Visualisation Logicielle pour la Maintenance

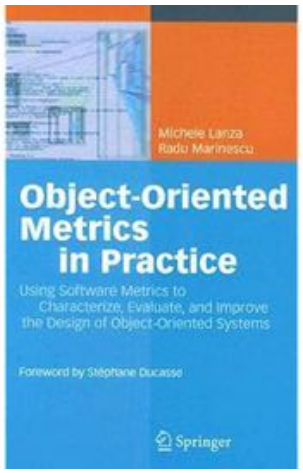
Sébastien Mosser

MGL7460- *Cours #10 - H20*

UQÀM | Département d'informatique

Crédit Images: Pixabay & Pexels

ace



Pragmatic Design Quality Assessment

Tudor Gîrba
University of Bern, Switzerland

Michele Lanza
University of Lugano, Switzerland

Radu Marinescu
Politehnica University of Timisoara, Romania

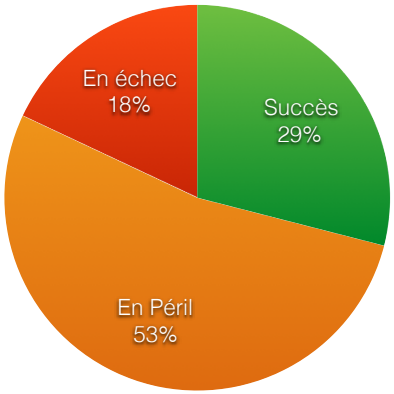
[PDQA]

https://fr.slideshare.net/girba/pragmatic-quality-assessment-tutorial-icse-2008

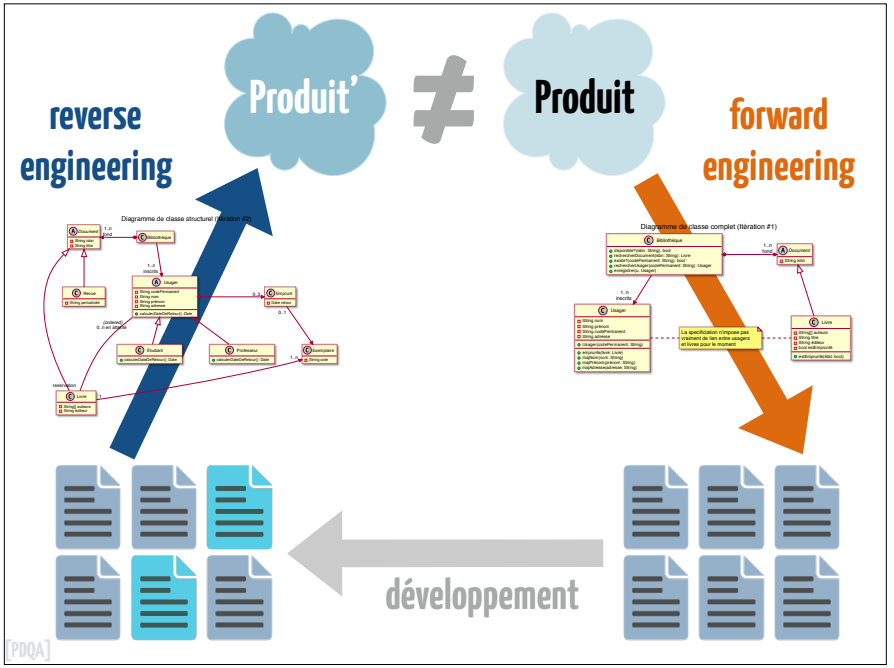
Épisodes Précédents

1

Le logiciel, cet objet complexe



[The Stanford Group, 2004]



You cannot **control**

what you cannot **measure**

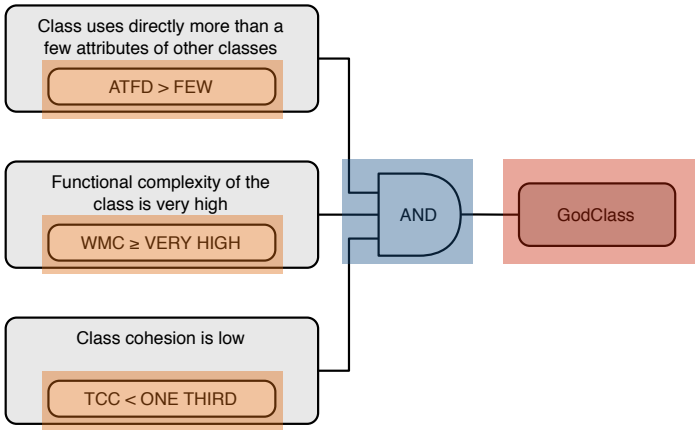
Tom de Marco

Métrique Logicielle (définition)

Les **métriques logicielles** sont des **mesures** associées à des **systèmes logiciels**, leurs **processus de développement** et les **documents associés**.

A God Class centralizes too much intelligence in the system.

Lanza, Marinescu 2006



Composition de métrique

Problème de conception

[PDQA]

Qui dit **métrique** dit aussi **bon sens** ...

Peut-on mesurer la beauté d'une oeuvre en comptant le nombre de couleurs utilisées ?

[PDQA]

Principes de Visualisation

2

Les métriques seules sont peu utile

Metric	Value	Remarks
No. of Lines of Code	223,068	including comments
No. of Source Files	1,209	*.java files
No. of Packages	99	–
No. of Classes	1,393	including 140 inner classes
No. of Methods	9,561	including accessor methods
No. of Attributes	3,358	all variables including static and local variables

On peut les combiner,
mais le résultat reste binaire

[PDQA]

Épidémie de Choléra à Londres (1854)

Quartier	|Puits|	|Malade|
A	1	Aucun
B	2	Aucun
C	1	Élevé
D	0	Élevé
E	0	Faible
F	2	Élevé
G	1	Faible
H	1	Faible

1854,
London,
cholera
epidemic

Quartier

Puits

[PDQA]

1854,
London,
cholera
epidemic

Foyer
Infectieux

Puits

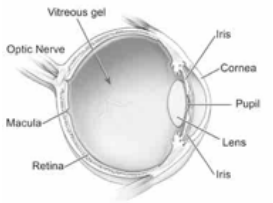
Épidémie de Choléra à Londres (1854)

Quartier	|Puits|	|Malade|
A	1	Aucun
B	2	Aucun
C	1	Élevé
D	3	Élevé
E	0	Faible
F	2	Élevé
G	1	Faible
H	1	Faible

La bonne visualisation est la distance au puit

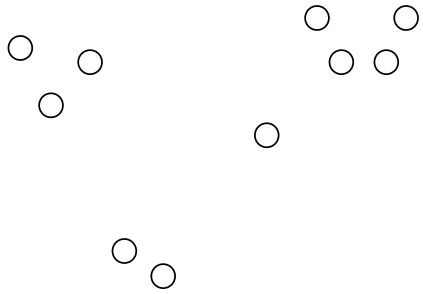
70%

of all external inputs come through the eyes



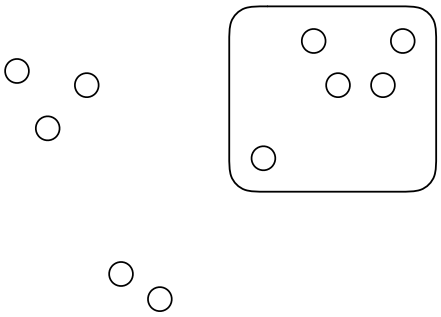
[PDQA]

Combien de groupes voyez vous ?



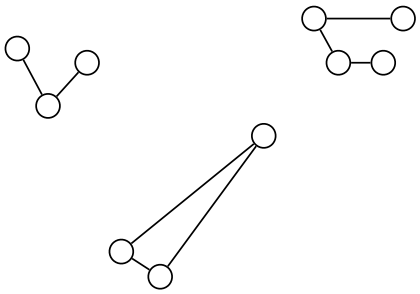
[PDQA]

Combien de groupes voyez vous ?

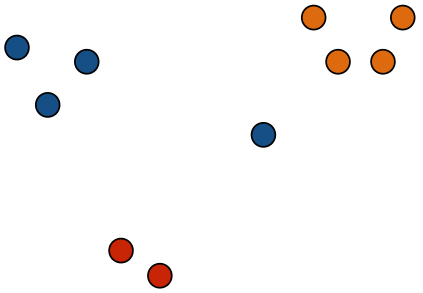


[PDQA]

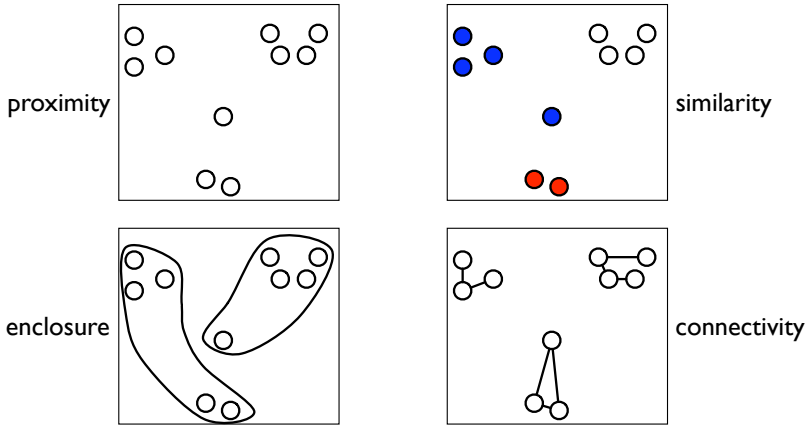
Combien de groupes voyez vous ?



Combien de groupes voyez vous ?



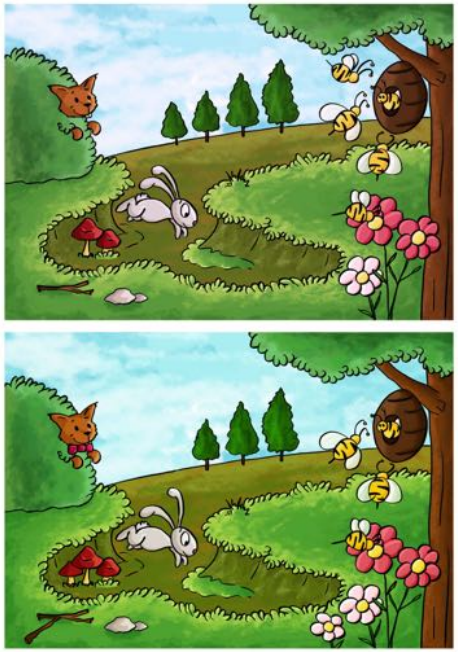
Principe de Gestalt



Visualiser = Exploiter la Pré-attention

8789364082376403128764532984732984732094873290845
389274-0329874-32874-231984750983409834098324098320
049823-0984903281453209481-0839393947896587436598

8789364082376403128764532984732984732094873290845
389274-0329874-32874-231984750983409834098324098320
049823-0984903281453209481-0839393947896587436598

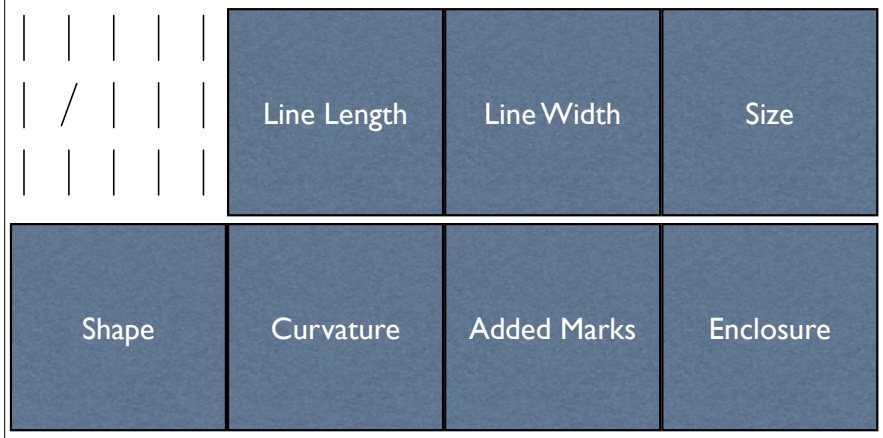


http://blog.dinett-illustration.com/

Attributes of Form

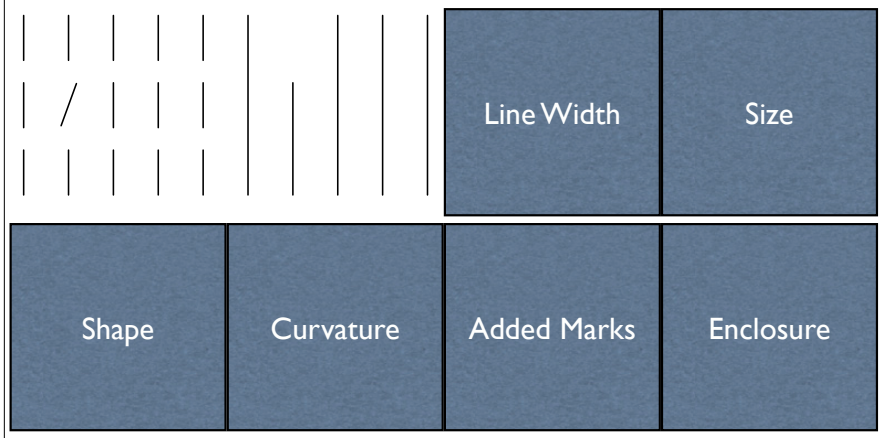
Orientation	Line Length	Line Width	Size
Shape	Curvature	Added Marks	Enclosure

Attributes of Form



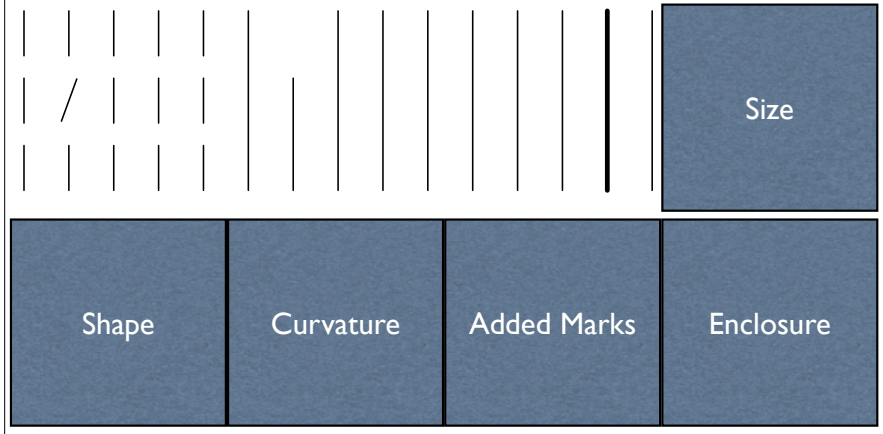
[PDQA]

Attributes of Form



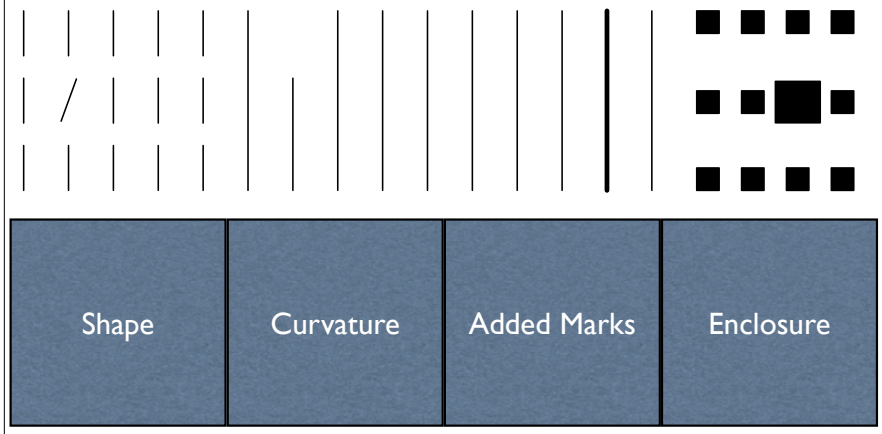
[PDQA]

Attributes of Form



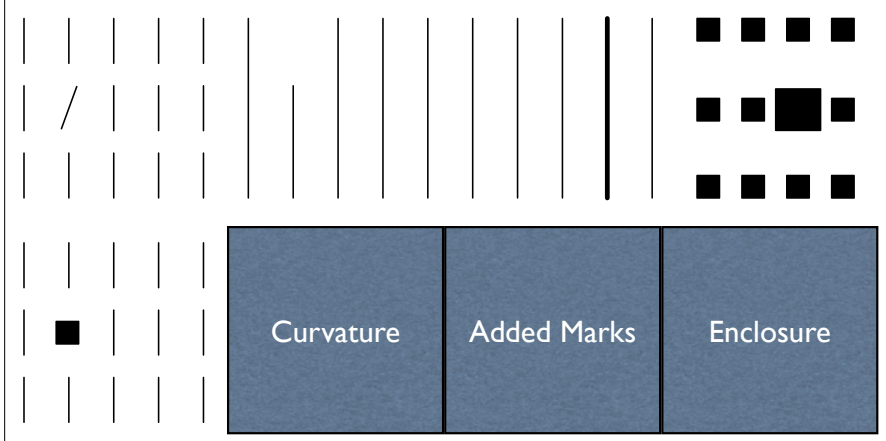
[PDQA]

Attributes of Form



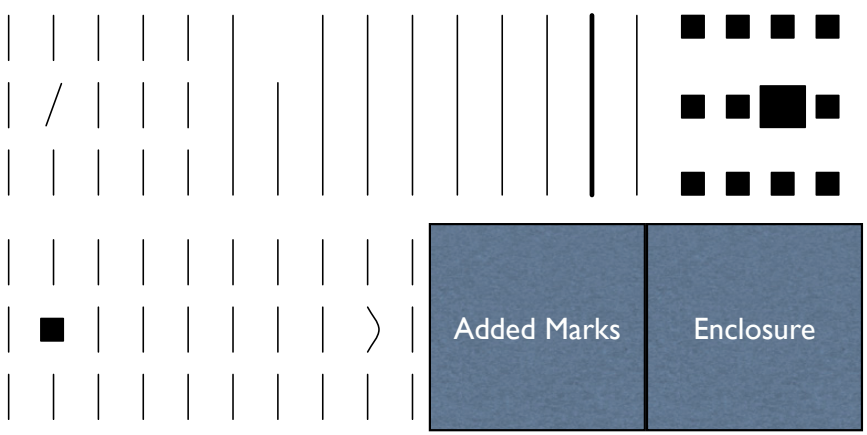
[PDQA]

Attributes of Form



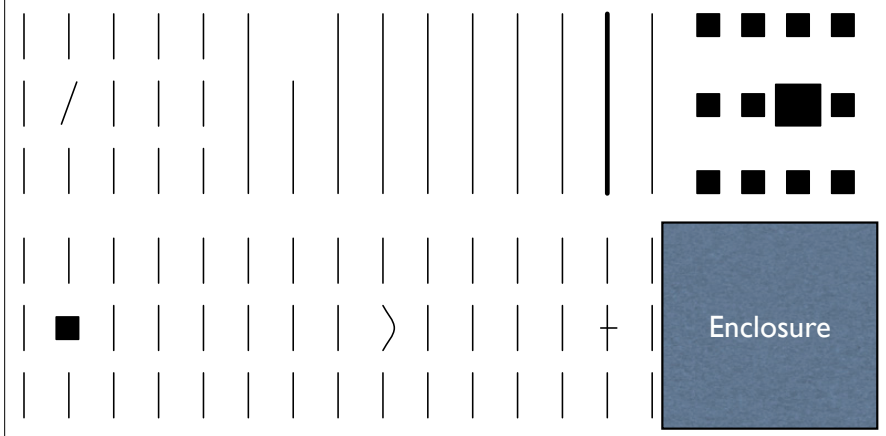
[PDQA]

Attributes of Form



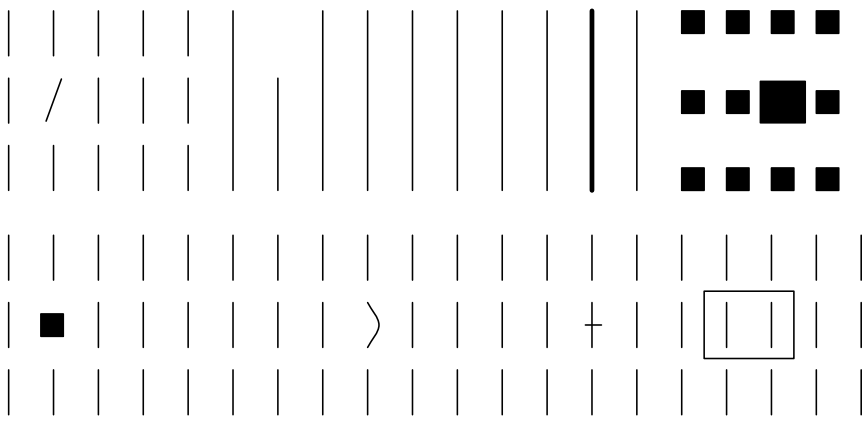
[PDQA]

Attributes of Form



[PDQA]

Attributes of Form

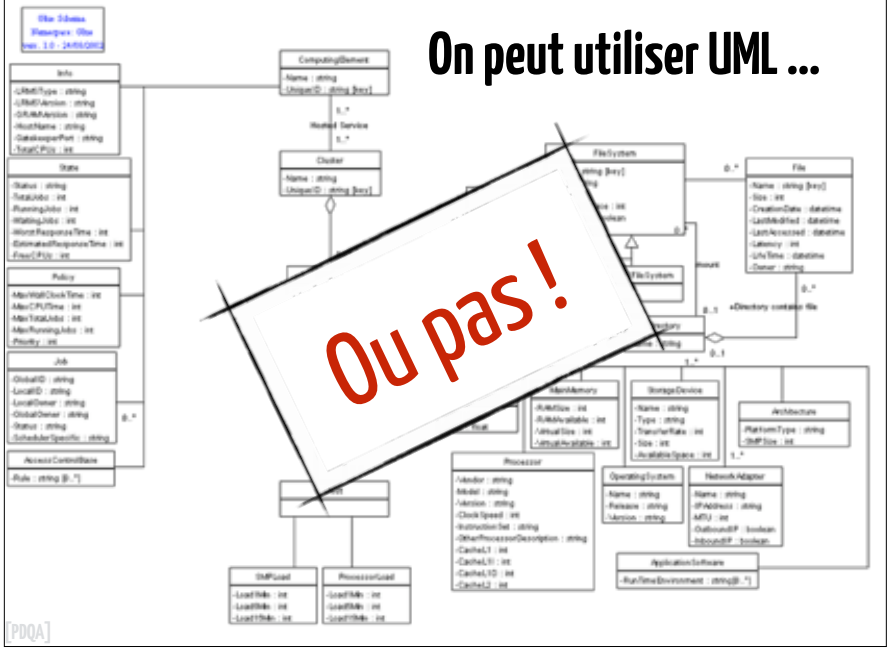


[PDQA]

Exemples de Visualisations

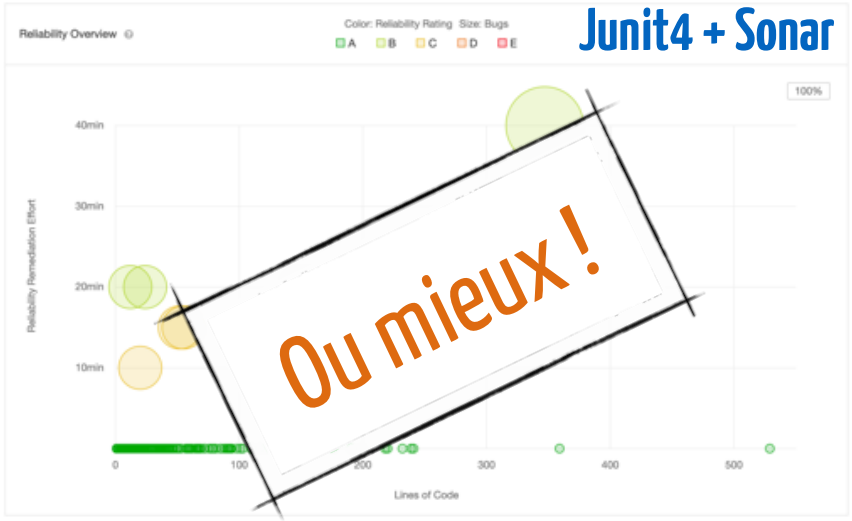
3

On peut utiliser UML ...



[PDQA]

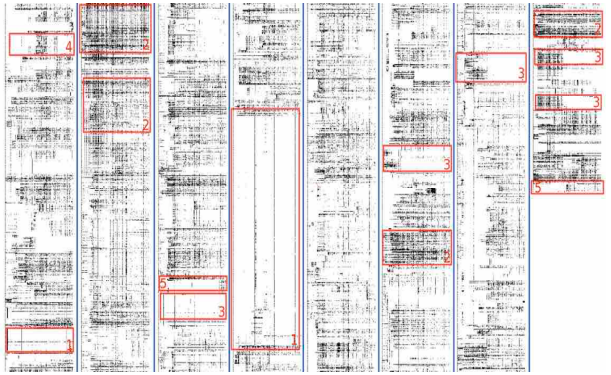
Junit4 + Sonar



On peut utiliser des graphes ...

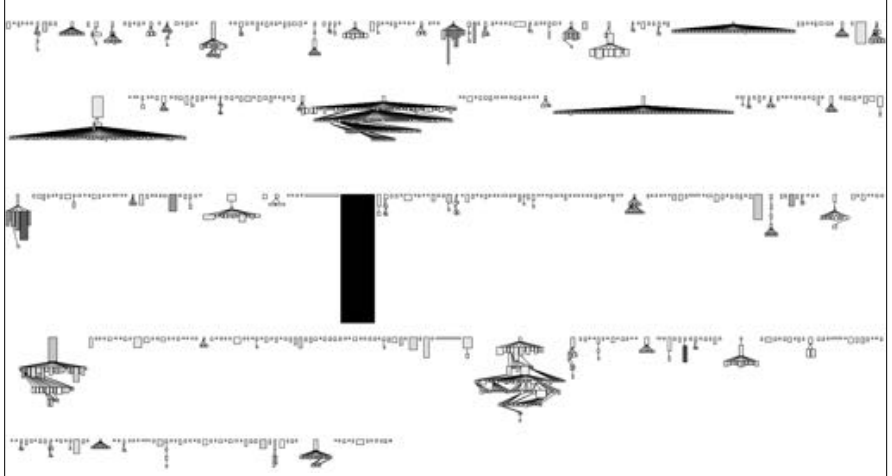
Software visualization is
the use of the crafts of typography, graphic design, animation, and cinematography with modern human-computer interaction and computer graphics technology to facilitate both the human understanding and effective use of computer software.

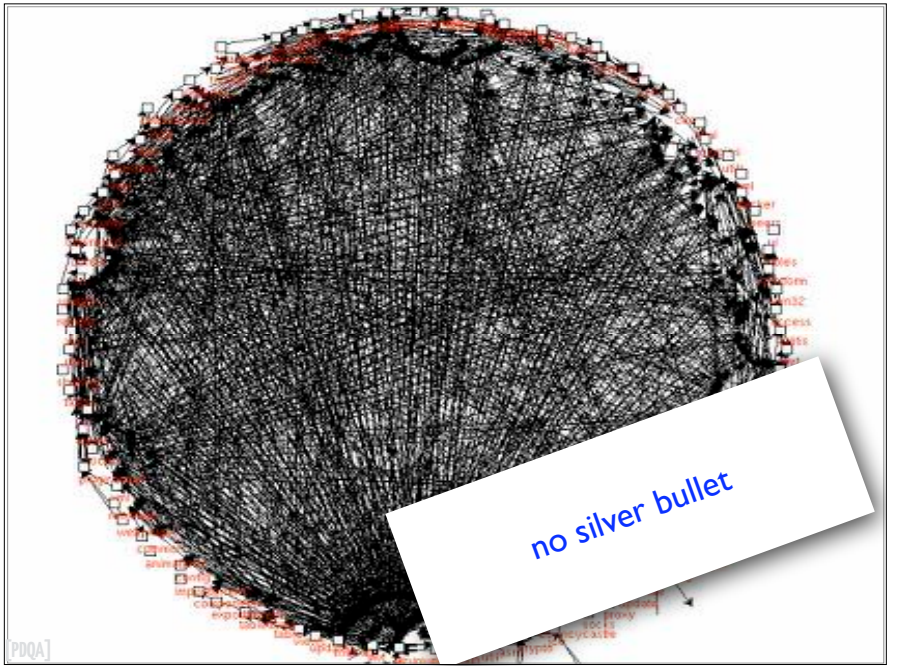
Price, Becker, Small



[PDQA]

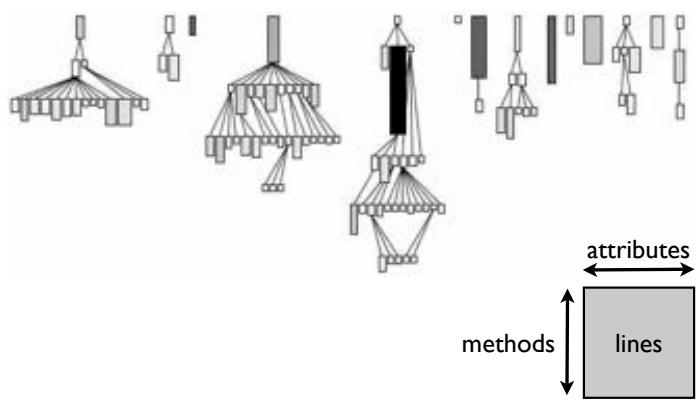
La visualisation rend la conception visible





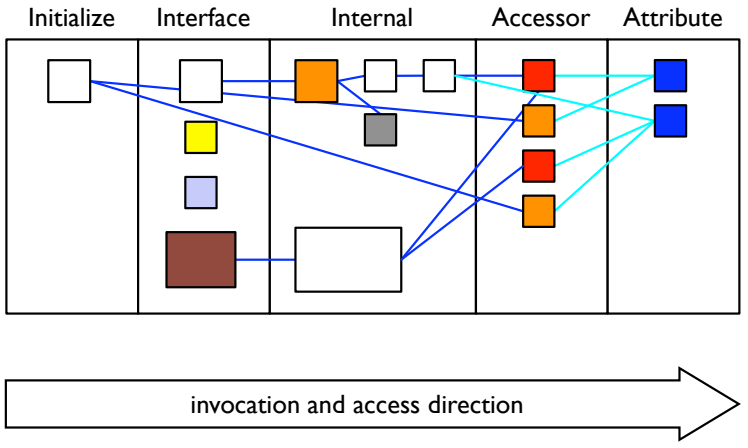
System Complexity shows class hierarchies.

Lanza, Ducasse, 2003

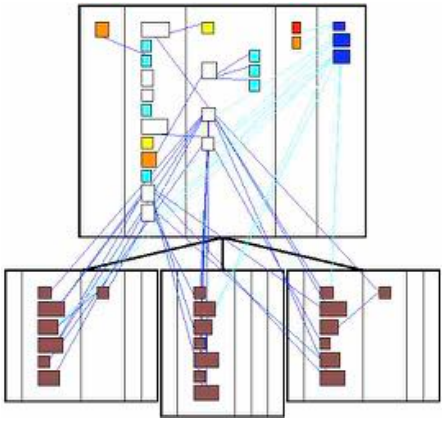


Class Blueprint shows class internals.

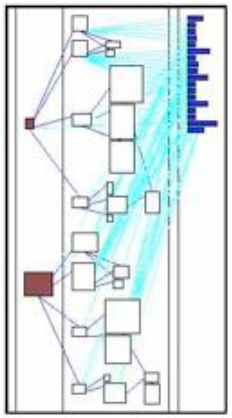
Lanza, Ducasse, 2005



Class Blueprint reveals patterns.



twin classes

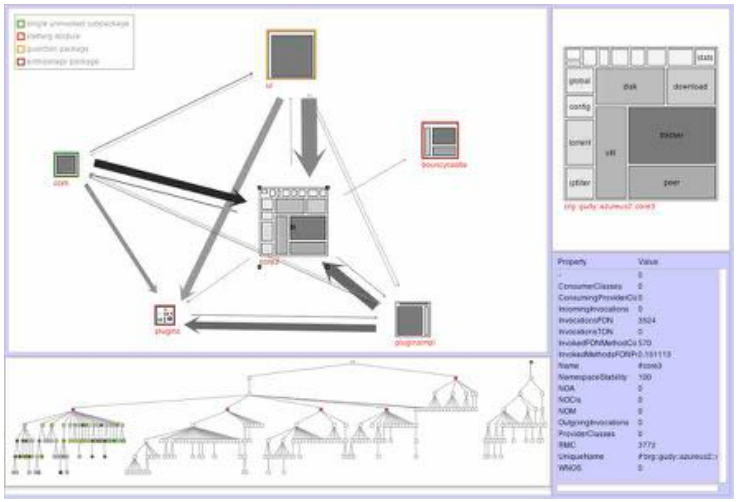


schizophrenic class

[PDQA]

Softwarenaut explores the package structure.

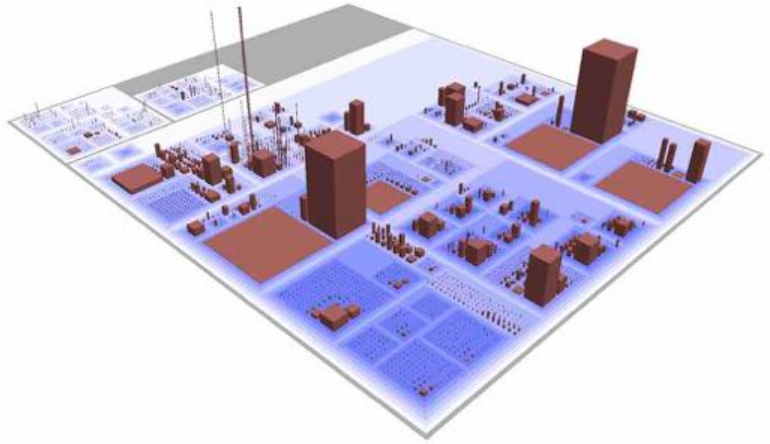
Lungu etal, 2006



[PDQA]

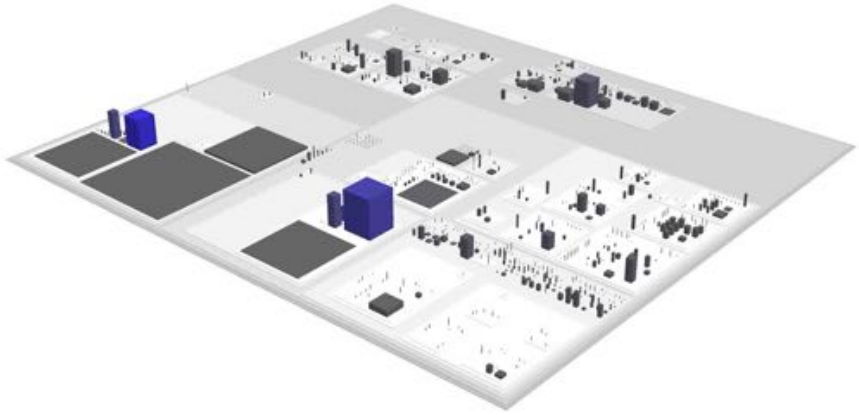
Code City shows where your code lives.

Wettel, Lanza, 2007

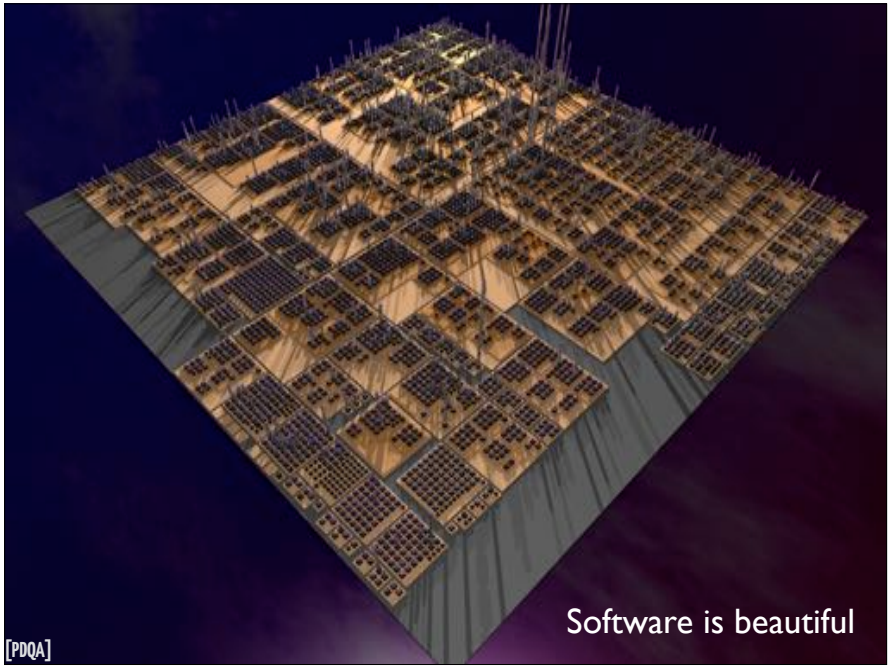


classes are buildings grouped in quarters of packages

[PDQA]



https://www.youtube.com/watch?v=U2MJ3La2y4U



Code as a Crime Scene

4

Embrace The Past

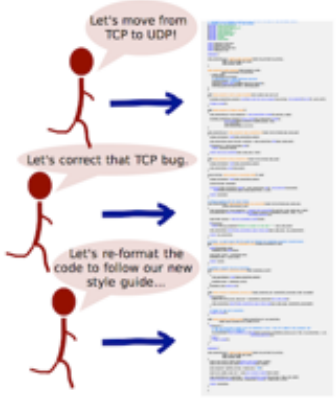
How Software Evolution Lets You Understand Large Codebases

empear

@AdamTornhill

adam.tornhill@empear.com
http://www.empear.com/

Is a Man-Month Still Mythical?



Responsible for the merge

https://pragprog.com/book/atcrime/your-code-as-a-crime-scene

empear

Why Do We Keep Repeating The Same Mistakes?

What's absent?

Time and Social Information!

```
;; We store all game documents in the following collection:
(defparameter *game-collection* "game")

;; We encapsulate all knowledge of the concrete storage
;; medium in the following functions:

(defun game->doc (game)
  ($ ($ "name" (name game))
     ($ "votes" (votes game))))

(defun doc->game (game-doc)
  (make-instance 'game :name (get-element "name" game-doc)
                       :votes (get-element "votes" game-doc)))

(defmethod vote-for :after (game)
  "In this method we update the votes in the persistent storage.
   An after method in CLOS gives us an Observer-like behaviour;
   once the primary method has run, CLOS invokes our after method."
  (let ((game-doc (game->doc game)))
    (db.update *game-collection* ($ "name" (name game)) game-doc)))

(defun game-from-name (name)
  "Queries the database for a game matching the
   given name.
   Note that db.find behaves like Mongo's findOne by default, so
   when we found-games we know there can be only one."
  (let ((found-games (docs (db.find *game-collection* ($ "name" name)))))
    (when found-games
      (doc->game (first found-games)))))

(defun game-stored? (name)
  (game-from-name name))

(defun games ()
  "Returns a sequence of all games, sorted on
```

@AdamTornhill

empear

Embrace The Past

Social information

Time

@AdamTornhill

empear

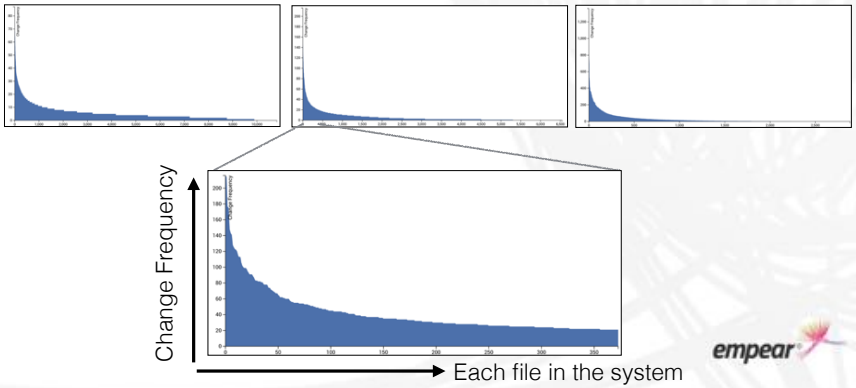
All Code is Equal

…but some Code is more equal than others*

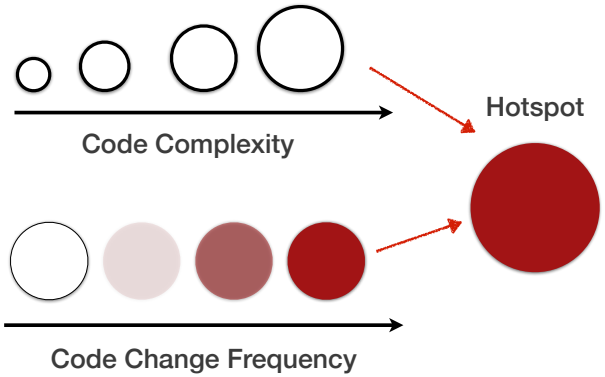
* Sorry, George Orwell

Change Distribution of Files

1 Year in Roslyn (C#, VB) | 6 Years of Erlang | 12 Years of Ruby on Rails

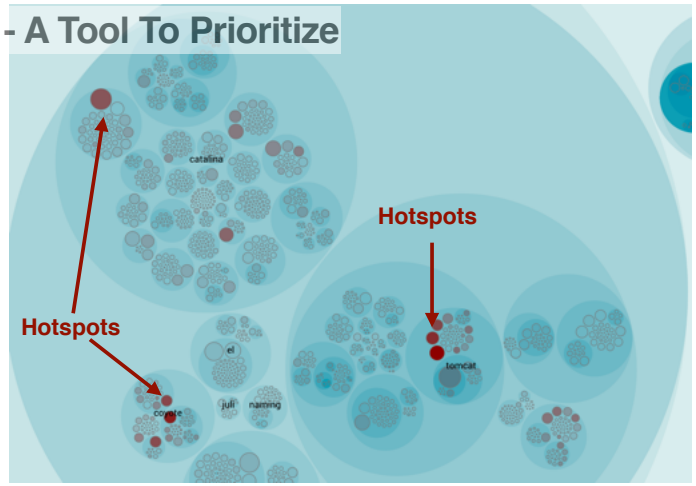


empear

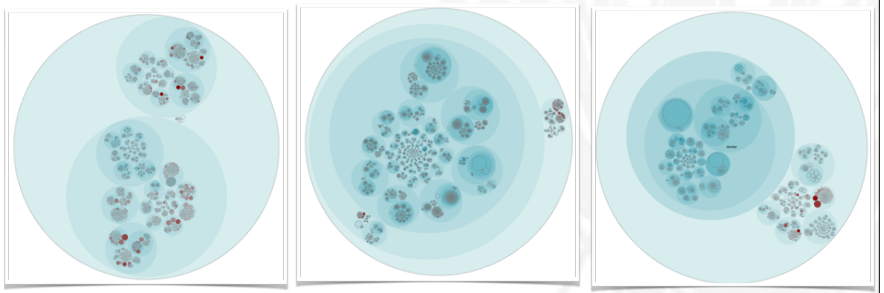


@AdamTornhill

Hotspots - A Tool To Prioritize



Focus on the Code that Matters

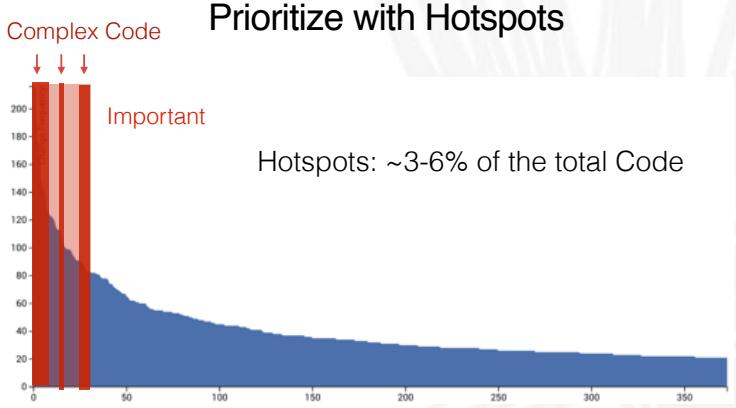


2.6% of the code, **15.5%** effort	**2.5%** of the code, **11.0%** effort	**2.4%** of the code, **13.3%** effort
ASP.NET MVC 200 kLoC	Mono 6.7 MLoC	Docker 500 kLoC

empear

@AdamTornhill

Prioritize with Hotspots

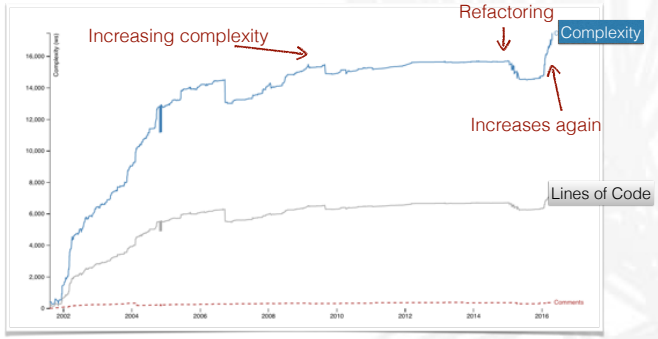


Hotspots: ~3-6% of the total Code

@AdamTornhill

empear

Supervise your Complexity Trends



@AdamTornhill

empear

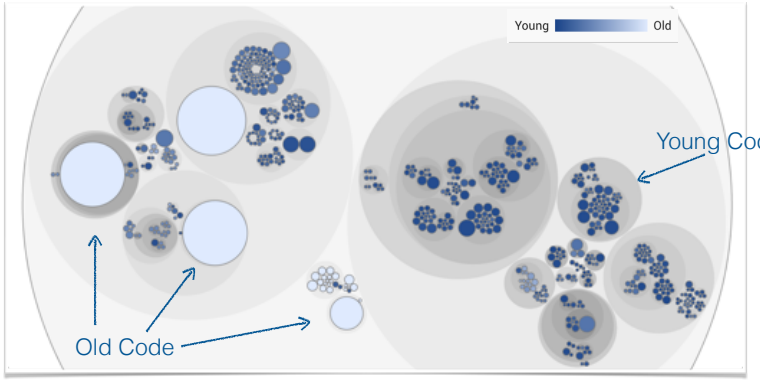
Software Half-Life

> "
> Reducing this half-life means any code you are looking at is likely to be either very recent or old.
>
> We discover that reasoning about code becomes harder when there is lots of code in the grey area between these two.
> "

Dan North https://leanpub.com/software-faster

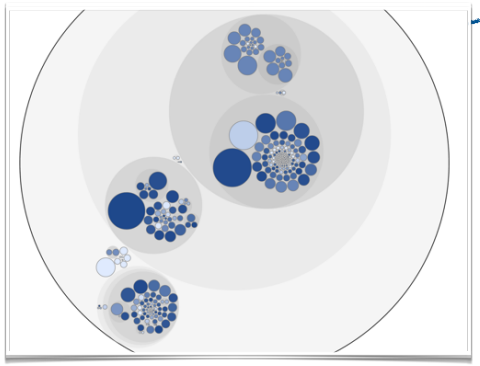
empear

Evolutionary Stable Code



@AdamTornhill

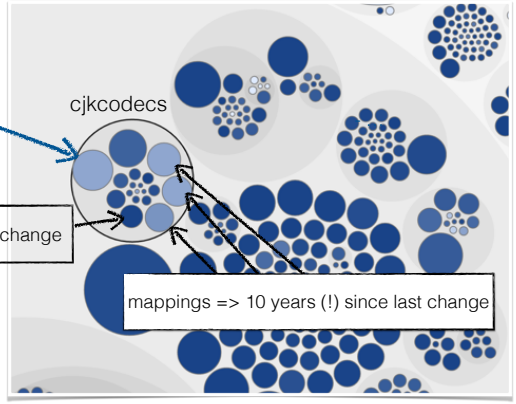
Case Study: Commodities in Clojure



Package: “asm”
- ClassReader.java
- MethodWriter.java
- …

@AdamTornhill

Case Study: CPython



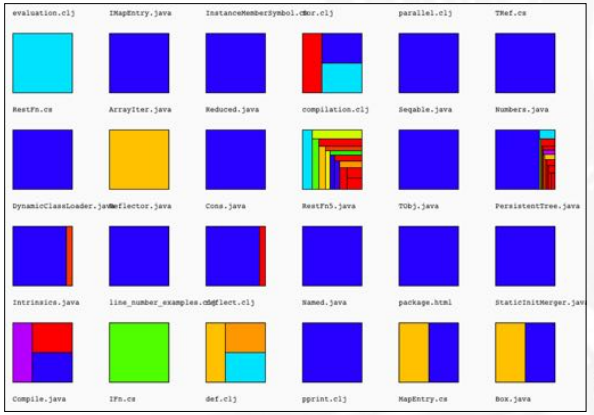
@AdamTornhill

Organizational problems are mistaken as technical issues

@AdamTornhill

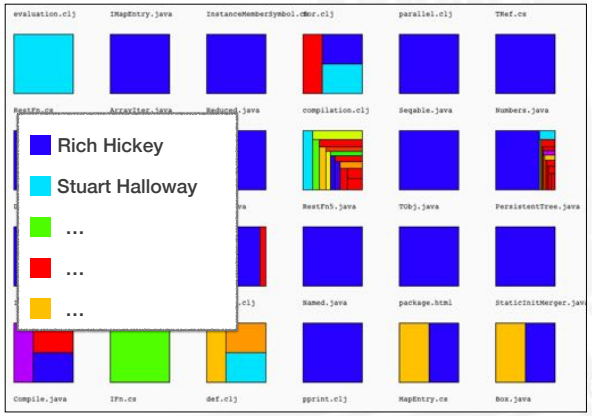
empear

A **Social** View of **Clojure**



empear

A **Social** View of **Clojure**



empear

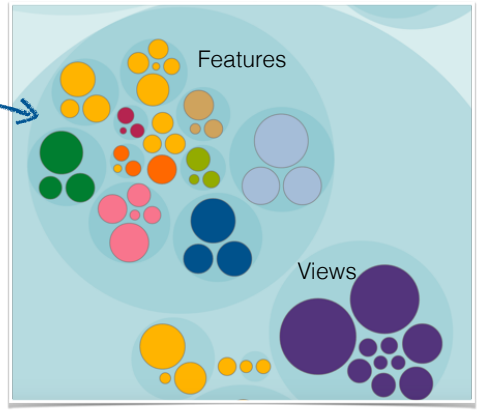
Conway's Law

Any organization that designs a system (defined broadly) will produce a design whose structure is a copy of the organization's communication structure.

https://en.wikipedia.org/wiki/Conway%27s_law

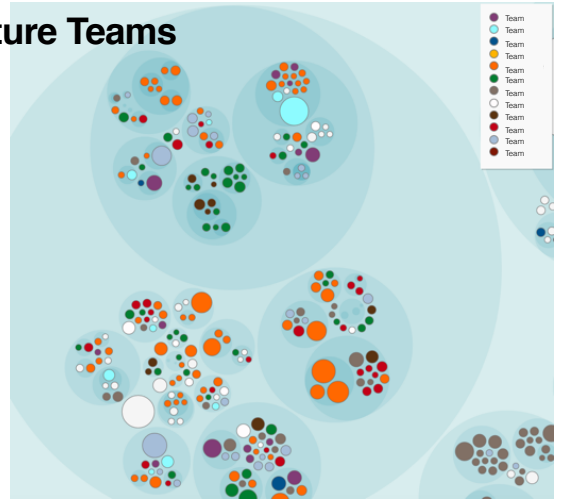
Measuring Conway's Law

Architectural Pattern: Package by Feature



@AdamTornhill

The Perils of Feature Teams



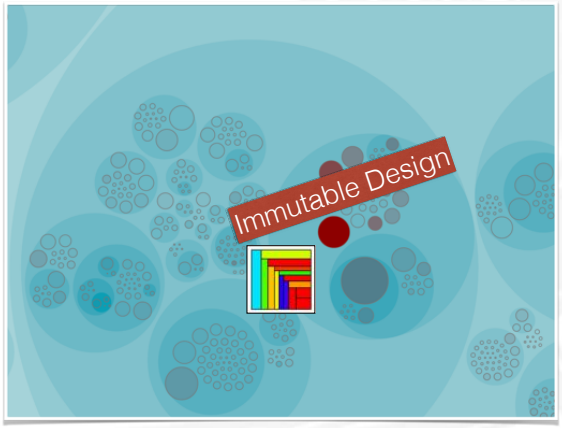
@AdamTornhill

Alternative: The Team as Gatekeeper

@AdamTornhill

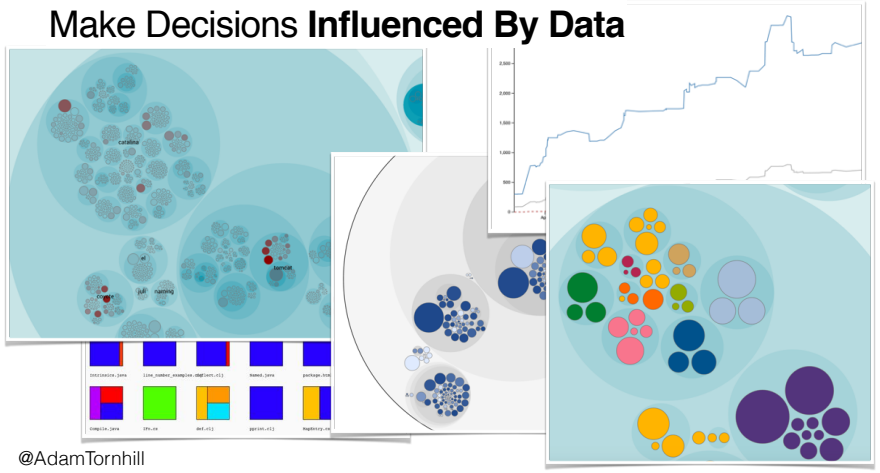
empear

Why Hotspots stay where they are



empear

Make Decisions **Influenced By Data**



@AdamTornhill

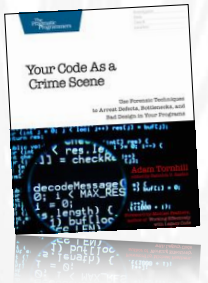
Read More

www.adamtornhill.com/articles/aspnetclones/killtheclones.html

www.adamtornhill.com/articles/socialside/socialsideofcode.htm

The Tools as a Service (work in progress)

https://codescene.io/



@AdamTornhill

adam.tornhill@empear.com
http://www.empear.com/

empear