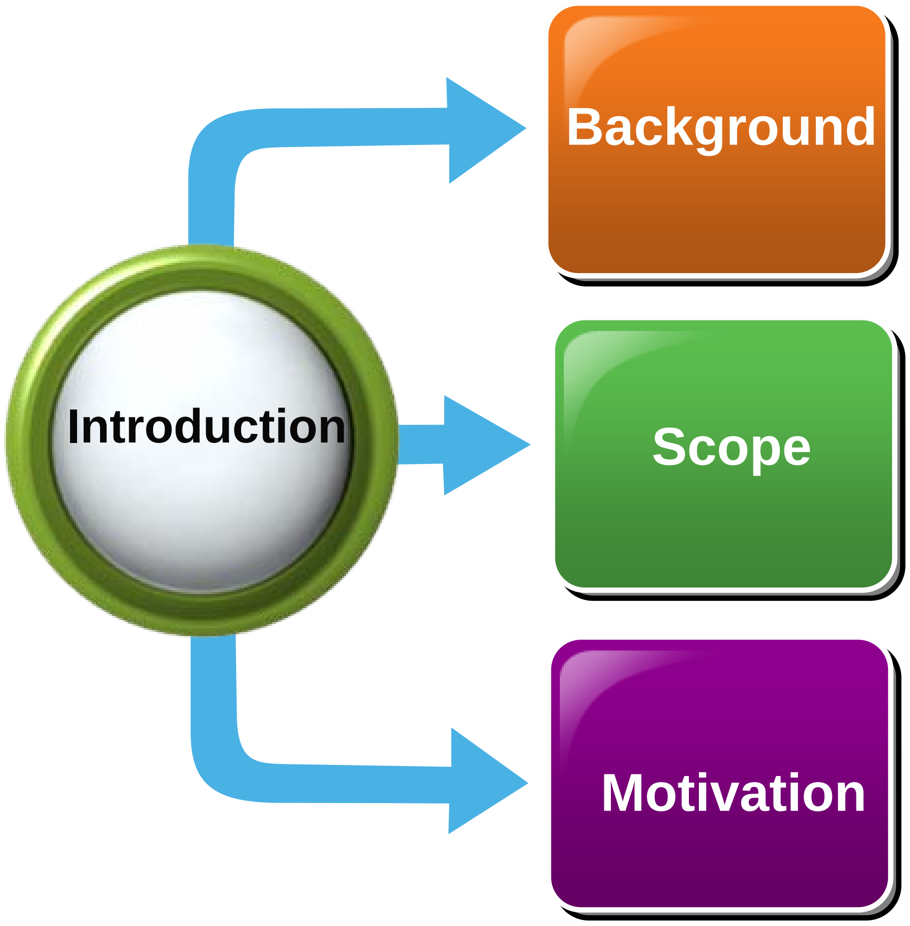
A Brief Survey of Code-Level Change Impact Analysis

Xiaobing Sun

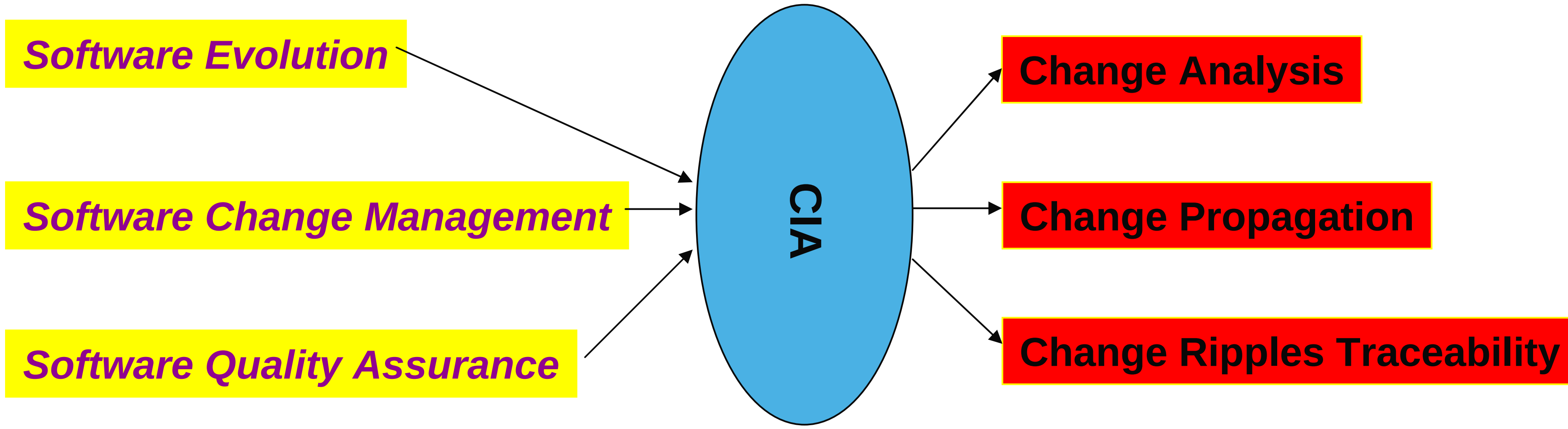
sundomore@163.com
School of Computer Science and Engineering
Southeast University

Contents

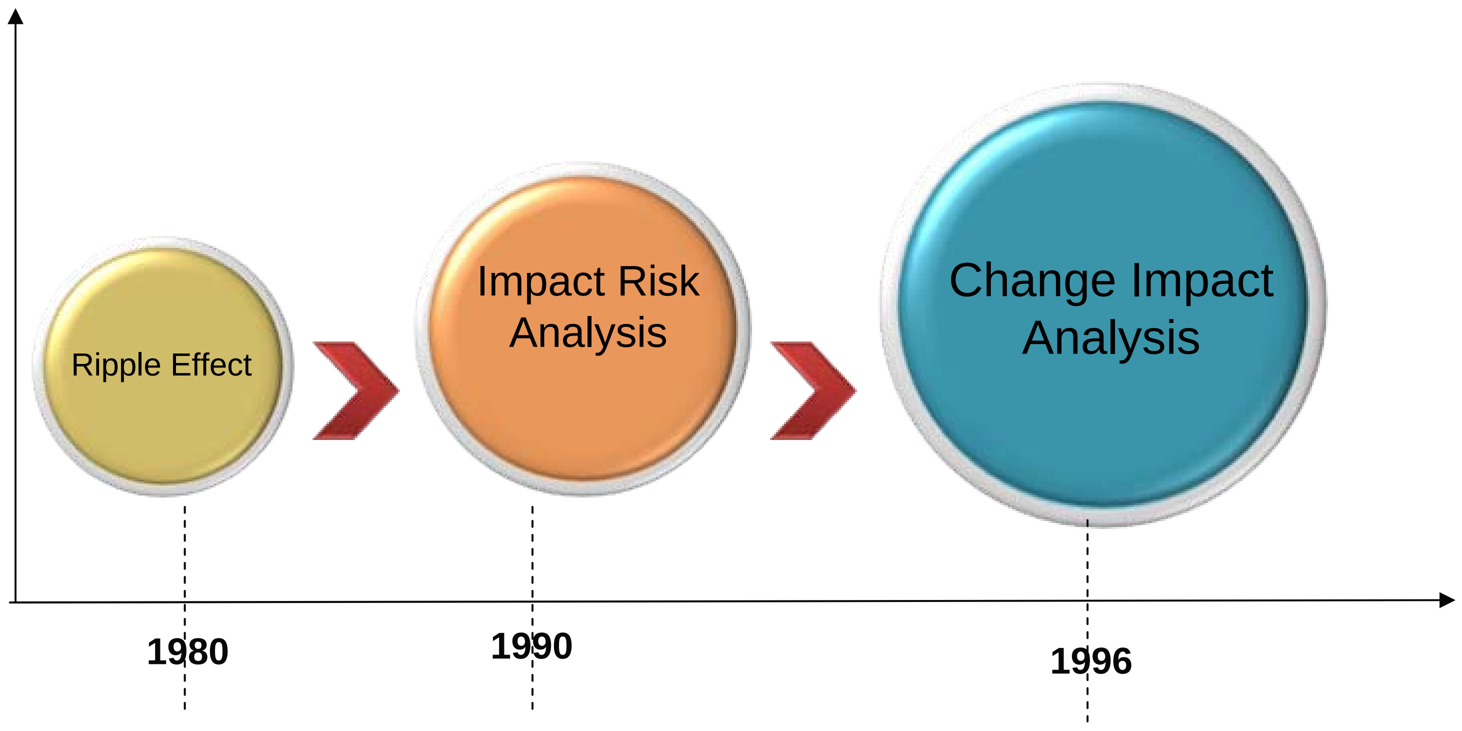
Introduction



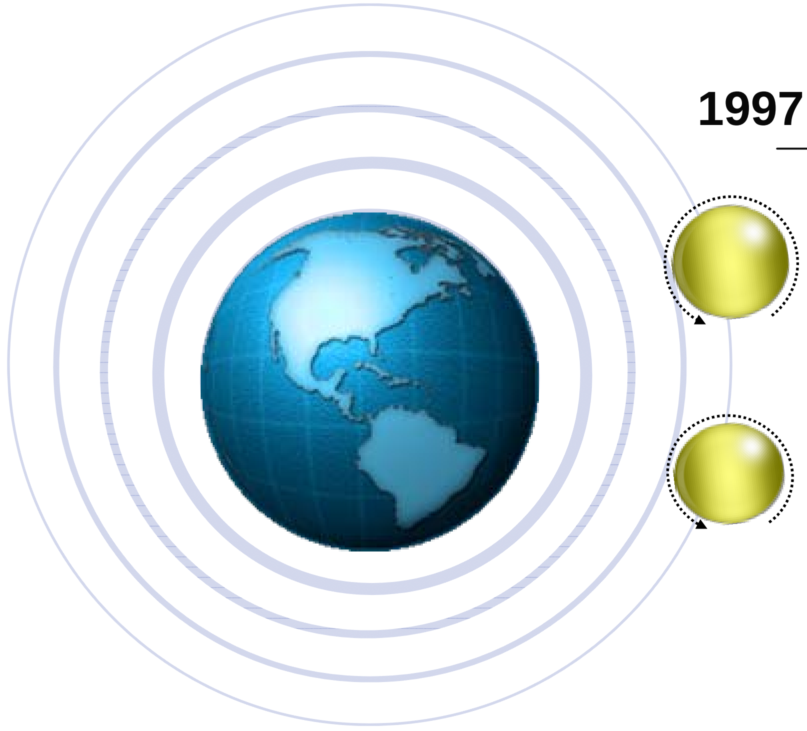
Background(1)


Software Evolution
Software Change Management
Software Quality Assurance
CIA
Change Analysis
Change Propagation
Change Ripples Traceability

Background(2)



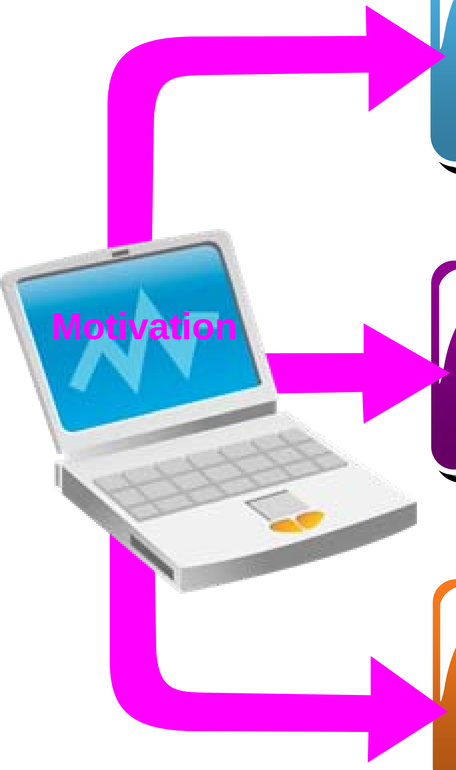
Scope



1997 → 2010

- Dependency Based Impact Analysis
- Traceability Based Impact Analysis

Motivation



CIA Technologies

1) identify key properties of CIA technique,
2) facilitate comparison of CIA techniques,
3) enable development of new CIA techniques,

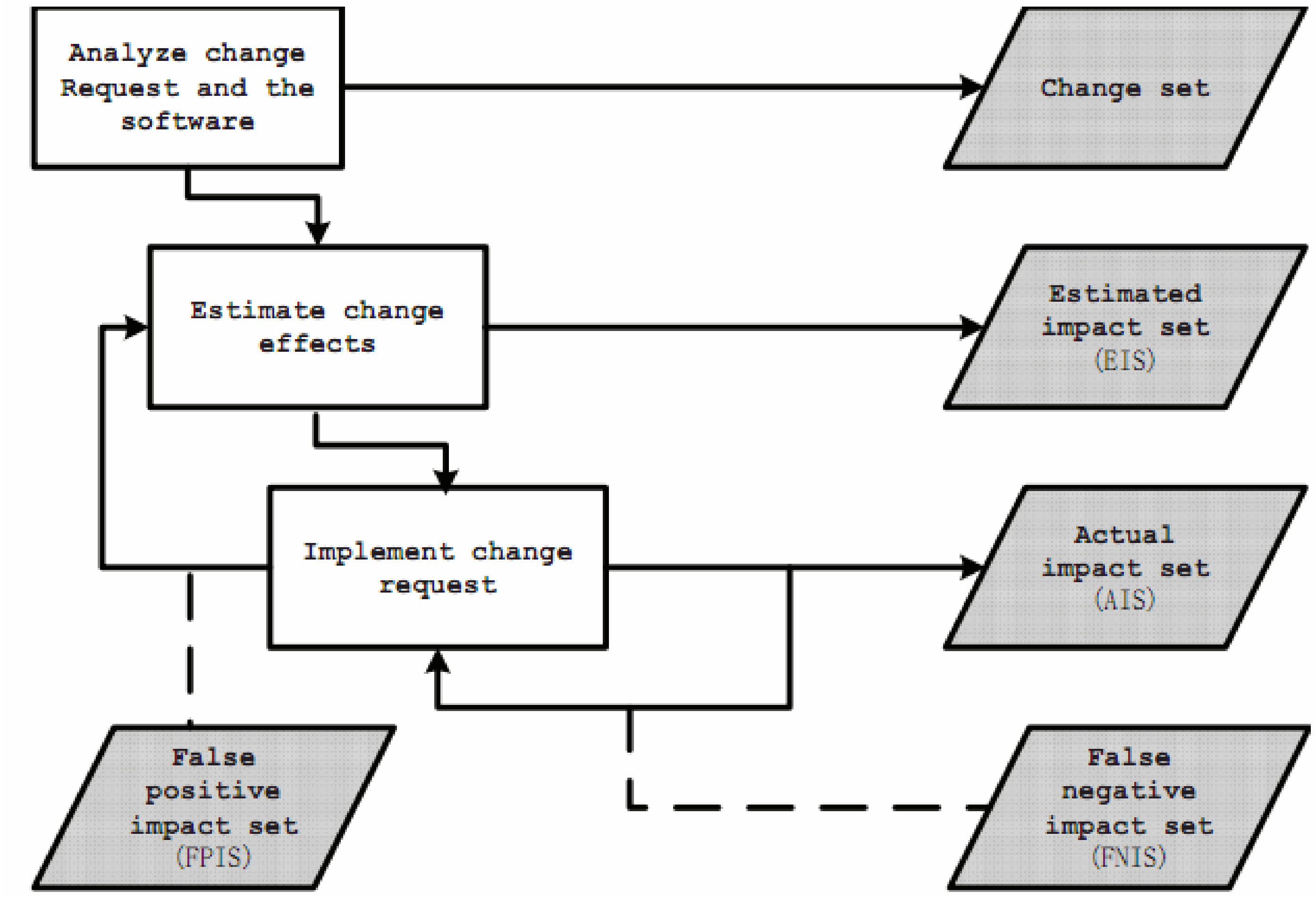
CIA Supporting Tools

select available CIA tool according to practical needs

CIA Applications

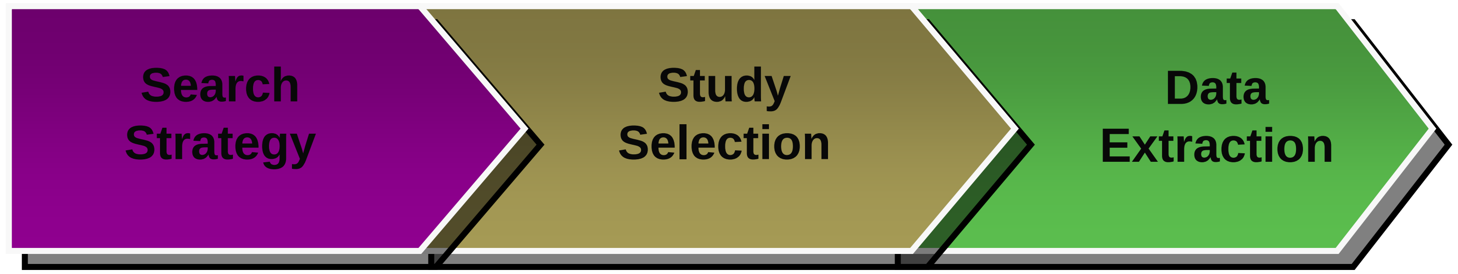
support maintainers to make decision among various change solutions, prepare change schedule, estimate resources and costs, and trace the change effect.

Preliminaries


Analyze change Request and the software
Change set
Estimate change effects
Estimated impact set (EIS)
Implement change request
Actual impact set (AIS)
False positive impact set (FPIS)
False negative impact set (FNIS)

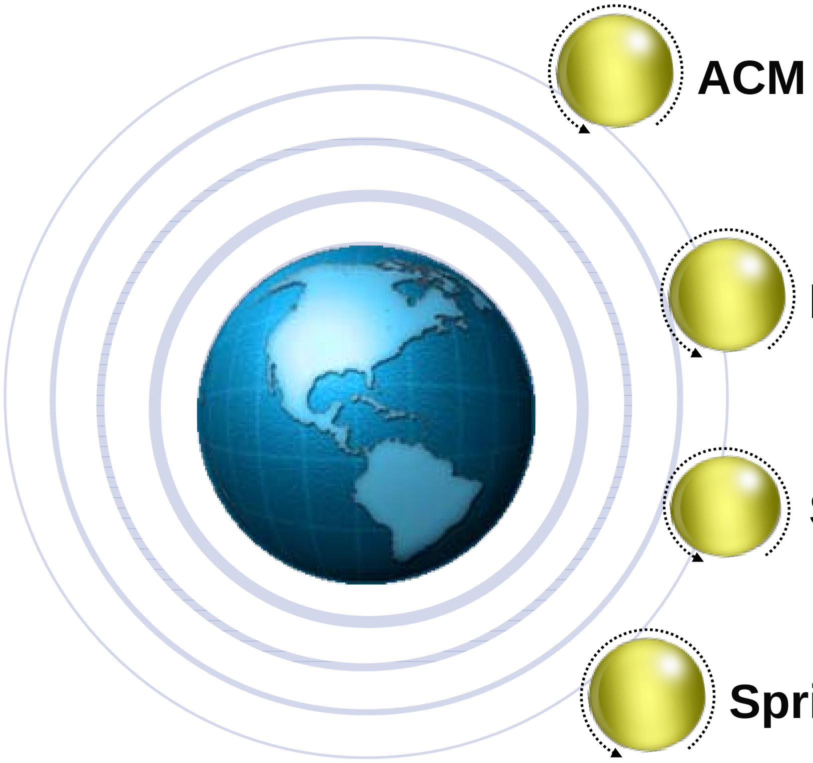
Survey Method

A Systematic Review Approach



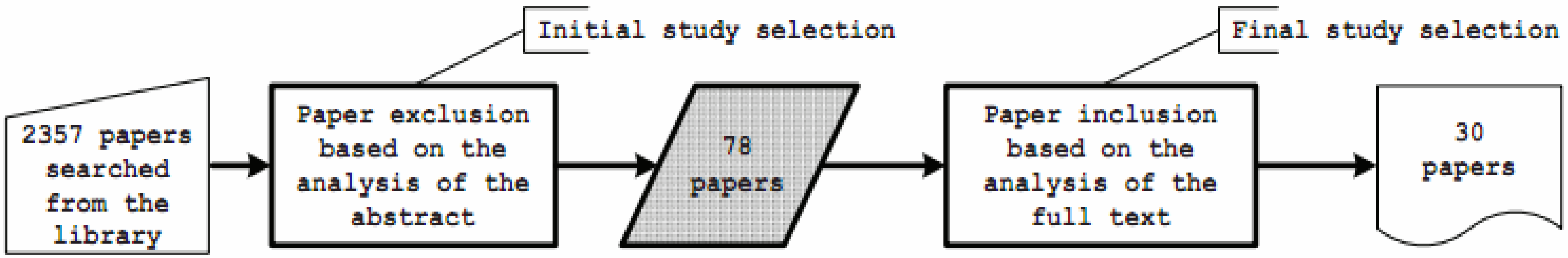
Survey detail is available online:
http://ise.seu.edu.cn/people/XiaobingSun/survey.xls

Search Strategy

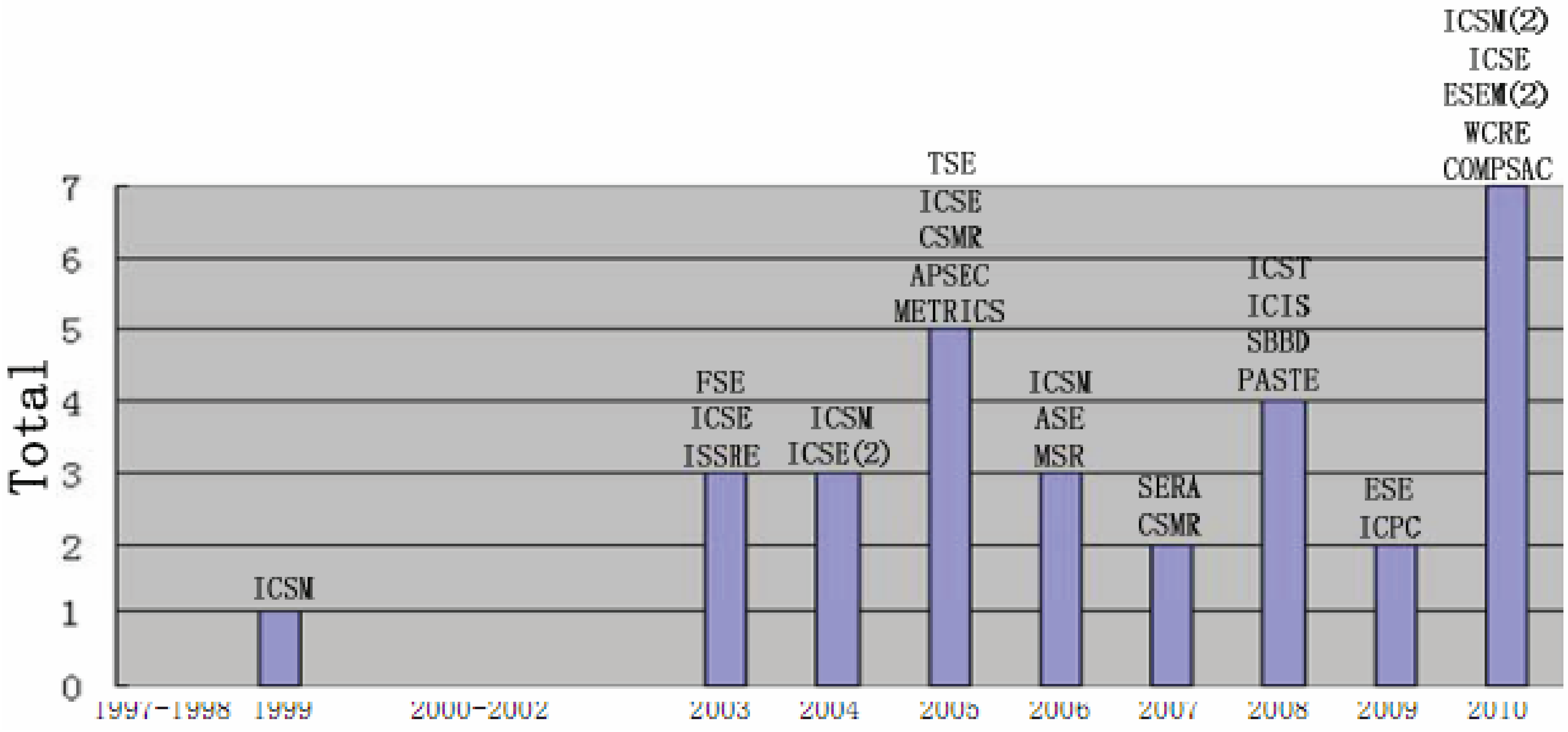


- ACM Digital Library
- IEEE Computer Society Digital Library
- Science@Direct
- Springer Link

Study Selection


Initial study selection
Final study selection
2357 papers searched from the library
Paper exclusion based on the analysis of the abstract
78 papers
Paper inclusion based on the analysis of the full text
30 papers

Data Extraction(1)

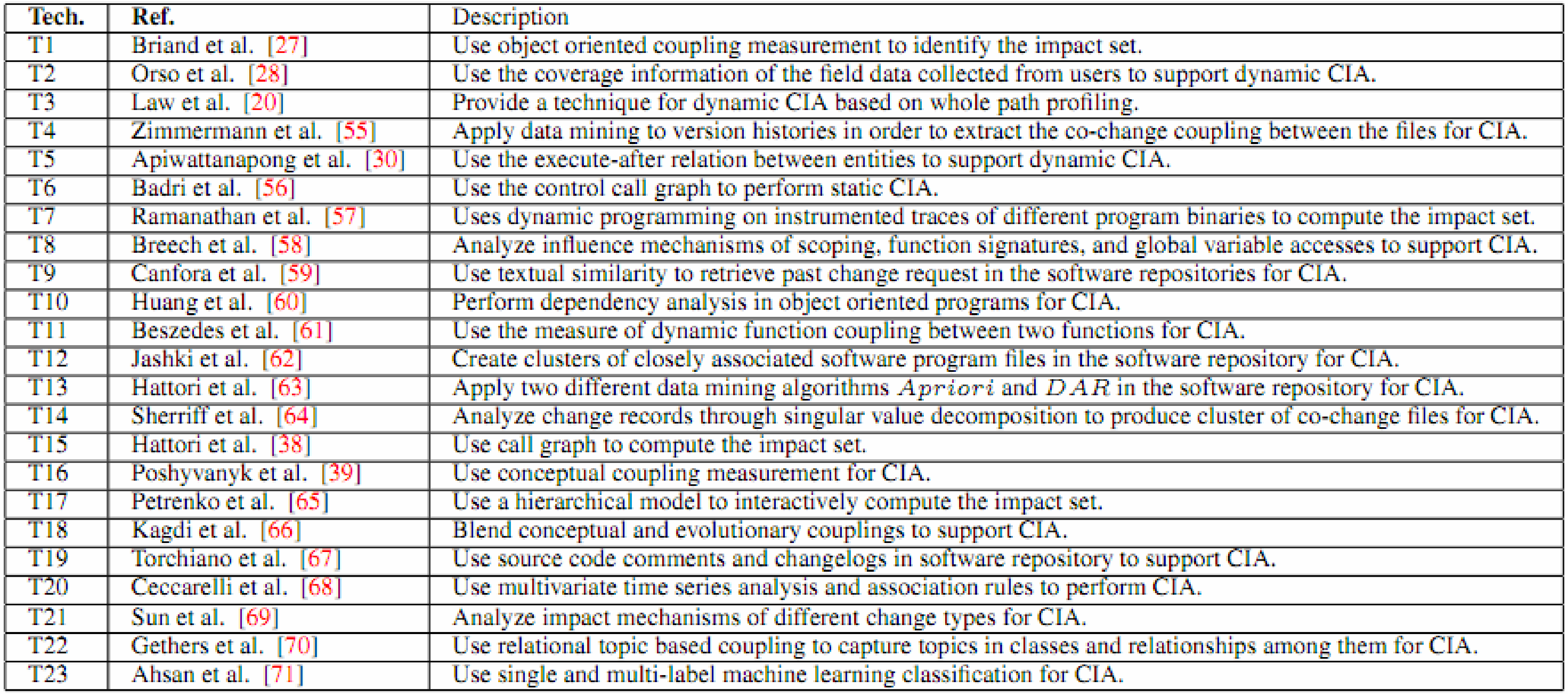


Data Extraction(2)

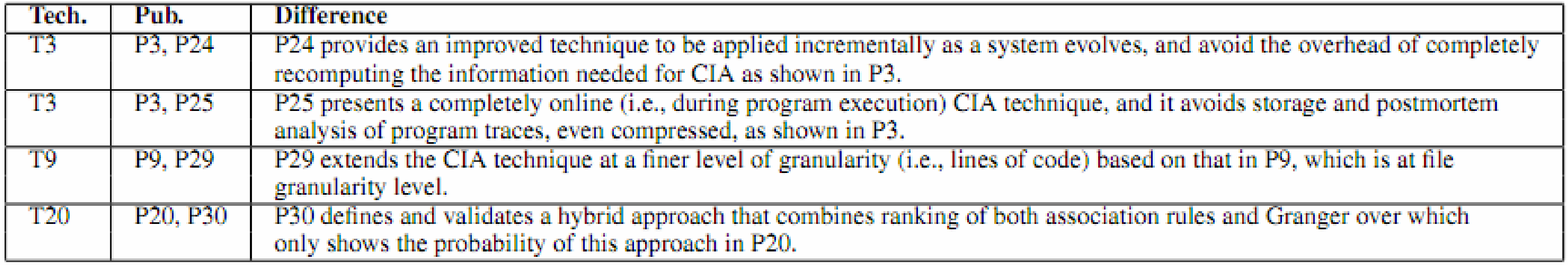
30 Publications

✓23 papers present 23 different CIA techniques with empirical studies
✓two papers cover the same CIA technique
✓four papers extend previous CIA techniques
✓two papers provide comparison of different CIA techniques

Data Extraction(3)

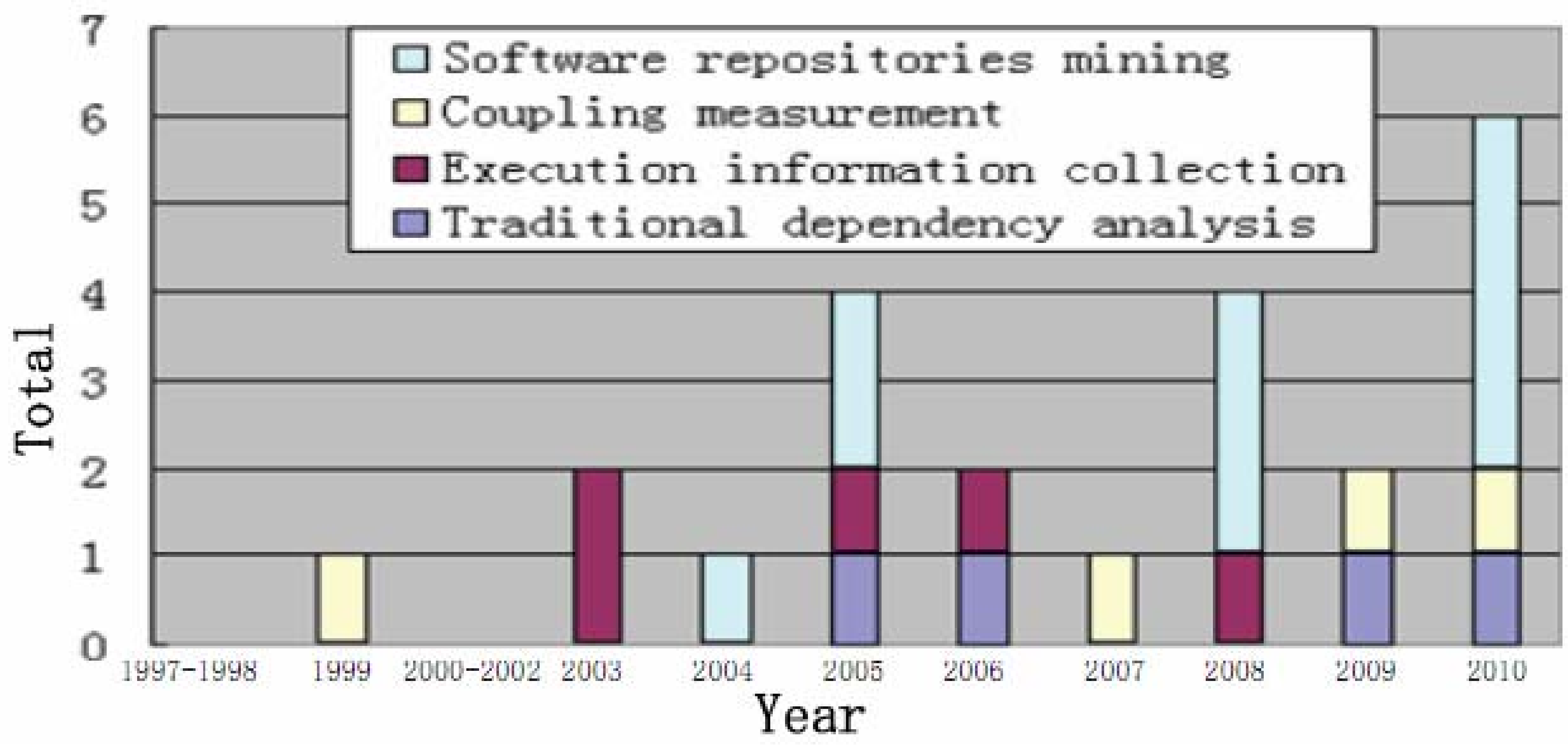


Tech.	Ref.	Description
T1	Briand et al. [27]	Use object oriented coupling measurement to identify the impact set.
T2	Orso et al. [28]	Use the coverage information of the field data collected from users to support dynamic CIA.
T3	Law et al. [20]	Provide a technique for dynamic CIA based on whole path profiling.
T4	Zimmermann et al. [55]	Apply data mining to version histories in order to extract the co-change coupling between the files for CIA.
T5	Apiwattanapong et al. [30]	Use the execute-after relation between entities to support dynamic CIA.
T6	Badri et al. [56]	Use the control call graph to perform static CIA.
T7	Ramanathan et al. [57]	Uses dynamic programming on instrumented traces of different program binaries to compute the impact set.
T8	Breech et al. [58]	Analyze influence mechanisms of scoping, function signatures, and global variable accesses to support CIA.
T9	Canfora et al. [59]	Use textual similarity to retrieve past change request in the software repositories for CIA.
T10	Huang et al. [60]	Perform dependency analysis in object oriented programs for CIA.
T11	Beszedes et al. [61]	Use the measure of dynamic function coupling between two functions for CIA.
T12	Jashki et al. [62]	Create clusters of closely associated software program files in the software repository for CIA.
T13	Hattori et al. [63]	Apply two different data mining algorithms $Apriori$ and DAR in the software repository for CIA.
T14	Sherriff et al. [64]	Analyze change records through singular value decomposition to produce cluster of co-change files for CIA.
T15	Hattori et al. [38]	Use call graph to compute the impact set.
T16	Poshyvanyk et al. [39]	Use conceptual coupling measurement for CIA.
T17	Petrenko et al. [65]	Use a hierarchical model to interactively compute the impact set.
T18	Kagdi et al. [66]	Blend conceptual and evolutionary couplings to support CIA.
T19	Torchiano et al. [67]	Use source code comments and changelogs in software repository to support CIA.
T20	Ceccarelli et al. [68]	Use multivariate time series analysis and association rules to perform CIA.
T21	Sun et al. [69]	Analyze impact mechanisms of different change types for CIA.
T22	Gethers et al. [70]	Use relational topic based coupling to capture topics in classes and relationships among them for CIA.
T23	Ahsan et al. [71]	Use single and multi-label machine learning classification for CIA.

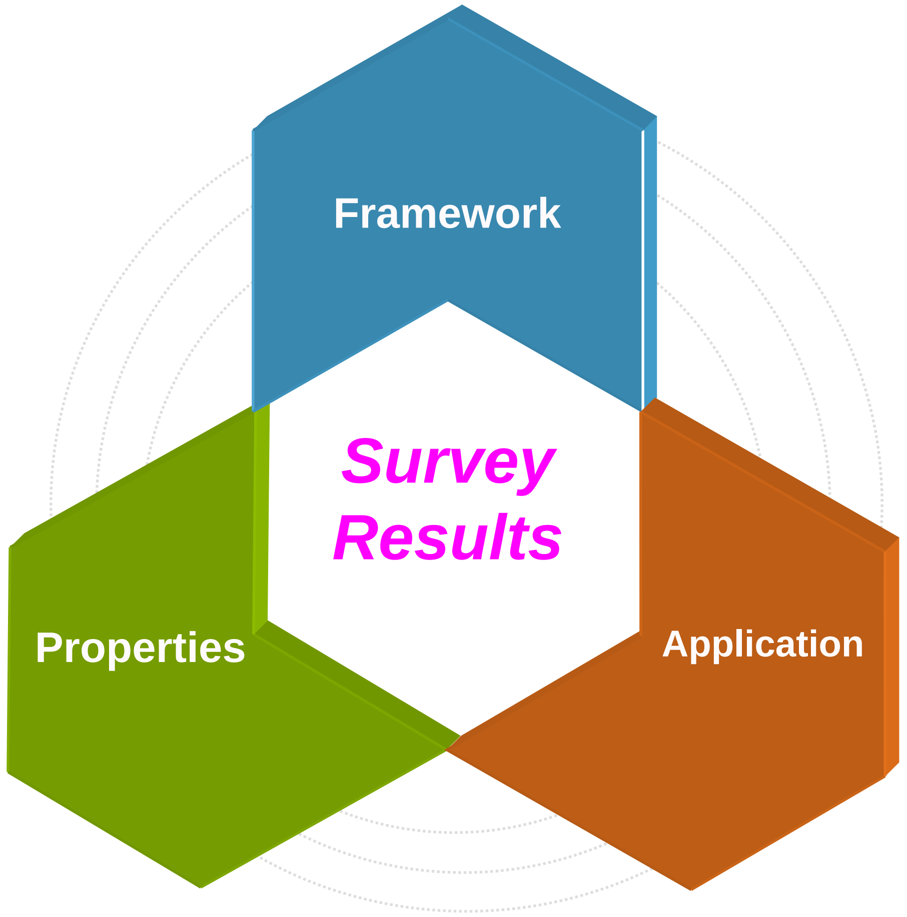


Tech.	Pub.	Difference
T3	P3, P24	P24 provides an improved technique to be applied incrementally as a system evolves, and avoid the overhead of completely recomputing the information needed for CIA as shown in P3.
T3	P3, P25	P25 presents a completely online (i.e., during program execution) CIA technique, and it avoids storage and postmortem analysis of program traces, even compressed, as shown in P3.
T9	P9, P29	P29 extends the CIA technique at a finer level of granularity (i.e., lines of code) based on that in P9, which is at file granularity level.
T20	P20, P30	P30 defines and validates a hybrid approach that combines ranking of both association rules and Granger over which only shows the probability of this approach in P20.

Data Extraction(4)



Survey Results



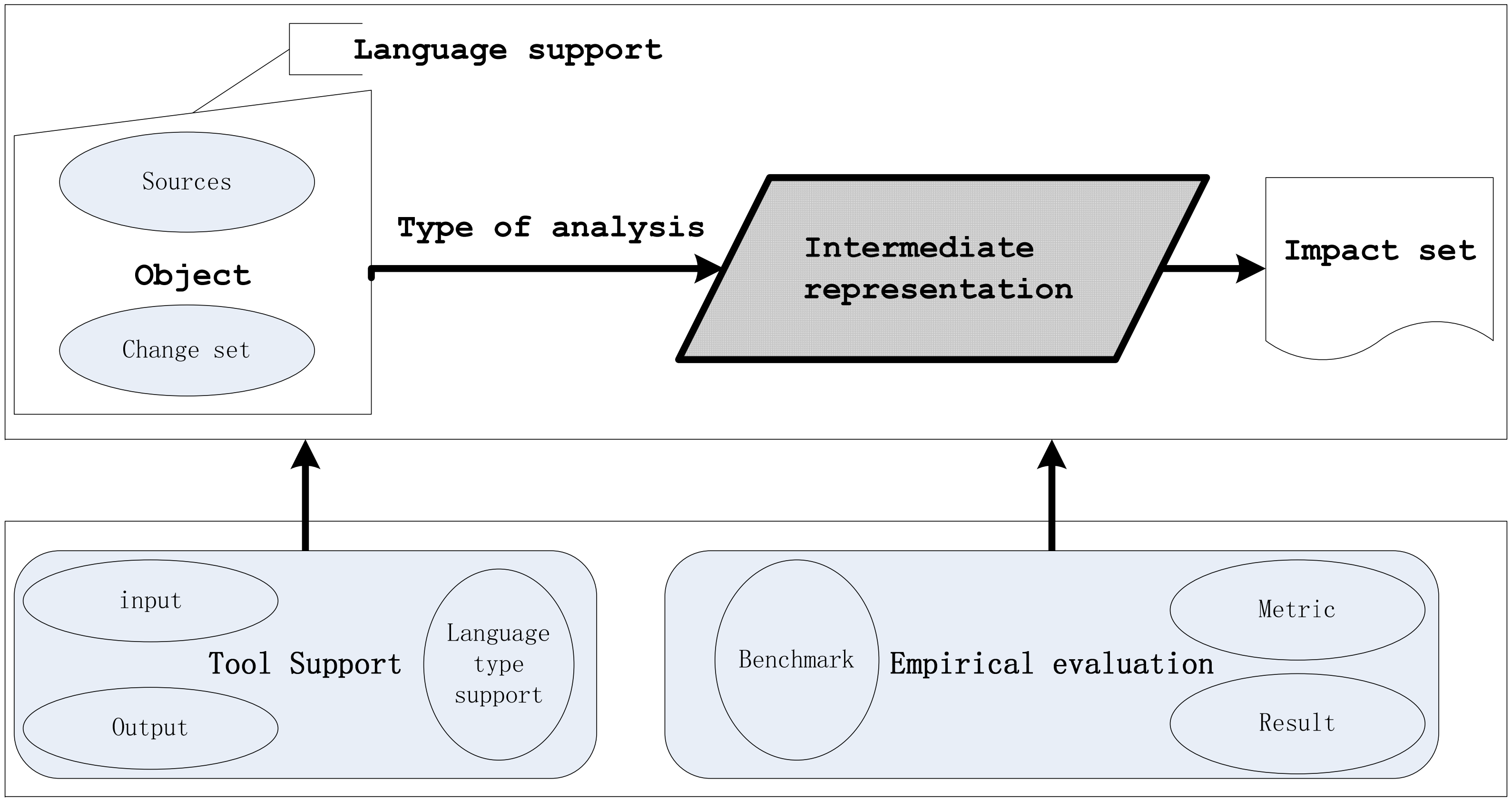
A Framework

- To characterize the CIA techniques.

- To support the identification and comparison of existing CIA techniques based on the specific needs of the user.

- To provide guidelines to support development of new CIA techniques.

A Framework


Language support
Sources
Object
Change set
Type of analysis
Intermediate representation
Impact set
input
Tool Support
Language type support
Output
Benchmark
Empirical evaluation
Metric
Result

Properties in the Framework

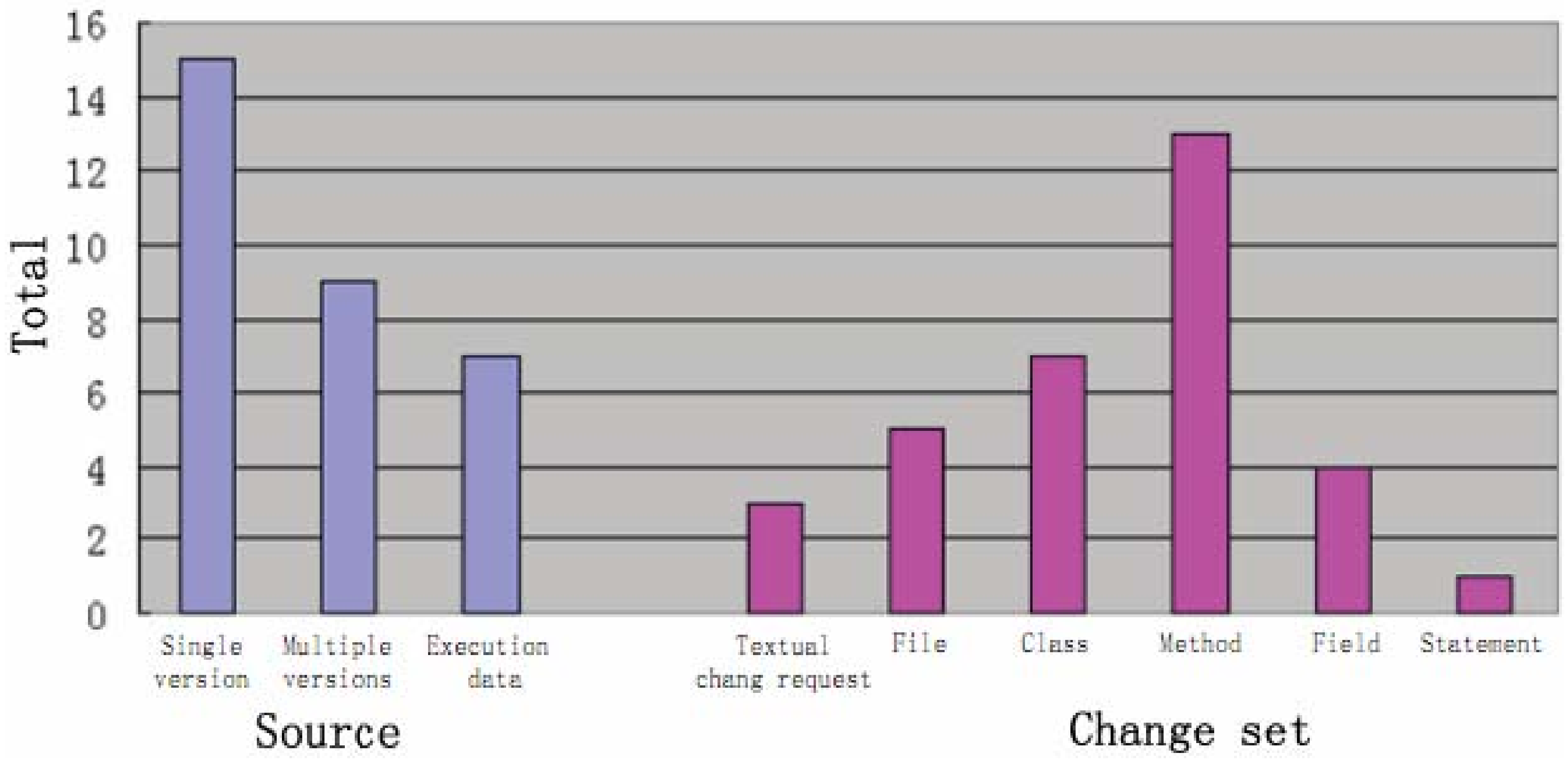
- **Object:** change set and the source (users' input)
- **Impact set:** output of the CIA (users' application)
- **Intermediate representation:** dependences between program elements (CIA's key)
- **Type of analysis:** static & dynamic (resource and user involvement)
- **Language support:** procedure-oriented programs, object-oriented programs and aspect-oriented program (application environment)
- **Tool support:** automation (availability)
- **Empirical evaluation:** assessment (comparison) of the CIA technique(s) (evidence)

Applications of the Framework

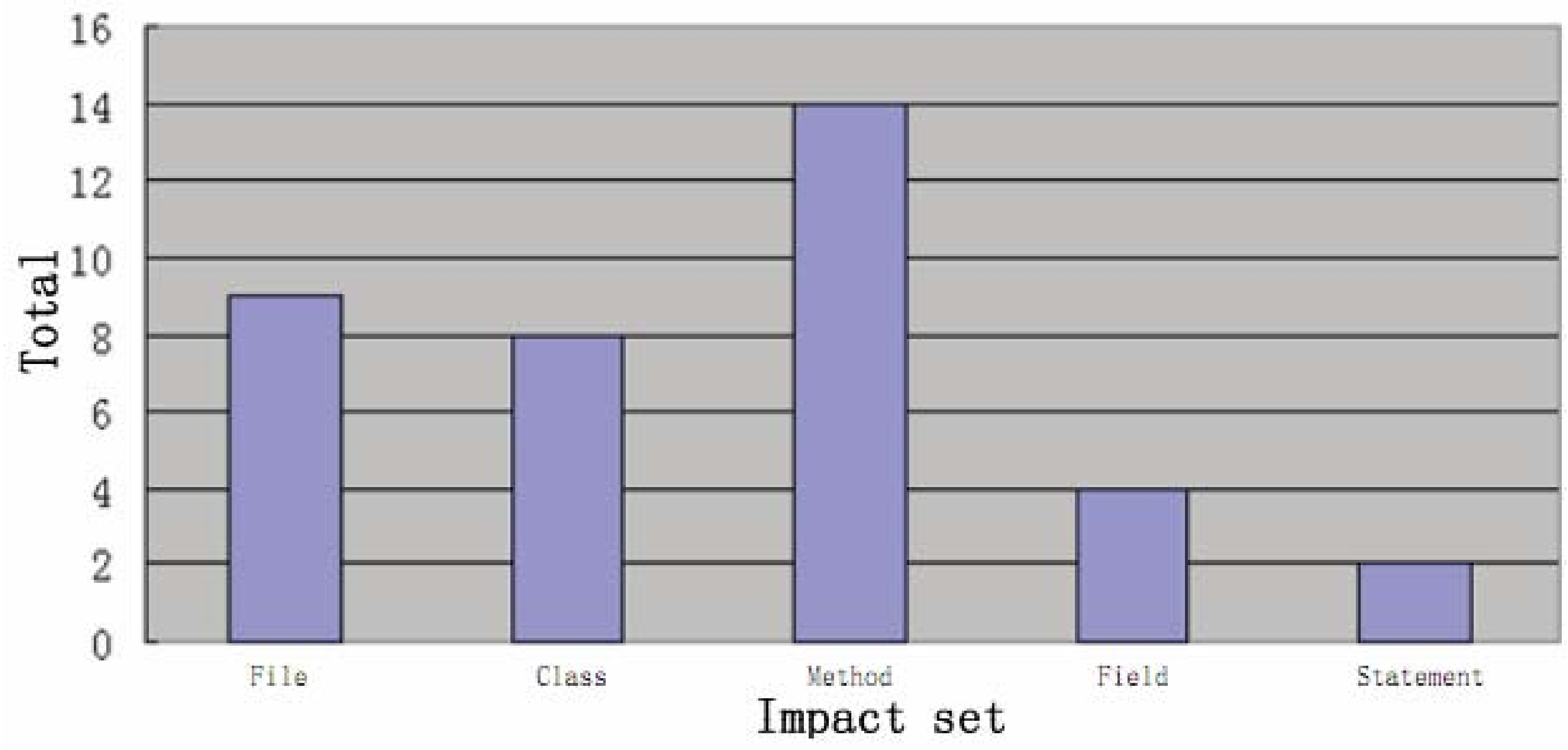
- Expressiveness: its ability to cover a wide spectrum of the CIA techniques.
- Effectiveness: the ease and comprehensiveness of comparison of the CIA techniques.

Using the proposed framework, the CIA technique that fits practical demands for a specific situation can be easily selected.

Object

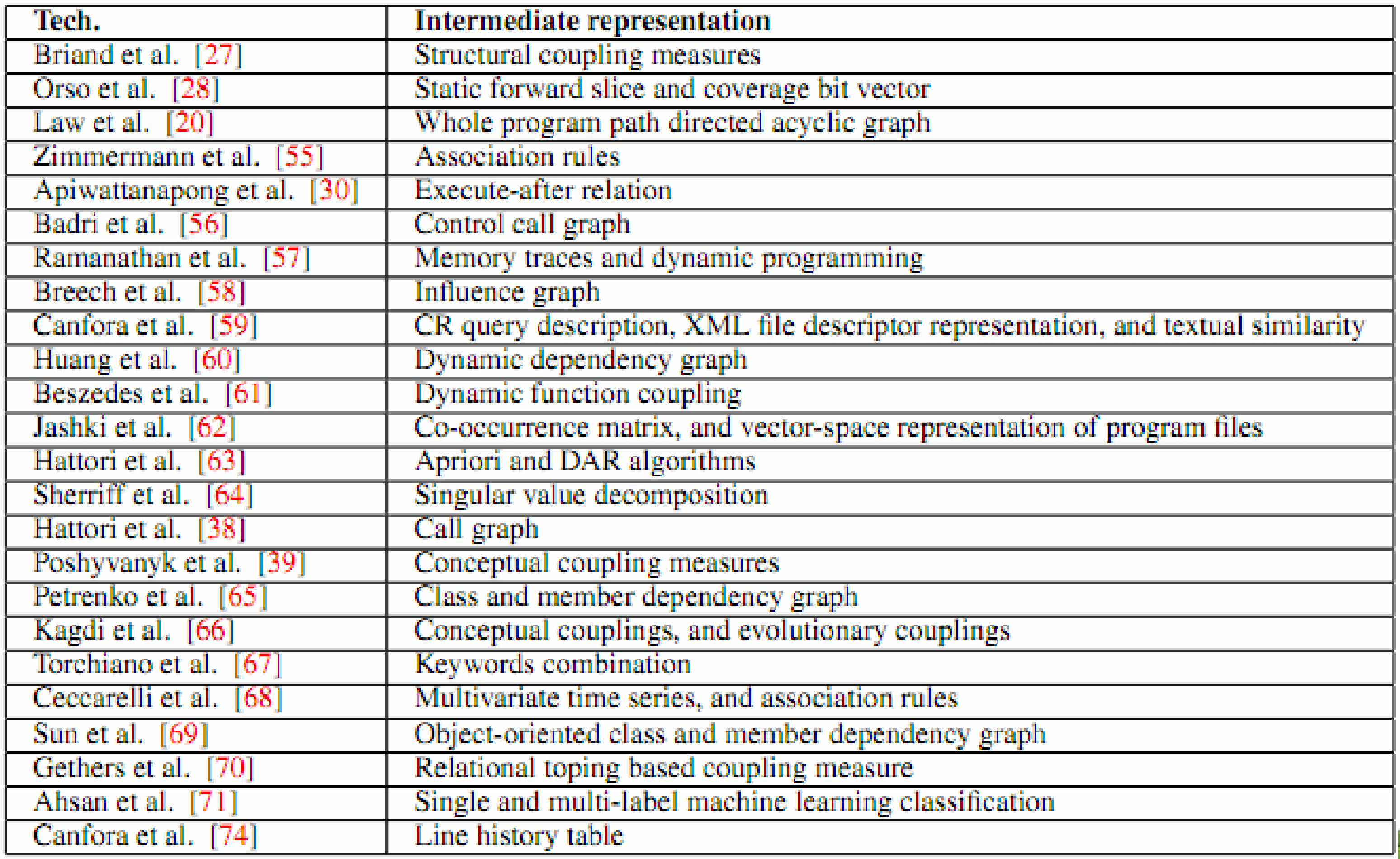

Total
0
2
4
6
8
10
12
14
16
Single version
Multiple versions
Execution data
Textual chang request
File
Class
Method
Field
Statement
Source
Change set

Impact Set



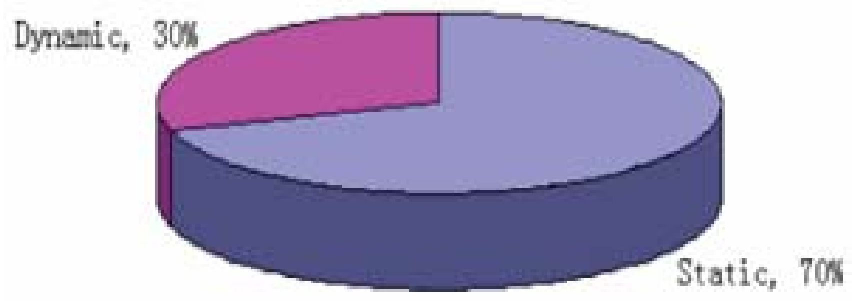
Total
16
14
12
10
8
6
4
2
0
File
Class
Method
Field
Statement
Impact set

Intermediate Representation

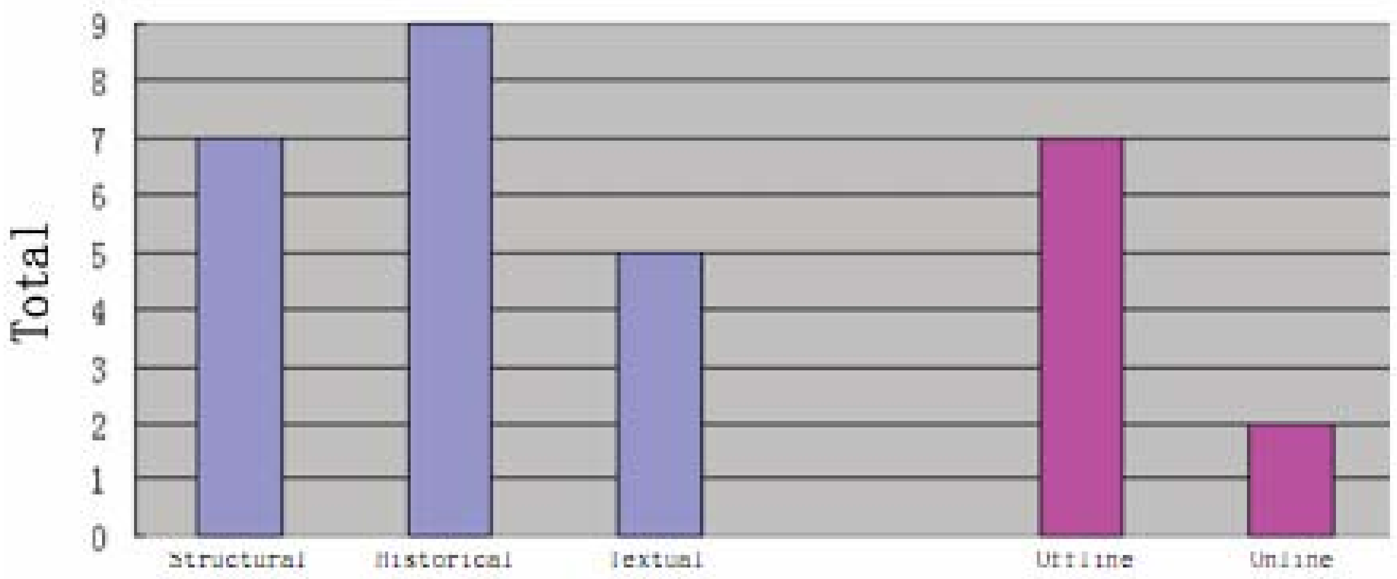


Tech.	Intermediate representation
Briand et al. [27]	Structural coupling measures
Orso et al. [28]	Static forward slice and coverage bit vector
Law et al. [20]	Whole program path directed acyclic graph
Zimmermann et al. [55]	Association rules
Apiwattanapong et al. [30]	Execute-after relation
Badri et al. [56]	Control call graph
Ramanathan et al. [57]	Memory traces and dynamic programming
Breech et al. [58]	Influence graph
Canfora et al. [59]	CR query description, XML file descriptor representation, and textual similarity
Huang et al. [60]	Dynamic dependency graph
Beszedes et al. [61]	Dynamic function coupling
Jashki et al. [62]	Co-occurrence matrix, and vector-space representation of program files
Hattori et al. [63]	Apriori and DAR algorithms
Sherriff et al. [64]	Singular value decomposition
Hattori et al. [38]	Call graph
Poshyvanyk et al. [39]	Conceptual coupling measures
Petrenko et al. [65]	Class and member dependency graph
Kagdi et al. [66]	Conceptual couplings, and evolutionary couplings
Torchiano et al. [67]	Keywords combination
Ceccarelli et al. [68]	Multivariate time series, and association rules
Sun et al. [69]	Object-oriented class and member dependency graph
Gethers et al. [70]	Relational toping based coupling measure
Ahsan et al. [71]	Single and multi-label machine learning classification
Canfora et al. [74]	Line history table

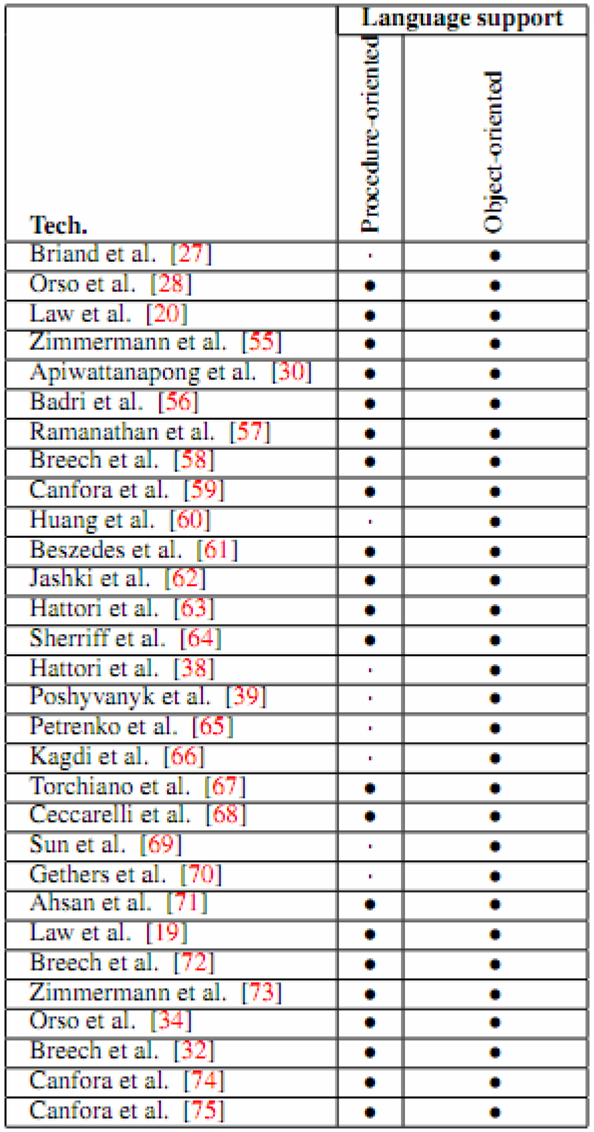
Type of Analysis



Type of Analysis

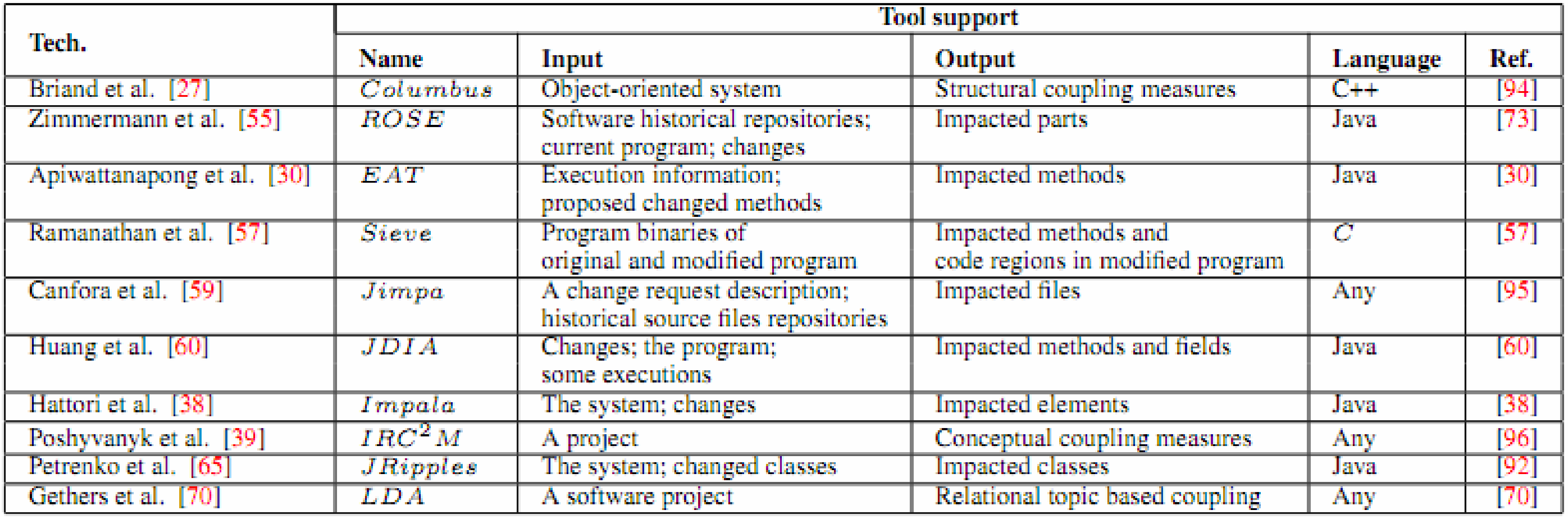


Language Support



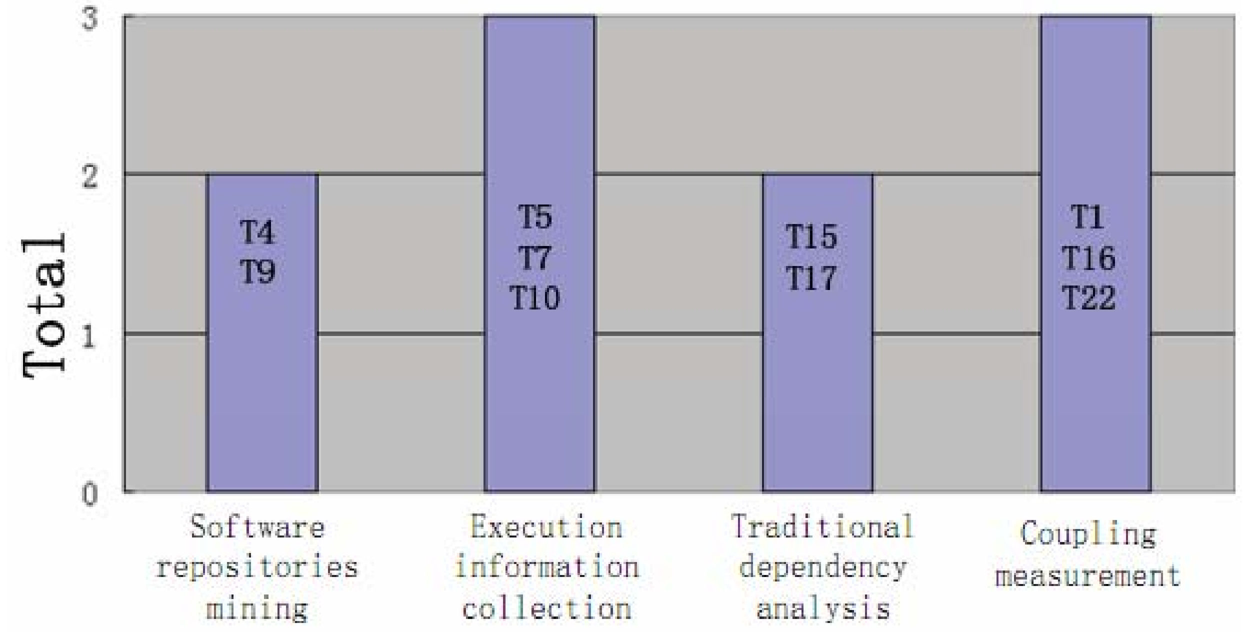
Tech.	Language support: Procedure-oriented	Language support: Object-oriented
Briand et al. [27]	·	●
Orso et al. [28]	●	●
Law et al. [20]	●	●
Zimmermann et al. [55]	●	●
Apiwattanapong et al. [30]	●	●
Badri et al. [56]	●	●
Ramanathan et al. [57]	●	●
Breech et al. [58]	●	●
Canfora et al. [59]	●	●
Huang et al. [60]	·	●
Beszedes et al. [61]	●	●
Jashki et al. [62]	●	●
Hattori et al. [63]	●	●
Sherriff et al. [64]	●	●
Hattori et al. [38]	·	●
Poshyvanyk et al. [39]	·	●
Petrenko et al. [65]	·	●
Kagdi et al. [66]	·	●
Torchiano et al. [67]	●	●
Ceccarelli et al. [68]	●	●
Sun et al. [69]	·	●
Gethers et al. [70]	·	●
Ahsan et al. [71]	●	●
Law et al. [19]	●	●
Breech et al. [72]	●	●
Zimmermann et al. [73]	●	●
Orso et al. [34]	●	●
Breech et al. [32]	●	●
Canfora et al. [74]	●	●
Canfora et al. [75]	●	●

Tool Support(1)

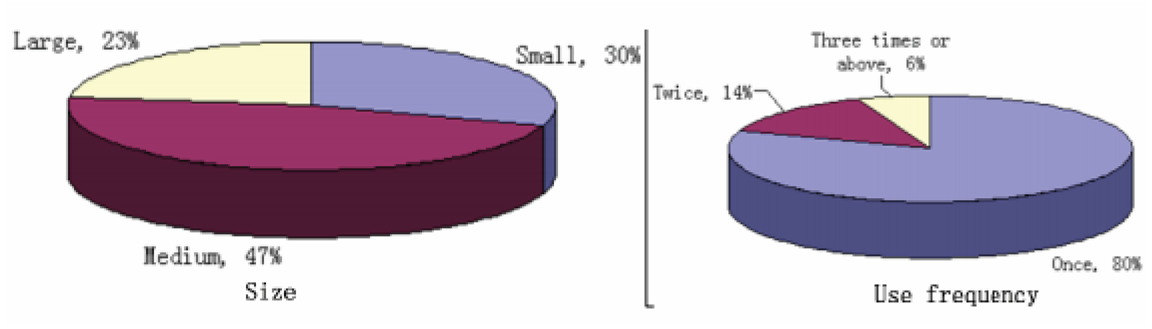


Tech.	Tool support				
	Name	Input	Output	Language	Ref.
Briand et al. [27]	*Columbus*	Object-oriented system	Structural coupling measures	C++	[94]
Zimmermann et al. [55]	*ROSE*	Software historical repositories; current program; changes	Impacted parts	Java	[73]
Apiwattanapong et al. [30]	*EAT*	Execution information; proposed changed methods	Impacted methods	Java	[30]
Ramanathan et al. [57]	*Sieve*	Program binaries of original and modified program	Impacted methods and code regions in modified program	*C*	[57]
Canfora et al. [59]	*Jimpa*	A change request description; historical source files repositories	Impacted files	Any	[95]
Huang et al. [60]	*JDIA*	Changes; the program; some executions	Impacted methods and fields	Java	[60]
Hattori et al. [38]	*Impala*	The system; changes	Impacted elements	Java	[38]
Poshyvanyk et al. [39]	IRC^2M	A project	Conceptual coupling measures	Any	[96]
Petrenko et al. [65]	*JRipples*	The system; changed classes	Impacted classes	Java	[92]
Gethers et al. [70]	*LDA*	A software project	Relational topic based coupling	Any	[70]

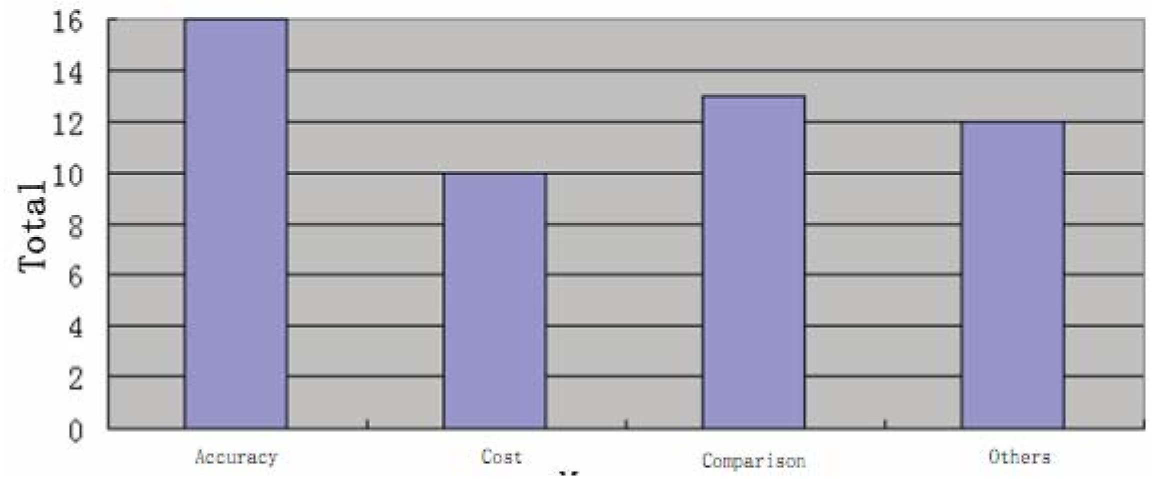
Tool Support(2)



Empirical Evaluation(1)

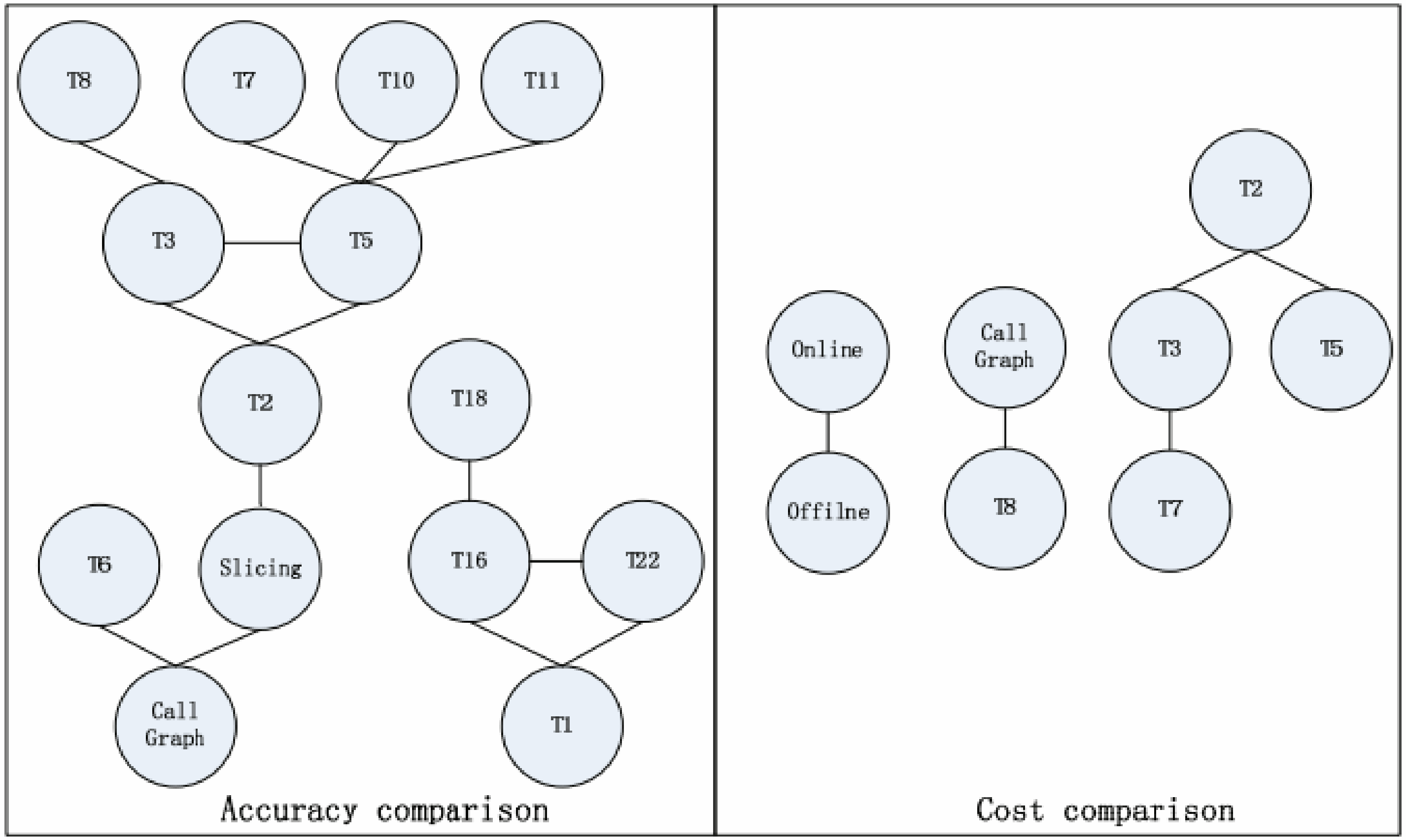


Benchmarks

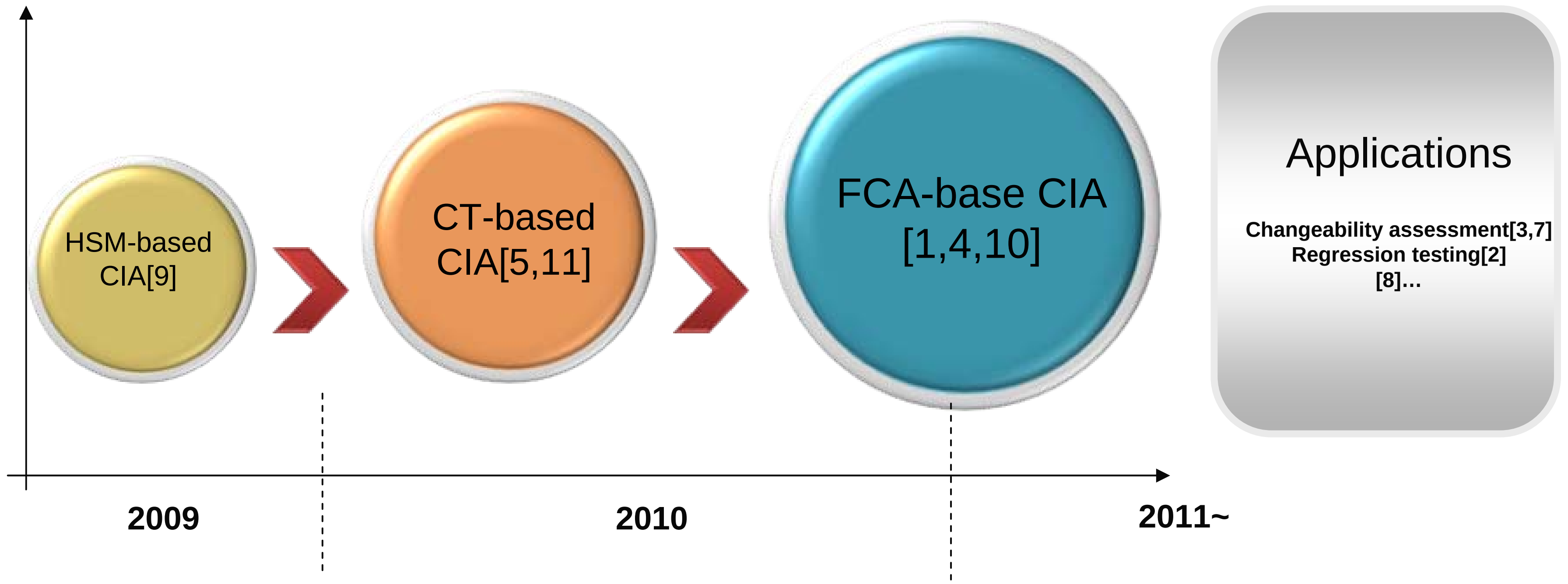


Measures

Empirical Evaluation(2)



Our Related Work



Our Related Work

1. Xiaobing Sun, Bixin Li, Sai Zhang. *FCA-based Change Impact Analysis for Object Oriented Program*. (under review)
2. Xiaobing Sun, Bixin Li, Sai *Zhang. A Novel Approach for Regression Testing Using FCA-based Change Impact Analysis.* (under review)
3. Xiaobing Sun, Bixin Li, *Sai Zhang. A Change Proposal Driven Approach for Changeability Assessment Using FormalConcept Analysis.* (under review)
4. Bixin Li, Xiaobing Sun, Hareton Leung. *Applying Formal Concept Analysis to Evaluating Impacts of Software Changes*. Submittd to Journal of System and Software (JSS) (under review).
5. Xiaobing Sun, Bixin Li, Chuanqi Tao, Wanzhi Wen, Sai Zhang. *Analyzing Impact Rules of Different Change Types to Support Change Impact Analysis.* International Journal of Software Engineering and Knowledge Engineering (IJSEKE)
6. Bixin Li, Xiaobing Sun, Hareton Leung *A Brief Survey and Comparative Classification of Vertical Change Impact Analysis Techniques.* Journal of Software Testing, Verification and Reliability (STVR)
7. 孙小兵,李必信,陶传奇. 基于LoCMD的软件修改分析技术. 软件学报,已录用(2011,6).
8. Xiaobing Sun, Bixin Li. *Using Formal Concept Analysis to Support Change Analysis.* In Proc. of 26th IEEE/ACM International Conference On Automated Software Engineering (ASE 2011), November 6-10, 2011, Lawrence, Kansas, USA.
9. Xiaobing Sun, Bixin Li, Chuanqi Tao, Sai Zhang. *HSM-based Change Impact Analysis of Object-Oriented Java Programs.* Chinese of Journal Electronics, Apr. 2011,20(2): 247-251. [SCI/EI]
10. Xiaobing Sun, Bixin Li, Sai Zhang and Chuanqi Tao. *Using Lattice of Class and Method Dependence for Change Impact Analysis of Object Oriented Programs.* In: Proc. of the 26th Symposium On Applied Computing (SAC 2011), Mar 21 - 24, 2011, TaiChung, Taiwan, ACM Computer Society Press [EI]
11. Xiaobing Sun, Bixin Li, Chuanqi Tao, Wanzhi Wen, Sai Zhang. *Change Impact Analysis Based on a Taxonomy of Change Types.* In Proc. of IEEE 23rd International Computer Software and Applications Conference (COMPSAC 2010), July 19-23, 2010, Seoul, South Korea. [EI]

Our Related Work

Framework of CIA

- Seven properties for CIA technique

CIA across different levels

- From Class-level change set to method-level impact set
- Consideration of the relationship between multiple changes
- A ranked list of impacted methods

Change proposal driven changeability assessment

- A metric to measure the changeability of a change proposal.

THANK YOU
.com
WELLS FARGO
THE WALK TO CURE DIABETES
JDRF 2005
We're on Track For a Cure

Q & A