Quantifying the Encapsulation of Implemented Software Architectures

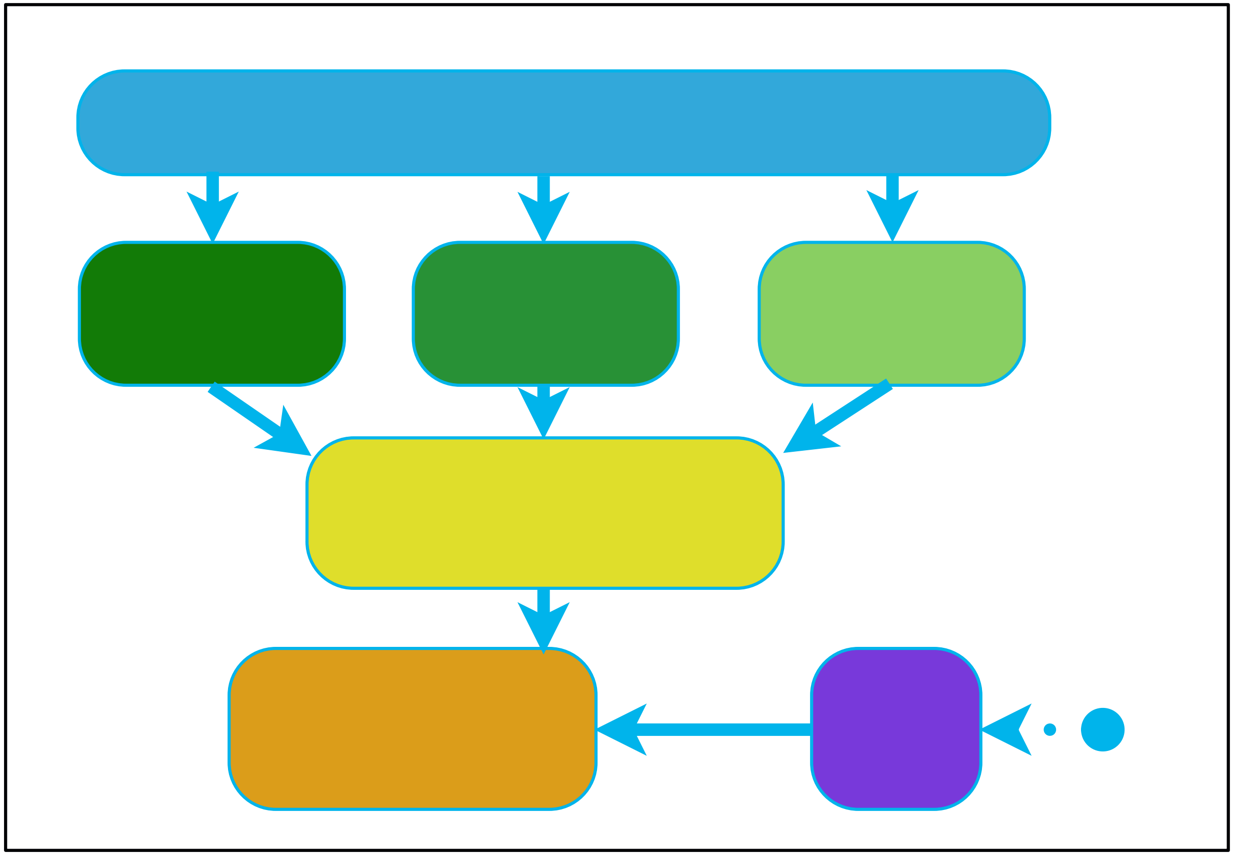
@ericbouwers
@avandeursen
@jstvssr

Software Improvement Group

Radboud University Nijmegen

TU Delft
Delft University of Technology

an *implemented* software architecture

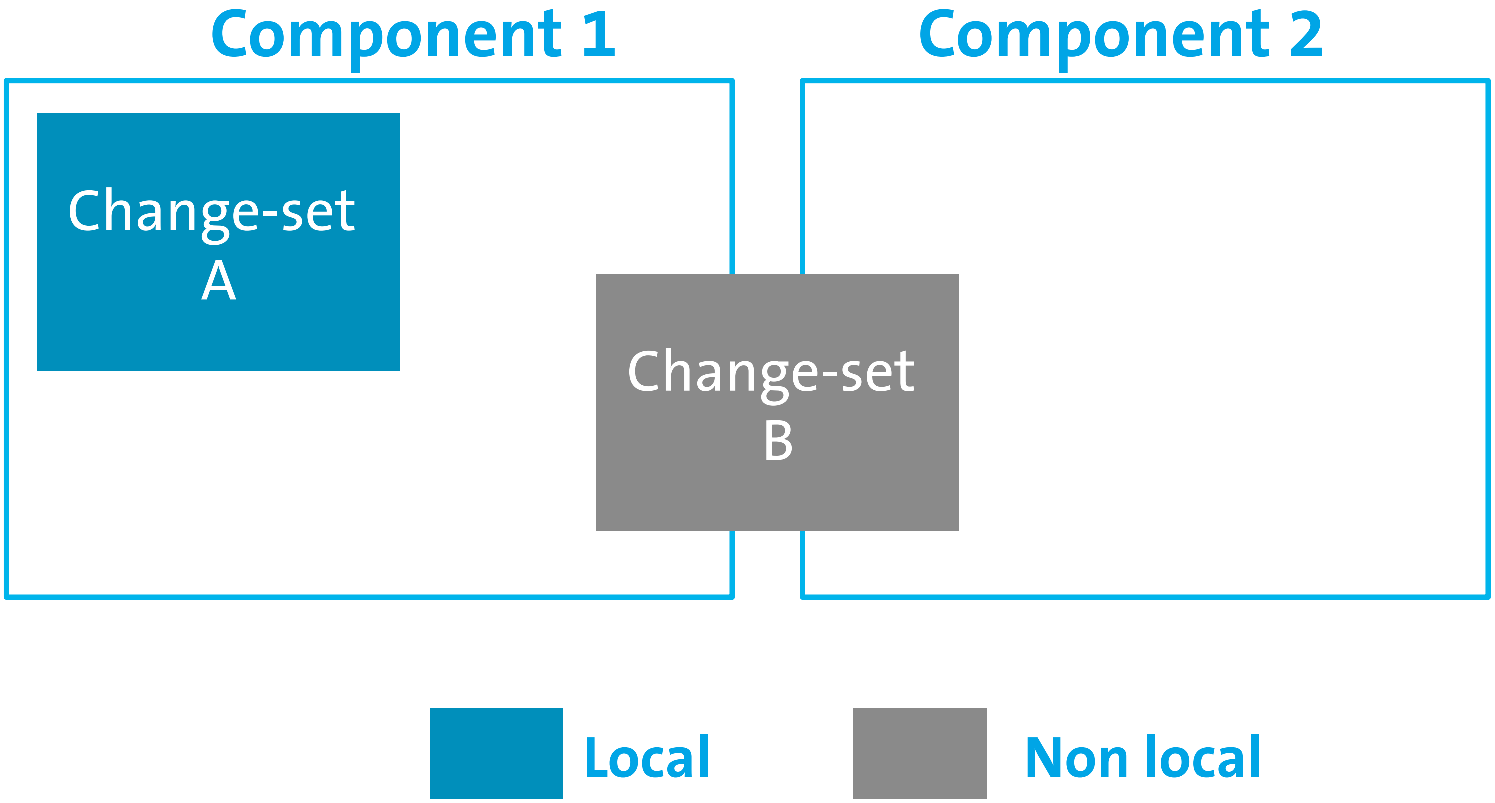


What is encapsulation?

“When applied correctly, the process of encapsulation ensures that the design decisions that are likely to change are localized”

G. Booch. *Object-oriented analysis and design with applications (2nd ed.)*. Benjamin-Cummings Publishing Co., Inc., Redwood City, CA, USA, 1994.

Encapsulation == local changes



Component 1

Component 2

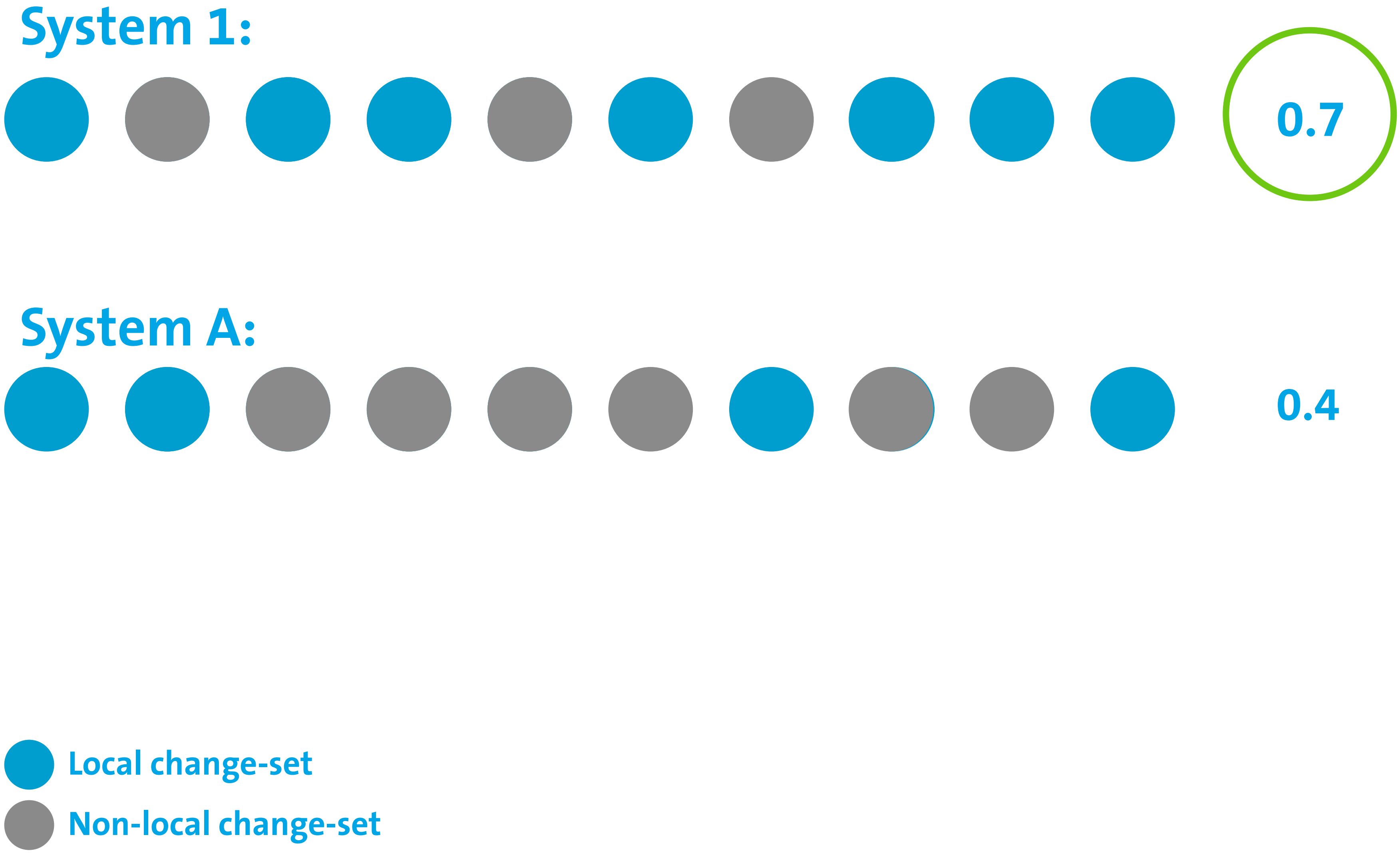
Change-set A

Change-set B

🟦 Local

▨ Non local

Historical encapsulation



System 1:

0.7

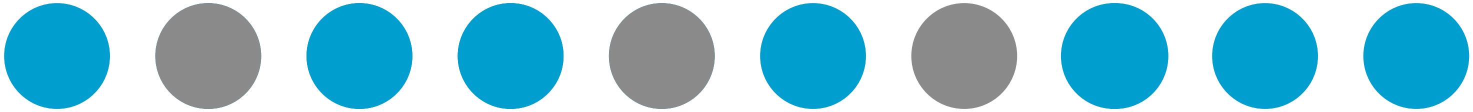
System A:

0.4

Local change-set

Non-local change-set

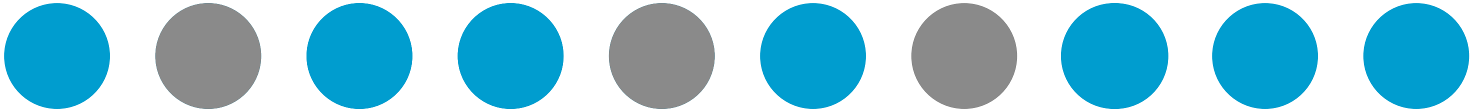
Problem?



You need to have the change-sets!

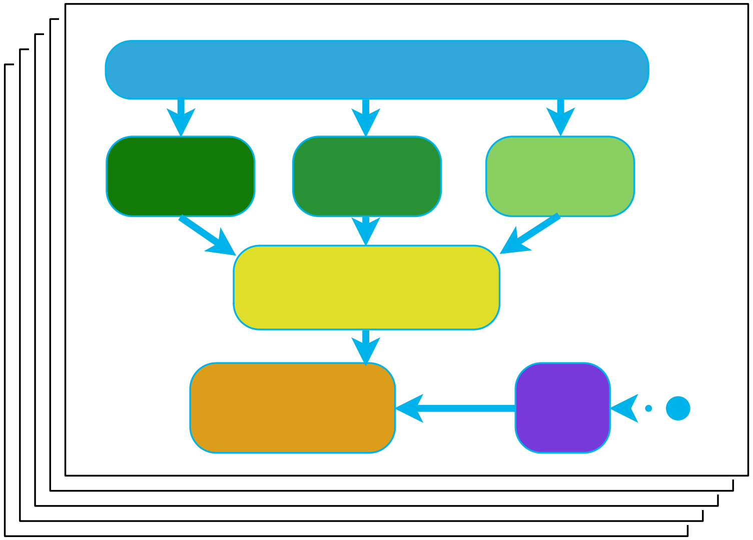
Solution: find related snapshot metric!

System 1, 2, ... N:



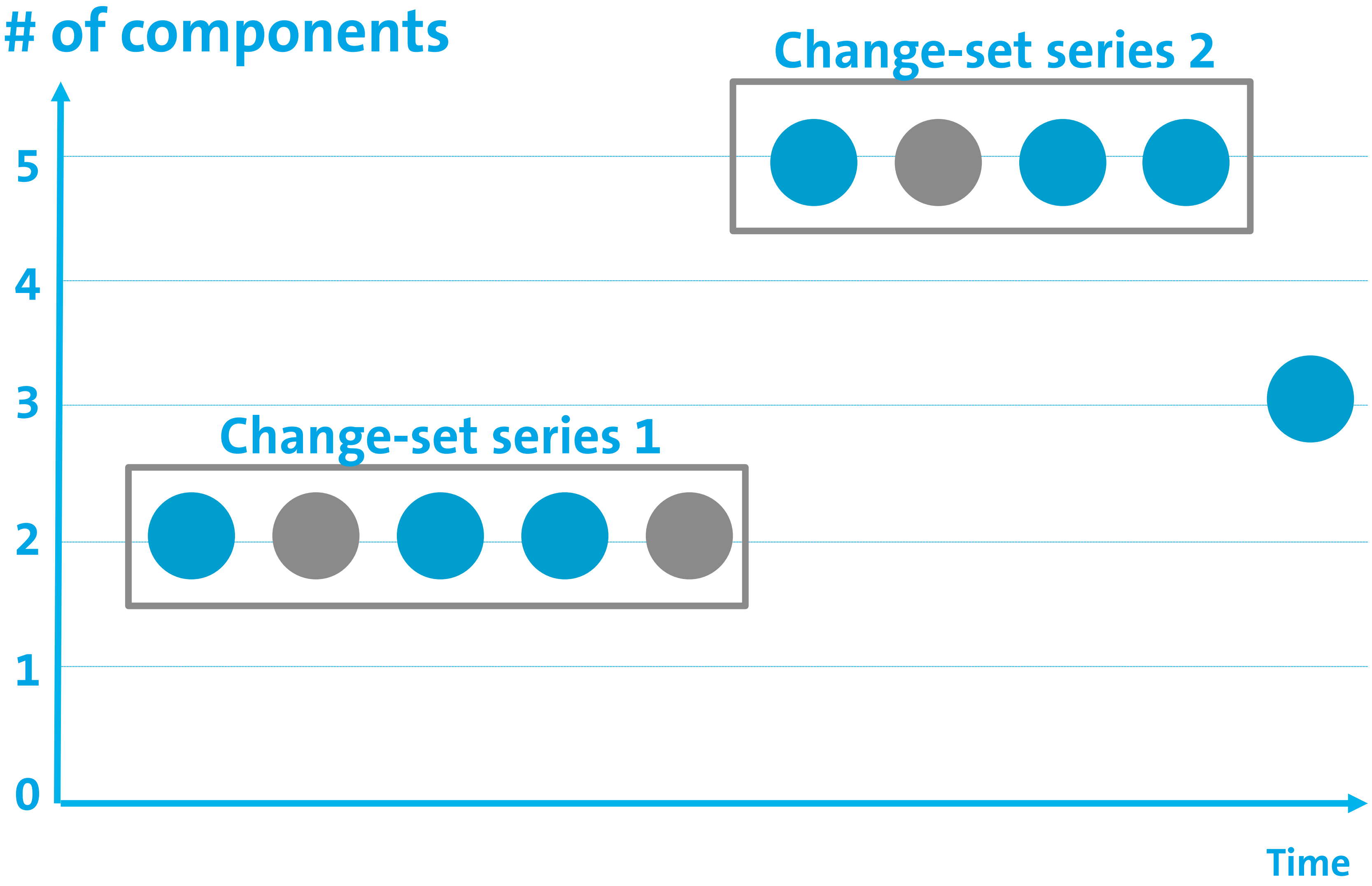
0.7

Correlation?

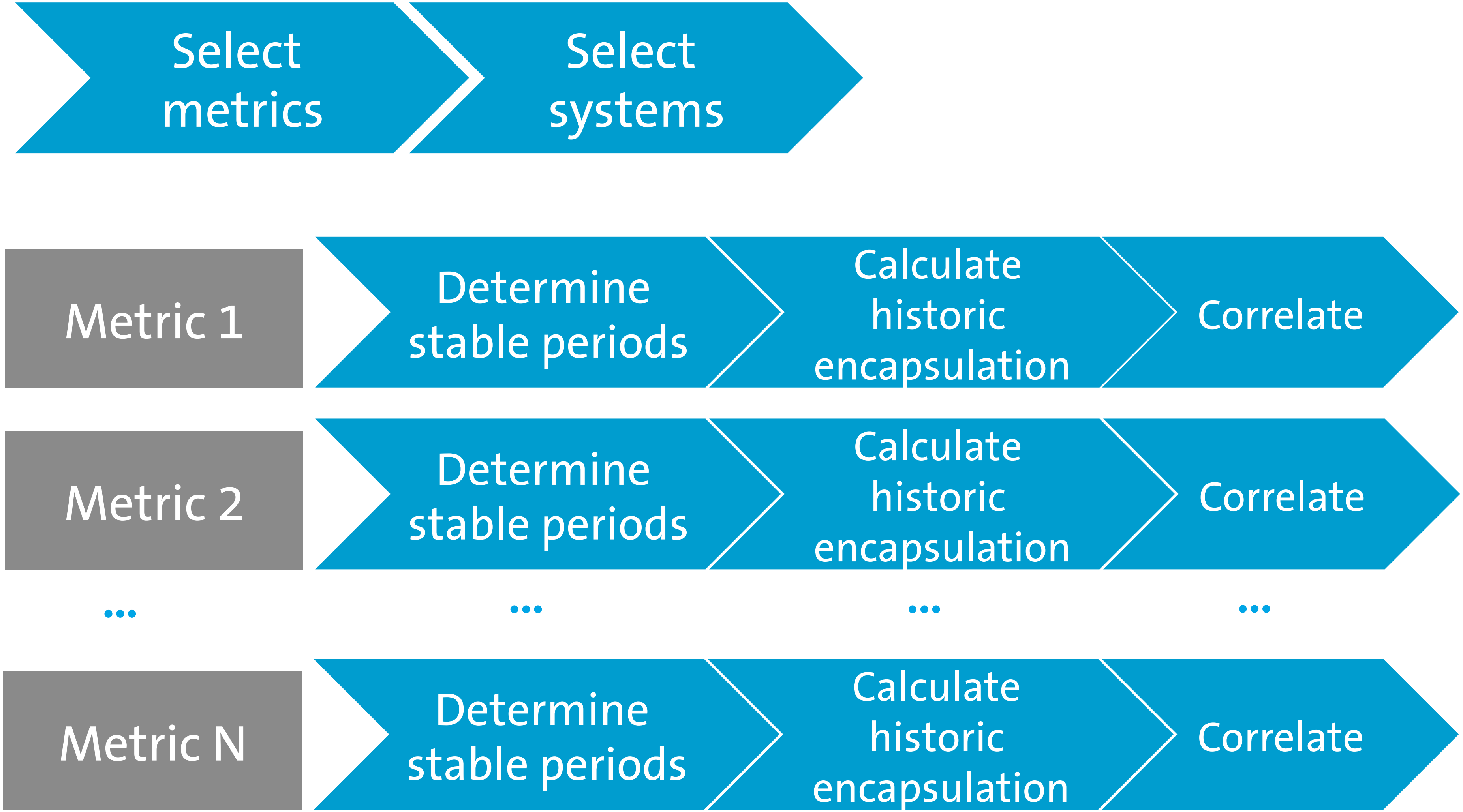


# components:	7
# dependencies:	8
....	
....	

What to correlate?



Experimental design



Select metrics

Select systems

Metric 1

Determine stable periods

Calculate historic encapsulation

Correlate

Metric 2

Determine stable periods

Calculate historic encapsulation

Correlate

…

…

…

…

Metric N

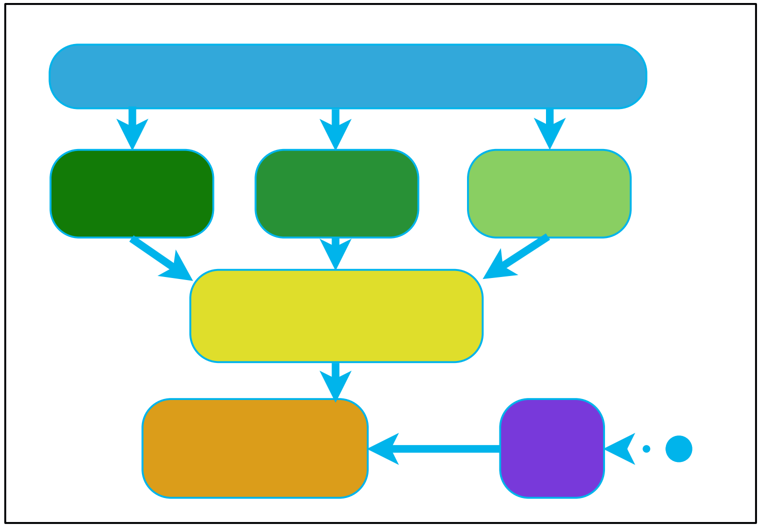
Determine stable periods

Calculate historic encapsulation

Correlate

Selected metrics

Ratio of Cohesive Interactions (RCI)
Cumulative Component dependency (CCD)
Average CCD (ACD)
Normalized CCD (NCD)
Cyclic Dependency Index (CDI)
Inbound Code (IBC)
Outbound Code (OBC)
Internal Code (IC)
Number of Binary Dependencies (NBD)



Component Balance (CB)
Module Size Uniformity (MSUI)
Number of Components (NC)

Selected systems

<APACHE ANT>

Apache Beehive Project

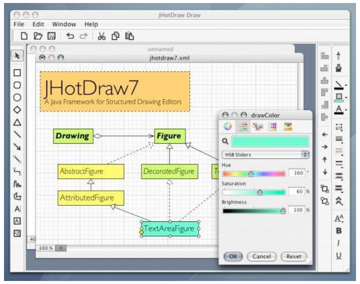
jE

JasperReports

Lucene

UNIVERSITY OF MARYLAND 18 56

FindBugs



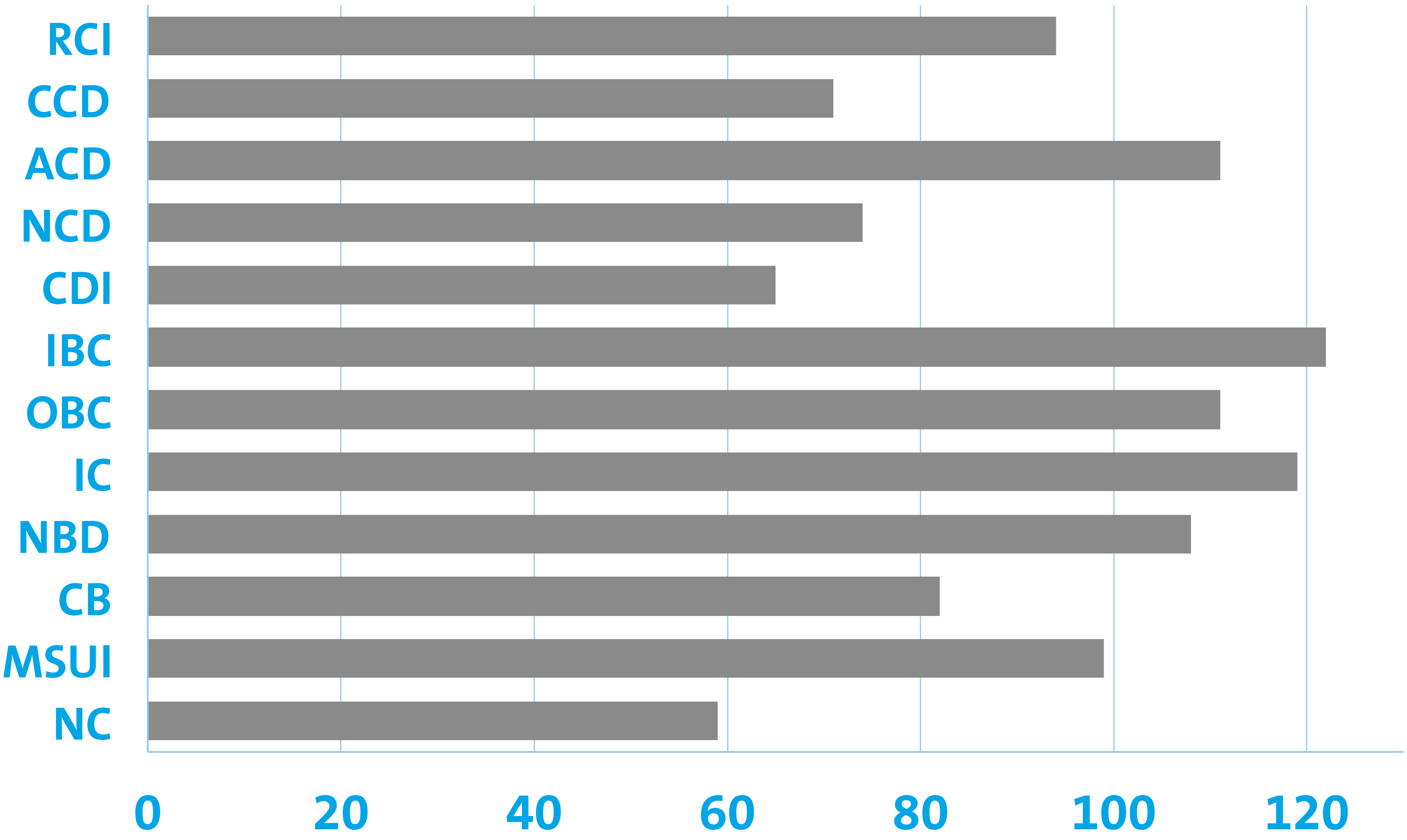
Struts 2

1+ year development

Subversion

Top-level package = component

Number of periods per metric



What are normal values?

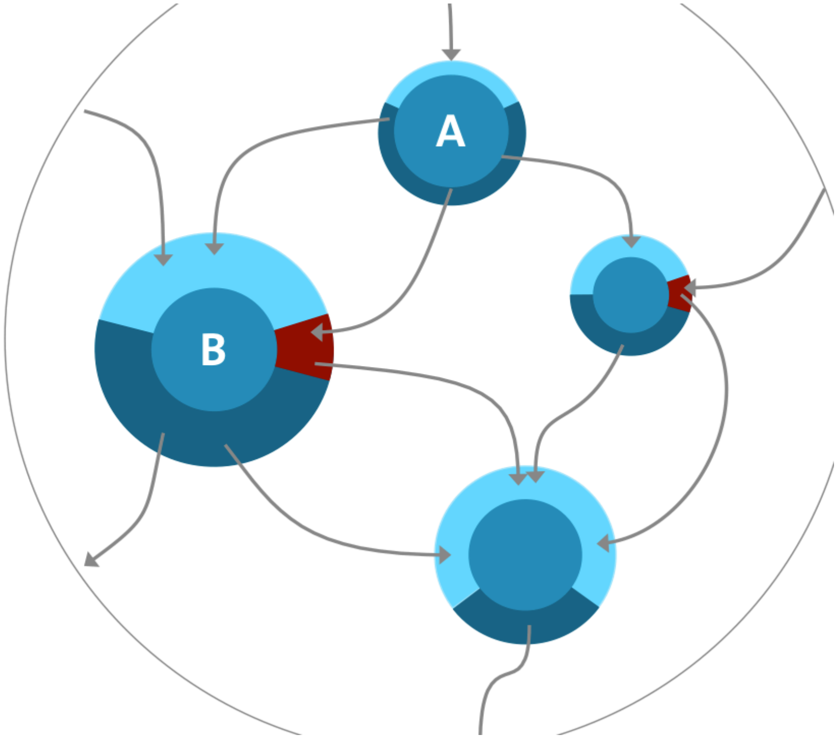
Metric	Median length (months)	Median ratio of local change
RCI	4.0	0.84
CCD	6.0	0.84
ACD	3.0	0.85
NCD	4.5	0.84
CDI	6.0	0.84
IBC	3.0	0.86
OBC	3.0	0.86
IC	2.0	0.86
NBD	3.0	0.84
CB	3.0	0.86
MSUI	3.0	0.84
NC	6.0	0.83

All data available on http://www.sig.eu/en/QuantifyingEncapSA

The correlation results

Metric	Correlation	P-value (corrected)
RCI	0.16	11.3
CCD	-0.27	0.13
ACD	-0.26	0.04
NCD	-0.19	0.59
CDI	0.32	11.94
IBC	-0.30	< 0.01
OBC	-0.31	< 0.01
IC	0.47	< 0.01
NBD	-0.22	0.14
CB	0.29	0.05
MSUI	-0.08	2.42
NC	-0.26	0.27

Dependency profiles



🟦	*Inbound*:	provides interface
🟦	*Internal*:	no dependencies
🟦	*Outbound*:	requires interface
🟥	*Transit*:	mixed inbound /outbound code

More internal code is related to more local change

E. Bouwers, et.al. *Dependency profiles for software architecture evaluations*. ICSM 2011

The interpretation

“The percentage of internal code can serve as an indicator for the success of encapsulation of an implemented software architecture.”

We @ International Conference of Software Maintenance and Evolution 2014

The threats

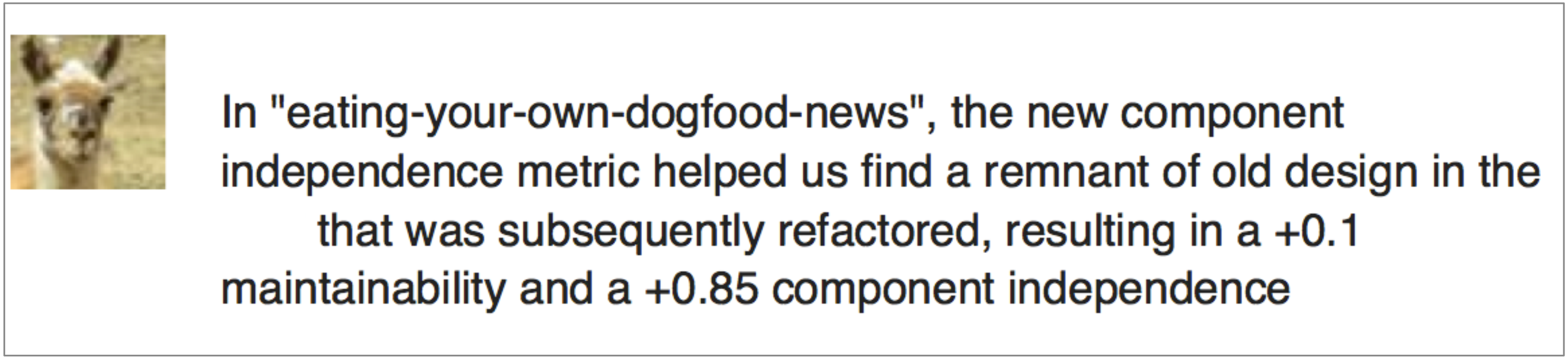
Local change = encapsulation?

When is a metric stable?

Top-level packages = components?

Is there really no relationship with the other metrics?

What did we do with the results?

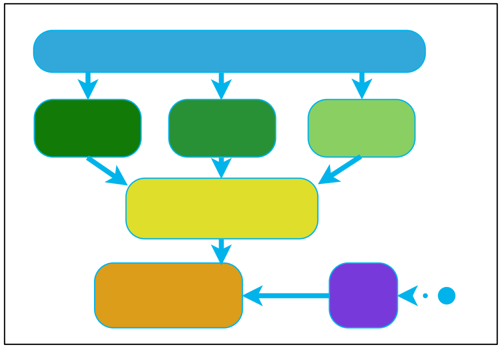


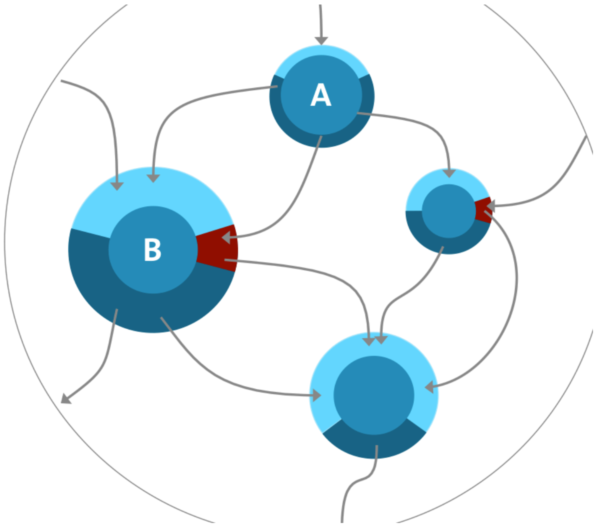
In "eating-your-own-dogfood-news", the new component independence metric helped us find a remnant of old design in the that was subsequently refactored, resulting in a +0.1 maintainability and a +0.85 component independence

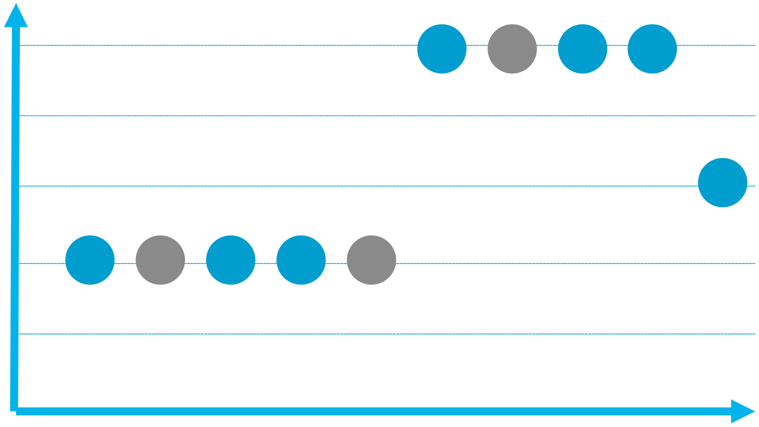
E. Bouwers, et.al. *Evaluating Usefulness of Software Metrics - An Industrial Experience Report,* ICSE SEIP 2013

Summary

VS







“Keep implementation details internal”

@EricBouwers
eric@sig.eu