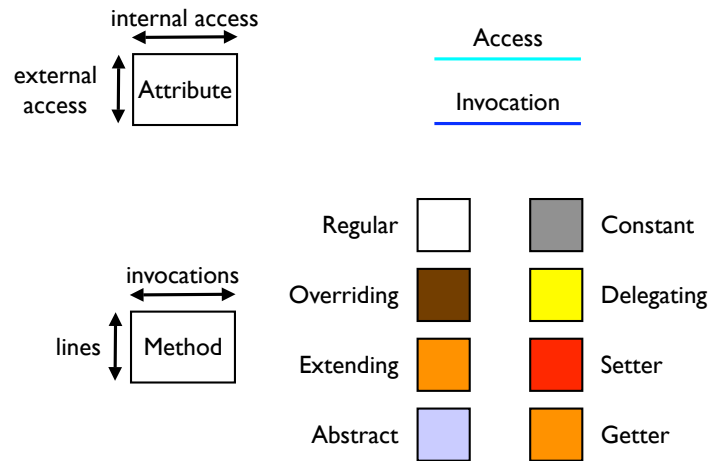


Class Blueprint shows class internals

Ducasse, Lanza, 2005



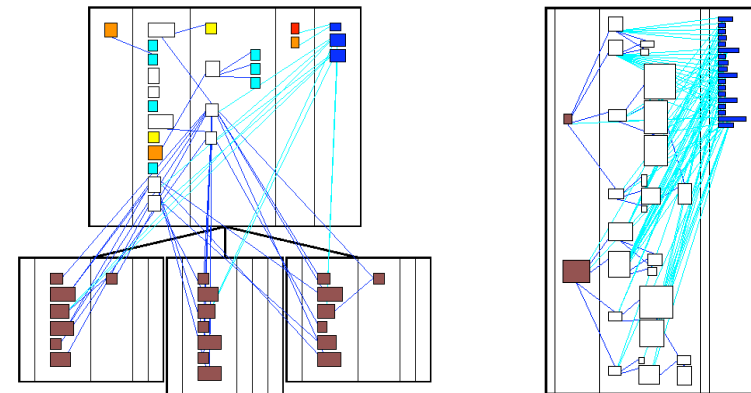
Class Blueprint has a rich vocabulary



Tudor Girba

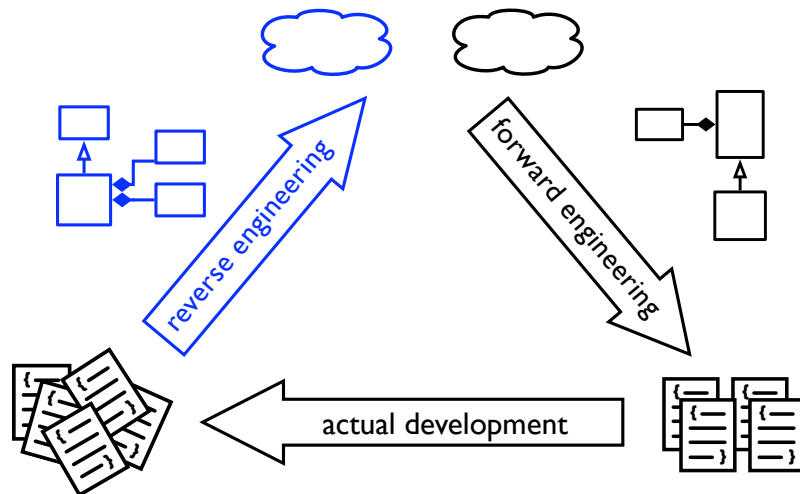
53

Class Blueprint shows class internals



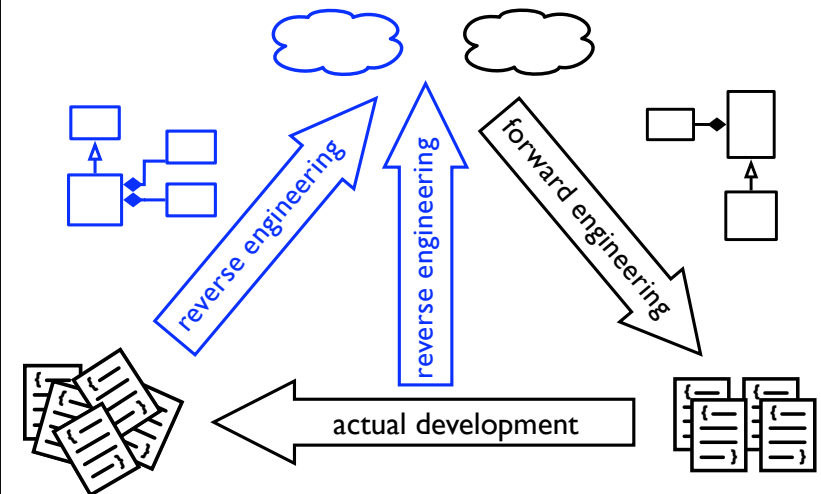
Tudor Girba

54



Tudor Girba

55



Tudor Girba

56

Spectrographs show change activity

Wu et al 2004

commit



time

Tudor Girba

57

Evolution Matrix shows class changes

Lanza, Ducasse, 2002

Idle class



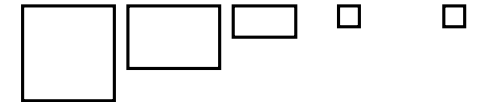
Pulsar class



Supernova class



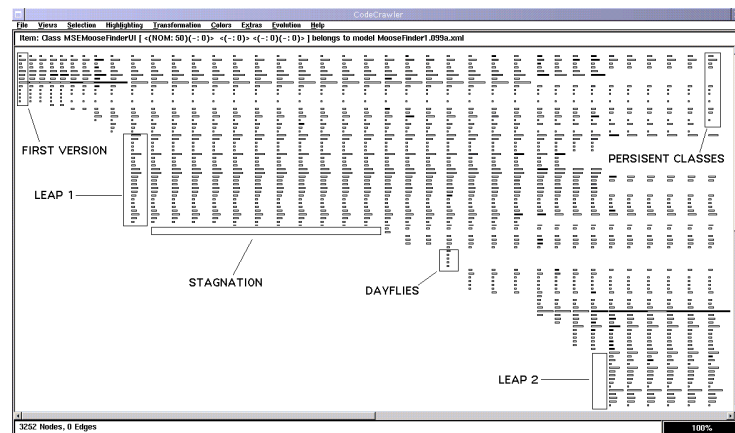
White dwarf class



Tudor Girba

58

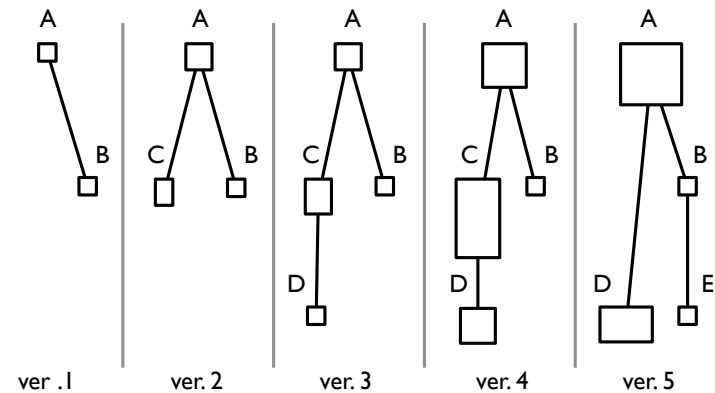
Evolution Matrix shows class changes



Tudor Girba

59

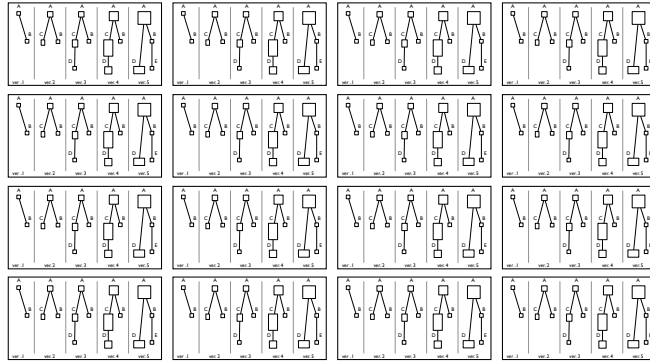
What happens with inheritance?



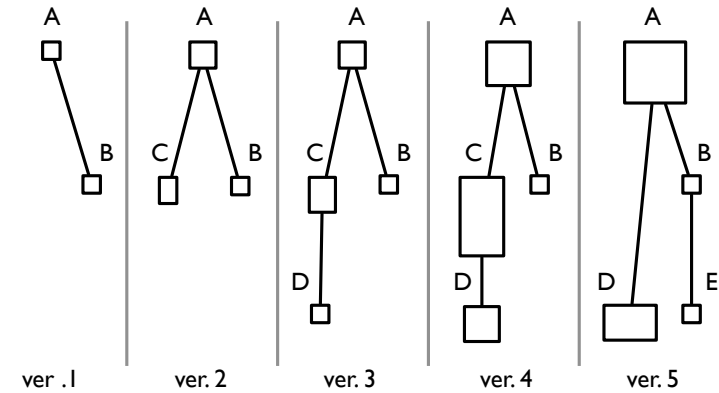
Tudor Girba

60

History contains too much data



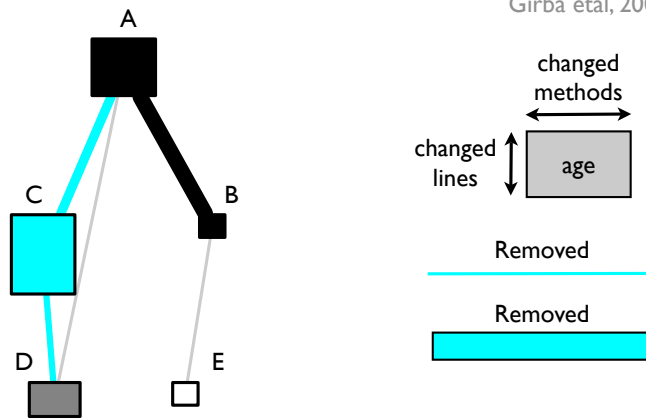
History contains too much data



A is persistent, B is stable, C was removed, E is newborn ...

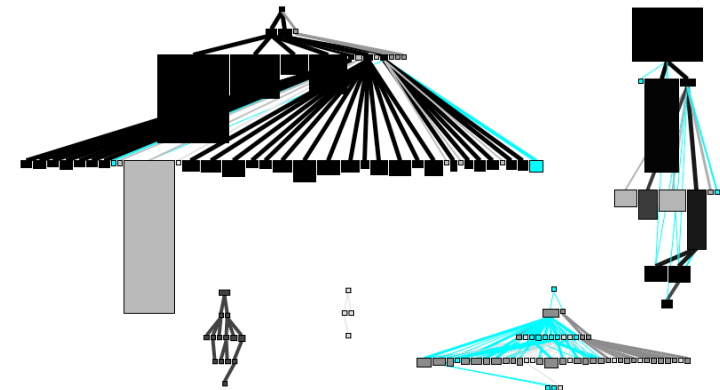
Hierarchy Evolution encapsulates time

Girba et al, 2005



A is persistent, B is stable, C was removed, E is newborn ...

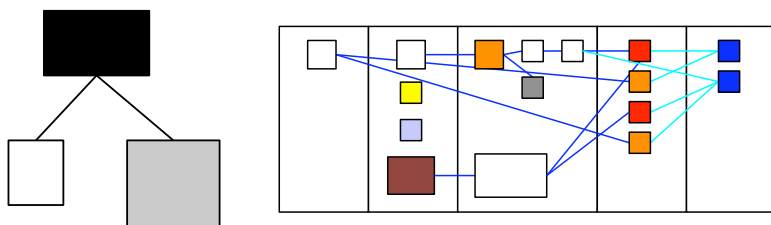
Hierarchy Evolution View reveals patterns



What to visualize?
How to visualize?

Why do we care?

System Complexity vs. Class Blueprint



A small experiment

1, 13, 27, 4, 96

What were the numbers?

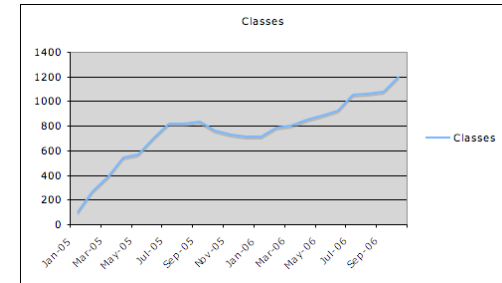
What was the last
advertisement you saw?

What to visualize?
How to visualize?

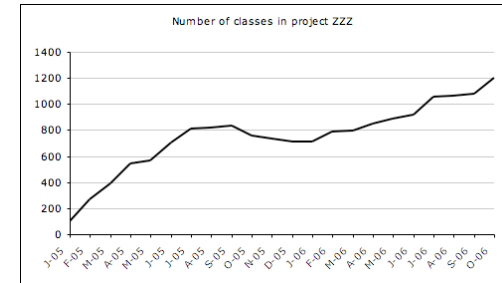
Principle: Minimize chart junk

Tufte, 1990

Excel default

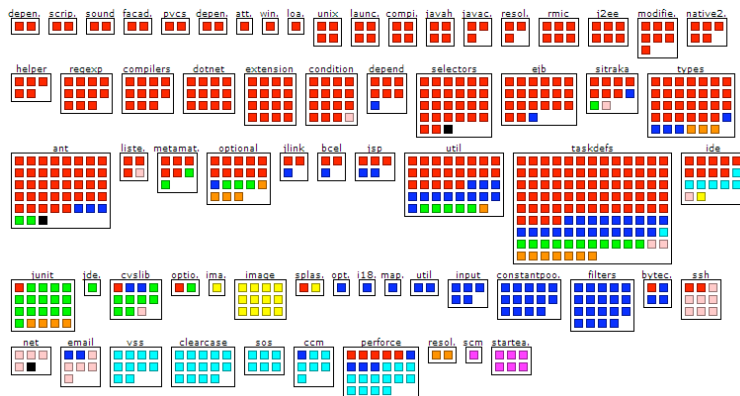


Excel default
+
6 actions



Distribution Map shows properties

Ducas et al, 2006

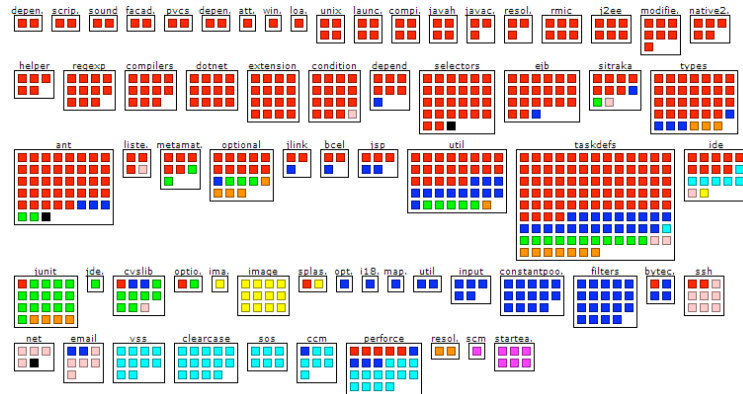


Principle: Maximize data ink

Tufte, 1990

Distribution Map shows properties in an order

Ducasse et al, 2006

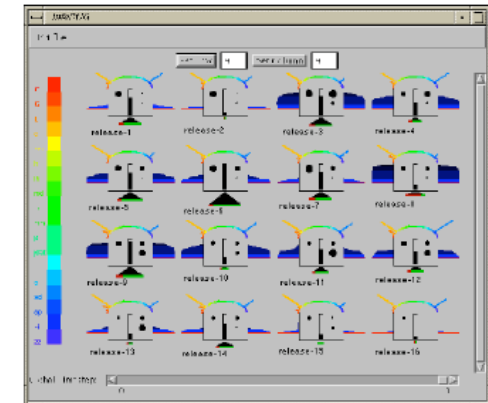


Tudor Girba

77

InfoBug presents software in a condensed and cute way

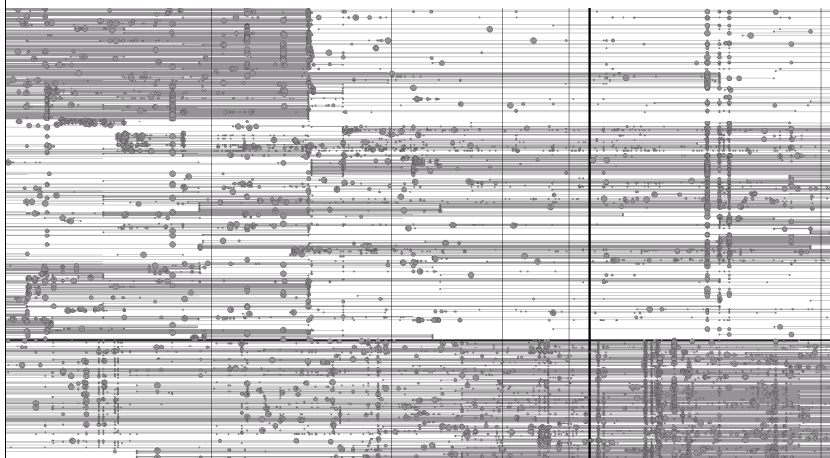
Chuah, Eick, 2006



Tudor Girba

78

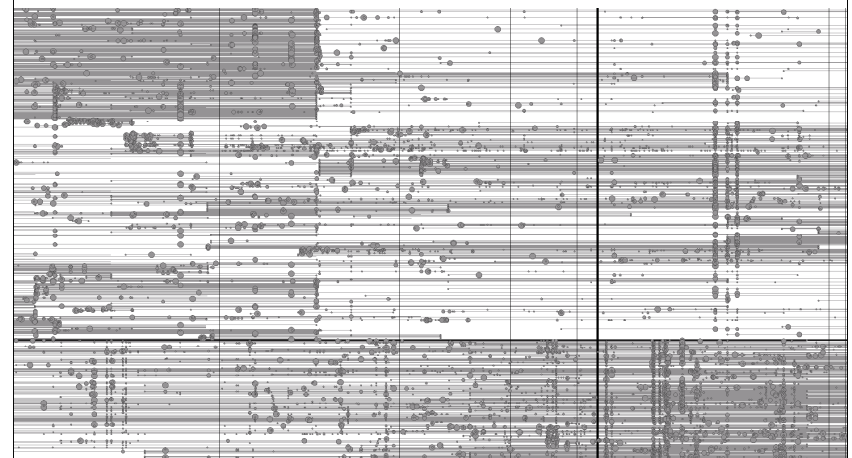
Example: CVS shows activity



Tudor Girba

79

Who did this?

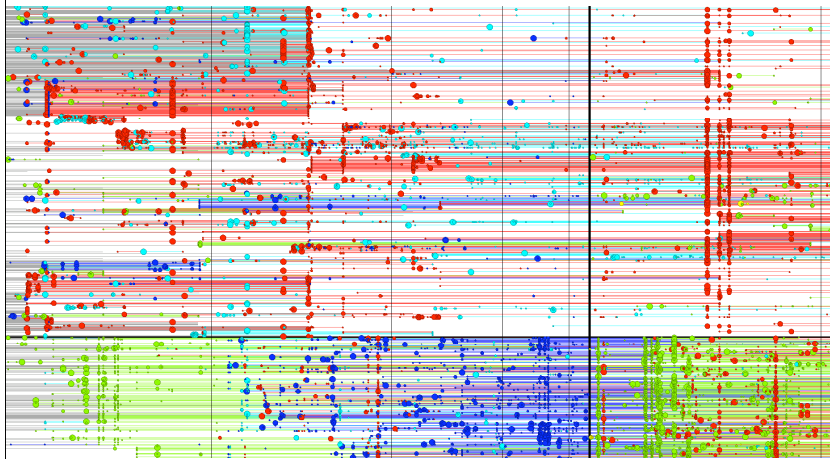


Tudor Girba

80

Alphabetical order is no order

Kuhn

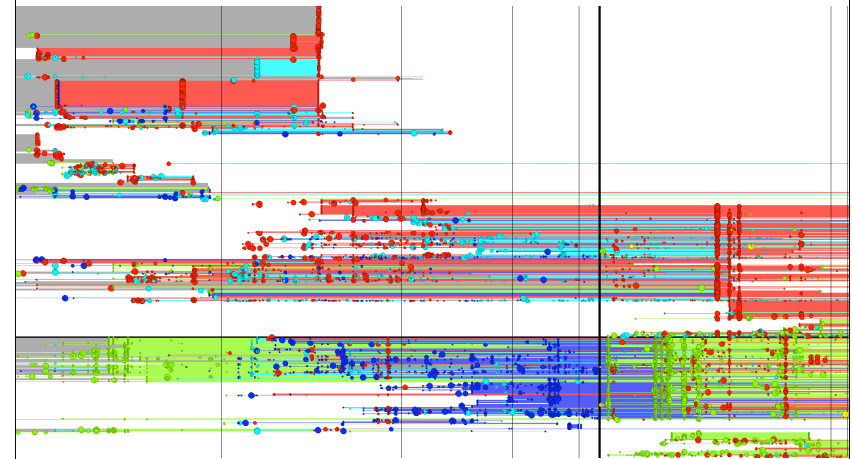


Tudor Girba

81

Ownership Map orders histories

Girba et al, 2006



Tudor Girba

82

Example: Who duplicated the code?

```
(john 23.06.03) public boolean stillValid (ToDoItem I, Designer dsgr) {  
(bill 09.01.05)   if (!isActive()) {  
(bill 09.01.05)     return false  
(bill 09.01.05)   }  
(steve 16.02.05)   List offs = i.getOffenders();  
(john 23.06.03)   Object dm = offs.firstElement();  
(steve 16.02.05)   ListSet newOffs = computeOffenders(dm);  
(john 23.06.03)   boolean res = offs.equals(newOffs);  
(john 23.06.03)   return res;  
}
```

```
(george 13.02.05) public boolean stillValid (ToDoItem I, Designer dsgr) {  
(bill 11.13.05)   if (!isActive()) {  
(bill 11.13.05)     return false  
(bill 11.13.05)   }  
(steve 16.02.05)   List offs = i.getOffenders();  
(george 13.02.05) Object dm = offs.firstElement();  
(steve 16.02.05)   ListSet newOffs = computeOffenders(dm);  
(george 13.02.05) boolean res = offs.equals(newOffs);  
(george 13.02.05) return res;  
}
```

Tudor Girba

83

What is useless?

```
(john 23.06.03) public boolean stillValid (ToDoItem I, Designer dsgr) {  
(bill 09.01.05)   if (!isActive()) {  
(bill 09.01.05)     return false  
(bill 09.01.05)   }  
(steve 16.02.05)   List offs = i.getOffenders();  
(john 23.06.03)   Object dm = offs.firstElement();  
(steve 16.02.05)   ListSet newOffs = computeOffenders(dm);  
(john 23.06.03)   boolean res = offs.equals(newOffs);  
(john 23.06.03)   return res;  
}
```

```
(george 13.02.05) public boolean stillValid (ToDoItem I, Designer dsgr) {  
(bill 11.13.05)   if (!isActive()) {  
(bill 11.13.05)     return false  
(bill 11.13.05)   }  
(steve 16.02.05)   List offs = i.getOffenders();  
(george 13.02.05) Object dm = offs.firstElement();  
(steve 16.02.05)   ListSet newOffs = computeOffenders(dm);  
(george 13.02.05) boolean res = offs.equals(newOffs);  
(george 13.02.05) return res;  
}
```

Tudor Girba

84

When did changes happen?

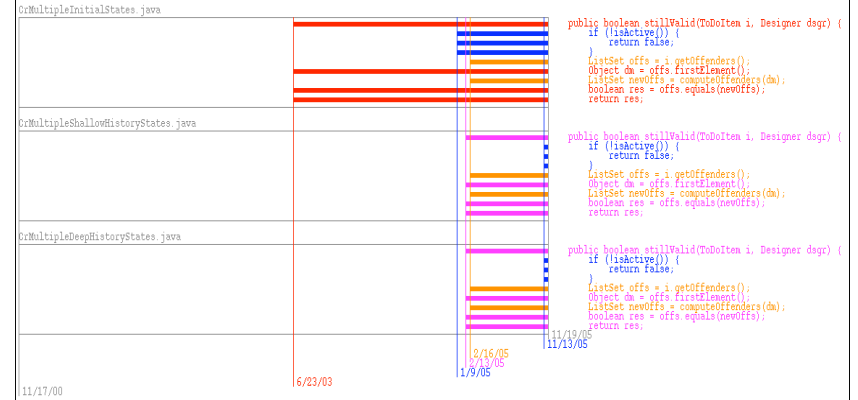
```

23.06.03      public boolean stillValid (ToDoItem I, Designer dsgr) {
09.01.05      if (!isActive()) {
09.01.05      return false
09.01.05      }
16.02.05      List offs = i.getOffenders();
23.06.03      Object dm = offs.firstElement();
16.02.05      ListSet newOffs = computeOffenders(dm);
23.06.03      boolean res = offs.equals(newOffs);
23.06.03      return res;

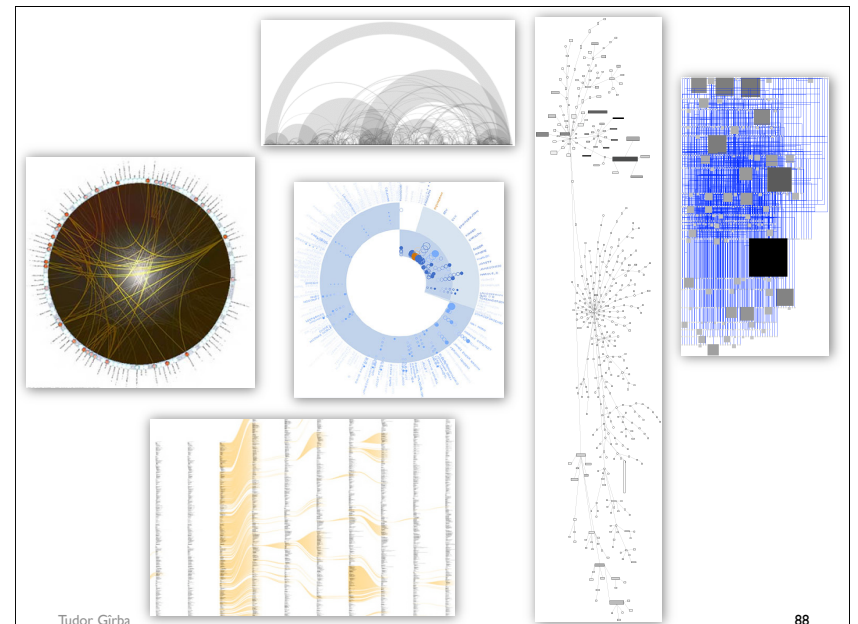
13.02.05      public boolean stillValid (ToDoItem I, Designer dsgr) {
11.13.05      if (!isActive()) {
11.13.05      return false
11.13.05      }
16.02.05      List offs = i.getOffenders();
13.02.05      Object dm = offs.firstElement();
16.02.05      ListSet newOffs = computeOffenders(dm);
13.02.05      boolean res = offs.equals(newOffs);
13.02.05      return res;
    
```

Clone Evolution shows how developers copy

Balint et al, 2006



Visualization does not guarantee understanding





Attribution-ShareAlike 2.5

You are free:

- to copy, distribute, display, and perform the work
- to make derivative works
- to make commercial use of the work

Under the following conditions:



Attribution. You must attribute the work in the manner specified by the author or licensor.



Share Alike. If you alter, transform, or build upon this work, you may distribute the resulting work only under a license identical to this one.

For any reuse or distribution, you must make clear to others the license terms of this work. Any of these conditions can be waived if you get permission from the copyright holder.

Your fair use and other rights are in no way affected by the above.