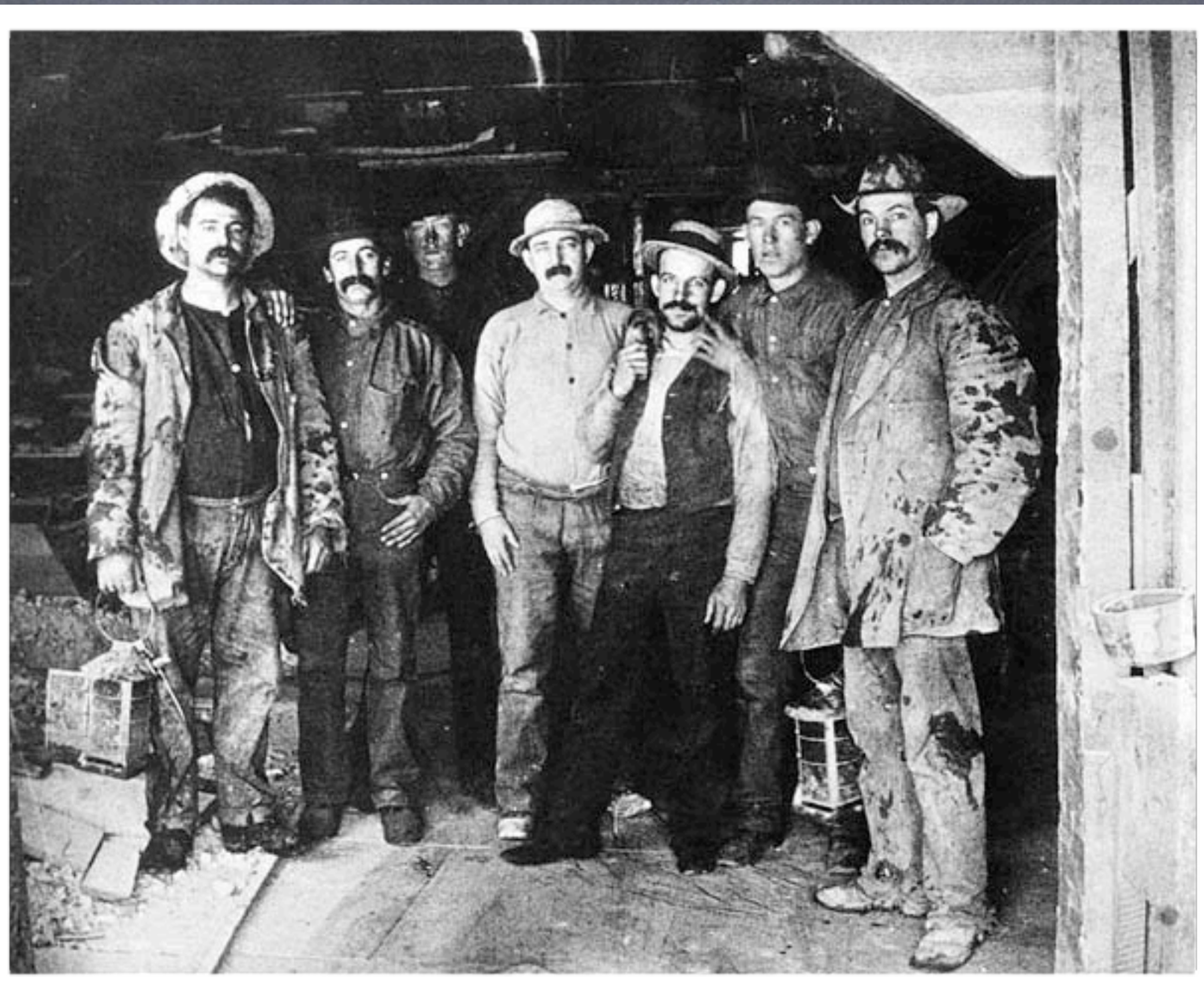
A broad introduction to Mining Software Repositories (and some related subjects)

Software Evolution 2012-2013

Jurgen Vinju

Mining Software Repositories

- Why?
- What?
- How?
- Example!



Repository Mining is a ...

- a potentially very fruitful
- but expensive
- and certainly non-trivial

software engineering activity

MSR Definition

"A broad class of investigations into the examination of software repositories. Here software repositories refer to artifacts that are produced and archived during software evolution. They include sources such as the information stored in source code version-control systems ... requirements/bug-tracking systems ..., and communication archives ...." [1]

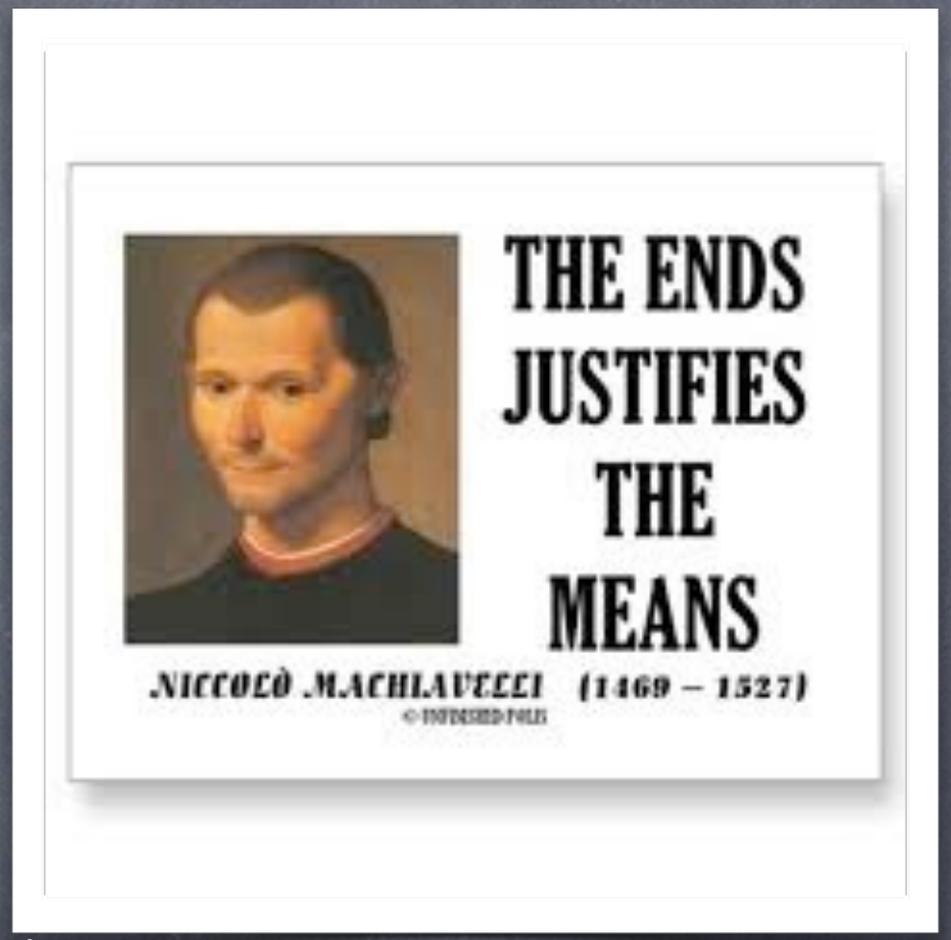
[1] Kadi et al. "A survey and taxonomy of approaches for mining software repositories in the context of software evolution" Journal of Software Maintenance and Evolution: Research and Practice. 2007; 19:77-131. Wiley InterScience

MSR is about evolution

- Software versions -> Time
- Changes -> Differences
- Data -> Source Code, Bug reports, ...
- Meta Data -> Who, when, where, why

MSR includes...

- data recovery
- data analysis
- data measurement
- data mining
- static source analysis
- statistical analysis (correlation)
- and then some...

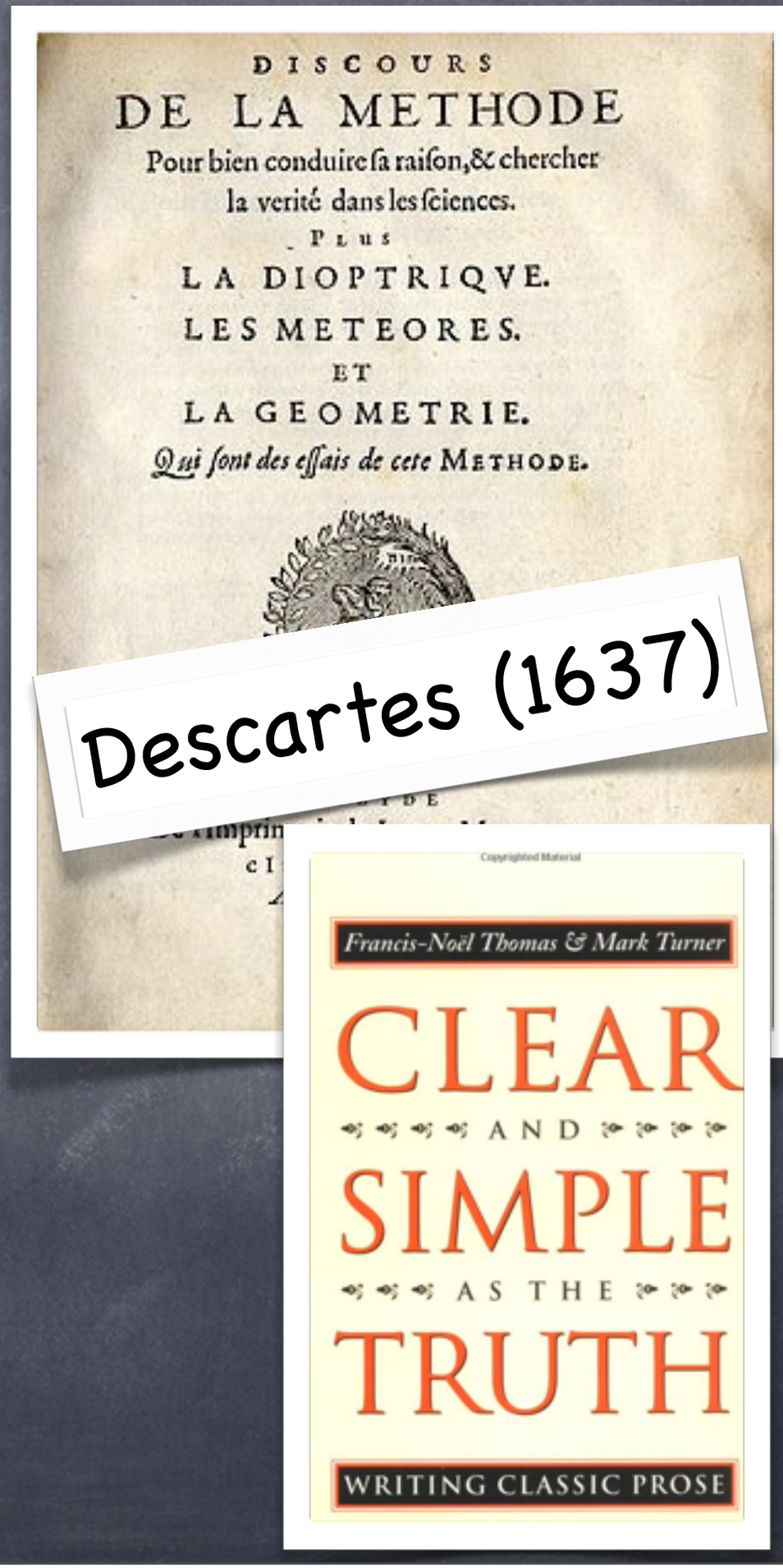


Why MSR?

- Research
  - To verify theories of software evolution in general
  - To search for general theories of software evolution
- Application
  - To exploit theories of software evolution for a particular system or family of systems

La methode scientifique

- Scepticism: doubt is the first certainty
- How to convince your sceptical self and your sceptical colleagues of a truth once discovered?
- As objective as possible
- As accessible as possible



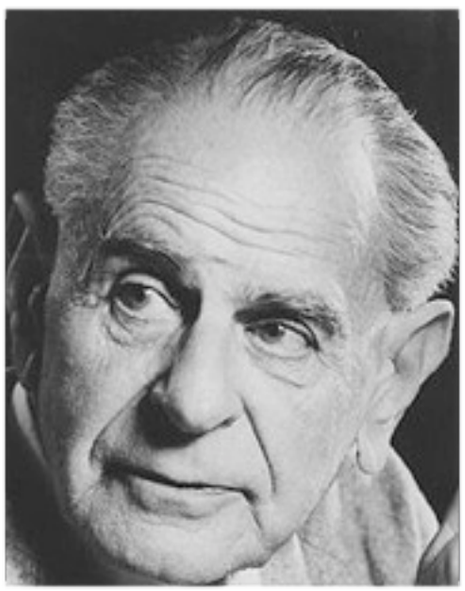
[2] Merriam Webster Dictionary

Scientific Methods

- A scientific method consists of the collection of data through observation and experimentation, and the formulation and testing of hypotheses [2]

- Three ingredients of scientific methods
  - Observation
  - Formulation of theory
  - Testing (falsification)

Karl Popper

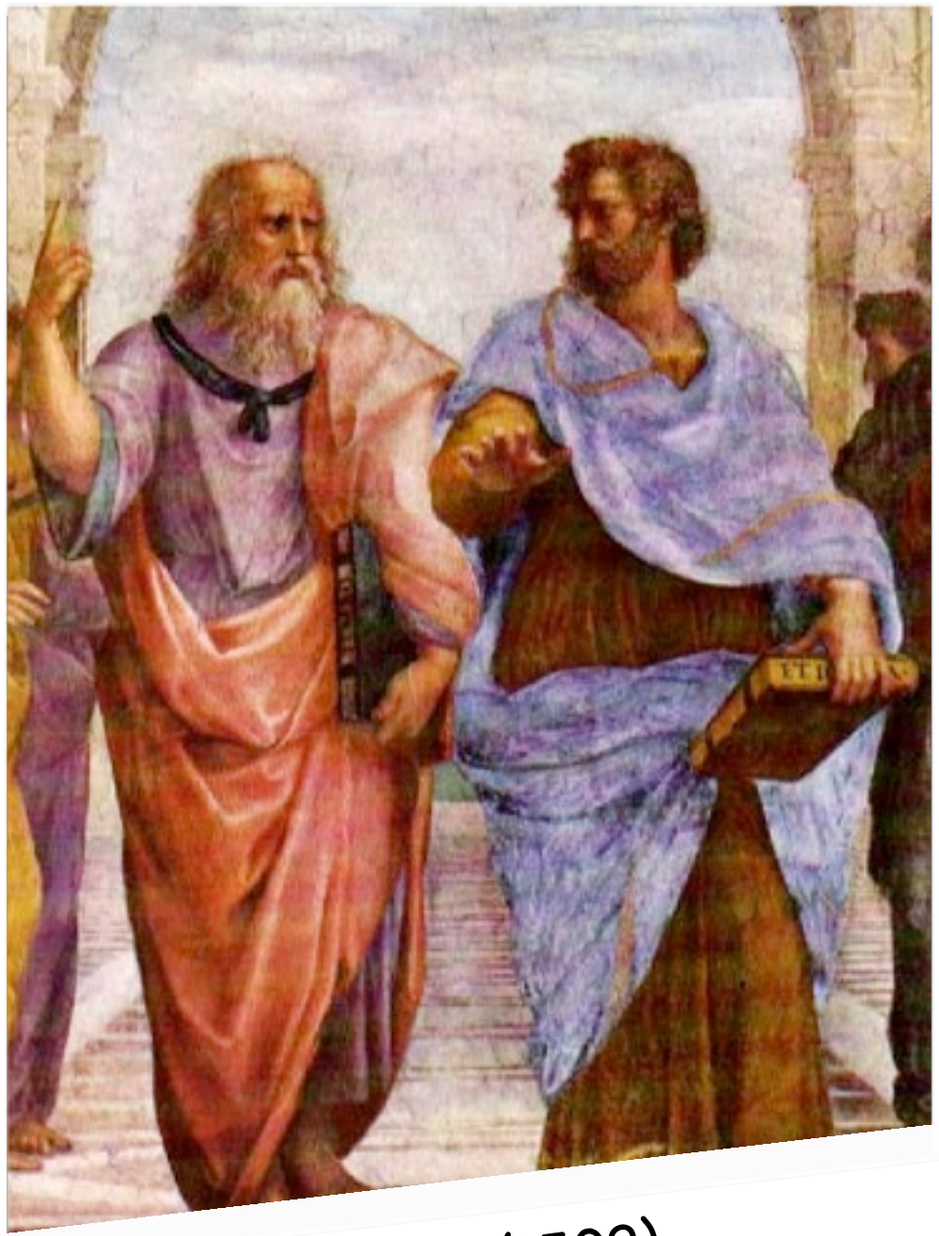


MSR + scientific method

- Software "exists" like butterflies in nature exist
- Study it using similar scientific method
- Reconstruct reality rather than construct new reality (more common in computer science)
- Weird for computer scientists (education!)
- Hard to ask the right questions

Questions? Theories?

- Science starts with good questions
- What do we want to know about software evolution anyway? What theories do we want tested?
- Please brainstorm about this for 5-10 minutes



Raphael (1509)
School of Athens (with Plato & Aristotle)

Question topics [3]

- Developer efforts and social networks
- Change impact and propagation (co-evolution)
- Trends and hotspots (risk management)
- Fault and defect analysis & prediction

[3] M. D'Ambros et al. Analysing Software Repositories to Understand Software Evolution

Example questions [1]

- Which coding style leads to more bugs?
- Which refactorings prevent design degredation?
- How and why does software grow/shrink?
- What is the relation between team size and velocity?
- What evolutionary couplings are there? (which classes change together?)
- When are bugs introduced? (on Friday?)
- Are source code clones bad for maintainability?

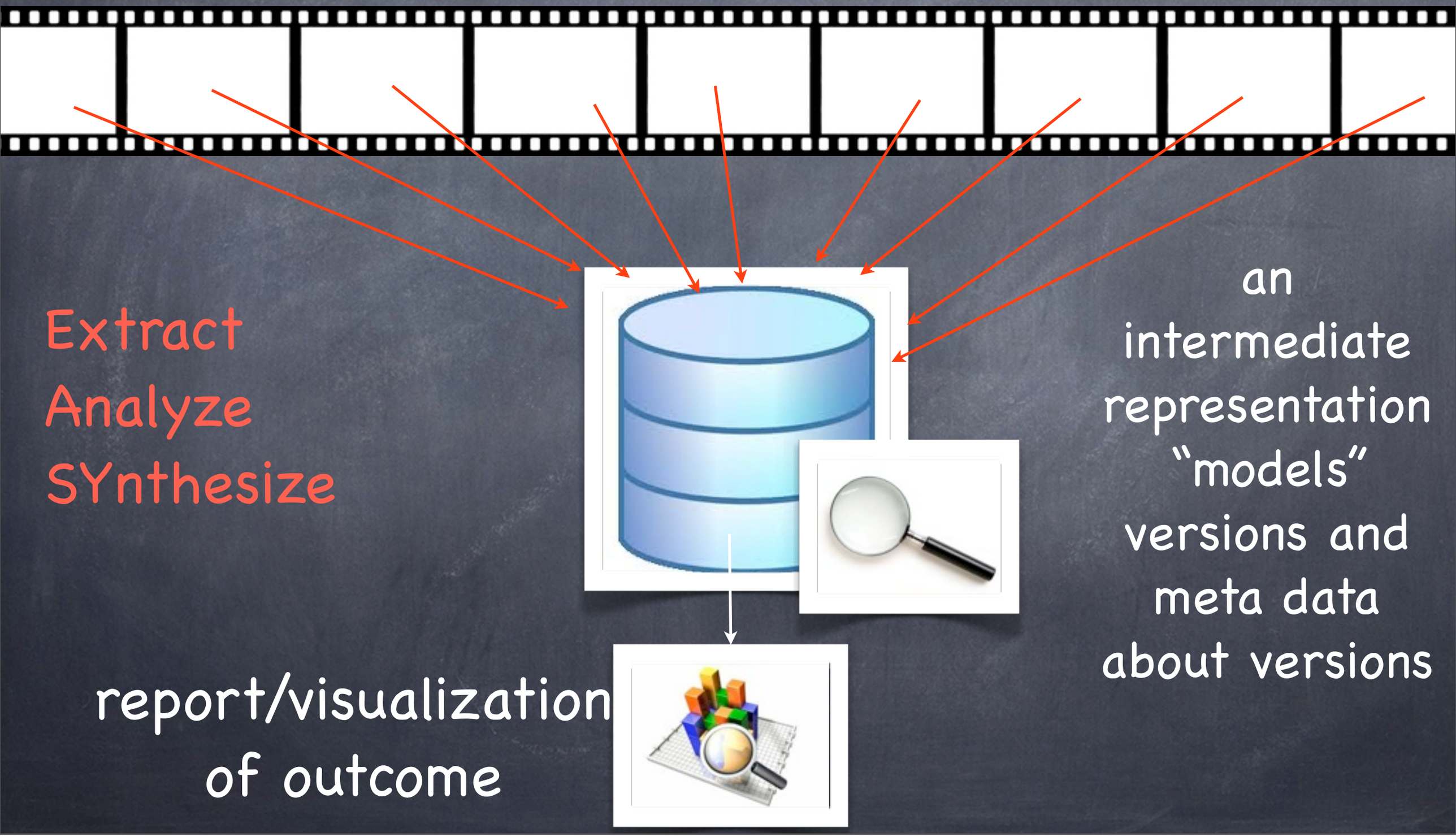
[1] Kadi et al. "A survey and taxonomy of approaches for mining software repositories in the context of software evolution" Journal of Software Maintenance and Evolution: Research and Practice. 2007; 19:77-131. Wiley InterScience

Hypotheses

- Take a question/theory, design an hypothetical MSR experiment which could invalidate it
- What are the major challenges in implementing this experiment?
- Take 10 minutes

"EASY MSR"

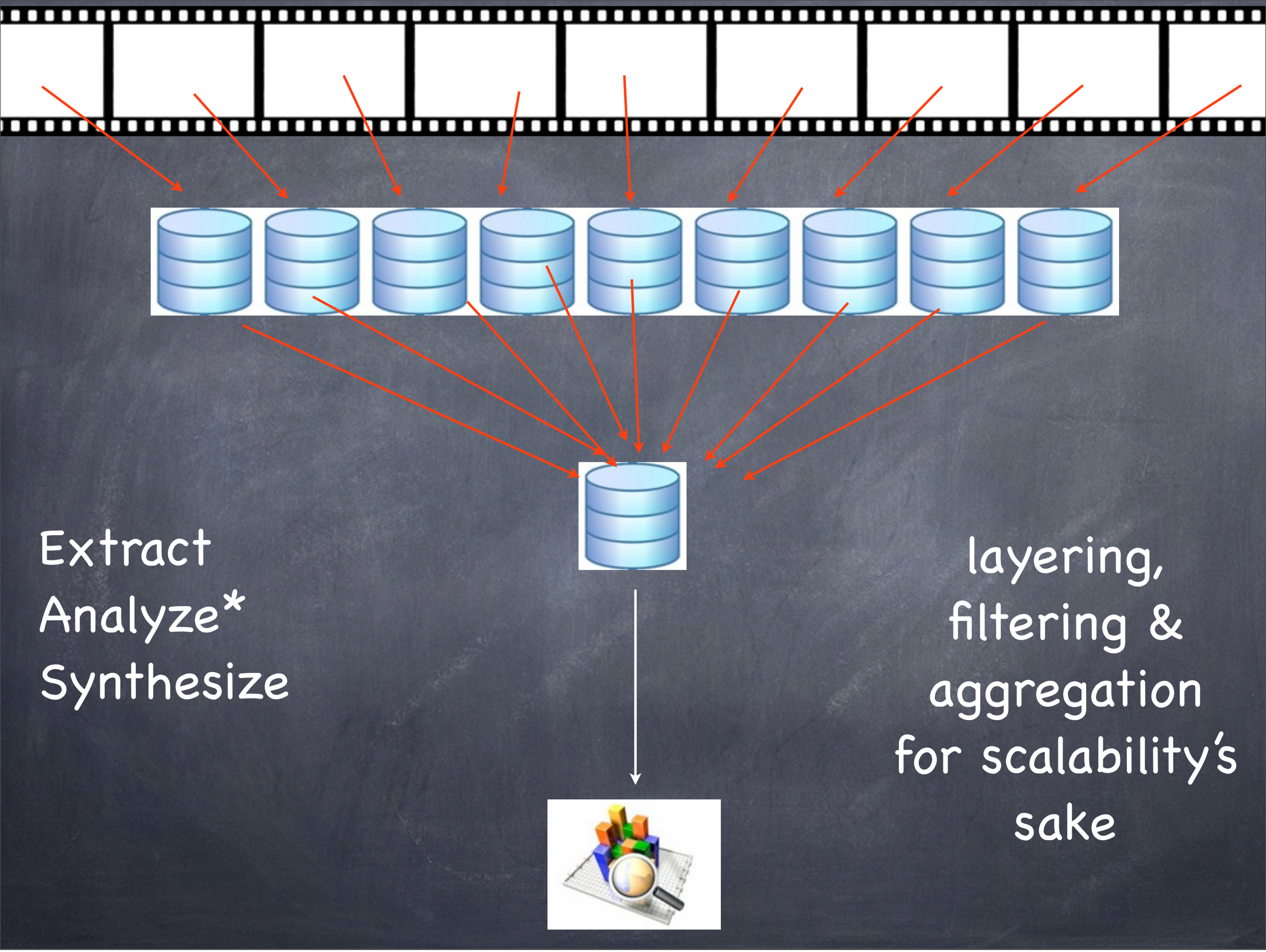
data of several versions is extracted



Extract
Analyze
SYnthesize

an intermediate representation "models" versions and meta data about versions

report/visualization of outcome



Extract
Analyze*
Synthesize

layering,
filtering &
aggregation
for scalability's
sake

HowTo MSR: Challenges

- Selection of input data
- Accuracy of input data
- Memory size
- Speed
- Traceability
- Reproducability
- Accuracy of analysis results

What is a version?

SVN, GIT, CVS, ...?

What is a change?

SVN, GIT, CVS, ...?

How to detect a ... ?

design pattern
refactoring
architectural change
clone
...

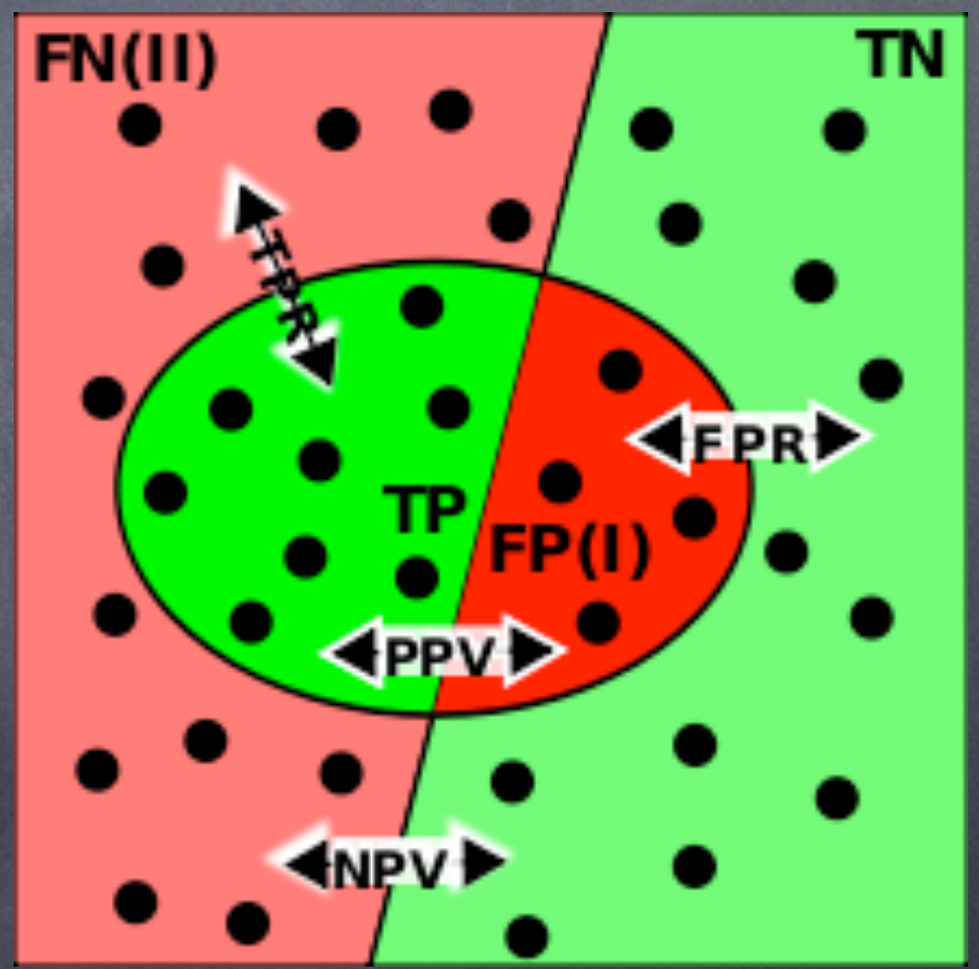
How to detect evolution of a ... ?

design pattern
refactoring
architectural change
clone
...

Hypothesis testing & binary classification

- How many times does your test get it right, how many times does it get it wrong?

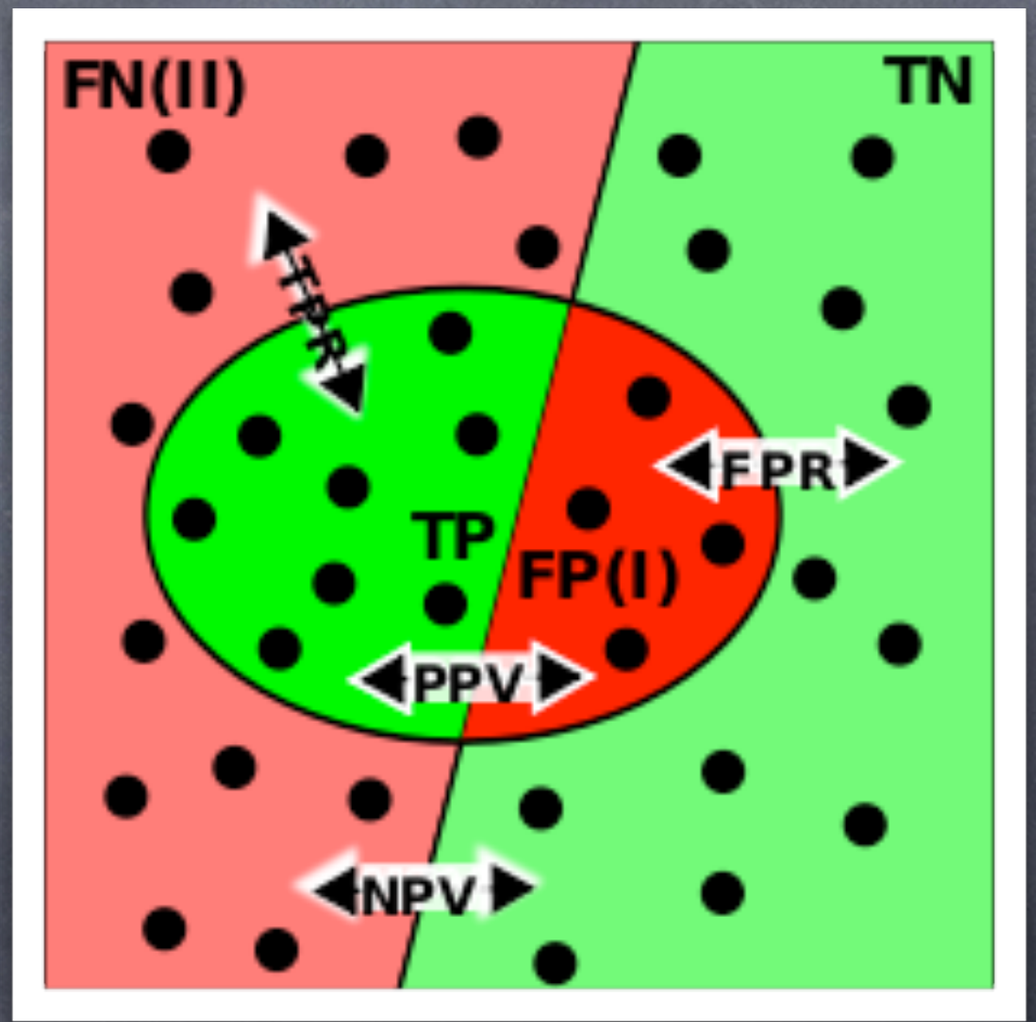
		actual value		
		p	*n*	total
prediction outcome	*p'*	True Positive	False Positive	P'
	n'	False Negative	True Negative	N'
	total	P	N	



[wikipedia]

Measuring accuracy

- TPR: **recall** (how many of the right results do you have)
- PPV: **precision** (how correct are the results that you have)
- You need a benchmark or a "golden standard"



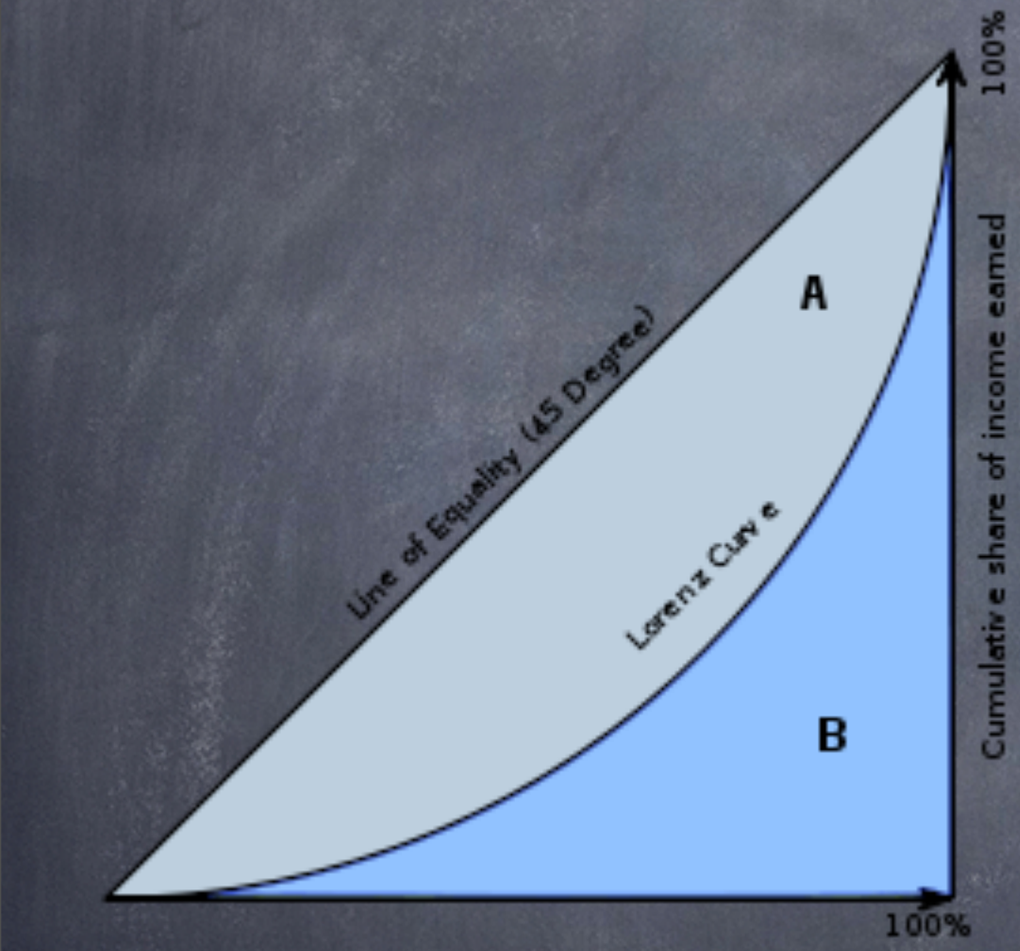
Measuring correlation

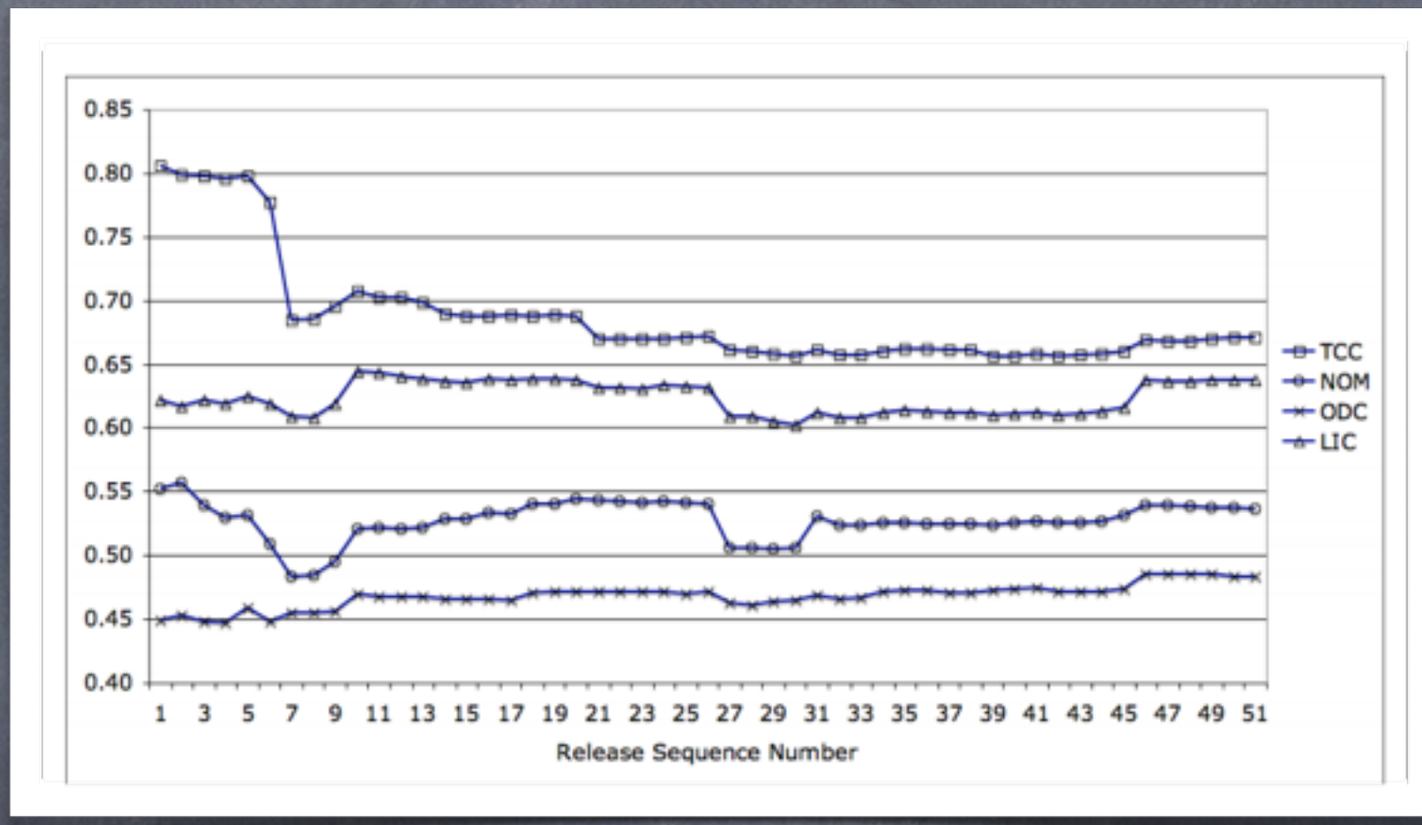
- Can be measured in particular cases (for example using Spearman's coefficient)
- Correlation does not imply causation, but it may indicate it
- Use statistical correlation to form hypotheses
- Correlation metrics may assume certain statistical models (be careful!)

Measuring evenness

- See how measurements are distributed (suchs as number of methods per class)
- Usually not "normal", so hard to measure in terms of averages and deviations
- Use Lorenz curves and Gini coefficients to summarize evenness does not assume statistical model
- Check out: Nierstraz et al. "Comparative Analysis of Evolving Software Systems Using the Gini. Coefficient" ICSM 2009

Lorenz & Gini





Gini = A / (A + B)

Gives you versions of interest

Dealing with uncertainty

- You have an interesting theory
- You have a lot of (messy) data
- You have a fuzzy detection method
- Problem: How to come to the right conclusion?
- Answer: focus on **invalidation** (objectivity)
- Answer: focus on **visualization** (accessibility)

One case of MSR

- An empirical study on the maintenance of source code clones [4]
- Suresh Thummalapenta, Luigi Cerulo, Lerina Aversano, Massimiliano Di Penta
- Questions:
  - How are clones maintained? (consistently/inconsistently, synchronous/asynchronous)
  - How do bugs correlate with these classes of clones?
  - How do we measure how clones are maintained?
  - How do clone detection algorithm parameters affect our results?

Tracing clone classes through revisions

- It all starts from "change sets" (logical commit)
- A clone class is a set of cloned pieces of code
- A clone class evolves with the code
- You need to invent a smart way to "diff" clone classes, given any weird change to the code
- Then you classify what happens to each class

Clone Evolution Patterns [4]

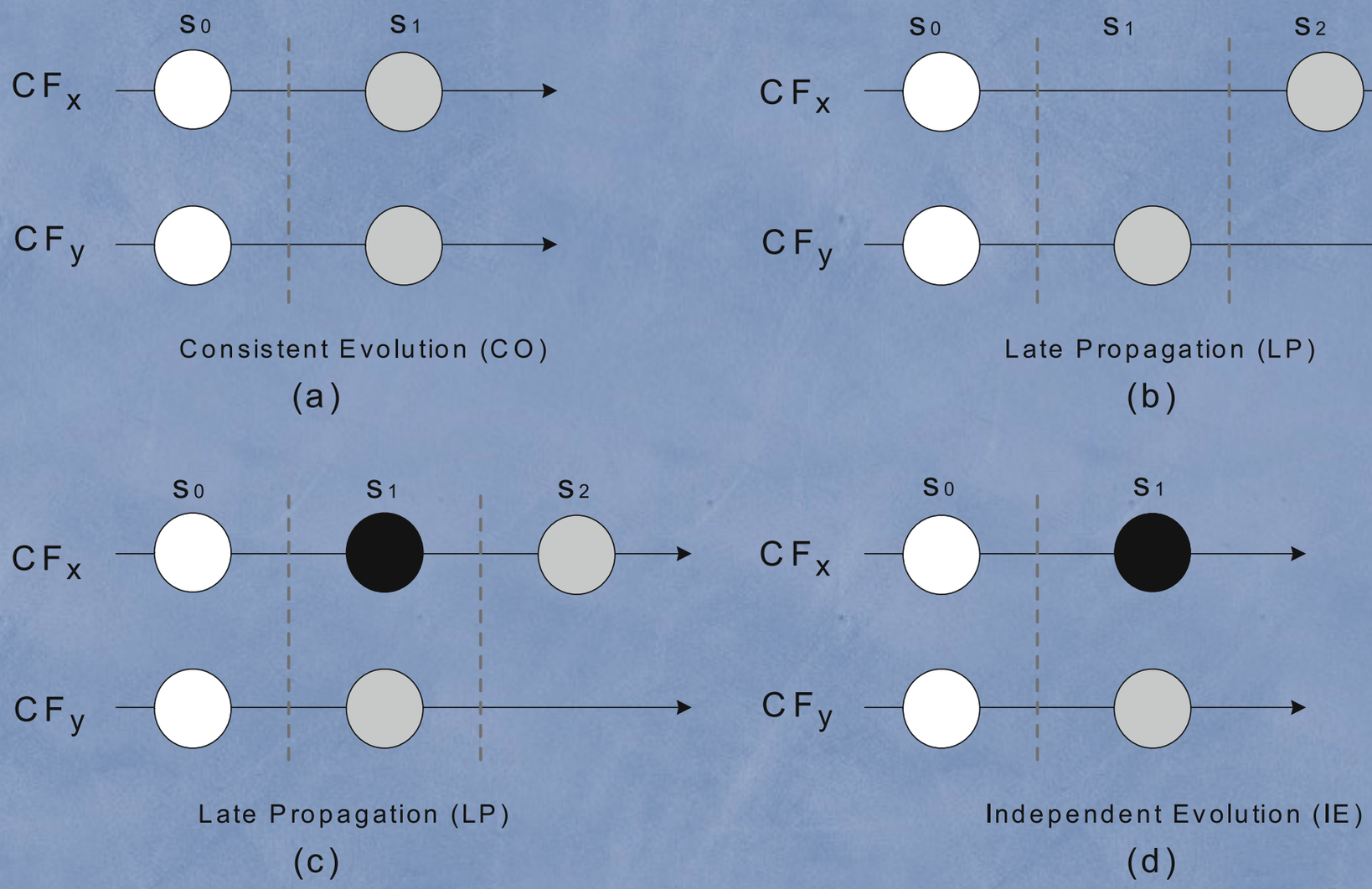


Fig. 2 **a–d** Clone evolution patterns for two clone fragments CF_x and CF_y

Clone Class Pairs

- Between each two consecutive versions find the pairs of identical clone classes
- Then decide in which evolution pattern they fall (previous slide)
- Then, over all versions, decide which is the dominant pattern for each clone class
- Then, analyse data, summarize and find correlations to confirm or reject hypotheses

Research Questions

- What is the percentage of clones following the different evolution patterns?
- Is there any relation between the clone finding parameters and the evolution pattern?
- Is there any relation between evolution patterns and bug fixing?

Results (from [4])

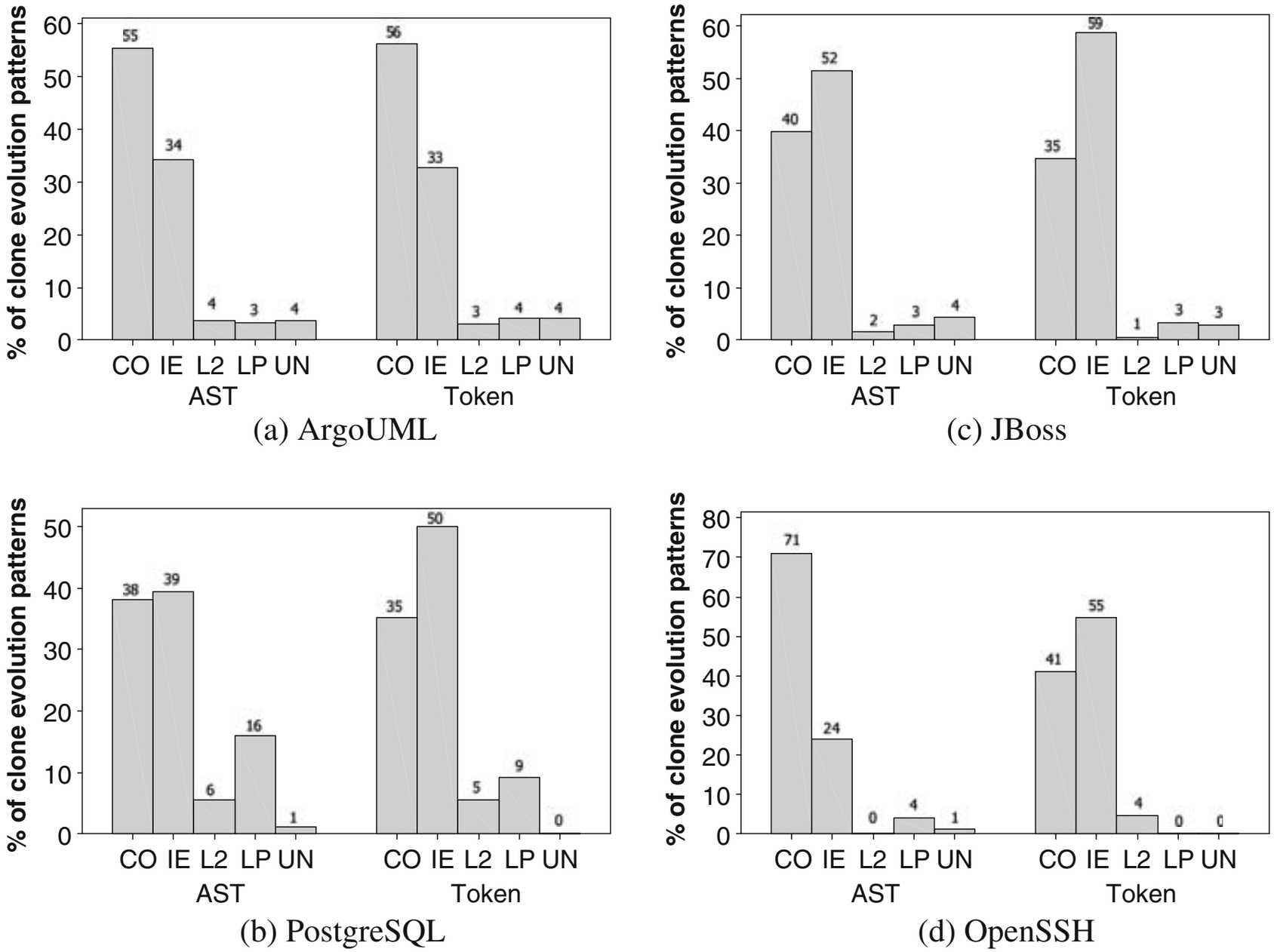


Fig. 5 a–d Clone evolution patterns among different clone detectors (*figures on bars* indicate percentages)

Conclusions

- Clones are often consistently edited immediately
- Clones are often used for templating (IE)
- Clone detection methods do not influence evolution pattern much
- Relation to bug fixes show observable differences between evolution patterns, but nothing conclusive.

Threats to validity

- construct validity: do they measure/observe the right things?
- internal validity: do they draw the right conclusions from the results?
- external validity: does this mean anything for anybody else?
- reliability validity: is this reproducible?

Threats to validity

- construct validity: do they measure/observe the right things?
- internal validity: do they draw the right conclusions from the results?
- external validity: does this mean anything for anybody else?
- reliability validity: is this reproducible?

Read the paper and think about it

MSR with Rascal

- Rascal has an experimental language independent model for repository analysis (SVN, CVS and GIT) available online
- Waruzj Shahbazian. "Rminer: A integrated model for repository mining using Rascal" 2010. Universiteit van Amsterdam
- It can be "easily" combined with other analysis written in Rascal

MSR with Rascal

- Rascal has an experimental language independent model for repository analysis (SVN, CVS and GIT) available online
- Waruzj Shahbazian. "Rminer: A integrated model for repository mining using Rascal" 2010. Universiteit van Amsterdam
- It can be "easily" combined with other analysis written in Rascal

Take home messages

- Empirical research = all about evidence, not proof
- Mining Software Repositories = Field of opportunity