A tertiary systematic literature review on Software Visualization

Laure Bedu, Olivier Tinh, Fabio Petrillo
Université du Québec à Chicoutimi
Chicoutimi, Canada
laure.bedu1@uqac.ca,olivier.tinh1@uqac.ca,fabio@petrillo.com

UQAC
Département d'informatique et mathématique

Introduction

- Even though the Software Visualization (SV) has been studied from **decades**, many aspects are still uncertain or unexplored
- It makes complex for **researchers and practitioners** to have a clear **landscape of SV**
- Our main goal is to identify the **consolidate** approaches and discussions on **SV**

Introduction

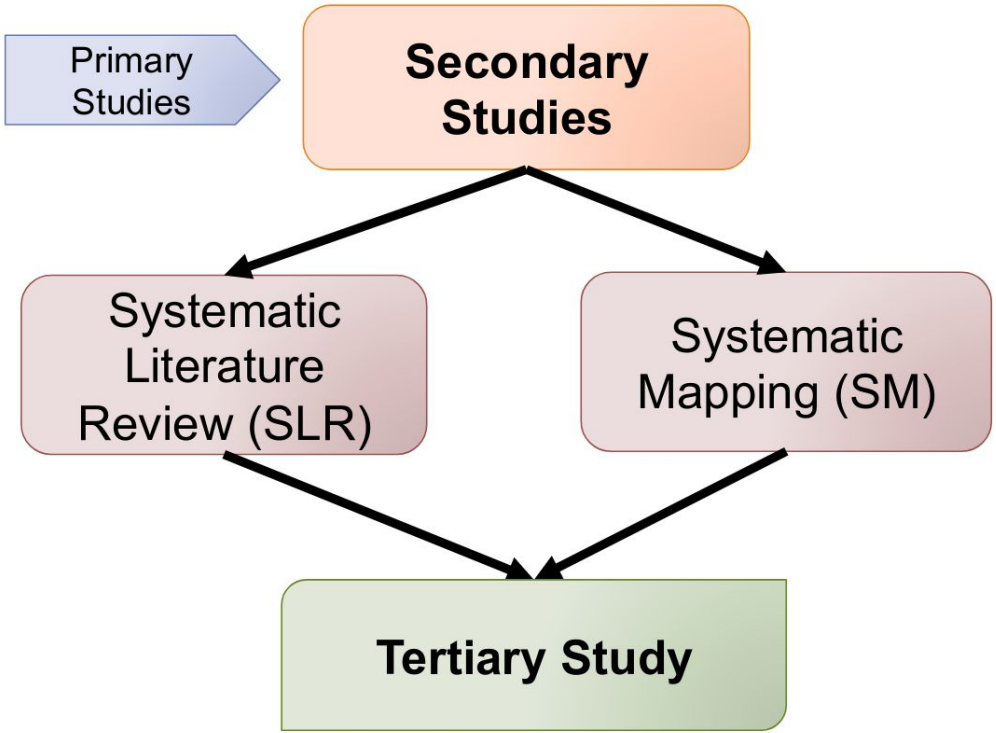
- The **main contributions** of this study are:
  - a **collection of 48 secondary studies** on SV
  - an up-to-date map of **state-of-the-art in SV** and its implications for future research
  - a set of **recommendations acting as sound guidelines** for future development in SV field

Introduction

- For achieving this goal, we applied the **tertiary systematic literature review** (SLR) methodology
- What is a SLR?
  - Systematic literature review (also referred to as a systematic review) is a form of **secondary study** that uses a **well-defined methodology** to identify, analyze and interpret all **available evidence** related to a specific **research question** in a way that is unbiased (to a degree) and **repeatable**. (Kitchenham's Guideline, 2007)

Introduction

- For achieving this goal, we applied the **tertiary systematic literature review** (SLR) methodology

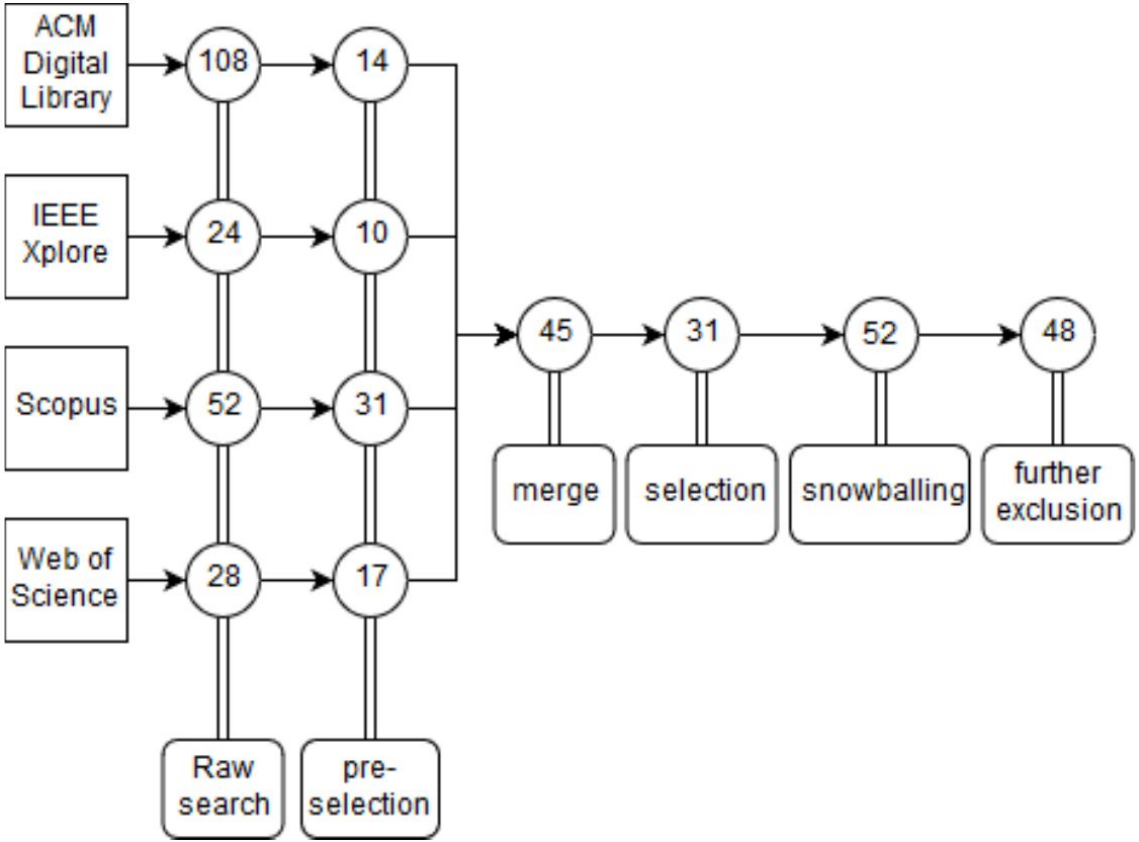


Study design

Main research questions

- RQ1 - What are the current **publication characteristics** in the software visualization research field (year, types, main topics)?
- RQ2 - What are the application **domain** of software visualization?
- RQ3 What are the main **issues** regarding software visualization application?

Search and Selection of Secondary Studies

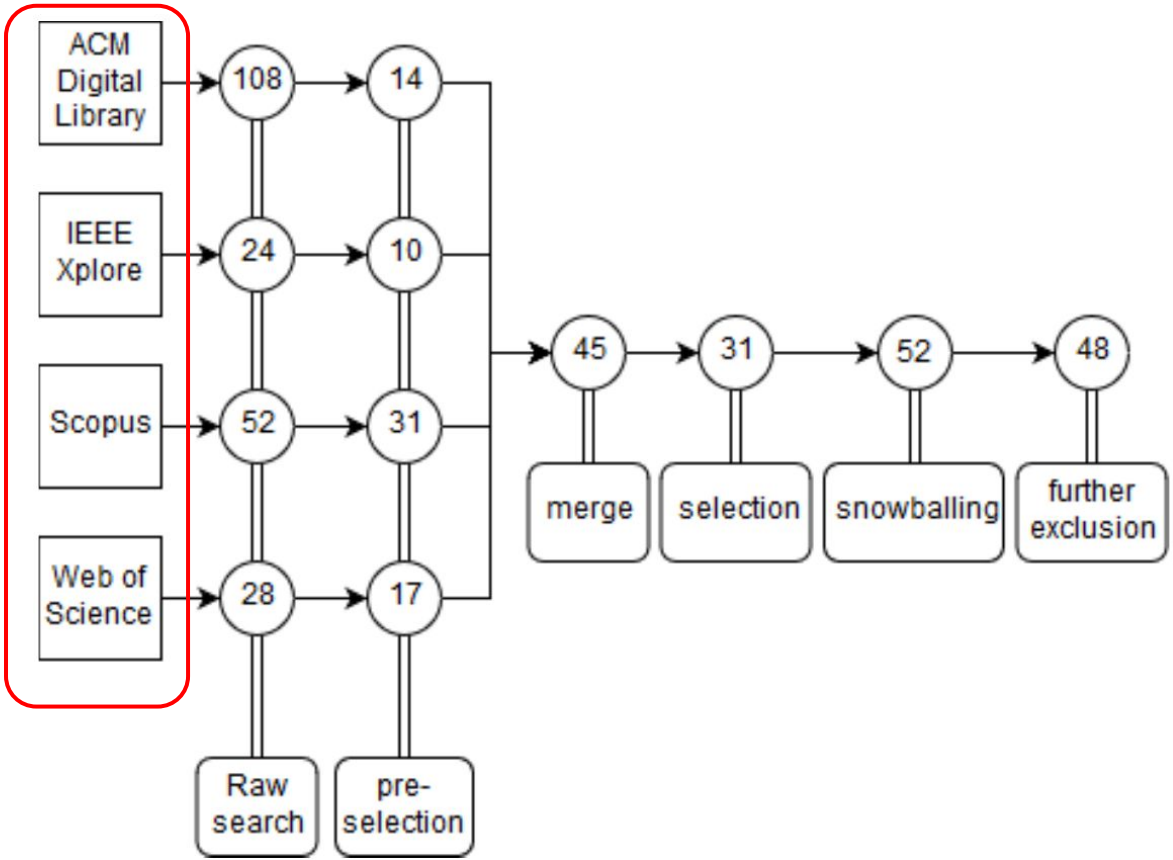


Search and Selection of Secondary Studies

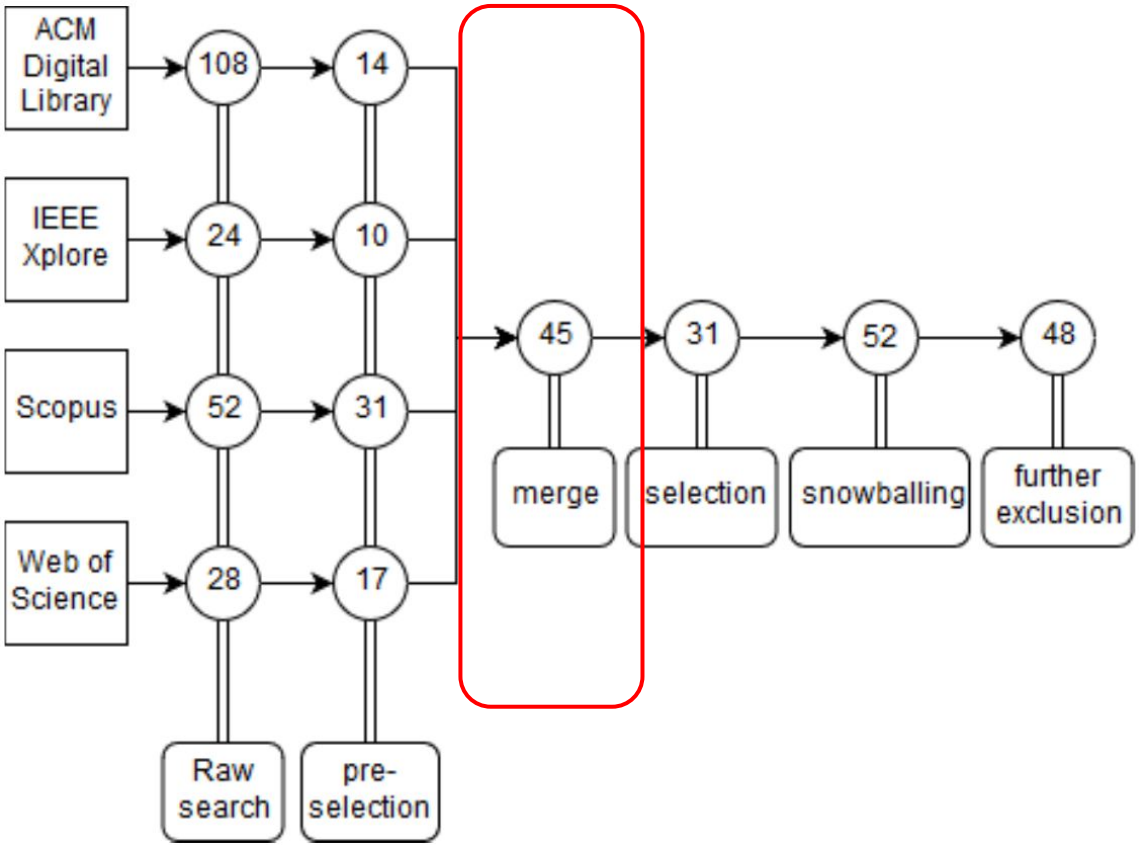
Listing 3 - Scopus

```
TITLE-ABS-KEY ("Software visualization")
AND TITLE-ABS-KEY ("systematic review"
OR "literature review"
OR "systematic mapping"
OR "mapping study"
OR "systematic map" OR "meta-analysis"
OR "survey" OR "literature analysis")
AND (LIMIT-TO(SUBJAREA, "COMP"))
```

Search and Selection of Secondary Studies



Search and Selection of Secondary Studies



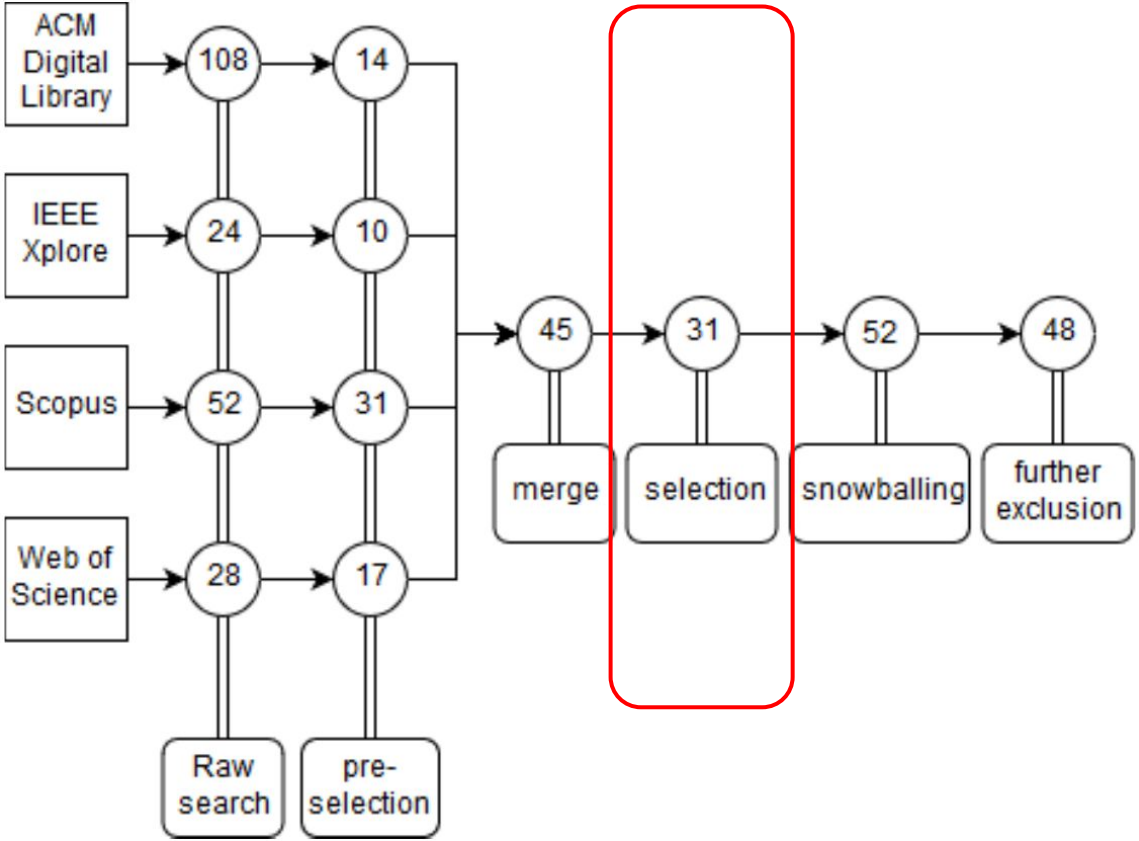
Search and Selection of Secondary Studies

- I1 - **Title, keyword list, and abstract** explicitly stating that the paper is related to SV.
- I2 - Studies conducting a **systematic mapping study or SLR** on SV or any sub-field.
- I3 - Studies providing a state of the art, taxonomy, review on SV, or any sub-field.

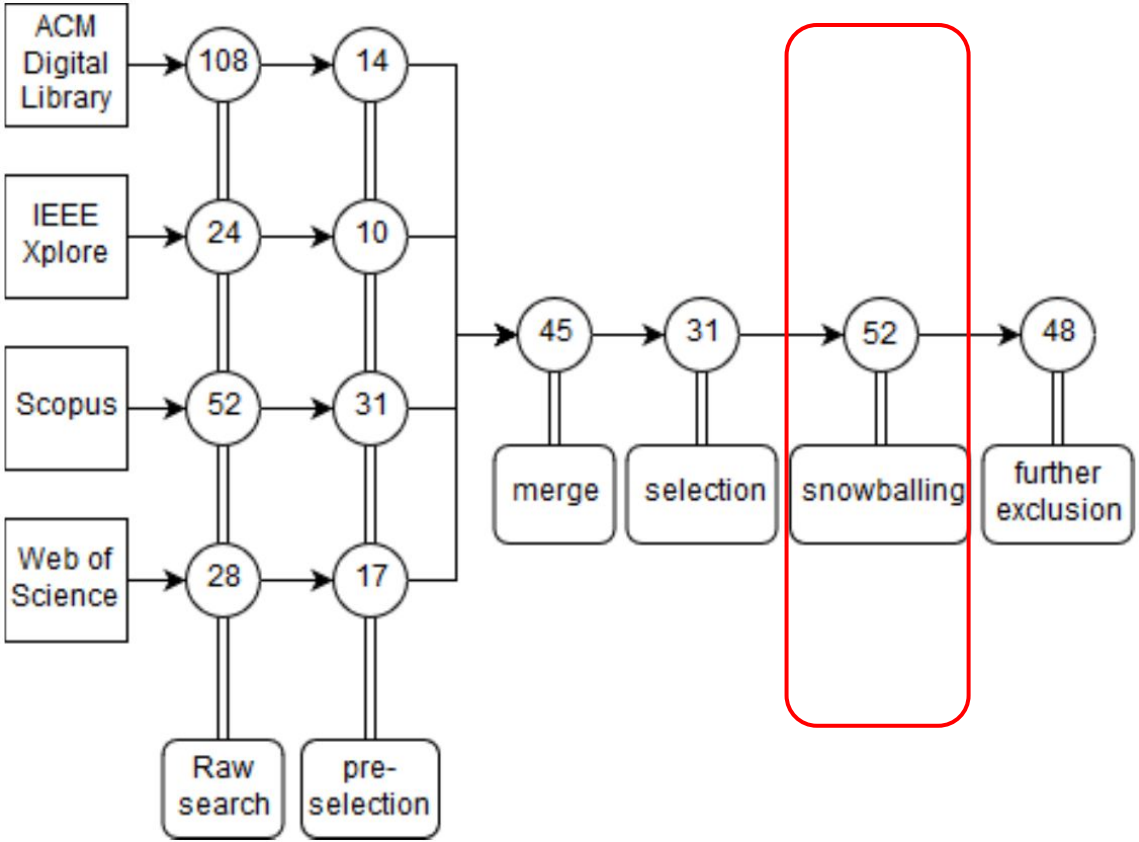
Search and Selection of Secondary Studies

- E1 - Conference proceedings, e-books, slideshows, or formats different from research papers.
- E2 - Papers focusing on a unique tool, technique, not broad enough.
- E3 - The paper's full text is not available for download.
- E4 - The paper is not in English.
- E5 - The publication year is lower than 2000**.
    - However, we included the [3] Price, B.A., Baecker, R.M., and Small, I.S. "A Principled Taxonomy of Software Visualization", Journal of Visual Languages and Computing, Volume 4, Issue 3, Pages 211-266, **1993**.

Search and Selection of Secondary Studies



Search and Selection of Secondary Studies



Search and Selection of Secondary Studies

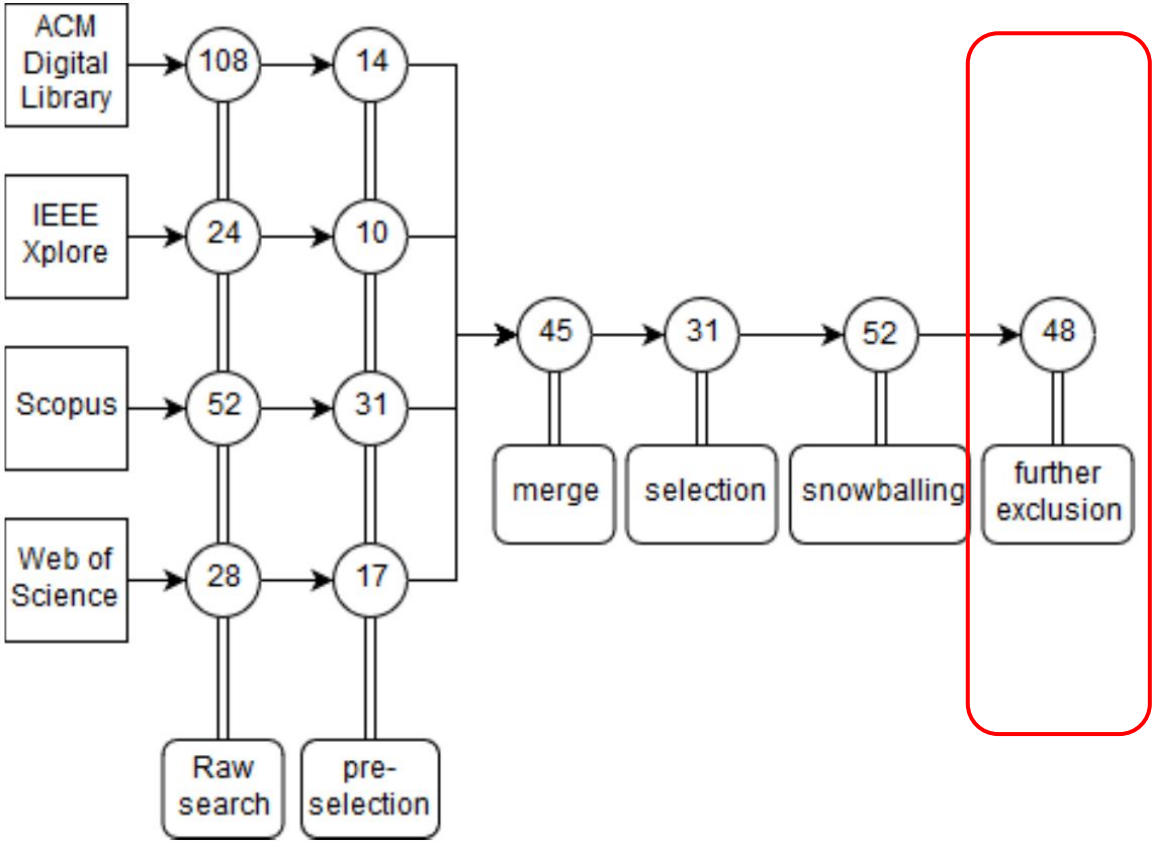


TABLE I: List of selected secondary studies

	Article	Topic	Year	Type
[3]	A principled taxonomy of software visualization	General	1993	Taxonomy
[35]	Software visualization tools: Survey and analysis	General	2001	Survey
[13]	A Task Oriented View of Software Visualization	General	2002	Classification
[20]	Documenting Software Systems with Views III: Towards a Task-oriented Classification of Program Visualization Techniques	General	2002	Classification
[32]	Software visualization	General	2005	Literature review
[26]	On the use of visualization to support awareness of human activities in software development: a survey and a framework	General	2005	Taxonomy
[37]	The paradox of software visualization	General	2005	Opinion paper
[48]	Visualization Techniques for Program Comprehension A Literature Review	General	2006	Literature review
[29]	Requirements of software visualization tools: A literature survey	General	2007	Literature review
[17]	An Overview of 3D Software Visualization	General	2009	Literature review
[25]	Mental imagery and software visualization in high-performance software development teams	General	2009	Survey
[46]	Visualization of the Static Aspects of Software: A Survey	General	2011	Literature review
[16]	An Information Visualization Feature Model for Supporting the Selection of Software Visualizations	General	2014	Literature review
[27]	Past, present, and future of 3D software visualization: A systematic literature analysis	General	2015	SLR[2]
[39]	To enlighten hidden facts in the code: A review of software visualization metaphors	General	2015	SLR
[34]	Software Visualization Today: Systematic Literature Review	General	2016	SLR
[40]	Towards Actionable Visualisation in Software Development	General	2016	SMS[1]
[43]	Visual augmentation of source code editors: A systematic mapping study	General	2018	Taxonomy
[22]	Exploring the Role of Visualization and Engagement in Computer Science Education	Education	2002	Taxonomy
[4]	A Review of Generic Program Visualization Systems for Introductory Programming Education	Education	2013	Literature review
[18]	Are Visualization Tools Used in Programming Education?: By Whom, How, Why, and Why Not?	Education	2014	Survey
[24]	Learning principles in program visualizations: A systematic literature review	Education	2016	SLR
[36]	Survey of software visualization systems to teach message-passing concurrency in secondary school	Education	2017	Literature review
[38]	Theoretical underpinnings of learner engagement in software visualization system: A systematic literature review protocol	Education	2018	SLR
[9]	A systematic literature review of student engagement in software visualization: a theoretical perspective	Education	2019	SLR
[14]	A Taxonomy of Computer Architecture Visualizations	Architecture	2002	Taxonomy
[1]	A Framework for Software Architecture Visualisation Assessment	Architecture	2005	Classification
[7]	A Survey Paper on Software Architecture Visualization	Architecture	2008	Literature review
[44]	Visualization and Evolution of Software Architectures	Architecture	2012	Literature review
[11]	A systematic review of software architecture visualization techniques	Architecture	2014	SLR
[5]	A Survey of Successful Evaluations of Program Visualization and Algorithm Animation Systems	Evaluation	2009	Literature review
[21]	Evaluation of Software Visualization Tools	Evaluation	2009	Literature review
[41]	Validation of Software Visualization Tools: A Systematic Mapping Study	Evaluation	2014	SMS
[42]	Validation of the City Metaphor in Software Visualization	Evaluation	2015	Survey + Experimental
[8]	A systematic literature review of software visualization evaluation	Evaluation	2018	SLR
[33]	Software visualization in software maintenance, reverse engineering, and re-engineering: a research survey	Maintenance	2003	Survey
[19]	Classifying Desirable Features of Software Visualization Tools for Corrective Maintenance	Maintenance	2008	Literature review
[6]	A survey on goal-oriented visualization of clone data	Maintenance	2015	Literature review
[28]	Program comprehension through reverse-engineered sequence diagrams: A systematic review	Maintenance	2018	SLR
[47]	Visualization Techniques for Application in Interactive Product Configuration	Product Line	2011	Literature review
[45]	Visualization for Software Product Lines: A Systematic Mapping Study	Product Line	2016	SMS
[10]	A systematic mapping study of information visualization for software product line engineering	Product line	2017	SMS
[2]	A meta-study of algorithm visualization effectiveness	Algorithm	2002	SMS
[15]	Algorithm Visualization: The State of the Field	Algorithm	2010	Literature review
[31]	Software evolution visualization: A systematic mapping study	Evolution	2013	SMS
[30]	Software evolution visualization techniques and methods - a systematic review	Evolution	2016	SLR
[12]	A Systematic Survey of Program Comprehension through Dynamic Analysis	Dynamic Analysis	2009	SLR
[23]	Information Visualization for Agile Software Development Teams	Process	2014	SMS

[1]SMS = Systematic Mapping Study, [2]SLR = Systematic Literature Review

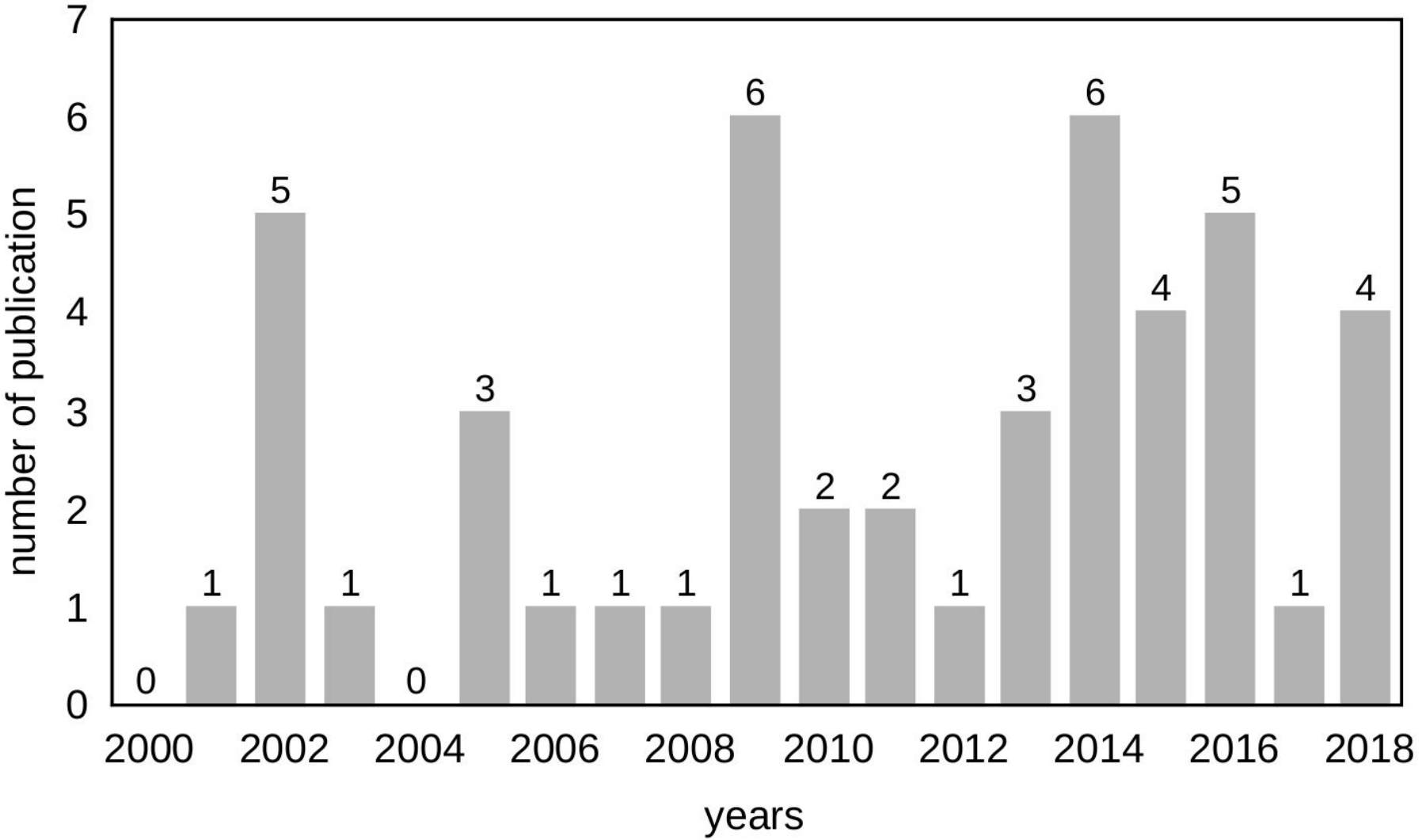
We organize a

catalogue of 48

Software Visualization

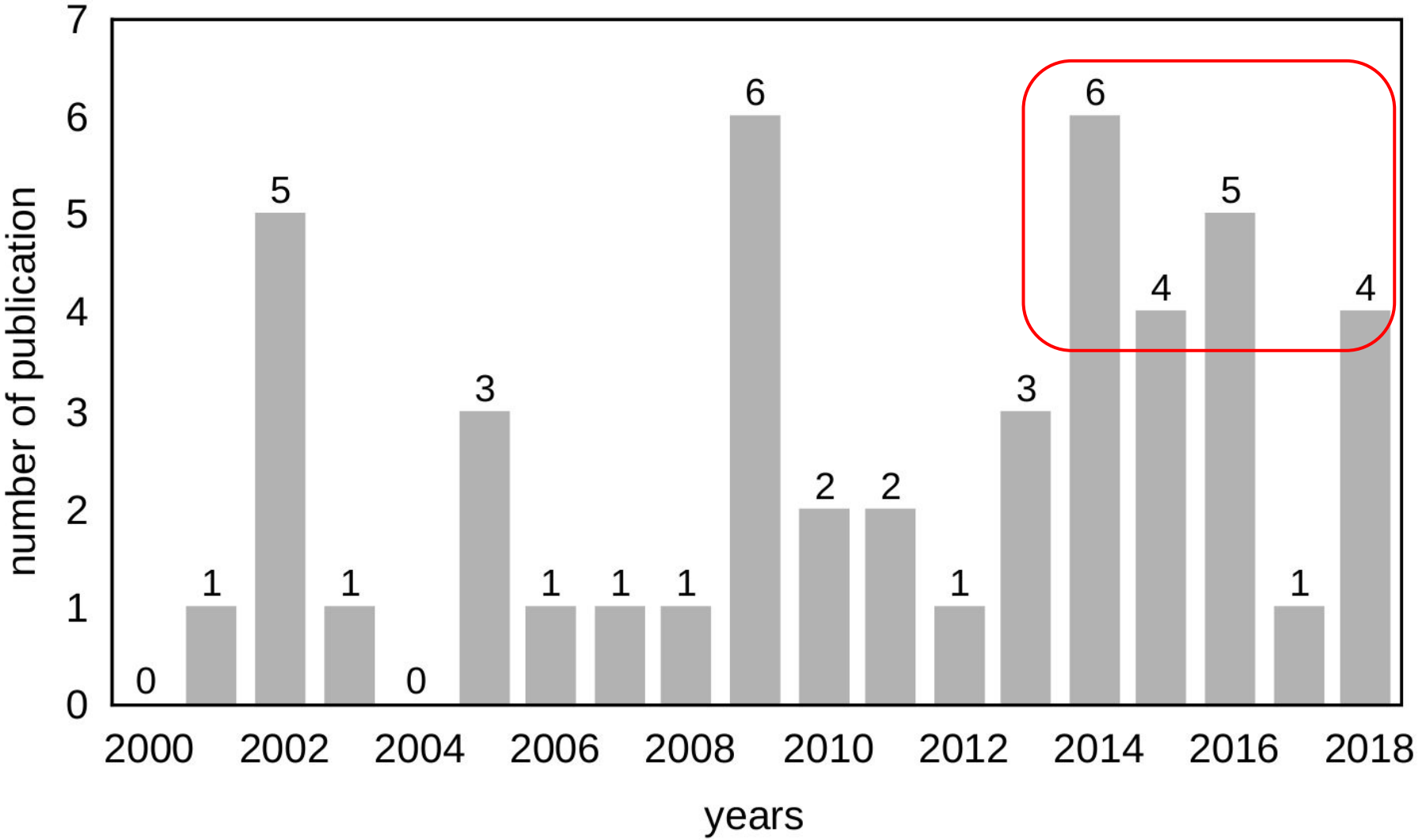
Secondary Studies

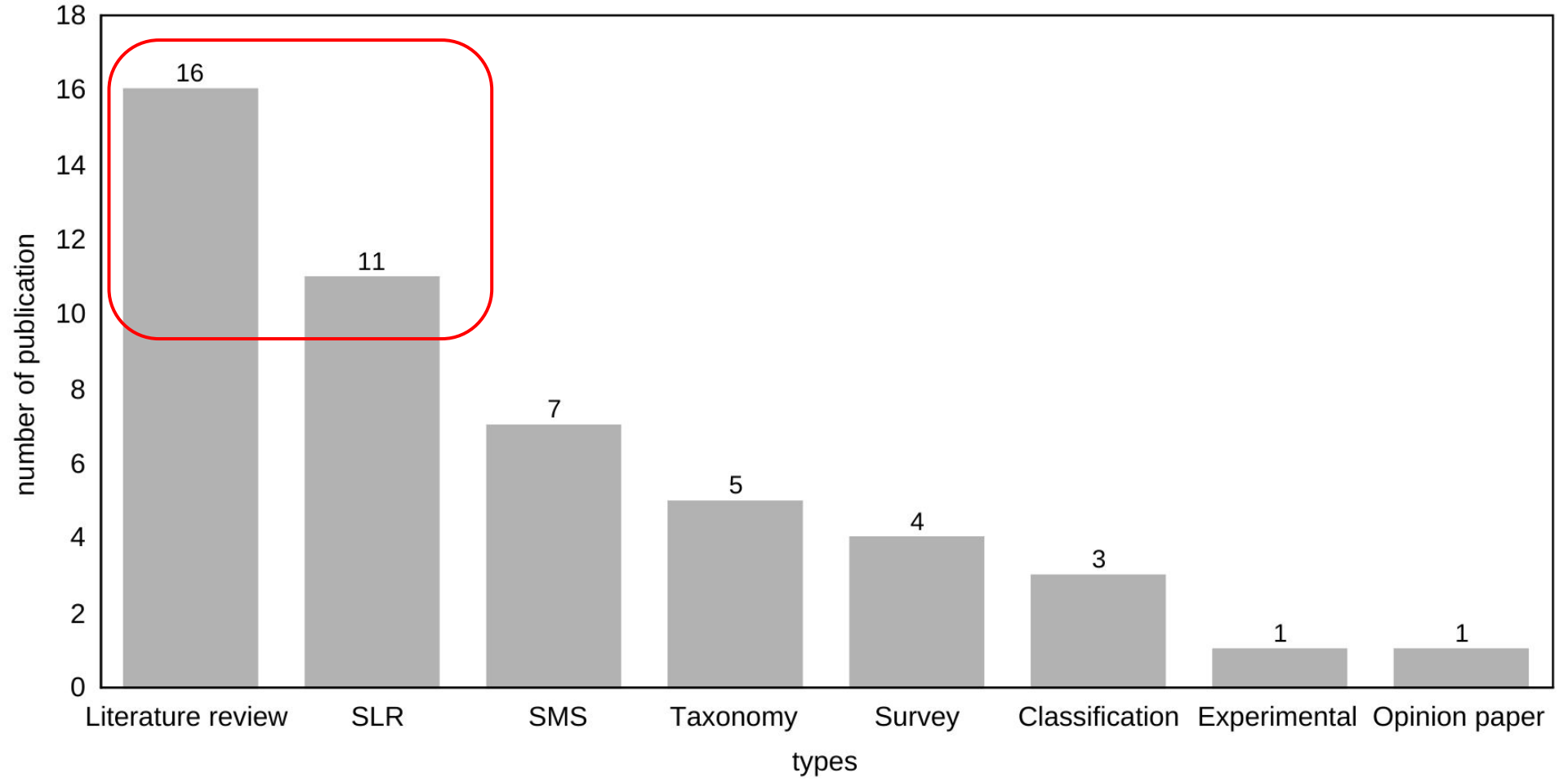
Results

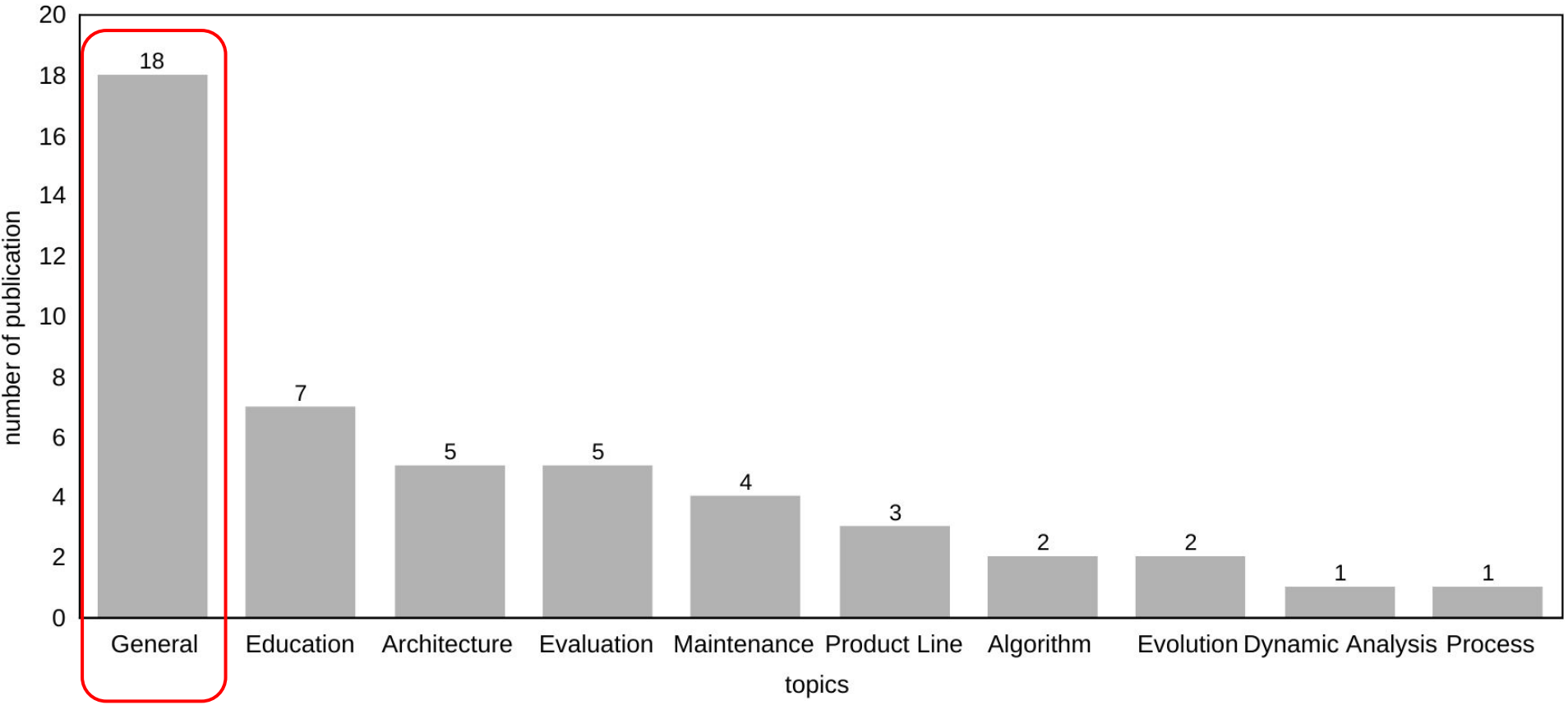

number of publication
years
0
1
2
3
4
5
6
7
2000
2002
2004
2006
2008
2010
2012
2014
2016
2018
0
1
5
1
0
3
1
1
1
6
2
2
1
3
6
4
5
1
4

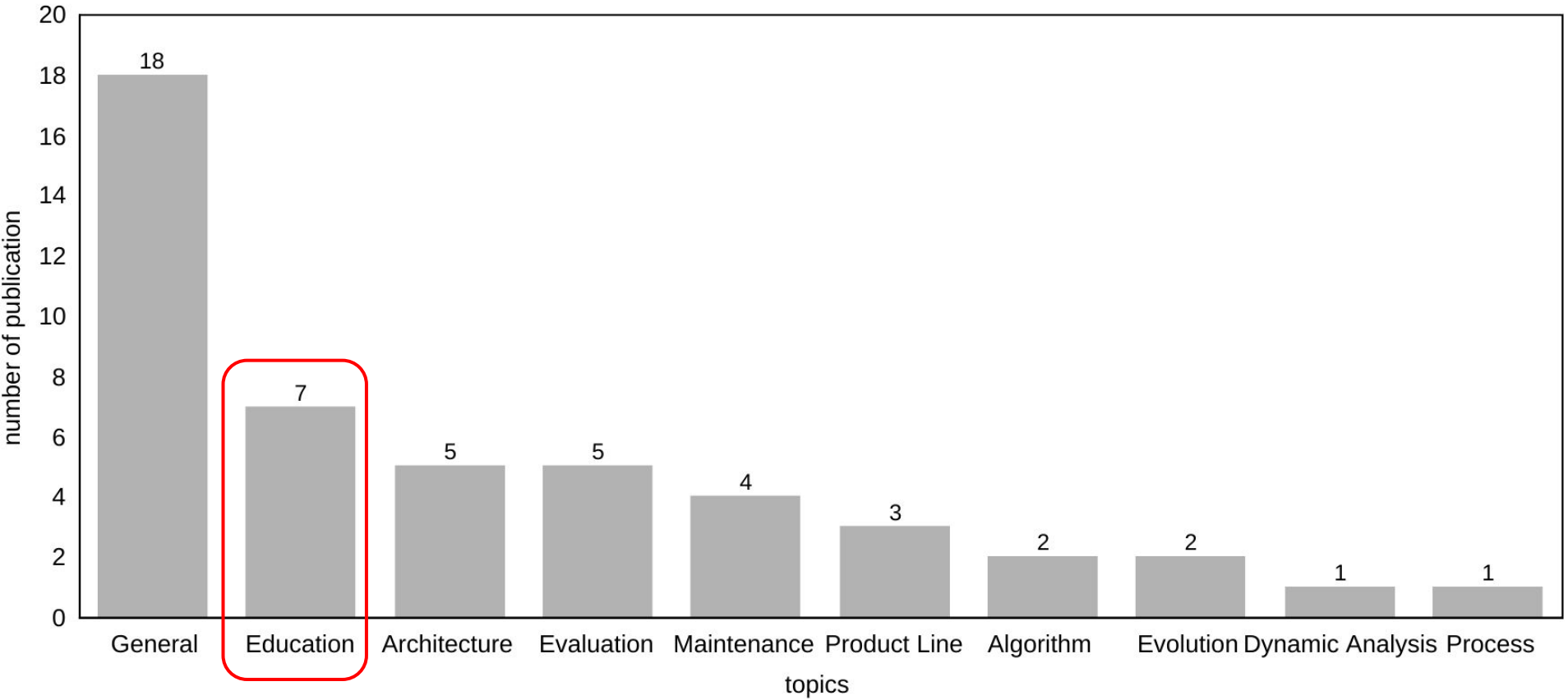
Publications per year

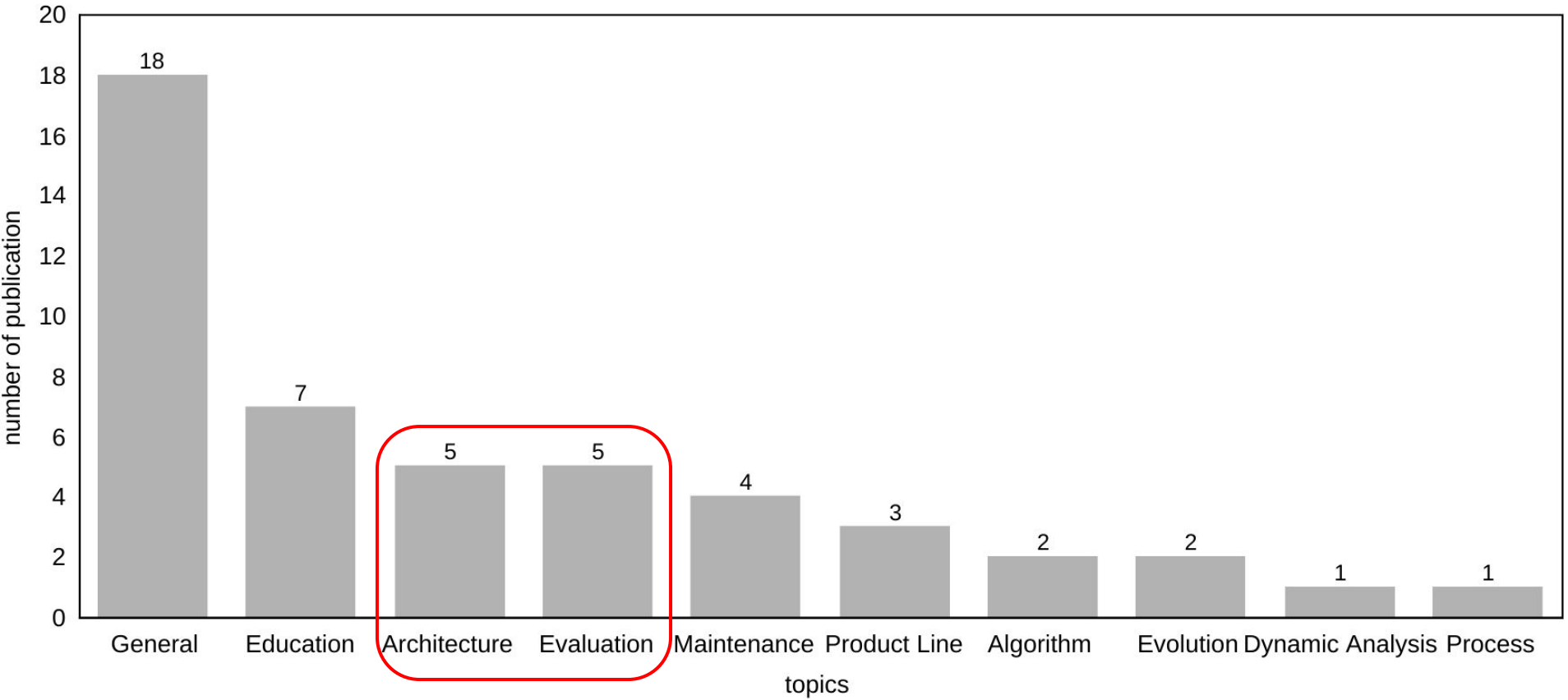
- We can note that the **scientific interest** on SV in the last years has **remained constant and steady**.
- We found secondary studies from **2001 to 2018**, and only 2004 had a discontinuity (**peaks** of 6 articles in **2009** and **2014**).
- Thus, we observed a **regular production of secondary in SV domain**.


number of publication
years
0
1
2
3
4
5
6
7
2000
2002
2004
2006
2008
2010
2012
2014
2016
2018
0
1
5
1
0
3
1
1
1
6
2
2
1
3
6
4
5
1
4


18
16
14
12
10
8
6
4
2
0
number of publication
16
11
7
5
4
3
1
1
Literature review
SLR
SMS
Taxonomy
Survey
Classification
Experimental
Opinion paper
types


number of publication
20
18
16
14
12
10
8
6
4
2
0
18
7
5
5
4
3
2
2
1
1
General
Education
Architecture
Evaluation
Maintenance
Product Line
Algorithm
Evolution
Dynamic Analysis
Process
topics


20
18
16
14
12
10
8
6
4
2
0
number of publication
18
7
5
5
4
3
2
2
1
1
General
Education
Architecture
Evaluation
Maintenance
Product Line
Algorithm
Evolution
Dynamic Analysis
Process
topics


20
18
16
14
12
10
8
6
4
2
0
number of publication
18
7
5
5
4
3
2
2
1
1
General
Education
Architecture
Evaluation
Maintenance
Product Line
Algorithm
Evolution
Dynamic Analysis
Process
topics

Discussion

(main insights)

The problem of scalability

- The challenge SV tools have to face is **scalability**
- Todays software systems are increasingly large and complex
- Software architecture is especially problematic
- It is an **open issue** yet

Evaluations lack rigor [7], [11]

- The evaluation of **SV still lacks rigor,** as this is the case for many SV tools
- It is essential to find **criteria** to determine what **makes an effective visualization**
- Paying attention to use more **objective evaluation methods** (e.g., **controlled experiments**) for providing convincing evidence to support its effectiveness
- Issue: controlled experiments are **very hard and expensive to perform**.

SV in Education [9],[24]

- We identify there is a growing area of research investigating the application of **3D graphics and algorithm animation** for **educational** purposes.
- Most of the effective SVs were built and guided by **cognitive theories** as c**onstructivism**, and **considering other learning theories** from different domains **could result in important contributions** in terms of **effectiveness**
- We also identify for PV to use **concrete visual allegories and gamification**

Software Maintenance Visualization

- The concerns of current SV tools for maintenance is that **they do not overcome** the domain specific issues of **scalability and complexity**
- Only appropriate for **small to medium size systems**
- SVs disconnect from **domain problems**

Trend -> real metaphors

- The trend in recent papers is to use **real metaphors**
- For example, **city metaphor** instead of abstract ones

Recommendations

SHARED RECOMMENDATIONS IDENTIFIED IN OUR SET

	Statement	Description	References	Total
R1	Conduct empirical studies to validate usefulness	On existing visualizations, or the ones being developed, to add them values and speed up the integration process. Controlled experiment, unbiased subjects, quantitative measures	[1]–[3], [7], [8], [10], [11], [21], [24], [26], [27], [30], [31], [33], [39], [41], [43], [44], [46], [47]	39.5% (19/48)
R2	Provide details on demand, avoid cognitive overload	Support human cognition, provide detail-on-demand interaction or higher level of abstraction	[6], [7], [13], [16], [20], [25], [28], [39], [44], [47], [48]	23% (11/48)
R3	Map techniques to meet specific goals, real problems	Embodying more knowledge of the application	[12], [15], [25], [31], [32], [37], [40], [44], [47], [48]	21% (10/48)
R4	Think about interoperability, community collaboration	Through exchange format, making tools available online (2, AV), reuse and use of recent techniques	[9], [14], [17], [26], [28], [30], [32], [35], [44], [45]	21% (10/48)
R5	Engage learner in activities	Allow them to construct their own visualization (active learning), student-centered vision	[1], [2], [4], [5], [9], [18], [22]	14.5% (7/48)
R6	Think about usability in the design	Keep interaction simple, provide help system	[5], [17], [20], [28], [35], [37]	12.5% (6/48)
R7	Have strong theoretical foundation	Psychology theories, other than constructivism for example	[1]–[3], [9], [38]	10.5% (5/48)
R8	Use multiple view	Ease selection of data, through source code view or more abstract ones	[3], [19], [20], [26], [32]	10.5% (5/48)
R9	Use automated tools	Reduce efforts to use the visualization	[1]–[3], [11]	8% (4/48)
R10	Exploit Virtual Environment	To extend visualization to perception, ease collaborative SE process, increase amount of data shown	[17], [32], [36], [46]	8% (4/48)
R11	Consider real needs of viewer	Meet stakeholders requirements, understand viewer objectives	[8], [26], [32], [44]	8% (4/48)
R12	Scale up to handle complexity of current software	Handle large amount of data, on production scale level	[10], [28], [37]	6% (3/48)
R13	State research method and questions	Besides, discuss the approach goals and validation strategy	[30], [33], [40]	6% (3/48)
R14	Enable customization	Permits to meet viewer-specific needs	[28], [37]	4% (2/48)
R15	Use Gamification	Contribute to effectiveness of learning tools	[9], [24]	4% (2/48)

We organise a catalogue of **15 recommendations** from secondary studies, classifying by number of occurrences in the studies.

R1 - Conduct **empirical studies** to validate **usefulness**

Cited by **40%** of papers (19/48)

R2 - Provide details on **demand**, avoid **cognitive overload**

Cited by 23% of papers (11/48)

R3 - Map techniques to meet **specific goals**, real problems

Cited by 21% of papers (10/48)

R4 - Think about interoperability, community collaboration

Cited by 21% of papers (10/48)

R5 - Engage learner in activities

Cited by 14.5% of papers (7/48)

R6 - Think about usability in the design

Cited by 12.5% of papers (6/48)

R7 - Have strong theoretical foundation

Cited by 10.5% of papers (5/48)

R8 - Use multiple views

Cited by 10.5% of papers (5/48)

Conclusion

Conclusions

- We **catalogued** and reviewed systematically **48 secondary** studies on **Software Visualization**
- We organised **15 main recommendations** to SV community from the studies
  - Conduct empirical studies to validade usefulness
  - Provide details on demand
  - Meet specific goals, real problems
- We can use those recommendations as a **guidelines** to further research projects

Conclusions

- We identified in the studies
  - the **lack of rigorous evaluation** or **theories** support to assess SV tools **effectiveness**
  - the **disconnection** between tool **design** and their **scope**
  - the **dispersal** of the research **community**

Conclusions

- We are **few** -- it is time to work **together** to construct and evaluate useful and focused SV
- It is time to **collaborate and consolidate** the most successful visualizations
- It is time to improve our **communication and "marketing strategy"** to become relevant and useful
- **Teach** the most successful and useful SV in our SE classes

A tertiary systematic literature review on Software Visualization

Laure Bedu, Olivier Tinh, Fabio Petrillo
Université du Québec à Chicoutimi
Chicoutimi, Canada
laure.bedu1@uqac.ca,olivier.tinh1@uqac.ca,fabio@petrillo.com

UQAC
Département d'informatique et mathématique