PPGCC
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
DCC-UFMG

Modeling and Predicting Evolution of Software Quality Internal Attributes

Bruno Luan de Sousa

Advisor: Mariza Andrade da Silva Bigonha (UFMG)

Co-Advisors: Kecia Aline Marques Ferreira (CEFET-MG), and Glaura da Conceição Franco (UFMG)

August 28th, 2020

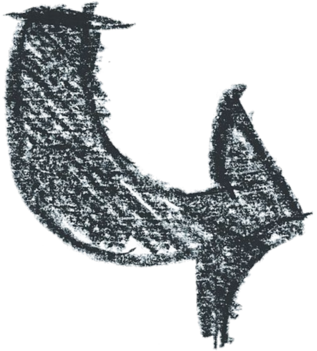
LLP Laboratório de Linguagens de Programação

Agenda

- Introduction
- Goal
- Systematic Literature Review
- Empirical Analysis
- Evolution Properties
- Next Steps
- Conclusion


INTRODUCTION


Change Request
Impact Analysis
Release Planning
System update
System Release
Software Evolution



85% - 90%

Motivation

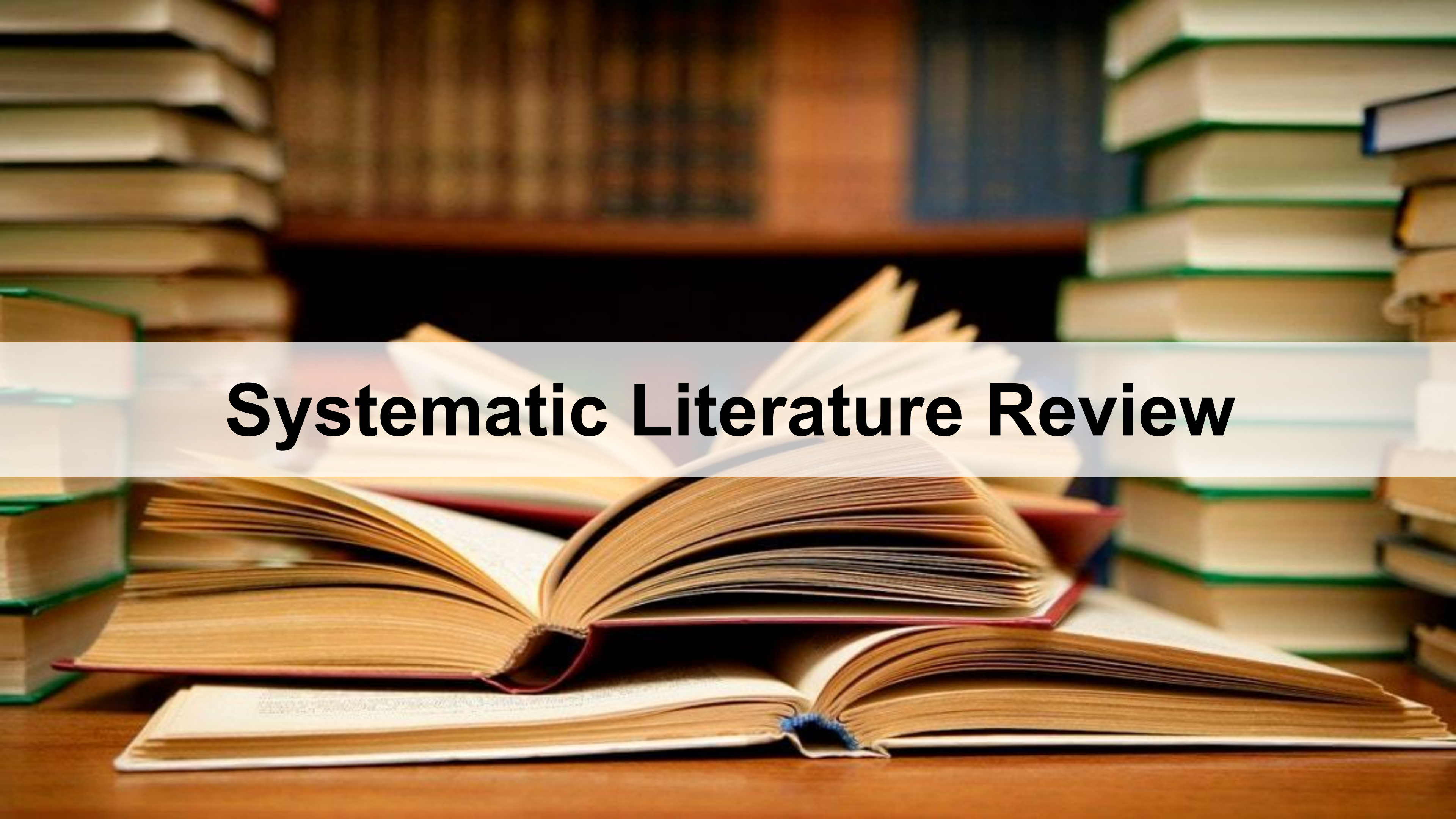
- Emergence of Lehman's laws
- Other studies started to investigate software evolution
- There is still no clear conclusion on how software evolves
- Need to understand how the internal dimensions of the software evolve
- Lack of a pattern explaining how software systems’ internal structure degrades

Goal

1. Provide a detailed view of the evolution of some internal dimensions in software systems

2. Build prediction models for software evolution

Systematic Literature Review



Why?

- Summarize a topic, technology or area
- Extract new lines of study
- **Aim:** Compile the body of knowledge on software evolution
- **Purpose:** Provide a background and support this thesis proposal

Systematic Literature Review Protocol

Planning → **Execution** → **Analysis of the Results**

Planning

- Research questions
- Search string
- Electronic databases
- Inclusion and exclusion criteria

Execution

- Search process
- Filtering process
- Data Extraction

Analysis of the Results

- Analysis of the extracted data
- Discussion of the research questions
- Summarization of the results

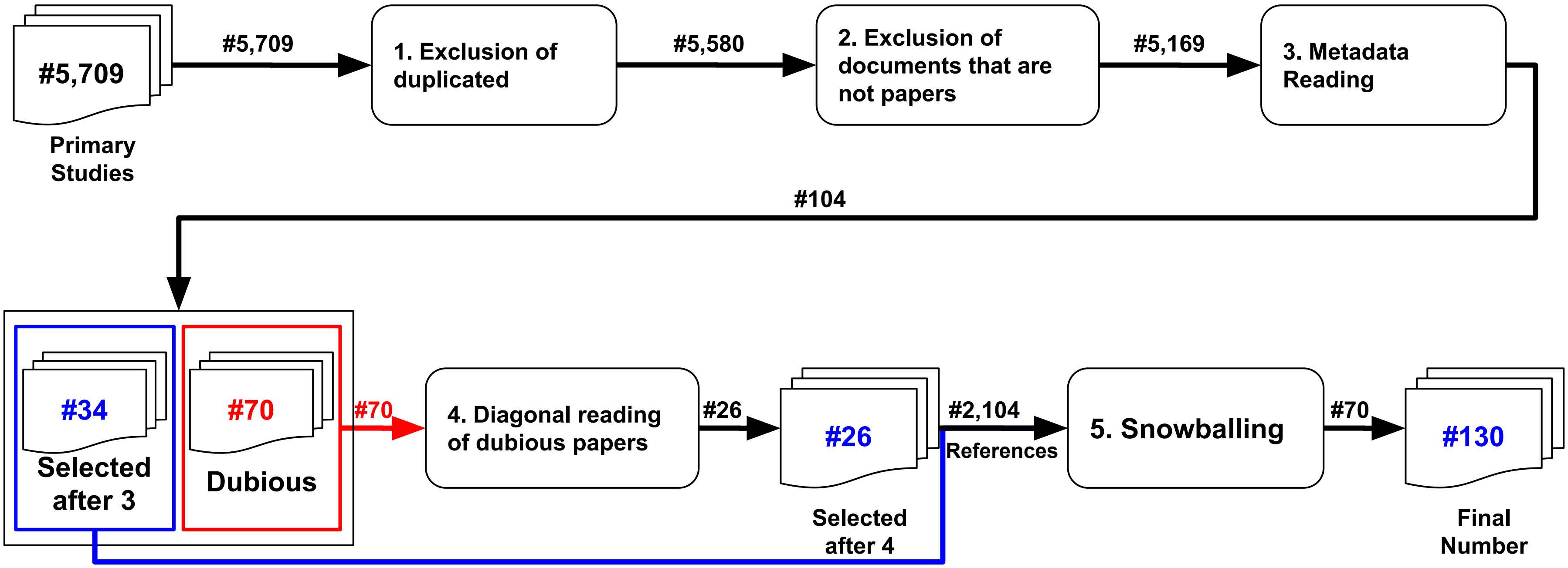
RQ1 (SLR): How has the literature approached studies on software evolution?

RQ2 (SLR): What are the main features of the datasets used in studies on software evolution?

Search Process

#	Database	Returned Studies
1	ACM Digital Library	101
2	Compendex (Engineering Village)	246
3	IEEE Xplore	150
4	Scopus	5,117
5	Web of Science	95
TOTAL		**5,709**

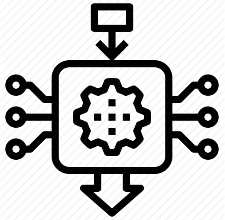
Selection Process



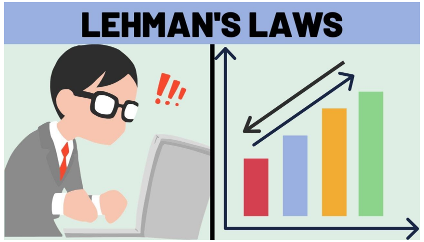
RQ1 (SLR): How has the literature approached studies on software evolution?

Application (34)

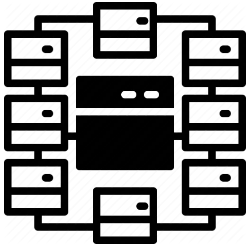
Evolution of the quality attributes (26)



Model (32)



Applicability of Lehman’s Laws (11)



Software Structure Evolution (27)

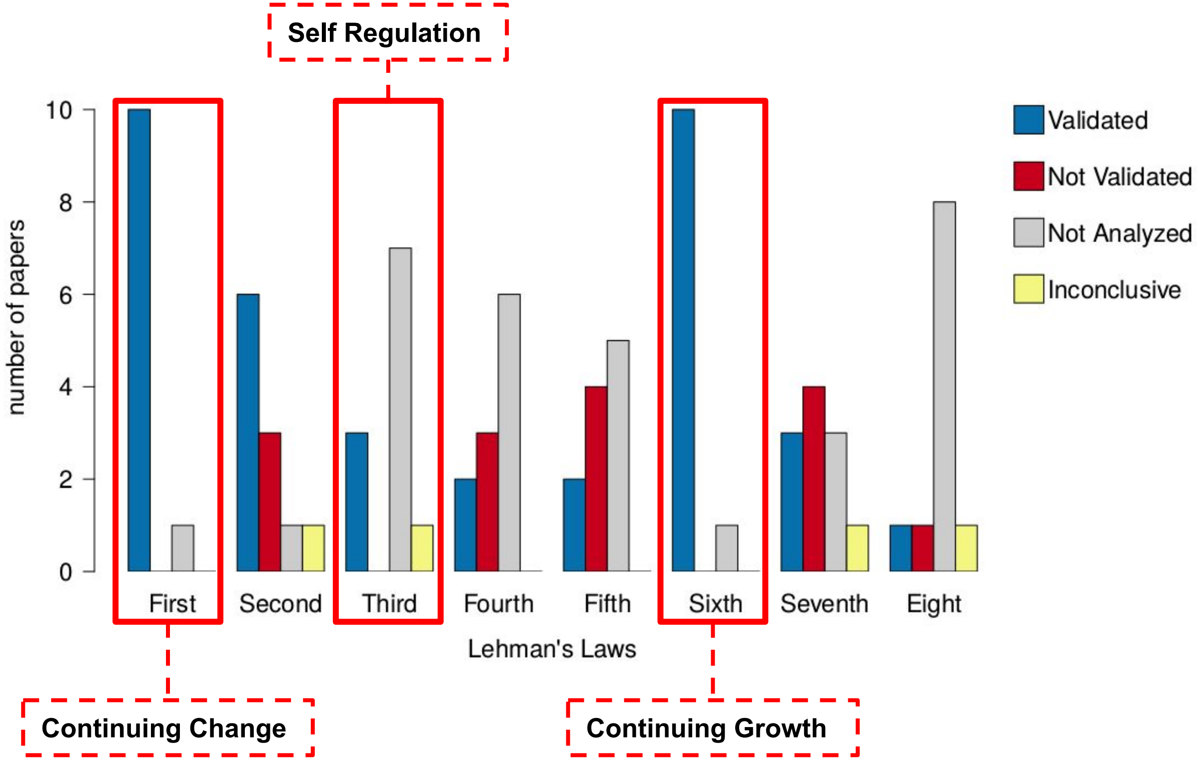
RQ2 (SLR): What are the main features of the datasets used in studies on software evolution?

- Datasets are not composed of large number of systems
  - In 75% of the cases, they presented no more than 14 systems

- Studies tend mostly to propose their **<u>own</u>** dataset
  - Few studies use third-party datasets

- In general, the evolution data of the systems concentrates between 2001 and 2010

Applicability of Lehman's Laws

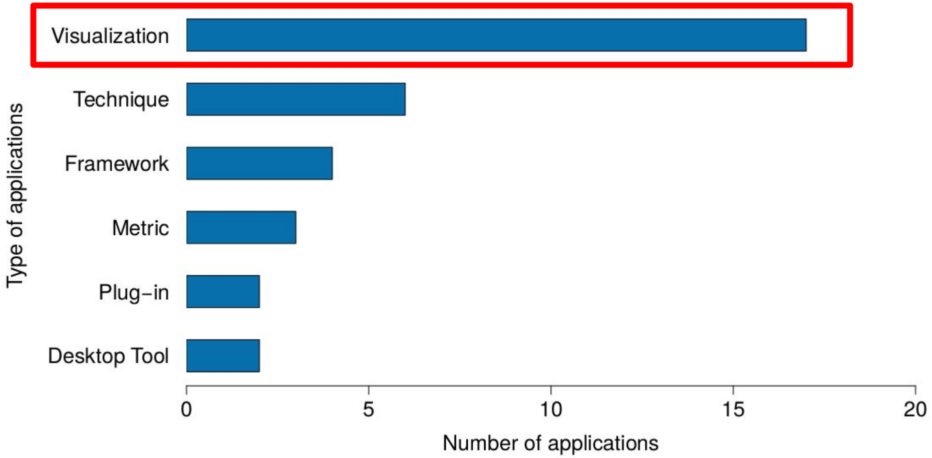


Application Contexts
Mobile Application
Open-source software
Eclipse third-party plugins
Proprietary software
C library

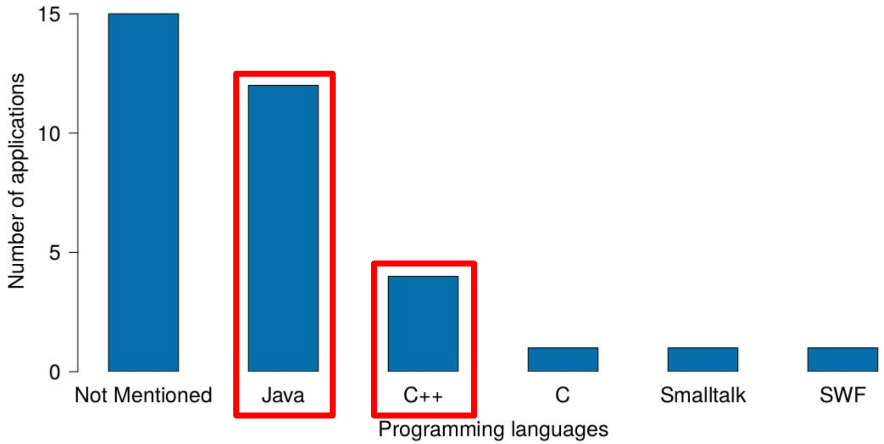
Application on Software Evolution

- A total of 33 applications was identified
  - 16 available and 17 unavailable

Type of Applications



Programming Languages



Evolution of Quality Attributes

- We identified 14 quality attributes
- **Size** and **complexity** are the most investigated attributes
- We found 72 metrics used to measure the attributes
- Relevant Mappings
  - Lines of Code (LOC) → Size (SI)
  - Number of Files (NOFL) → Size (SI)
  - McCabe's complexity (VG) → Complexity (CP)

Software Structure Evolution

Dimensions
Internal Structure
Bad Smell
Architectural design
Co-evolution between types of coupling
Faults
Function side effects
Internal quality
Project inter-dependencies
Technical debt
Vulnerabilities

- Compilation of the main studies results
- Most systems connectivity concentrates on a few components over their evolution.
- Bad Smells grow over the software evolution
  - Feature Envy, Switch Statements, Long Method, and God Class
- Architectural design
  - Systems have increased in size
  - Divergence in the growth pattern

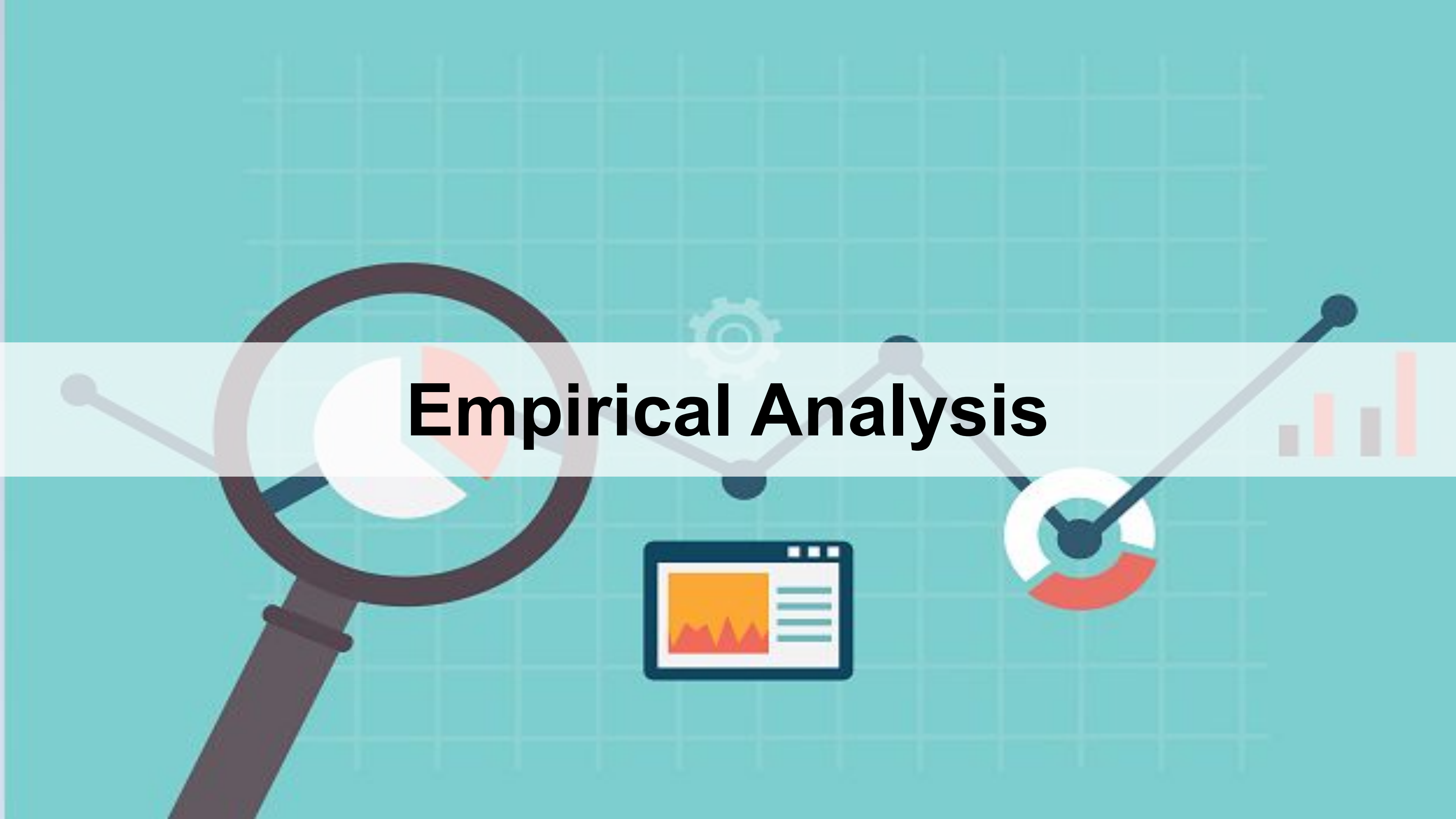
Model

- Types of Models
    - **Prediction**: defects
    - **Characterization**: represent the evolution of the internal structure of the systems
    - **Description**: explain how the software systems’ intrinsic properties behave
    - **Simulation**: study and compare the evolution of software systems’ internal quality attributes

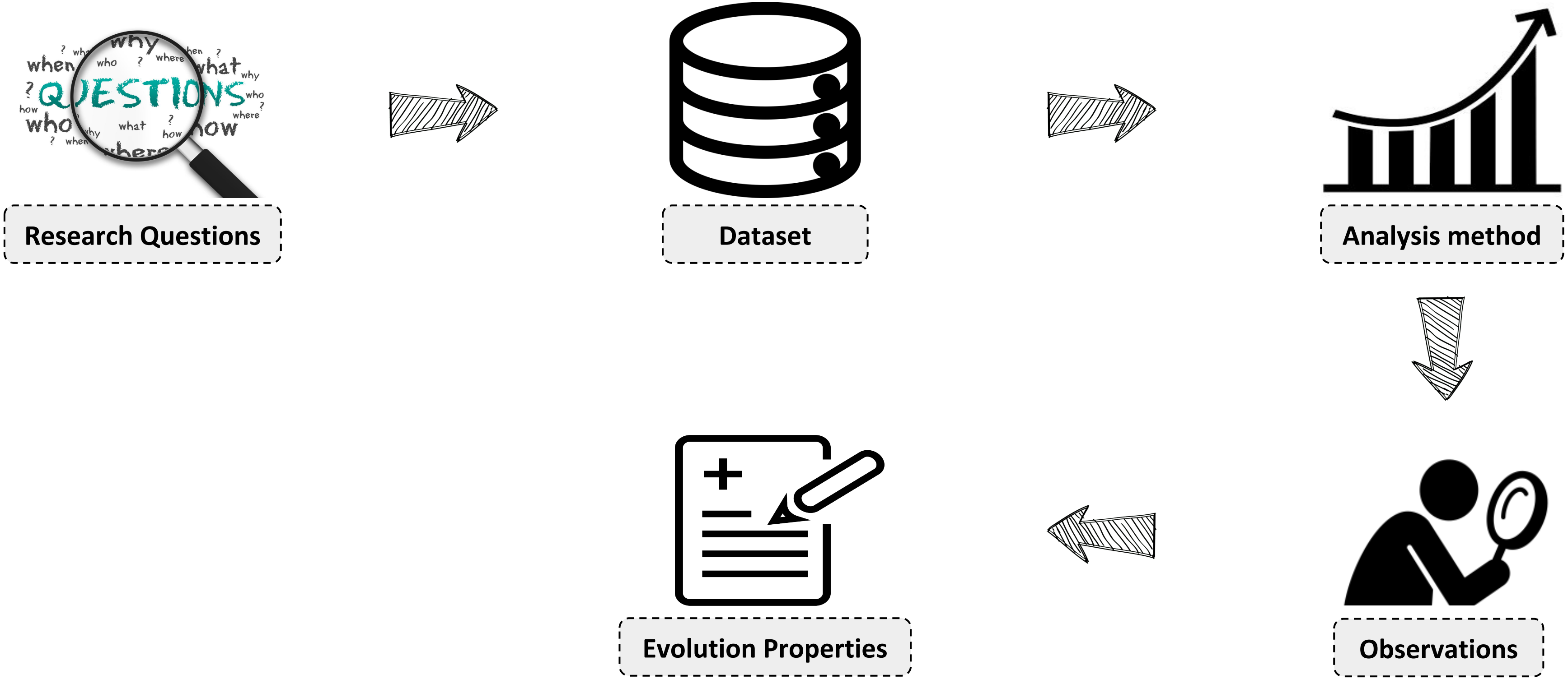
Final Remarks

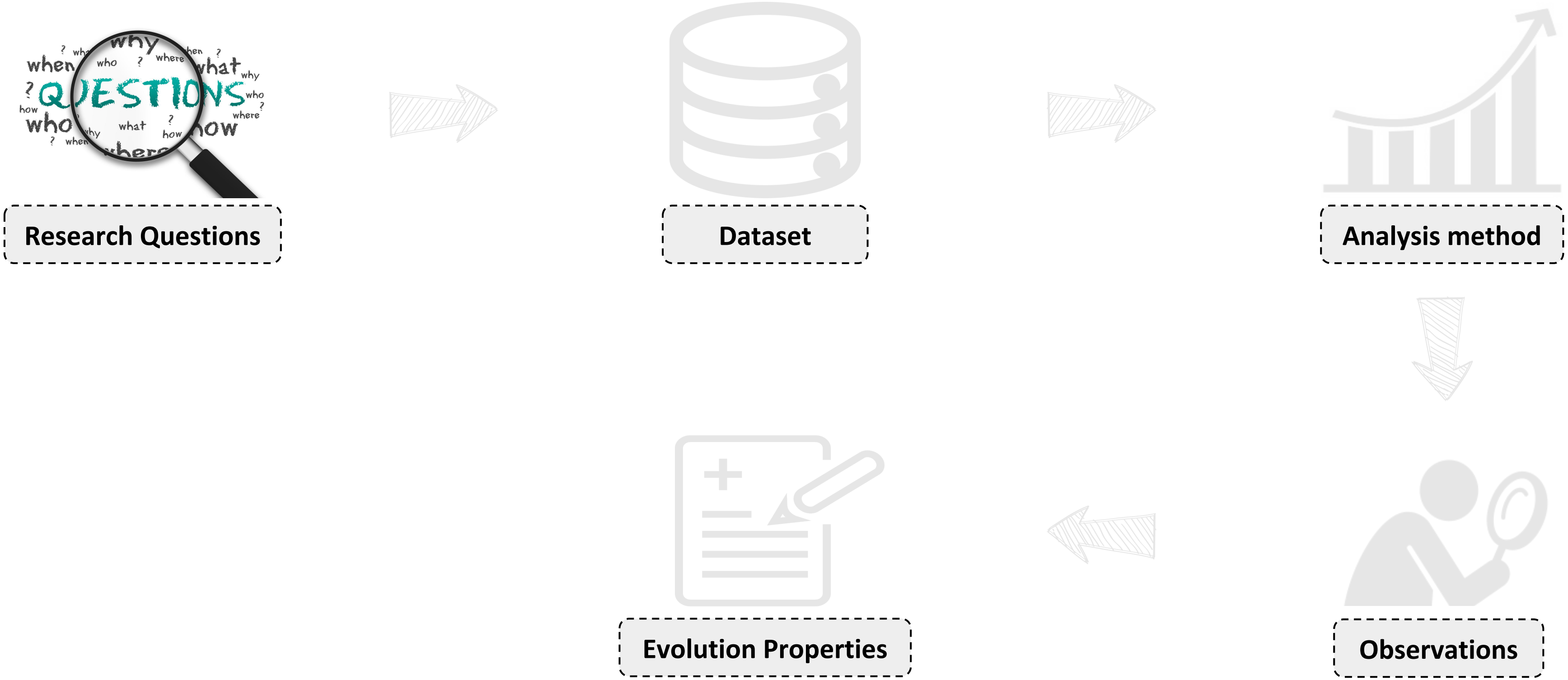
- Compilation of the knowledge on software evolution
- Lack of software evolution dataset containing recent information
- Need for detailing the software evolution from the perspective of the internal dimensions
- Lack of models that help developers to project the evolution of the internal dimensions of the software

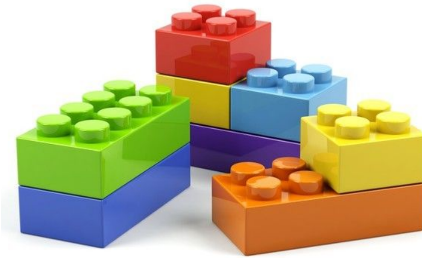
Empirical Analysis



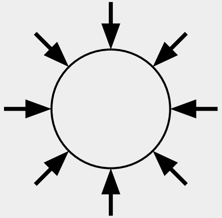
Study Design



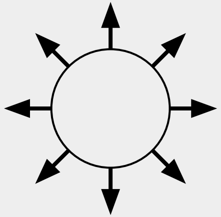

QUESTIONS
Research Questions
Dataset
Analysis method
Observations
Evolution Properties



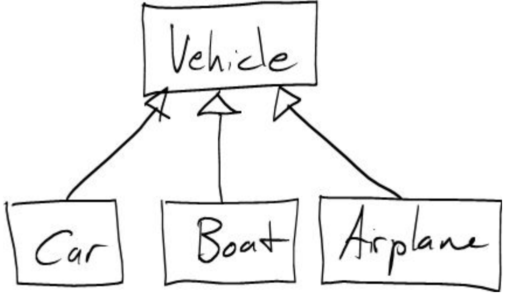
Coupling



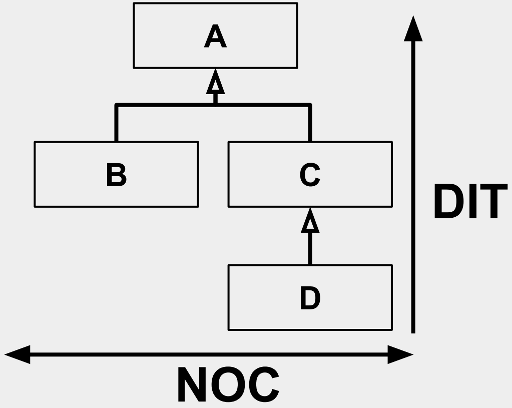
Fan-in



Fan-out



Inheritance Hierarchy

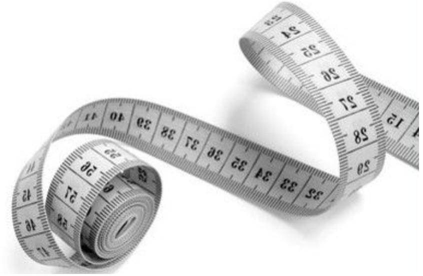


DIT (A) = 0

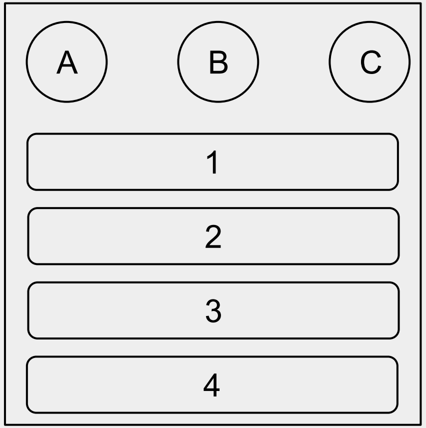
NOC (A) = 2

DIT (C) = 1

NOC (C) = 1



Size of Classes



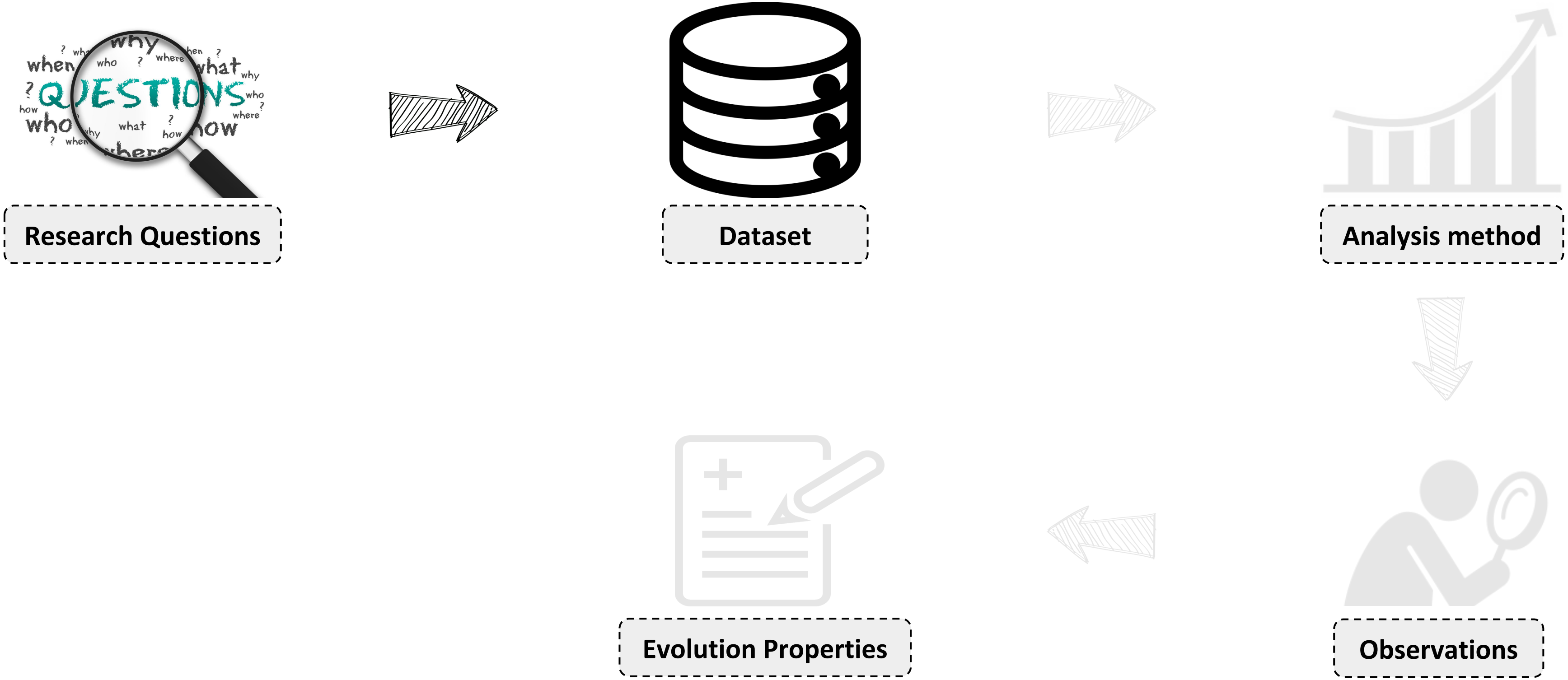
NOA = 3

NOM = 4

RQ1: Which model better describes the evolution of the dimensions in software systems?

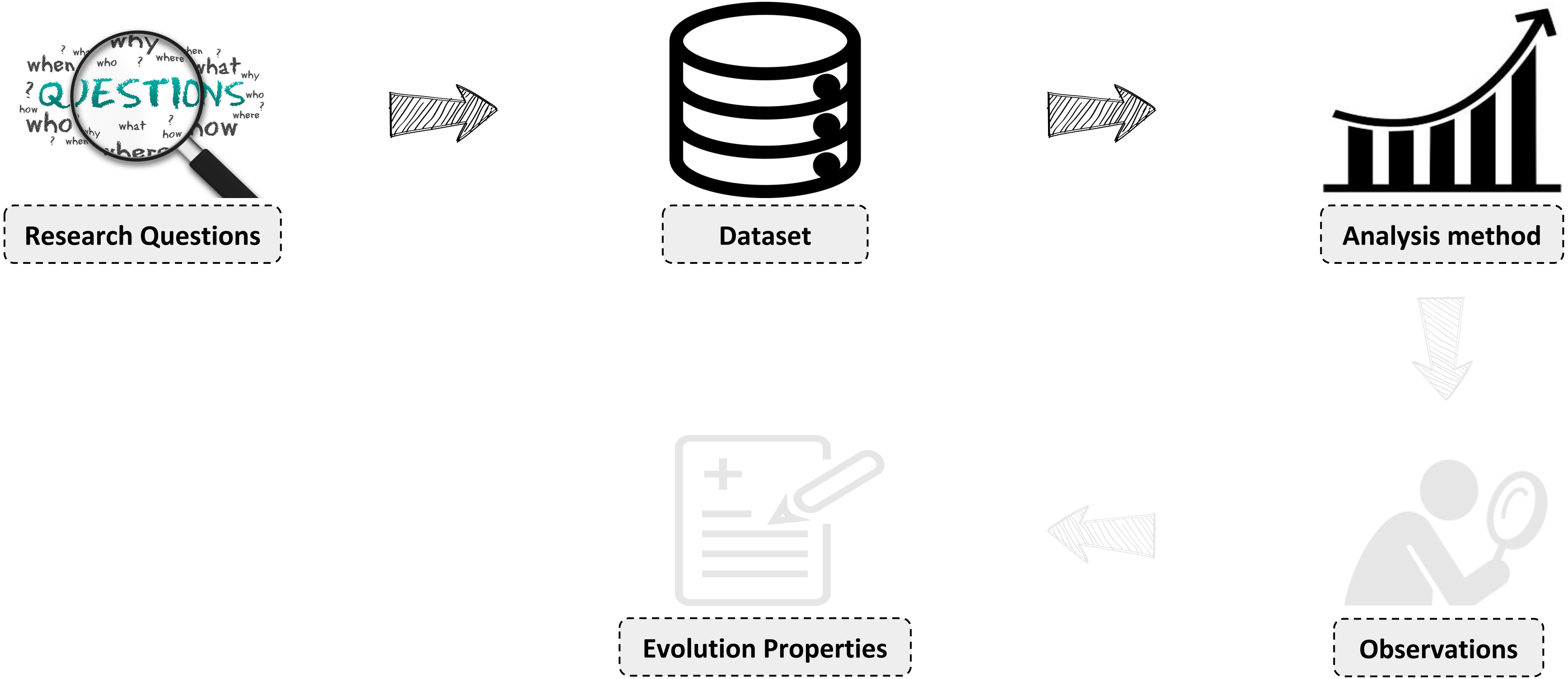
RQ2: How does the relation between dimension metrics behave throughout the evolution of software systems?

RQ3: What set of classes within the software system affects the dimensions of growth/decrease and how these classes evolve?


QUESTIONS
Research Questions
Dataset
Analysis method
Observations
Evolution Properties

COMETS Dataset

#	System Name	Description	Time Frame	# Versions
1	Eclipse JDT Core	Compiler and other tools for Java	2001-07-01 – 2008-06-14	183
2	Eclipse PDE UI	Set of tools to create, develop, test, debug and deploy Eclipse plug-ins, fragments, features, update sites and RCP products	2001-06-01 – 2008-09-06	191
3	Equinox Framework	OSGi application implementor	2005-01-01 – 2008-06-14	91
4	Hibernate Core	Database persistence framework	2007-06-13 – 2011-03-02	98
5	JabRef	Bibliography reference manager	2003-10-14 – 2011-11-11	212
6	Lucene	Search software and document indexing API	2005-01-01 – 2008-10-04	99
7	Pentaho Console	Software for business intelligence	2008-04-01 – 2010-12-07	72
8	PMD	Source code analyzer	2002-06-22 – 2011-12-11	248
9	Spring Framework	Java application development framework	2003-12-17 – 2009-11-25	156
10	TV-Browser	Electronic TV guide	2003-04-23 – 2011-08-27	221


Research Questions
Dataset
Analysis method
Observations
Evolution Properties

Behavior Analysis

Time series normalization → Application of linear regression → Improvement the model fit → Model evaluation

Time series normalization → Application of linear regression → Improvement the model fit → Model evaluation

System

class	v01	v02	v03	v04	…	vY
A	XXX	XXX	XXX	XXX		XXX
B	XXX	XXX	XXX	XXX		XXX
C	XXX	XXX	XXX	XXX		XXX
D	XXX	XXX	XXX	XXX		XXX
E	XXX	XXX	XXX	XXX		XXX
F	XXX	XXX	XXX	XXX		XXX
G	XXX	XXX	XXX	XXX		XXX
⋮						
AXZ	XXX	XXX	XXX	XXX		XXX

How can this problem be solved?

Time series normalization → Application of linear regression → Improvement the model fit → Model evaluation

System

class	v01	v02	v03	v04	…	vY
A	XXX	XXX	XXX	XXX		XXX
B	XXX	XXX	XXX	XXX		XXX
C	XXX	XXX	XXX	XXX		XXX
D	XXX	XXX	XXX	XXX		XXX
E	XXX	XXX	XXX	XXX		XXX
F	XXX	XXX	XXX	XXX		XXX
G	XXX	XXX	XXX	XXX		XXX
⋮						
AXZ	XXX	XXX	XXX	XXX		XXX

Global Measure

SUM

- Fan-in
- Fan-out
- NOA
- NOM

Arithmetic average

- DIT
- NOC

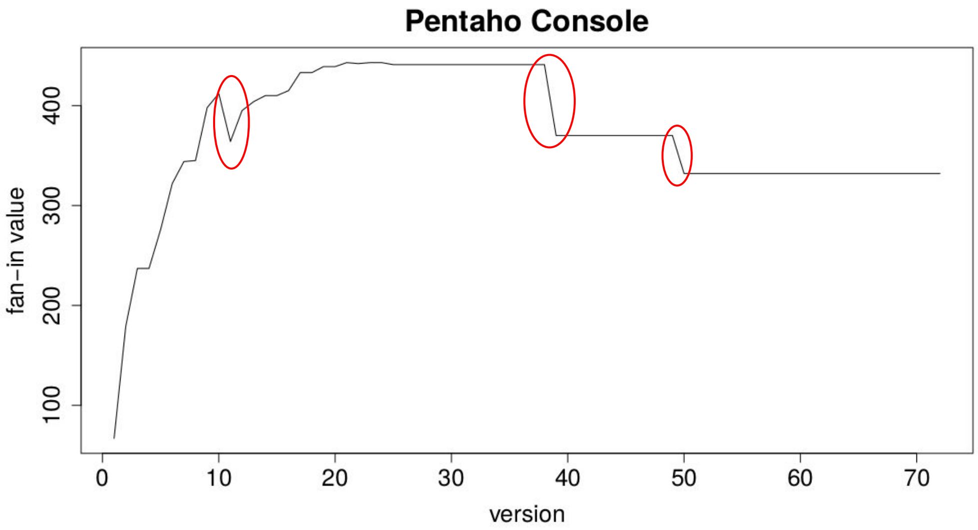
Time series normalization → Application of linear regression → Improvement the model fit → Model evaluation

- Evaluated models
    - Linear
    - Quadratic (polynomial at Degree 2)
    - Cubic (polynomial at Degree 3)
    - Logarithmic at Degree 1
    - Logarithmic at Degree 2
    - Logarithmic at Degree 3

Time series normalization → Application of linear regression → Improvement the model fit → Model evaluation

- Intervention analysis

- Error modeling
  - Autoregression



Time series normalization → Application of linear regression → Improvement the model fit → Model evaluation

Metric

Adjusted R^2

Evaluation Protocol

Relevance

Model with adjusted $R^2 \geq 90\%$

Coverage

Type with most number of relevant adjusted R^2

Simplicity

lin. > quad. > cub. > log. 1 > log. 2 > log. 3

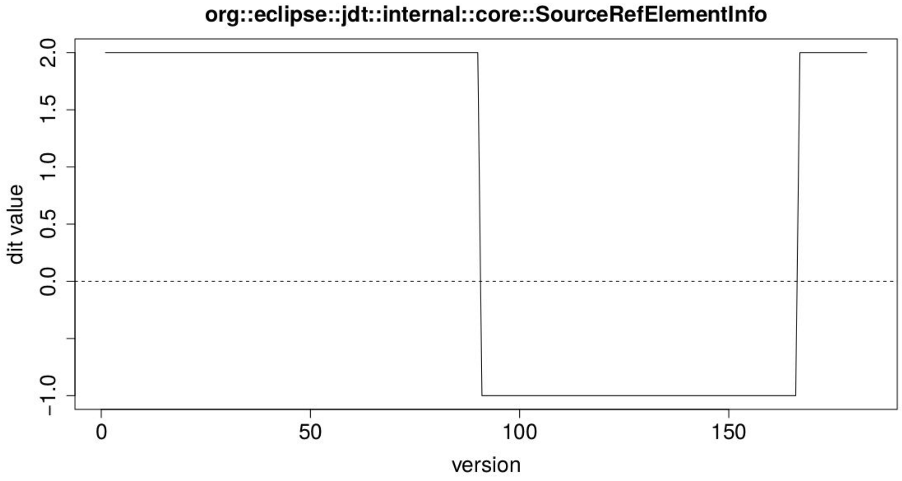
Trend Analysis

Time series reorganization → Application of trend tests → Identification of trends → Classification of trends

Time series reorganization → Application of trend tests → Identification of trends → Classification of trends

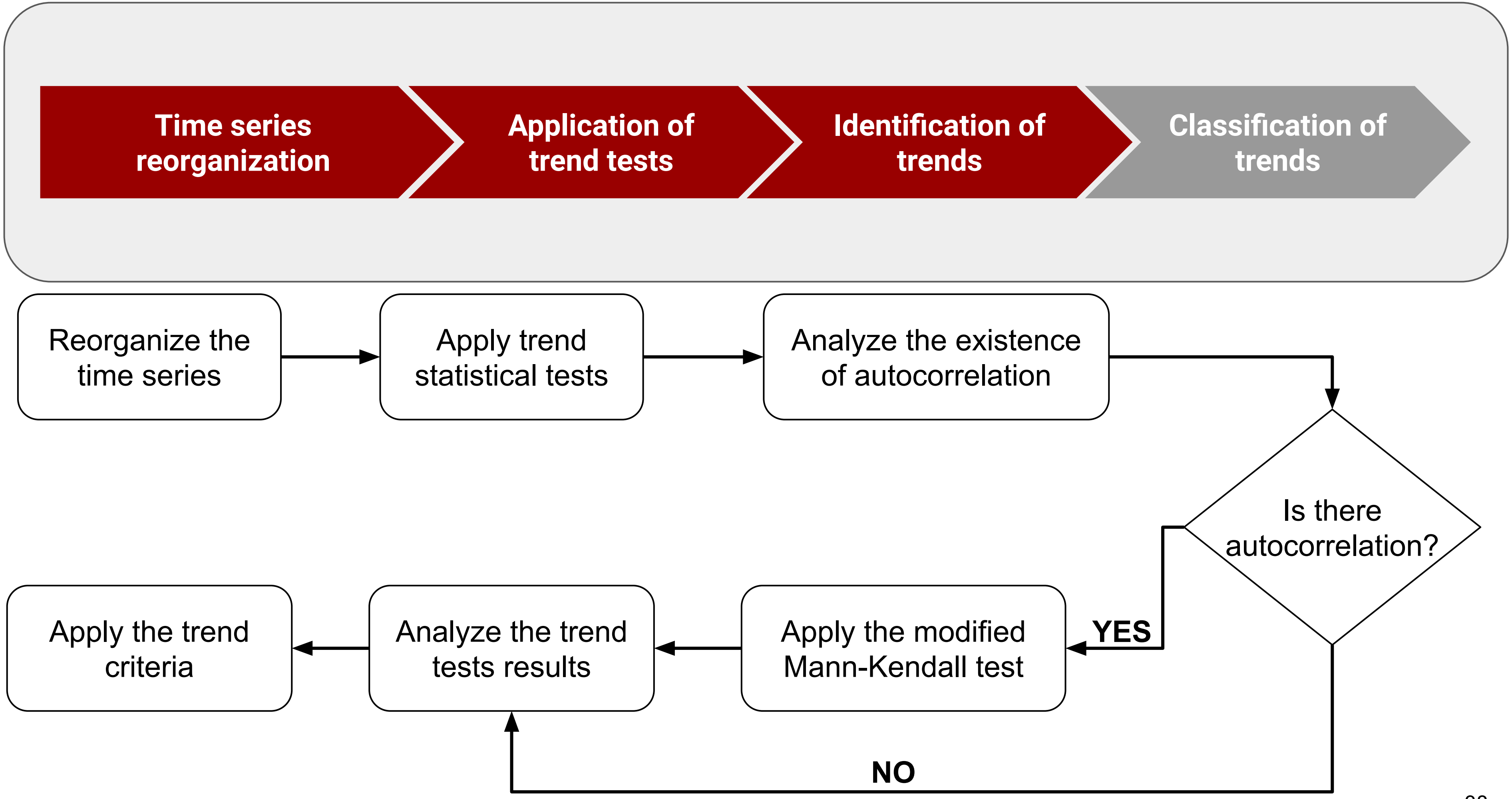
- Removal of -1 values

- Removal of ghost classes



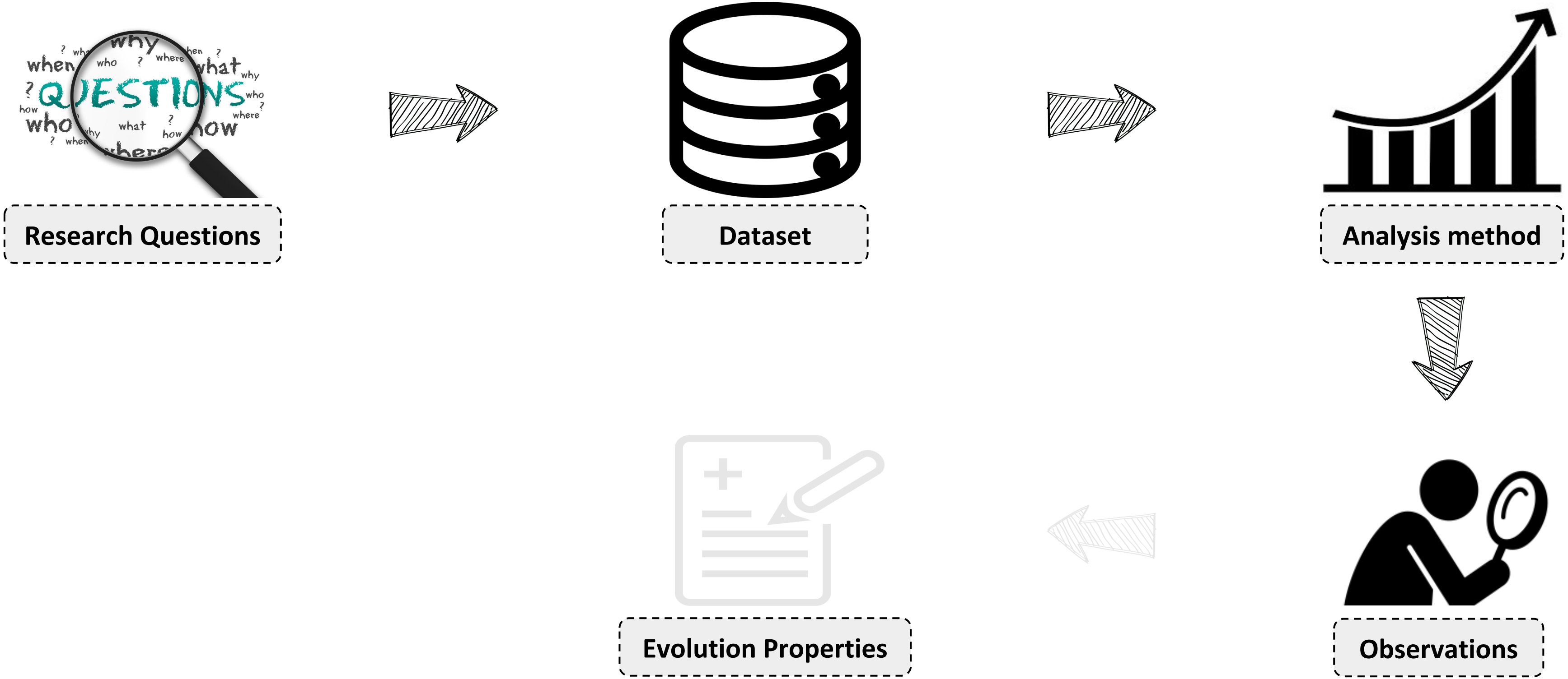
Time series reorganization → Application of trend tests → Identification of trends → Classification of trends

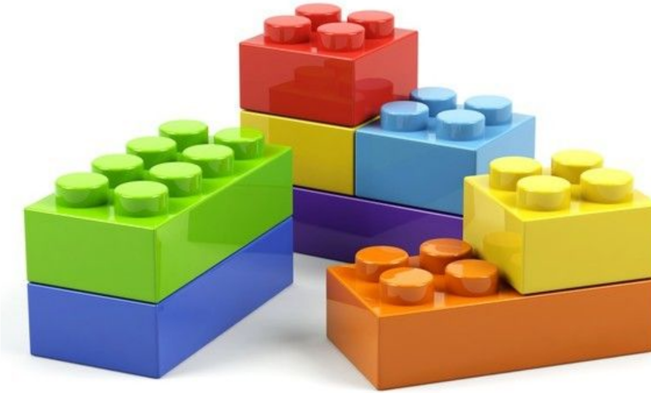
- Cox-Stuart
- Wald-Wolfowitz
- Mann-Kendall
  - <u>Original:</u> non-autocorrelated time series
  - <u>Modified:</u> autocorrelated time series


Time series reorganization
Application of trend tests
Identification of trends
Classification of trends
Reorganize the time series
Apply trend statistical tests
Analyze the existence of autocorrelation
Is there autocorrelation?
YES
Apply the modified Mann-Kendall test
Analyze the trend tests results
Apply the trend criteria
NO

Time series reorganization → Application of trend tests → Identification of trends → Classification of trends

- Upward Trend
- Downward Trend
- Undefined Trend


Research Questions
Dataset
Analysis method
Observations
Evolution Properties

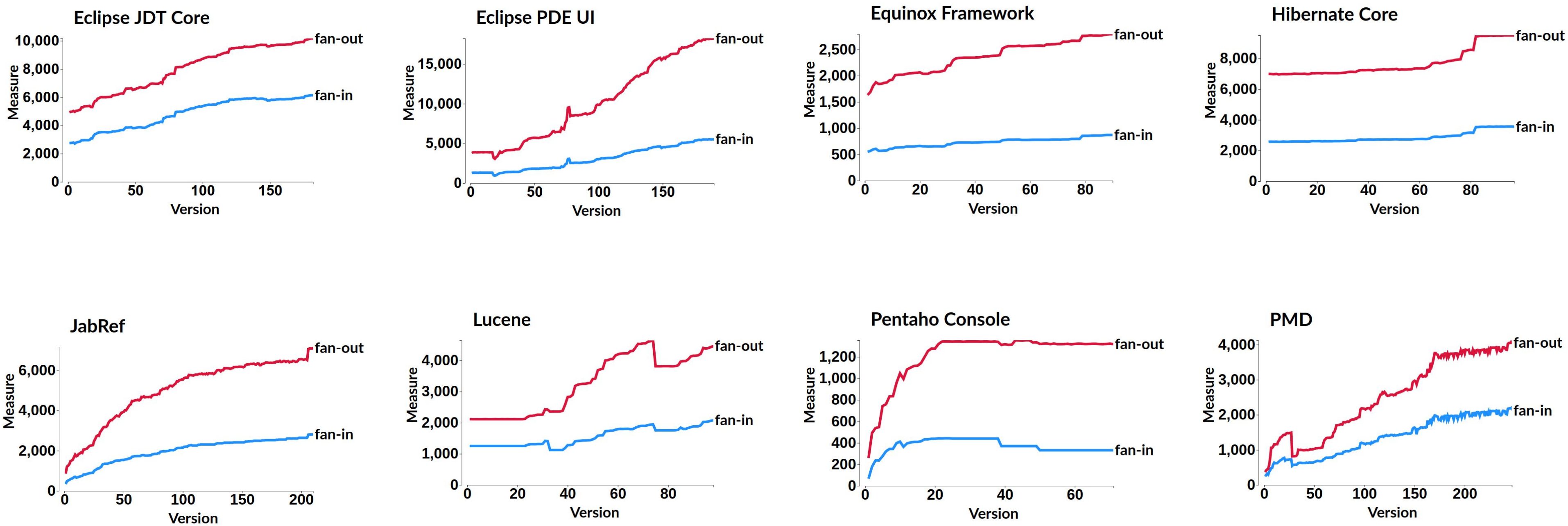


Coupling

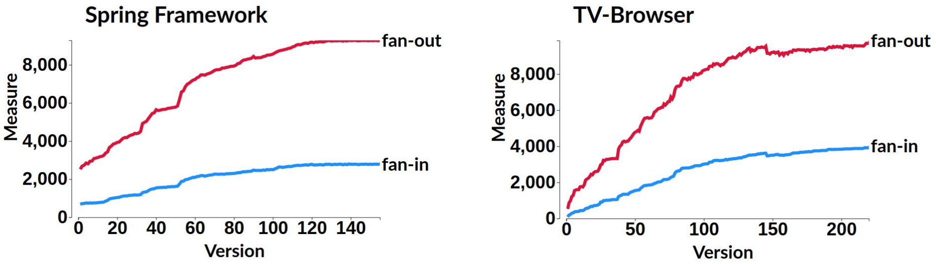
RQ1: Which model better describes the evolution of the dimensions in software systems?

RQ2: How does the relation between dimension metrics behave throughout the evolution of software systems?

RQ3: What set of classes within the software system affects the dimensions of growth/decrease and how these classes evolve?



Eclipse JDT Core
Eclipse PDE UI
Equinox Framework
Hibernate Core
JabRef
Lucene
Pentaho Console
PMD
fan-out
fan-in
Measure
Version



Spring Framework
TV-Browser
fan-out
fan-in
Measure
Version

Adjusted R^2 values computed from the fan-in models

System	lin.	quad.	cub.	log. 1	log. 2	log. 3
Eclipse JDT Core	99.84%	99.84%	99.84%	99.82%	99.82%	99.77%
Eclipse PDE UI	99.72%	99.72%	99.73%	99.55%	99.57%	99.57%
Equinox Framework	98.58%	98.57%	98.57%	98.44%	98.44%	98.44%
Hibernate Core	98.55%	98.58%	-	98.71%	98.73%	-
JabRef	99.88%	99.88%	99.88%	99.86%	99.86%	99.86%
Lucene	97.65%	97.63%	97.61%	97.28%	-	97.24%
Pentaho Console	92.64%	93.66%	95.11%	85.70%	89.40%	94.94%
PMD	99.25%	99.24%	-	99.27%	99.30%	-
Spring Framework	99.87%	99.88%	99.83%	99.86%	99.86%	99.87%
TV-Browser	99.94%	99.94%	-	99.89%	99.66%	99.87%

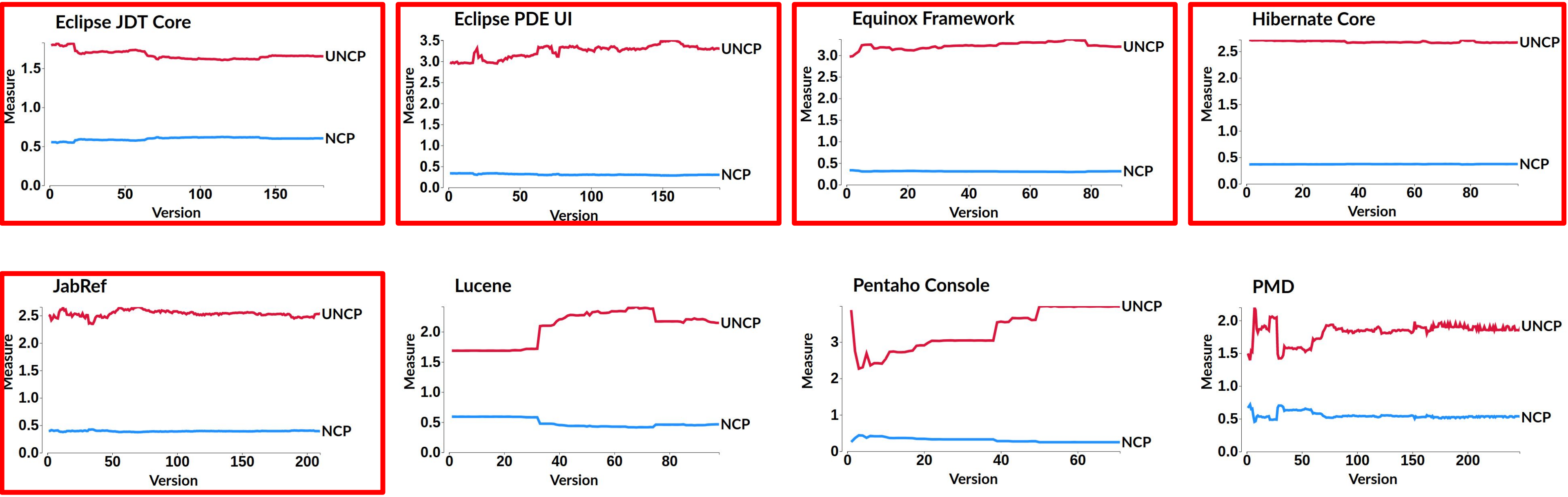
Adjusted R^2 values computed from the fan-out models

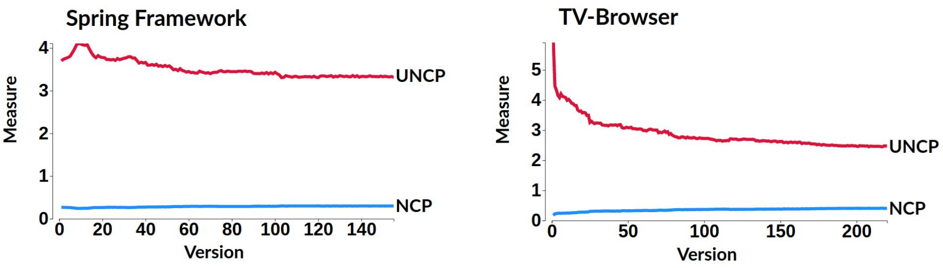
System	lin.	quad.	cub.	log. 1	log. 2	log. 3
Eclipse JDT Core	99.85%	99.85%	99.83%	99.83%	-	99.85%
Eclipse PDE UI	99.84%	99.84%	99.74%	99.75%	99.76%	99.85%
Equinox Framework	99.39%	99.42%	99.31%	99.38%	99.37%	99.41%
Hibernate Core	98.63%	98.68%	98.78%	98.79%	-	-
JabRef	99.84%	99.84%	99.68%	99.70%	99.78%	99.85%
Lucene	98.67%	98.69%	99.05%	99.09%	99.09%	98.68%
Pentaho Console	98.27%	98.25%	97.52%	97.50%	97.58%	98.28%
PMD	99.55%	99.55%	99.03%	99.04%	99.06%	99.57%
Spring Framework	99.92%	99.92%	99.89%	99.89%	99.89%	99.92%
TV-Browser	99.89%	99.90%	99.61%	99.75%	99.86%	99.90%

RQ1: Which model better describes the evolution of the dimensions in software systems?

RQ2: How does the relation between dimension metrics behave throughout the evolution of software systems?

RQ3: What set of classes within the software system affects the dimensions of growth/decrease and how these classes evolve?





$$\frac{fan\text{-}in}{fan\text{-}out}$$

Necessary Coupling

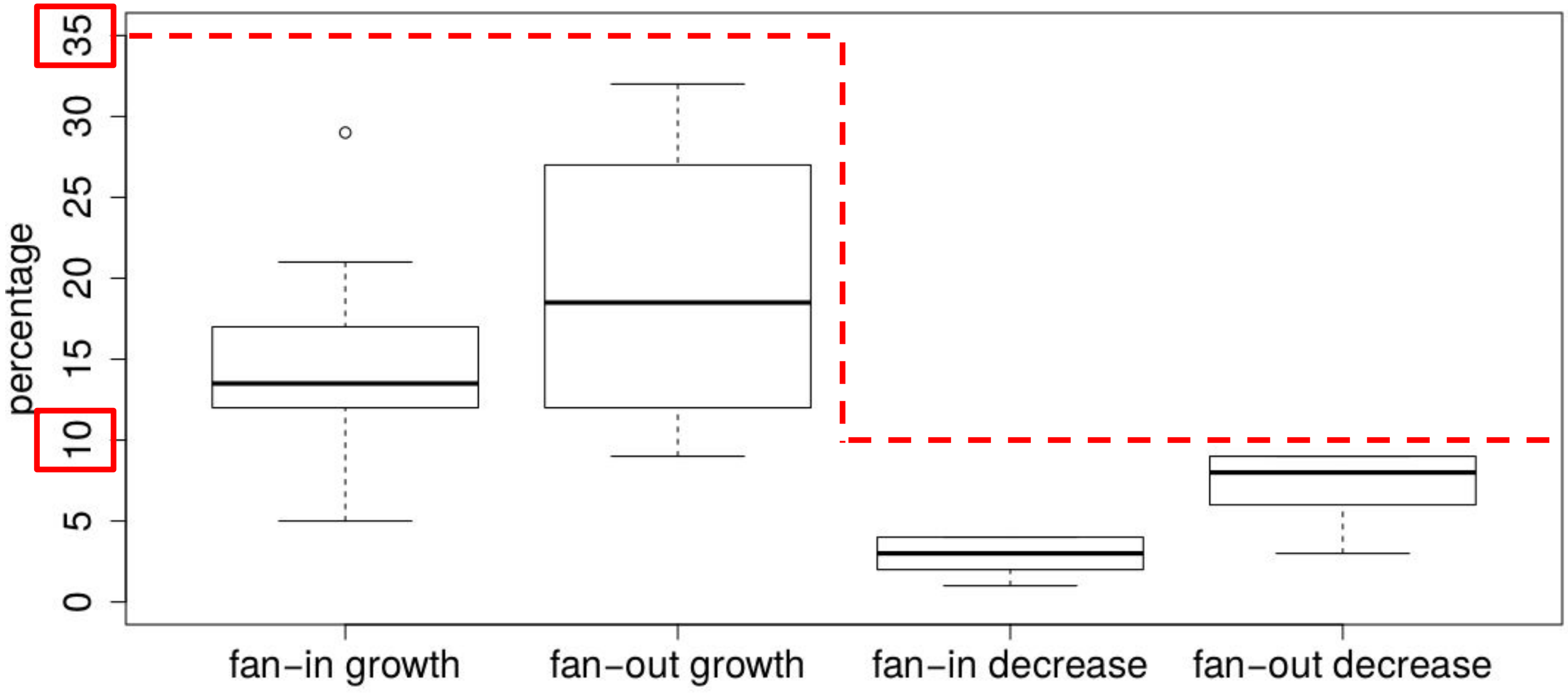
$$\frac{fan\text{-}out}{fan\text{-}in}$$

Unnecessary Coupling

RQ1: Which model better describes the evolution of the dimensions in software systems?

RQ2: How does the relation between dimension metrics behave throughout the evolution of software systems?

RQ3: What set of classes within the software system affects the dimensions of growth/decrease and how these classes evolve?


percentage
0
5
10
15
20
25
30
35
fan−in growth
fan−out growth
fan−in decrease
fan−out decrease

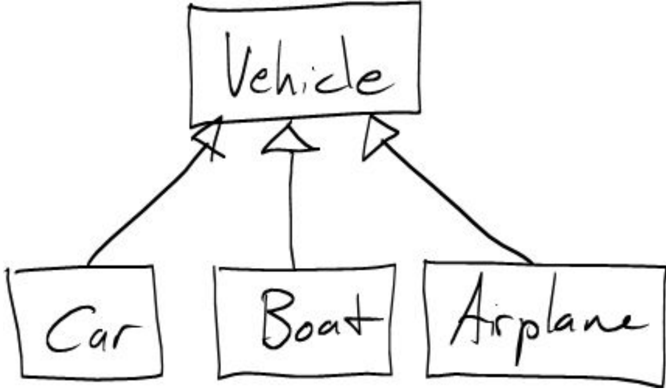
Intersection from the system perspective

System	i	ii	iii	iv
Eclipse JDT Core	18%	1%	3%	1%
Eclipse PDE UI	7%	1%	1%	1%
Equinox Framework	7%	0%	1%	2%
Hibernate Core	4%	0%	1%	0%
JabRef	8%	0%	1%	1%
Lucene	4%	0%	1%	1%
Pentaho Console	1%	0%	2%	0%
PMD	4%	1%	1%	1%
Spring Framework	7%	1%	2%	1%
TV-Browser	14%	1%	1%	1%

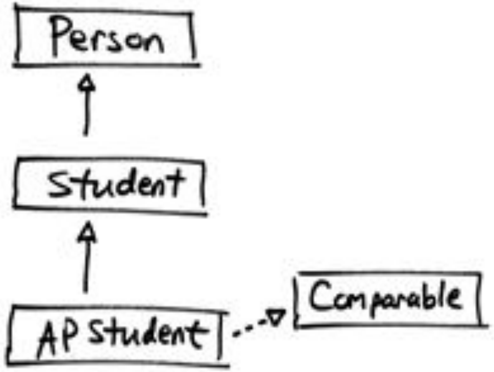
Intersection from the trend class perspective

System	i	ii	iii	iv
Eclipse JDT Core	35%	2%	6%	2%
Eclipse PDE UI	21%	2%	4%	2%
Equinox Framework	20%	1%	3%	5%
Hibernate Core	15%	1%	2%	2%
JabRef	19%	1%	3%	3%
Lucene	15%	1%	3%	4%
Pentaho Console	4%	0%	7%	0%
PMD	15%	2%	2%	2%
Spring Framework	15%	2%	4%	3%
TV-Browser	29%	2%	3%	3%

Case i	Case ii	Case iii	Case iv
fan-in ↑ fan-out ↑	fan-in ↓ fan-out ↓	fan-in ↑ fan-out ↓	fan-in ↓ fan-out ↑



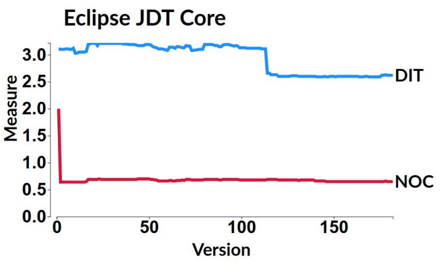
Inheritance Hierarchy

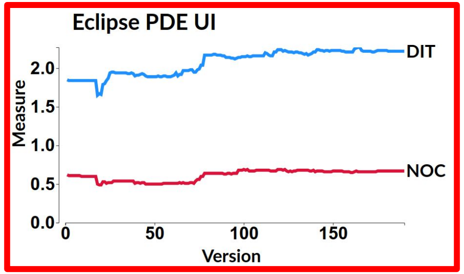


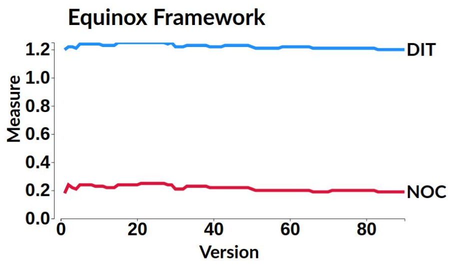
RQ1: Which model better describes the evolution of the dimensions in software systems?

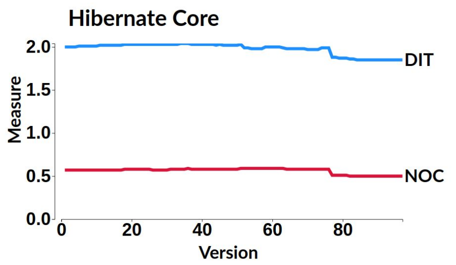
RQ3: What set of classes within the software system affects the dimensions of growth/decrease and how these classes evolve?

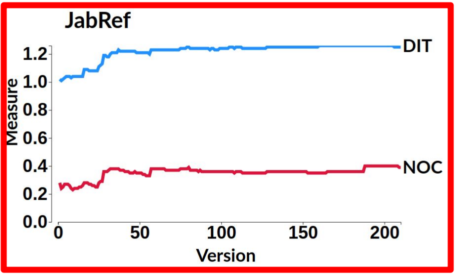
DIT

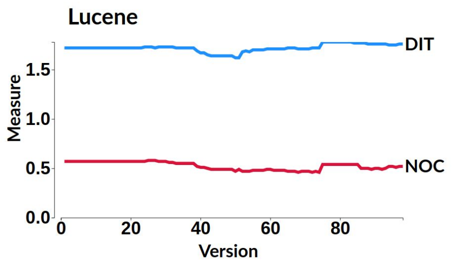

Eclipse JDT Core
Measure
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
50
100
150
Version
DIT
NOC

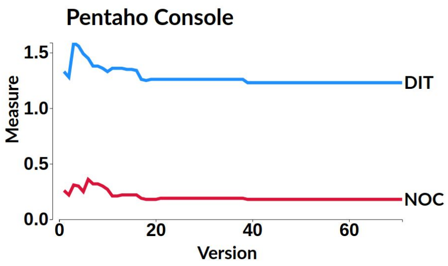

Eclipse PDE UI
Measure
2.0
1.5
1.0
0.5
0.0
0
50
100
150
Version
DIT
NOC

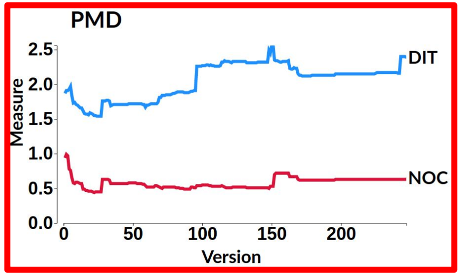

Equinox Framework
Measure
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
20
40
60
80
Version
DIT
NOC

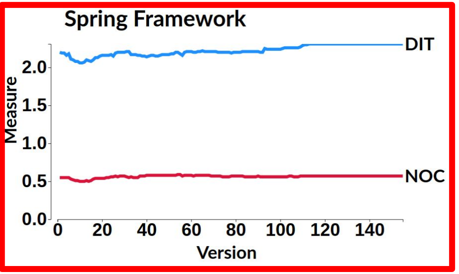

Hibernate Core
Measure
2.0
1.5
1.0
0.5
0.0
0
20
40
60
80
Version
DIT
NOC

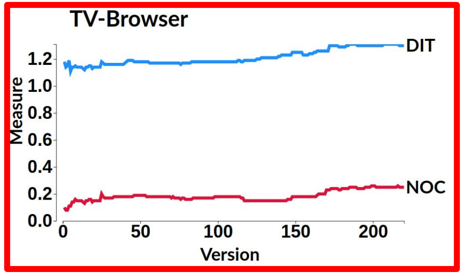

JabRef
Measure
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150
200
Version
DIT
NOC


Lucene
Measure
1.5
1.0
0.5
0.0
0
20
40
60
80
Version
DIT
NOC

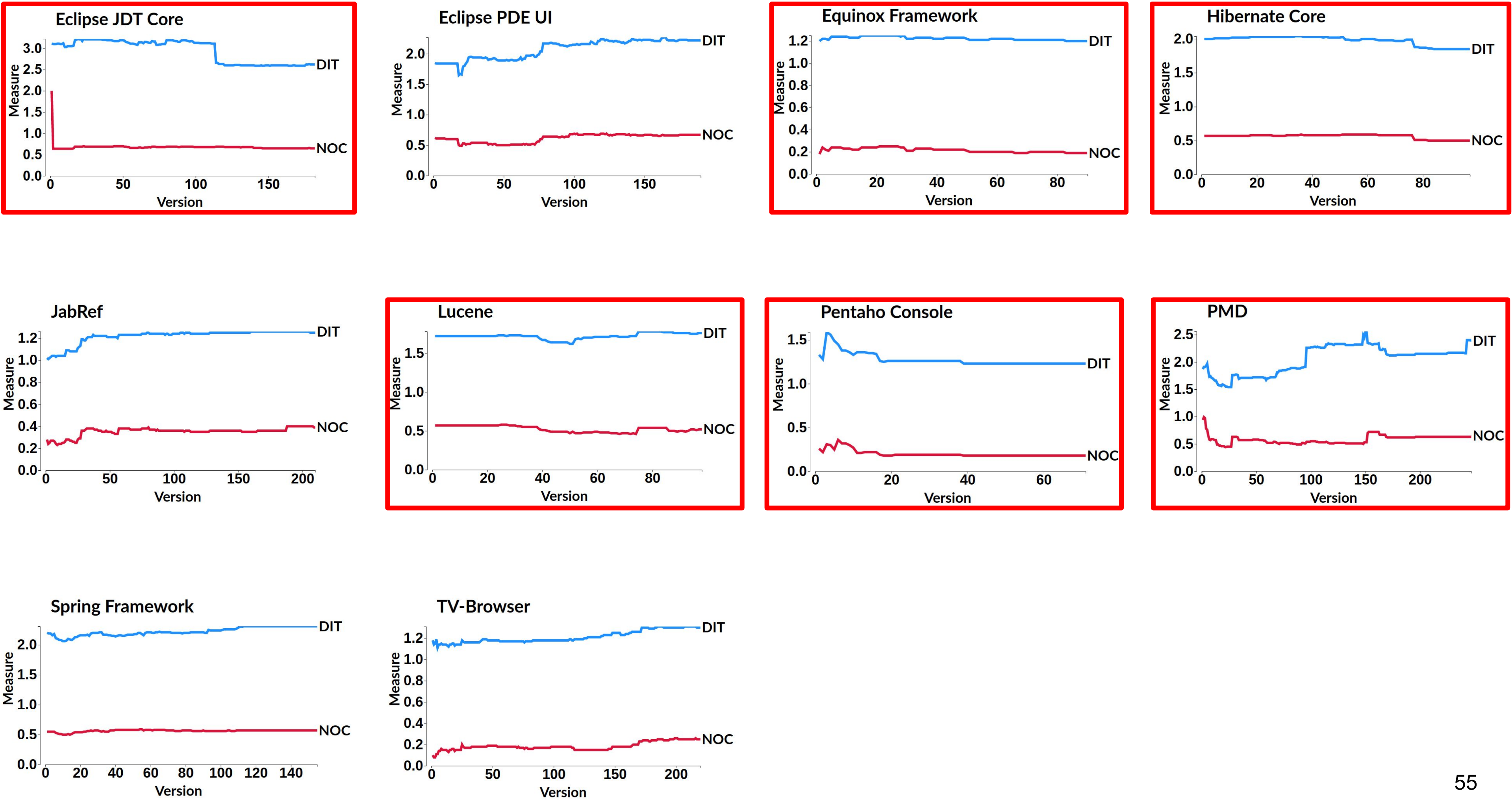

Pentaho Console
Measure
1.5
1.0
0.5
0.0
0
20
40
60
Version
DIT
NOC


PMD
Measure
2.5
2.0
1.5
1.0
0.5
0.0
0
50
100
150
200
Version
DIT
NOC


Spring Framework
Measure
2.0
1.5
1.0
0.5
0.0
0
20
40
60
80
100
120
140
Version
DIT
NOC


TV-Browser
Measure
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150
200
Version
DIT
NOC

NOC


Eclipse JDT Core
Eclipse PDE UI
Equinox Framework
Hibernate Core
JabRef
Lucene
Pentaho Console
PMD
Spring Framework
TV-Browser
DIT
NOC
Measure
Version

Adjusted R^2 values computed from the DIT models

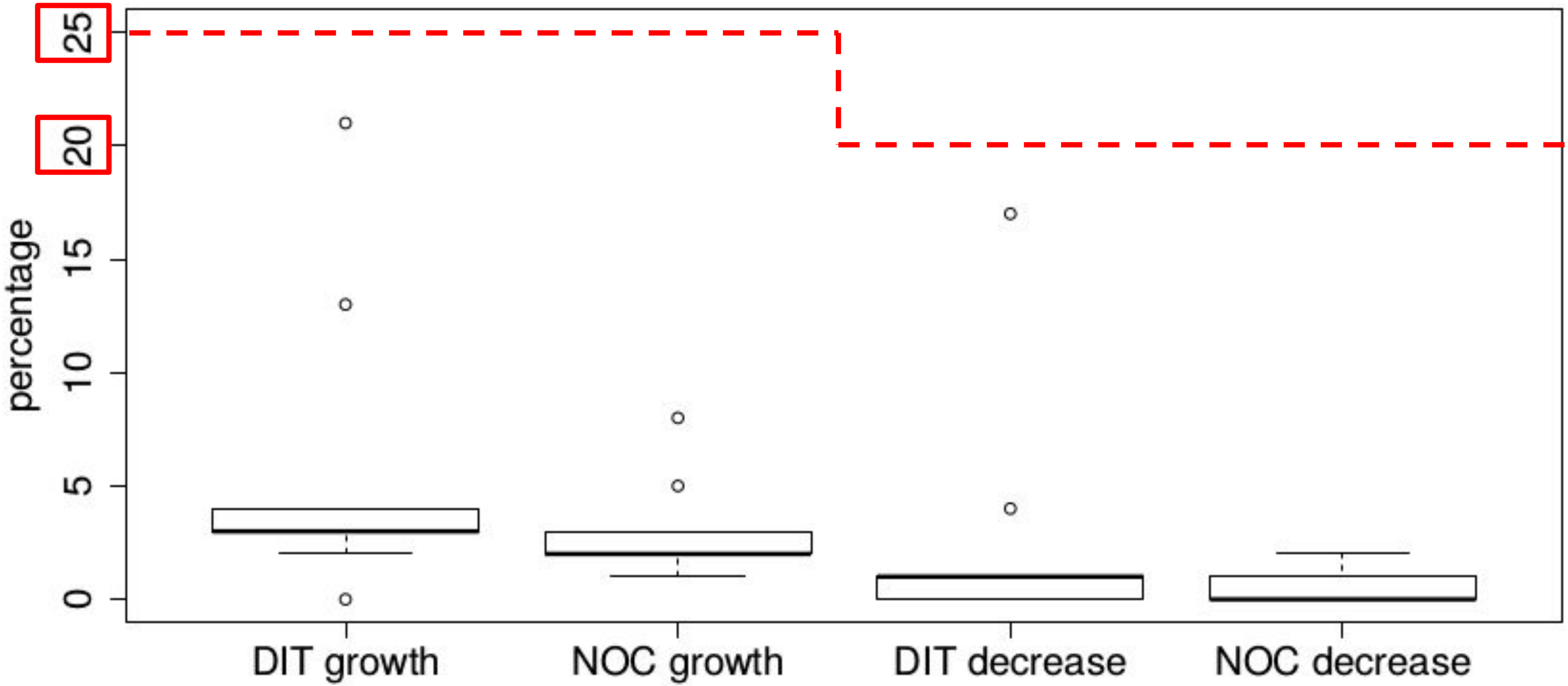
System	lin.	quad.	cub.	log. 1	log. 2	log. 3
Eclipse JDT Core	98.00%	97.41%	98.28%	98.07%	97.48%	98.35%
Eclipse PDE UI	97.81%	97.87%	97.83%	97.45%	97.53%	97.52%
Equinox Framework	86.89%	85.01%	86.49%	86.95%	85.10%	86.54%
Hibernate Core	96.54%	96.67%	96.58%	96.58%	96.71%	96.59%
JabRef	98.69%	98.69%	98.79%	98.68%	98.67%	98.64%
Lucene	92.23%	92.21%	89.52%	92.13%	92.11%	89.42%
Pentaho Console	75.49%	83.43%	90.92%	77.50%	85.05%	91.62%
PMD	97.61%	97.64%	96.52%	97.79%	97.82%	96.84%
Spring Framework	97.21%	97.04%	97.08%	97.09%	96.98%	-
TV-Browser	98.24%	98.09%	97.94%	98.11%	97.97%	97.81%

System	lin.	quad.	cub.	log. 1	log. 2	log. 3
Eclipse JDT Core	88.99%	91.61%	91.99%	89.08%	91.72%	92.11%
Eclipse PDE UI	97.13%	96.69%	97.16%	96.92%	96.46%	96.95%
Equinox Framework	86.05%	67.85%	85.70%	86.68%	68.08%	86.28%
Hibernate Core	94.67%	94.93%	-	94.80%	95.12%	-
JabRef	94.81%	94.79%	95.05%	94.71%	94.70%	94.99%
Lucene	96.07%	92.03%	-	95.86%	91.43%	-
Pentaho Console	76.80%	76.83%	78.40%	79.45%	81.35%	82.52%
PMD	93.83%	93.83%	93.43%	93.86%	93.84%	93.83%
Spring Framework	90.57%	90.67%	91.48%	90.76%	90.91%	91.63%
TV-Browser	96.57%	96.74%	96.74%	95.08%	94.79%	95.26%

Adjusted R^2 values computed from the NOC models

RQ1: Which model better describes the evolution of the dimensions in software systems?

RQ3: What set of classes within the software system affects the dimensions of growth/decrease and how these classes evolve?



percentage
0
5
10
15
20
25
DIT growth
NOC growth
DIT decrease
NOC decrease

Intersection from the system perspective

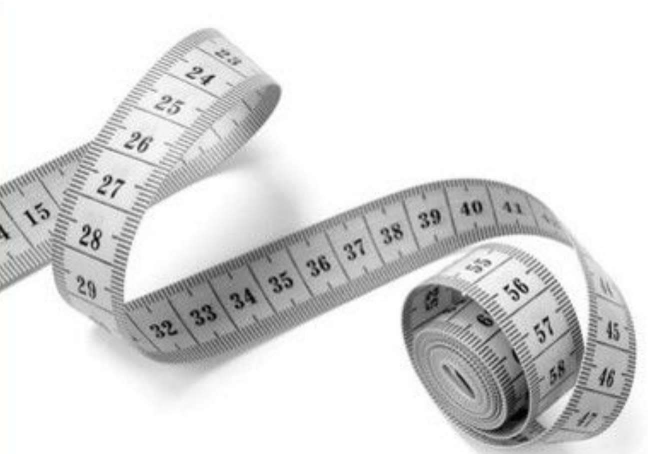
System	i	ii	iii	iv
Eclipse JDT Core	1%	0%	0%	3%
Eclipse PDE UI	0%	0%	0%	0%
Equinox Framework	0%	0%	0%	0%
Hibernate Core	0%	0%	0%	0%
JabRef	0%	0%	0%	0%
Lucene	0%	0%	0%	0%
Pentaho Console	0%	0%	0%	0%
PMD	0%	0%	0%	0%
Spring Framework	1%	0%	0%	0%
TV-Browser	0%	0%	0%	0%

Intersection from the trend class perspective

System	i	ii	iii	iv
Eclipse JDT Core	3%	1%	1%	15%
Eclipse PDE UI	4%	0%	0%	5%
Equinox Framework	0%	0%	0%	0%
Hibernate Core	0%	0%	0%	3%
JabRef	0%	0%	0%	0%
Lucene	4%	0%	0%	0%
Pentaho Console	0%	0%	0%	0%
PMD	1%	0%	0%	0%
Spring Framework	6%	0%	1%	0%
TV-Browser	0%	0%	1%	0%

Case i	Case ii	Case iii	Case iv
DIT ↑ NOC ↑	DIT ↓ NOC ↓	DIT ↑ NOC ↓	DIT ↓ NOC ↑

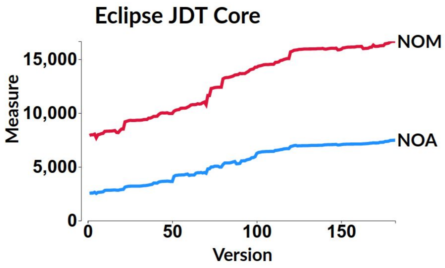
Size of Classes

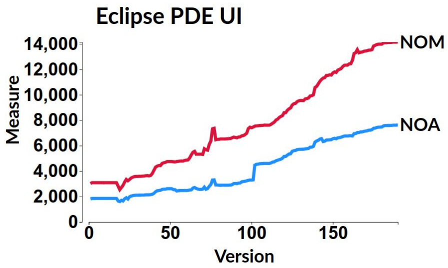


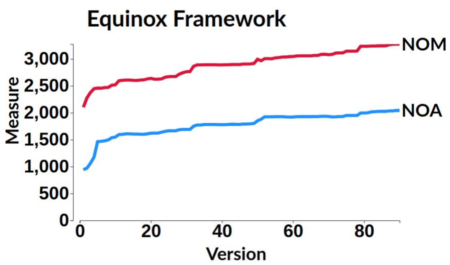
RQ1: Which model better describes the evolution of the dimensions in software systems?

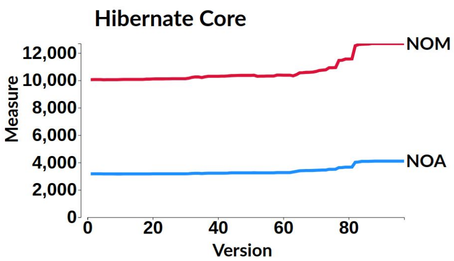
RQ2: How does the relation between dimension metrics behave throughout the evolution of software systems?

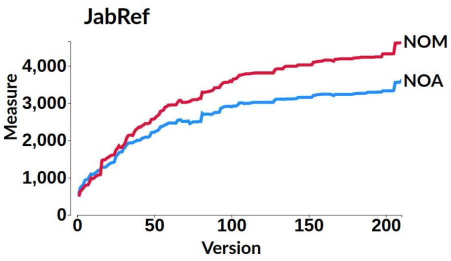
RQ3: What set of classes within the software system affects the dimensions of growth/decrease and how these classes evolve?

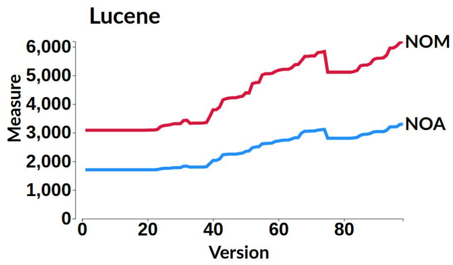

Eclipse JDT Core
Measure
Version
0
5,000
10,000
15,000
0
50
100
150
NOM
NOA

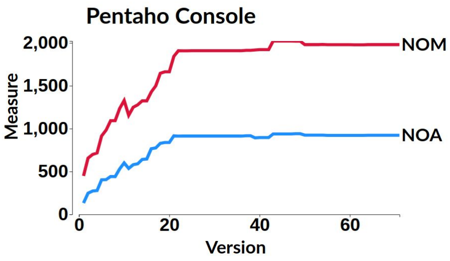

Eclipse PDE UI
Measure
Version
0
2,000
4,000
6,000
8,000
10,000
12,000
14,000
0
50
100
150
NOM
NOA

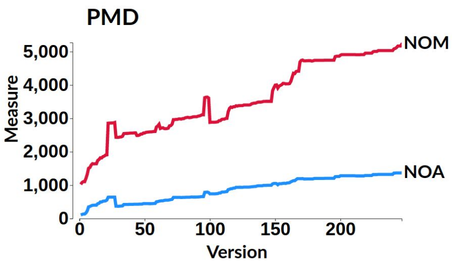

Equinox Framework
Measure
Version
0
500
1,000
1,500
2,000
2,500
3,000
0
20
40
60
80
NOM
NOA

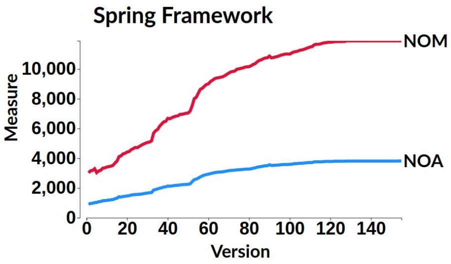

Hibernate Core
Measure
Version
0
2,000
4,000
6,000
8,000
10,000
12,000
0
20
40
60
80
NOM
NOA

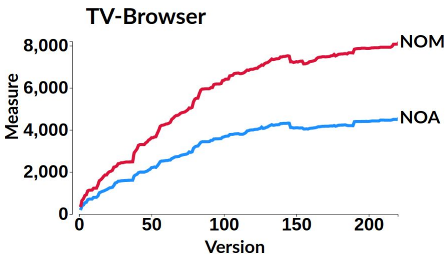

JabRef
Measure
Version
0
1,000
2,000
3,000
4,000
0
50
100
150
200
NOM
NOA


Lucene
Measure
Version
0
1,000
2,000
3,000
4,000
5,000
6,000
0
20
40
60
80
NOM
NOA


Pentaho Console
Measure
Version
0
500
1,000
1,500
2,000
0
20
40
60
NOM
NOA


PMD
Measure
Version
0
1,000
2,000
3,000
4,000
5,000
0
50
100
150
200
NOM
NOA


Spring Framework
Measure
Version
0
2,000
4,000
6,000
8,000
10,000
0
20
40
60
80
100
120
140
NOM
NOA


TV-Browser
Measure
Version
0
2,000
4,000
6,000
8,000
0
50
100
150
200
NOM
NOA

System	lin.	quad.	cub.	log. 1	log. 2	log. 3
Eclipse JDT Core	99.82%	99.82%	99.83%	99.77%	99.79%	99.79%
Eclipse PDE UI	99.73%	99.73%	99.73%	99.61%	99.56%	99.51%
Equinox Framework	98.25%	98.24%	98.22%	97.68%	97.86%	97.84%
Hibernate Core	98.75%	98.79%	-	98.86%	98.88%	-
JabRef	99.81%	99.81%	99.82%	99.70%	99.71%	99.77%
Lucene	99.30%	98.99%	99.29%	99.44%	99.03%	99.45%
Pentaho Console	98.11%	98.09%	98.25%	97.52%	97.49%	97.90%
PMD	99.51%	99.51%	99.53%	98.98%	98.92%	98.62%
Spring Framework	99.93%	99.94%	99.92%	99.91%	99.91%	99.91%
TV-Browser	99.90%	99.90%	99.90%	99.42%	99.58%	99.85%

Adjusted R^2 values computed from the NOA models

System	lin.	quad.	cub.	log. 1	log. 2	log. 3
Eclipse JDT Core	99.80%	99.81%	99.81%	99.76%	99.77%	99.77%
Eclipse PDE UI	99.85%	99.85%	99.85%	99.78%	99.80%	-
Equinox Framework	99.29%	99.31%	99.31%	99.30%	99.33%	99.33%
Hibernate Core	98.61%	98.67%	-	98.71%	98.73%	-
JabRef	99.84%	99.84%	99.86%	99.67%	99.67%	99.70%
Lucene	99.09%	98.79%	99.08%	99.28%	98.90%	99.29%
Pentaho Console	98.41%	98.42%	98.52%	97.99%	97.98%	98.09%
PMD	99.23%	99.20%	99.16%	98.96%	98.96%	98.96%
Spring Framework	99.93%	99.94%	99.91%	99.88%	99.88%	99.89%
TV-Browser	99.92%	99.93%	99.93%	99.50%	99.69%	99.80%

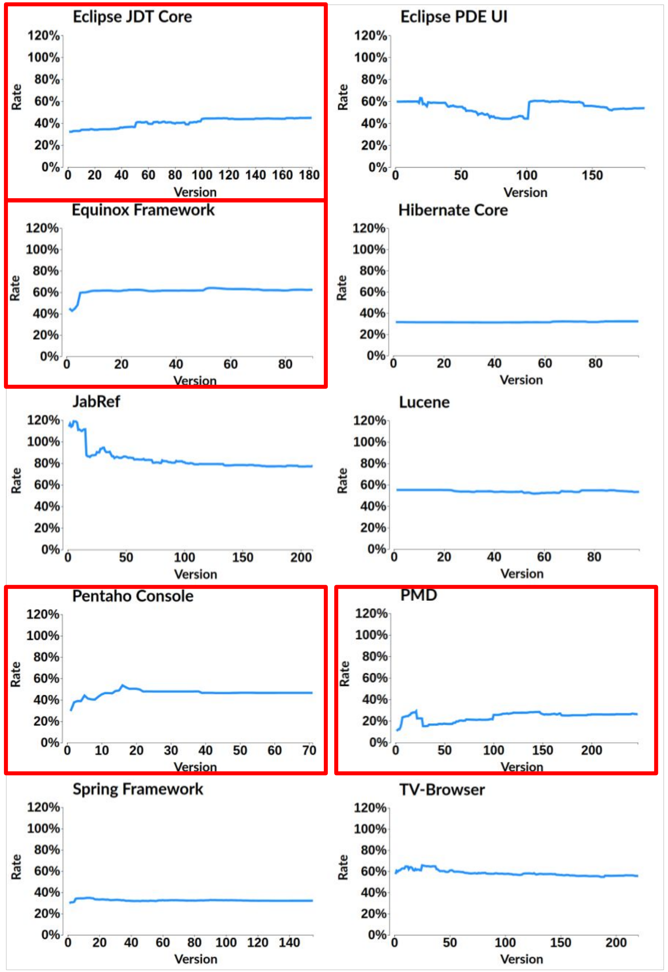
Adjusted R^2 values computed from the NOM models

RQ1: Which model better describes the evolution of the dimensions in software systems?

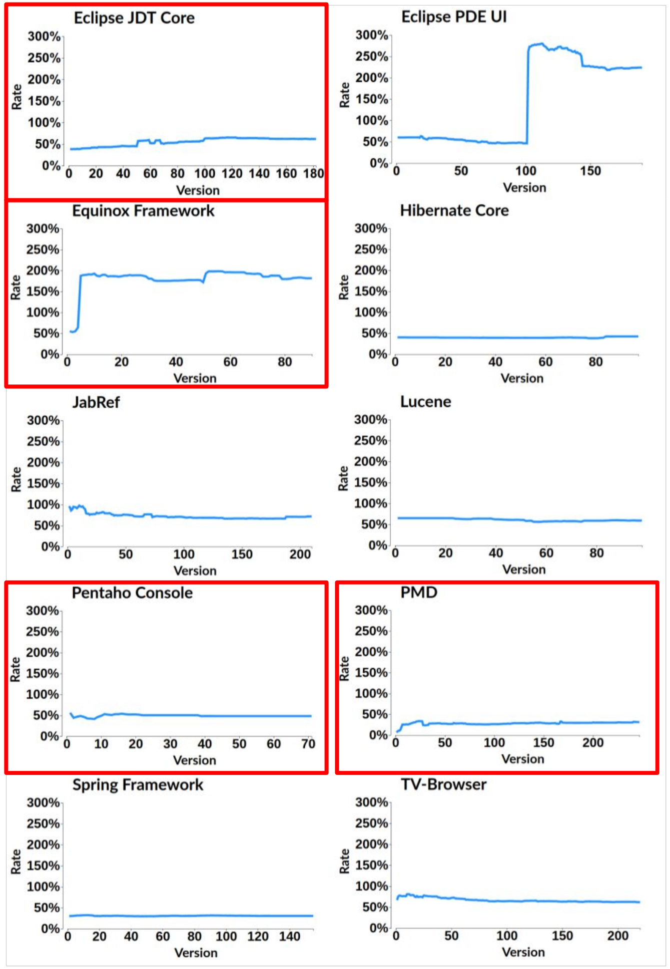
RQ2: How does the relation between dimension metrics behave throughout the evolution of software systems?

RQ3: What set of classes within the software system affects the dimensions of growth/decrease and how these classes evolve?

Global NOA/NOM proportion



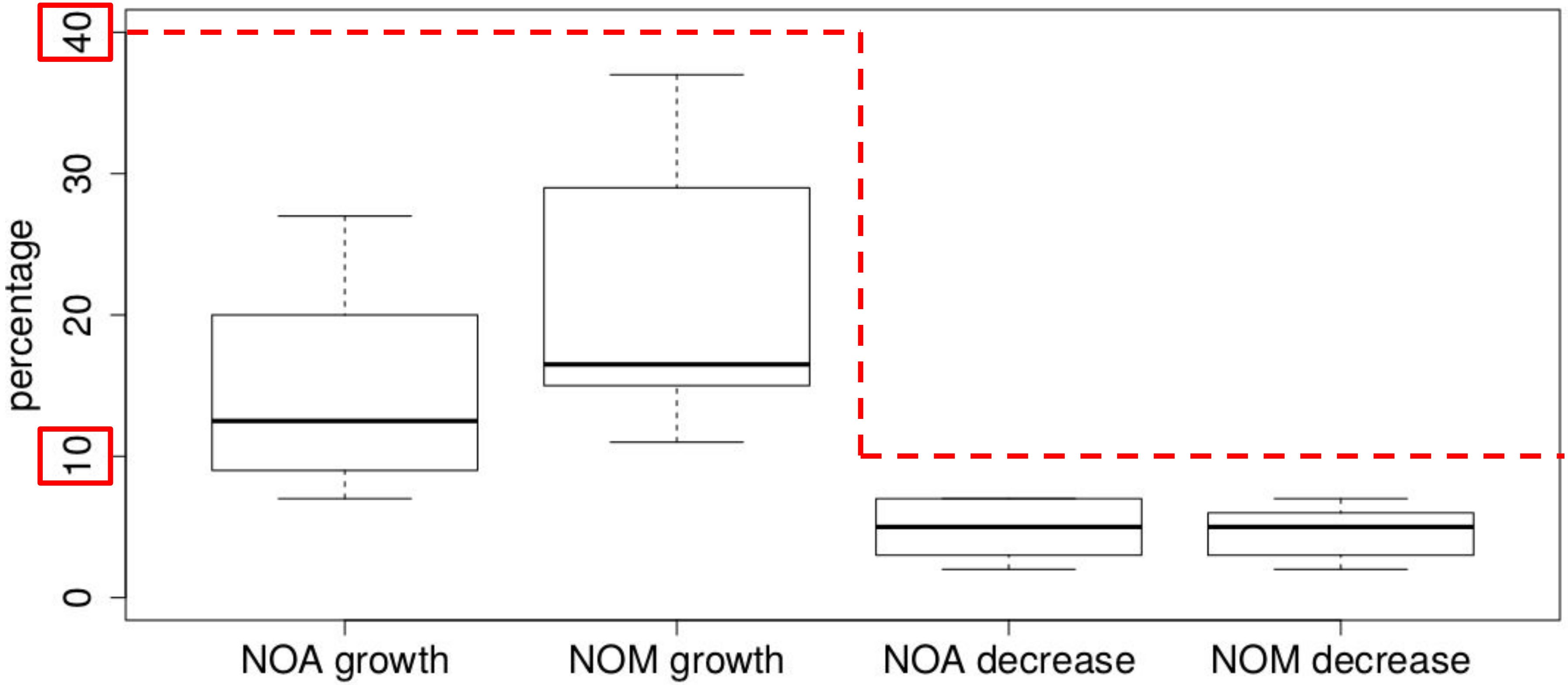
AVG of class-by-class NOA/NOM proportion



RQ1: Which model better describes the evolution of the dimensions in software systems?

RQ2: How does the relation between dimension metrics behave throughout the evolution of software systems?

RQ3: What set of classes within the software system affects the dimensions of growth/decrease and how these classes evolve?


40
30
20
10
0
percentage
NOA growth
NOM growth
NOA decrease
NOM decrease

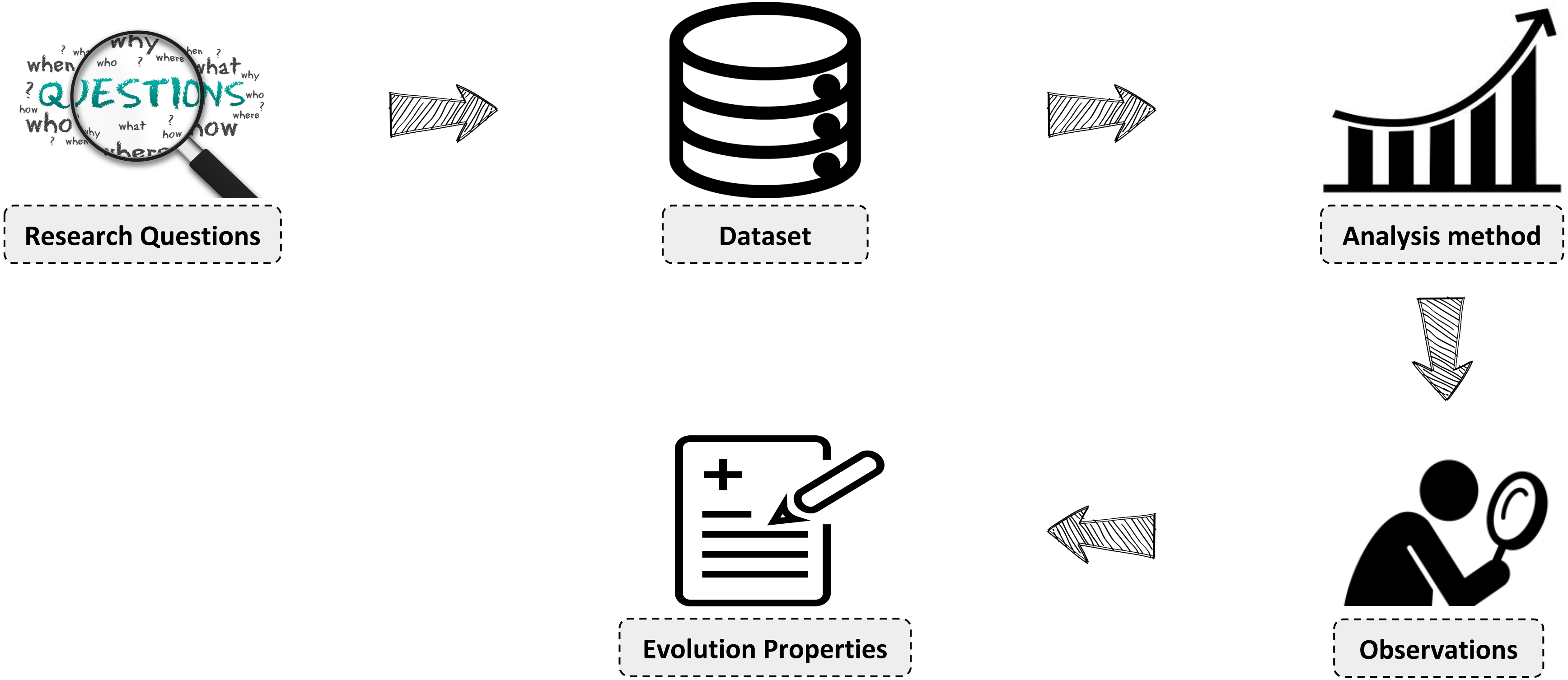
Intersection from the system perspective

System	i	ii	iii	iv
Eclipse JDT Core	22%	2%	1%	3%
Eclipse PDE UI	7%	3%	1%	2%
Equinox Framework	11%	1%	1%	0%
Hibernate Core	7%	1%	0%	0%
JabRef	10%	1%	0%	1%
Lucene	7%	1%	0%	1%
Pentaho Console	5%	1%	0%	1%
PMD	7%	1%	1%	0%
Spring Framework	18%	4%	1%	1%
TV-Browser	16%	3%	1%	1%

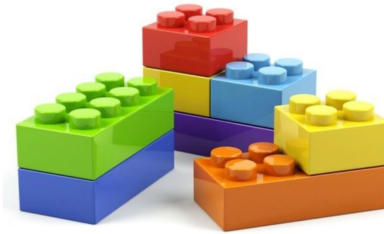
Intersection from the trend class perspective

System	i	ii	iii	iv
Eclipse JDT Core	43%	5%	2%	5%
Eclipse PDE UI	28%	11%	2%	7%
Equinox Framework	38%	4%	3%	2%
Hibernate Core	35%	6%	2%	1%
JabRef	39%	4%	2%	6%
Lucene	31%	3%	1%	3%
Pentaho Console	24%	6%	1%	4%
PMD	33%	5%	3%	1%
Spring Framework	42%	9%	1%	3%
TV-Browser	40%	9%	2%	3%

Case i	Case ii	Case iii	Case iv
NOA ↑ NOM ↑	NOA ↓ NOM ↓	NOA ↑ NOM ↓	NOA ↓ NOM ↑

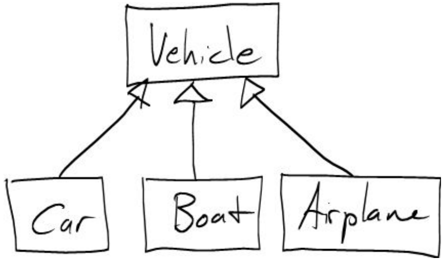

Research Questions
Dataset
Analysis method
Observations
Evolution Properties

Coupling



1. Coupling grows linearly over time
2. Unnecessary coupling is continuously higher than necessary coupling
3. A small group of classes have high coupling
4. Complexity is introduced since the first versions of a system
5. Legacy classes mainly contribute to coupling evolution
6. There is no association between fan-in and fan-out

Inheritance Hierarchy



1. Inheritance hierarchy tends to increase in depth and decrease in breadth over time

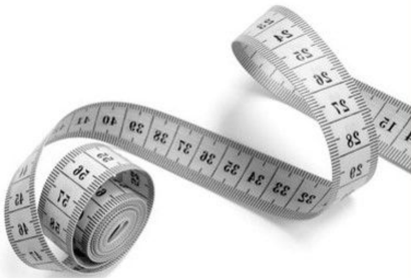
2. Inheritance hierarchy depth grows according to a linear model

3. Inheritance hierarchy breadth decreases according to a quadratic model

4. A small part of the system influences the growth and the decrease of the inheritance hierarchy

5. There is no association between the depth and the breadth of a class

Size of Classes



1. The size of classes grows according to the linear model

2. The proportion of the number of attributes in relation to the number of methods in a class grows over time

3. A small group of classes affects the growth of system size

4. The evolution of the number of attributes and the number of methods are correlated

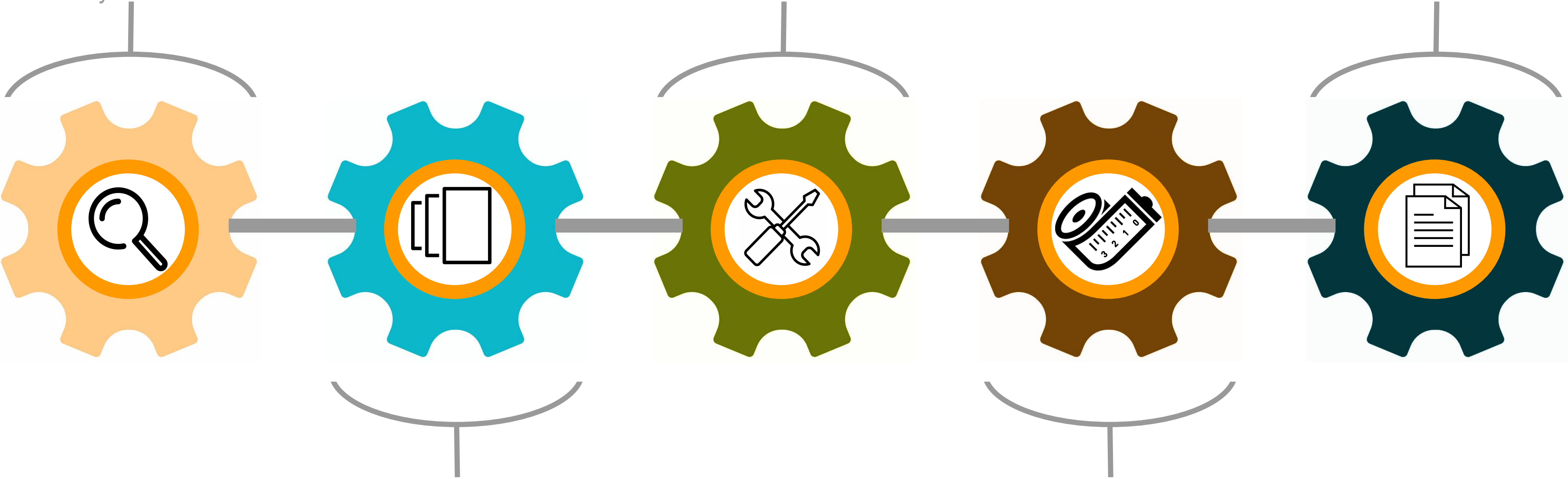

NEXT STEPS

Novel Dataset

Step 1
Definition of software systems

Step 3
Identify tools for collecting software metrics

Step 5
Organize the collected metrics as time series



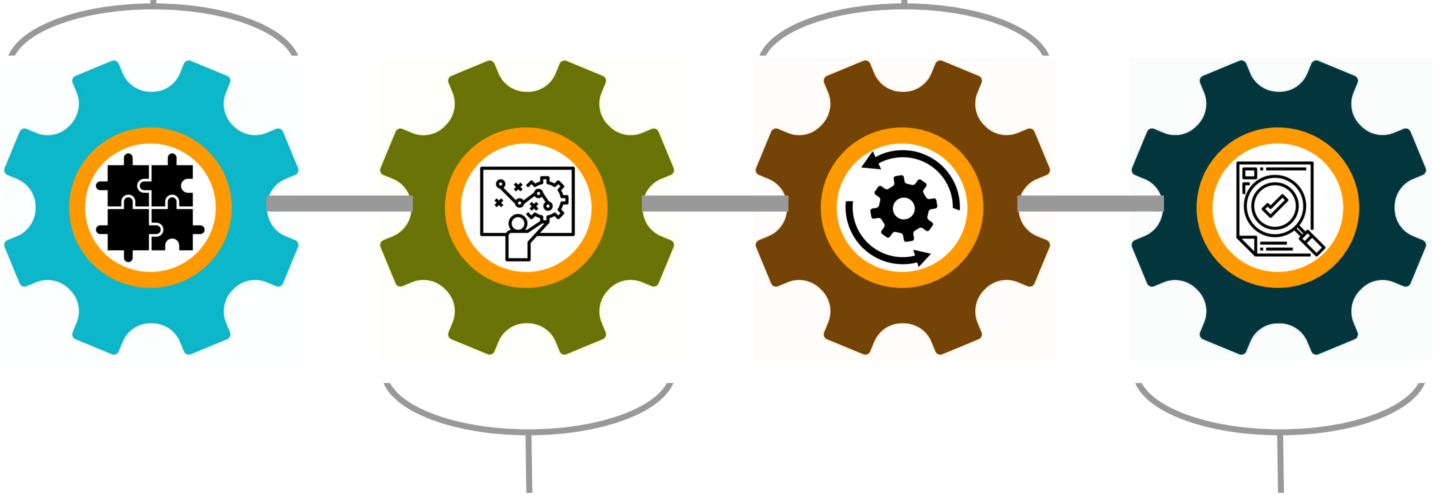
Step 2
Extract the software releases from their repository

Step 4
Collect the measures from the versions of the software systems

Prediction Method

Step 1
Use of the behavior analysis approach as the foundation

Step 3
Automation of the selection strategy



Step 2
Definition of a selection strategy for prediction models

Step 4
Evaluation of the prediction method


conclusion

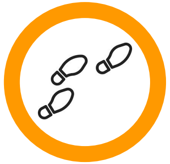
Conclusion

Part 1

- ➔ Compilation of the body of knowledge on software evolution
- ➔ Identification of five lines of study
- ➔ Characterization of the software evolution datasets
- ➔ Detailing of the lines

Part 2

- ➔ Definition of a novel analysis method based on time series
- ➔ Analysis of the evolution of three software dimensions
- ➔ Extraction of a set composed of 15 properties



Part 3

- ➔ Construction of a novel software evolution dataset
- ➔ Definition of a prediction software evolution method

Publications

- *Software Engineering Evolution: The History Told by ICSE* - SBES 2019 (Short paper)

- *Analysis of Coupling Evolution on Open Source Systems* - SBCARS 2019

- *A Comprehensive Systematic Literature Review of Software Evolution* - Submitted to an international journal (Under Review)

- *Evolution of Size and Inheritance in Object-Oriented Software - A Time Series Based Approach* - International journal

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
DCC-UFMG

Modeling and Predicting Evolution of Software Quality Internal Attributes

Bruno Luan de Sousa

Advisor: Mariza Andrade da Silva Bigonha (UFMG)

Co-Advisors: Kecia Aline Marques Ferreira (CEFET-MG), and
Glaura da Conceição Franco (UFMG)

August 28th, 2020

LLP Laboratório de Linguagens de Programação