Context-Based Code Smells Prioritization for Prefactoring

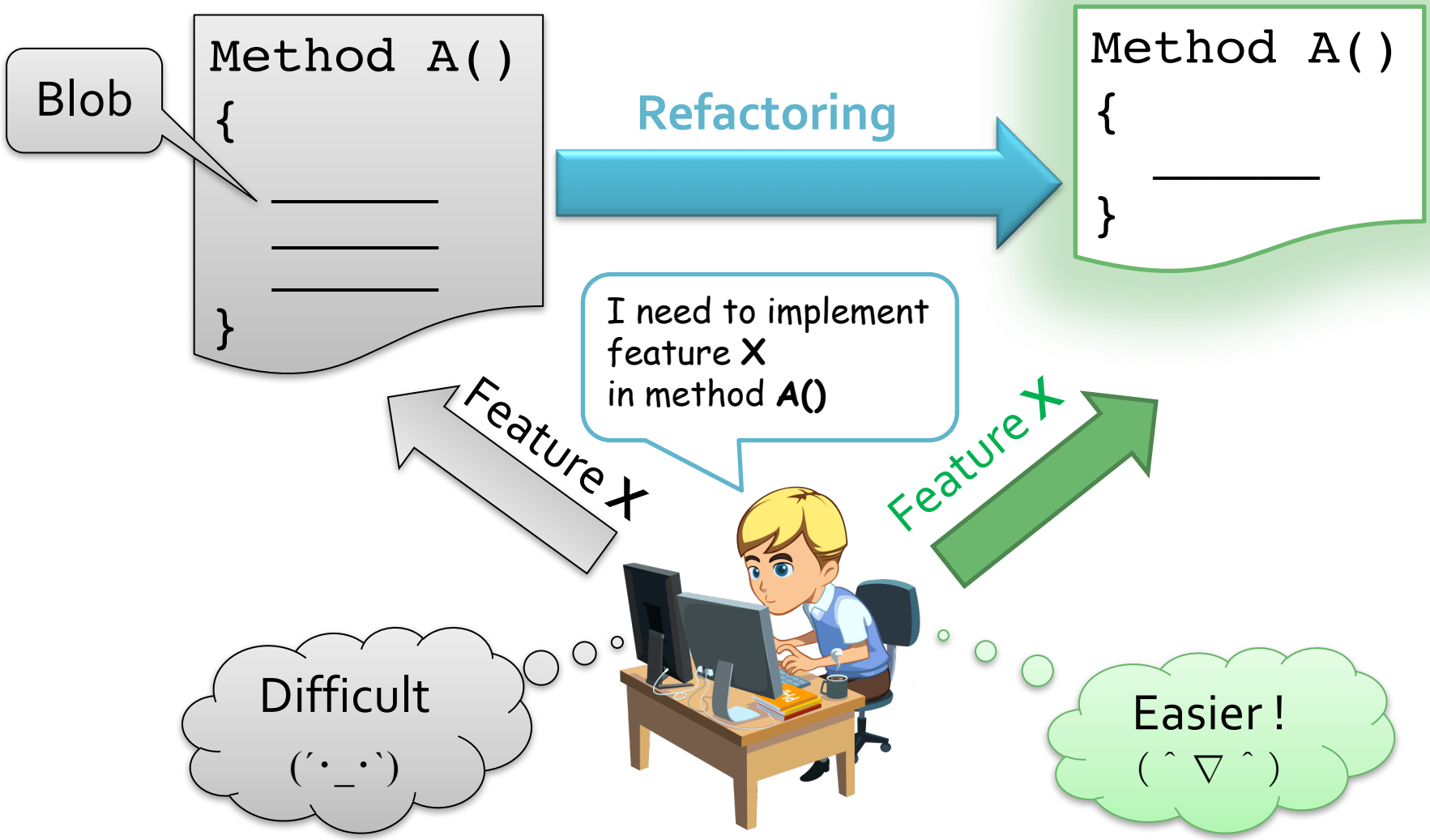
Natthawute Sae-Lim, Shinpei Hayashi, and Motoshi Saeki



Department of Computer Science
Tokyo Institute of Technology

INTRODUCTION

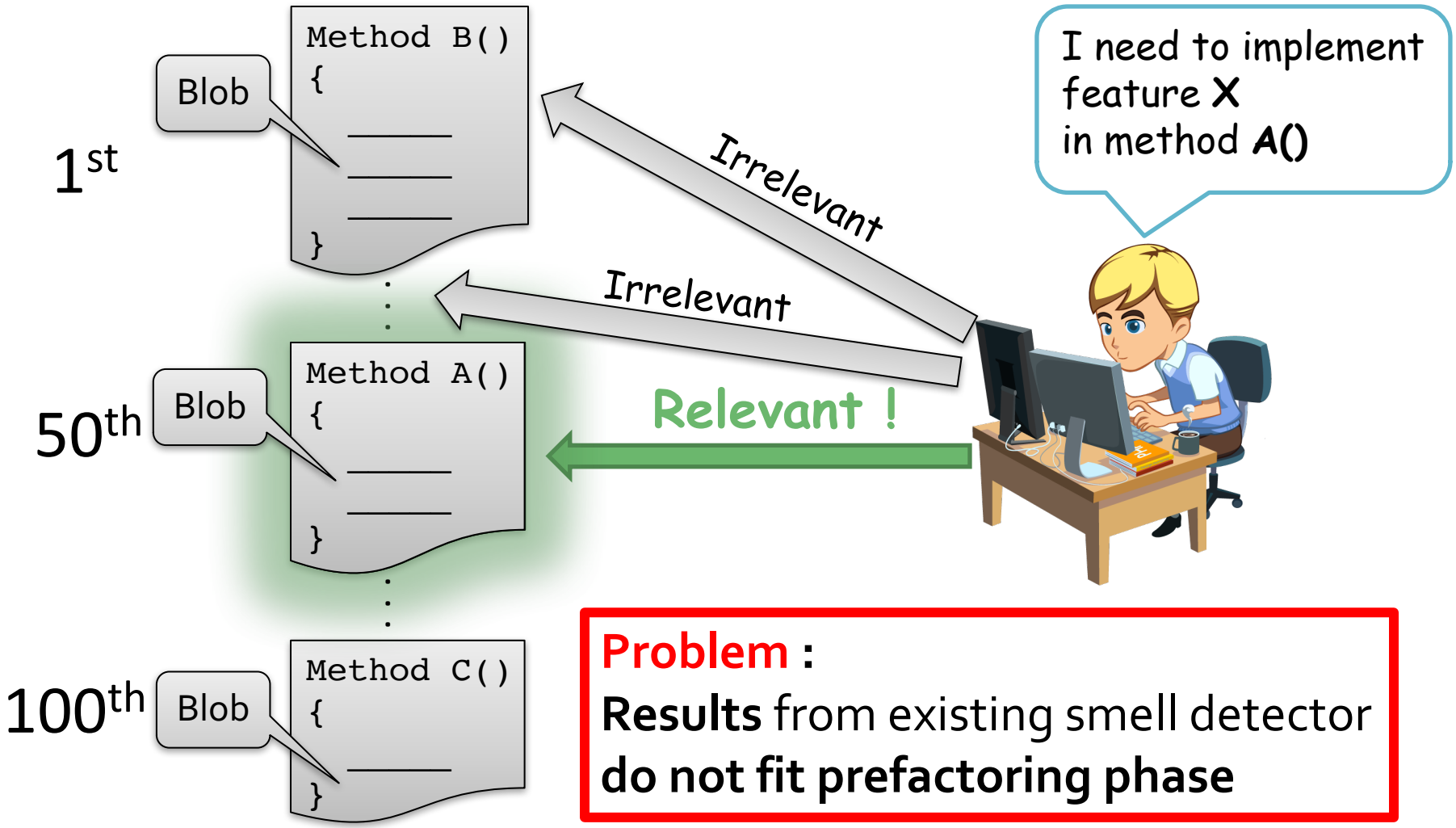
Prefactoring[1]



[1] V. Rajlich, Software Engineering: The Current Practice. Chapman and Hall/CRC, 2011

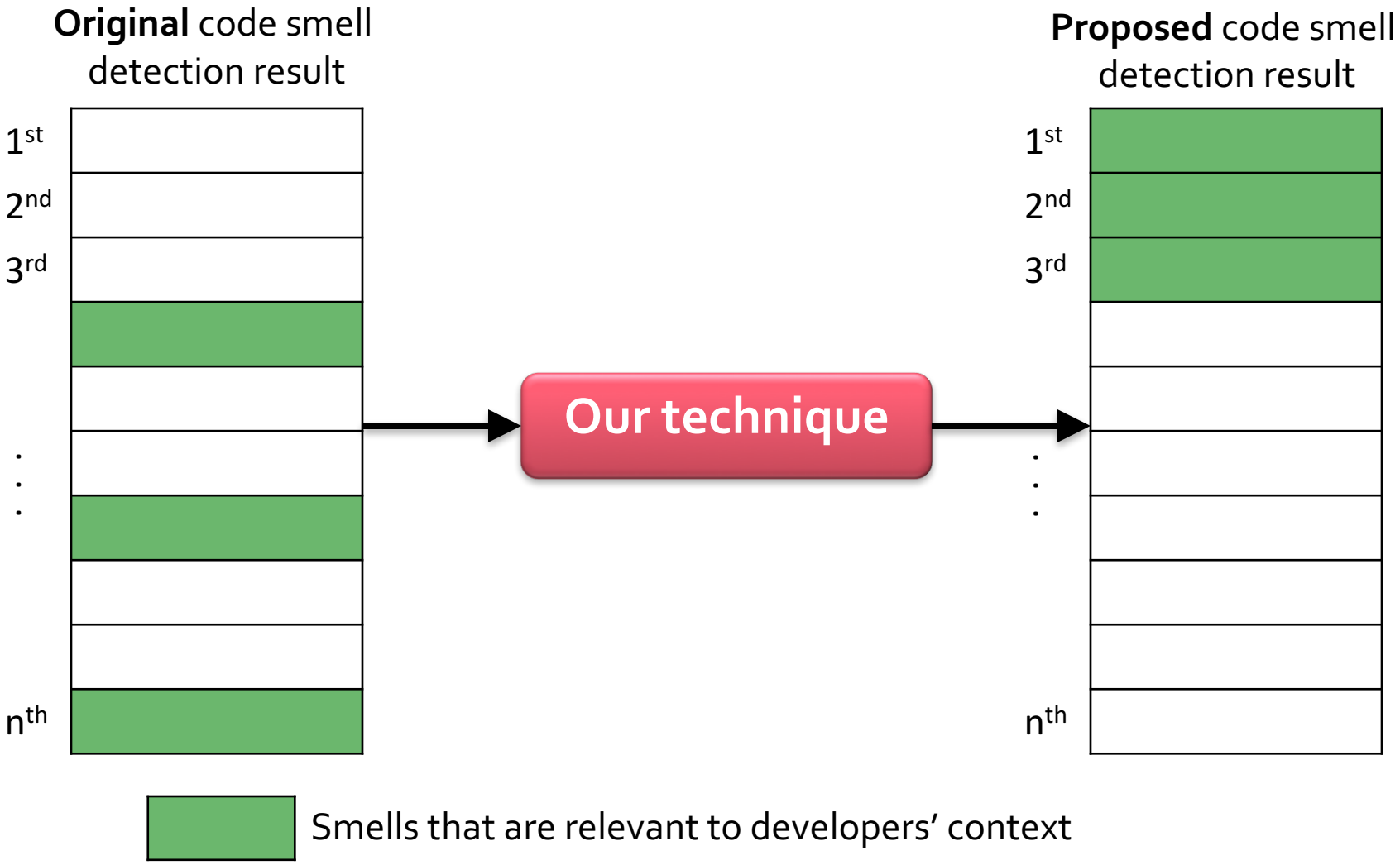
Problem

Code smell detection results



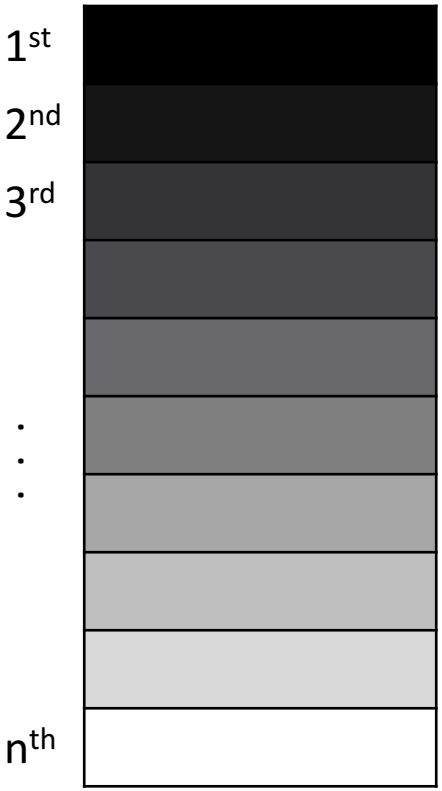
Problem :
Results from existing smell detector **do not fit prefactoring phase**

Goal


Original code smell detection result
1st
2nd
3rd
nth
Our technique
Proposed code smell detection result
1st
2nd
3rd
nth
Smells that are relevant to developers' context

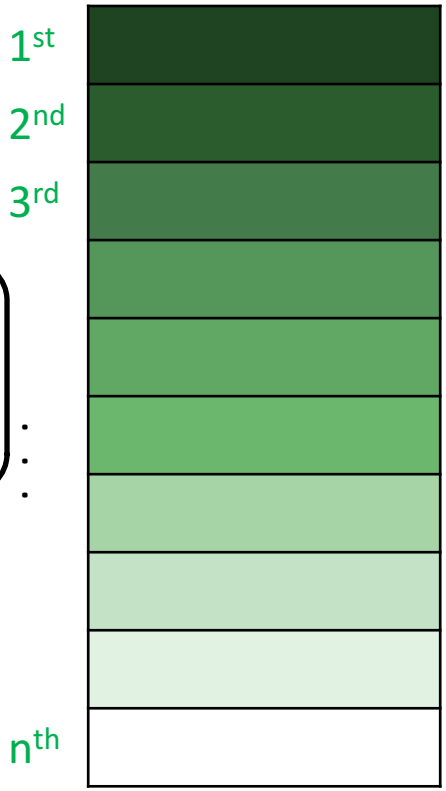
Key Idea

Severity-based prioritization[1][2]



I need to improve **overall quality** of the source code

Context-based prioritization



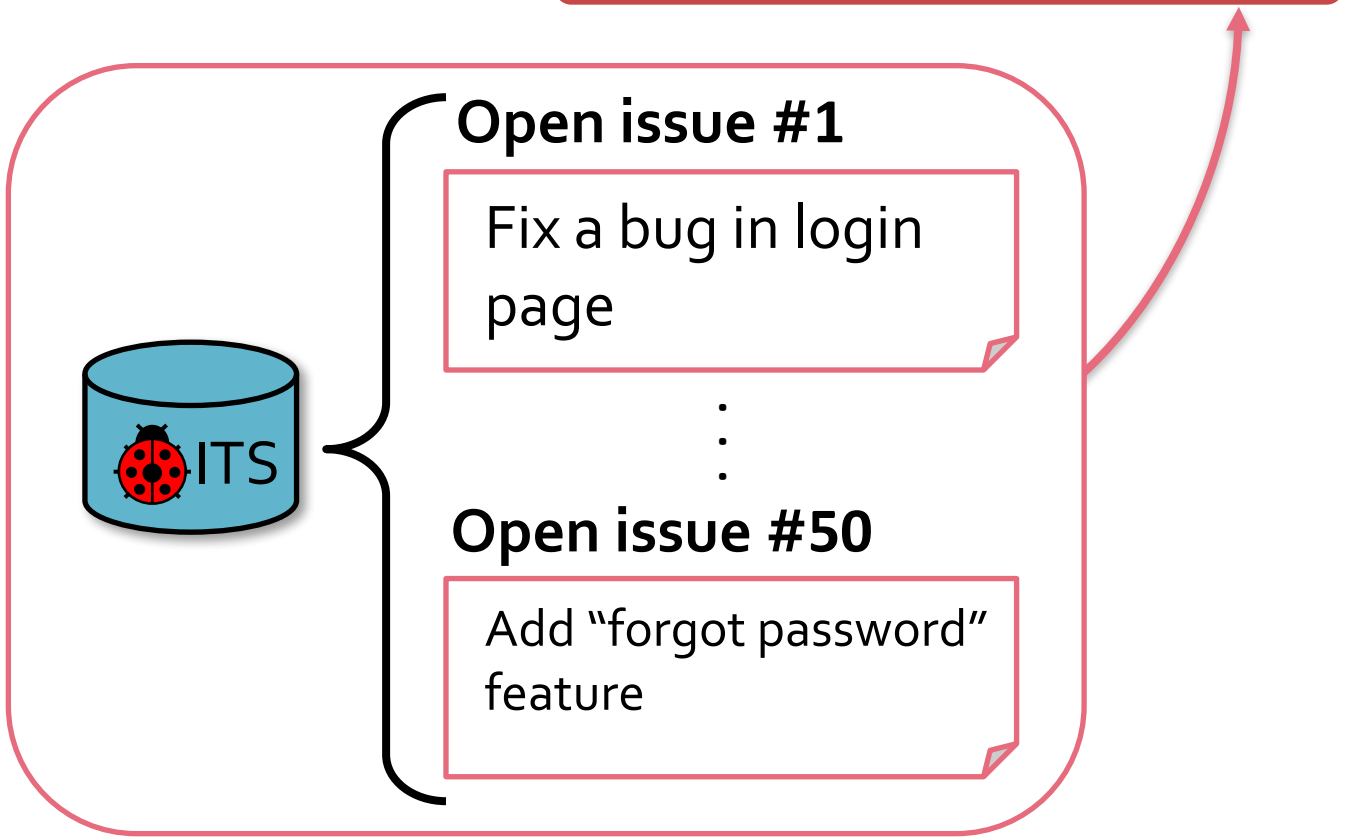
I need to implement feature **X** in method **A()**

[1] R. Marinescu, "Assessing technical debt by identifying design flaws in software systems," IBM Journal of Research and Development, 2012
[2] F. A. Fontana, V. Ferme, M. Zanoni, and R. Roveda, "Towards a Prioritization of Code Debt : A Code Smell Intensity Index, " MTD2015

PROPOSED TECHNIQUE

Developers' context

- Developers' context = modules to be modified



  - This list is used to estimate developers' context

Impact analysis

- Identify modules in source code that are likely to be affected by the changes
- Impact analysis → Change prediction

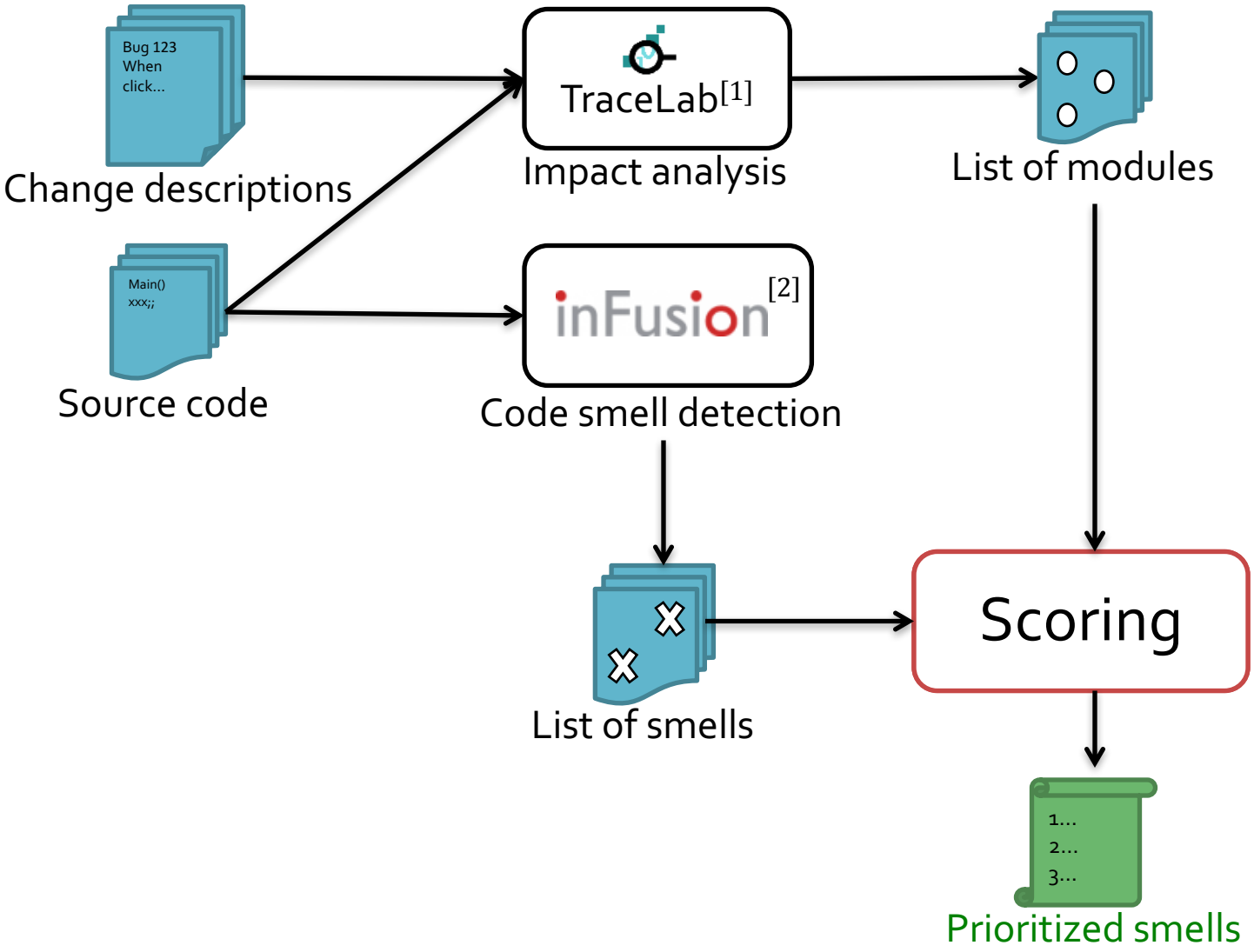
Change description #1

There is a bug in login page that user can login successfully if leave the password field blank.

Relevant modules

```
LoginPage.login()
LoginPage.Reset()
UserPage.setPassword()
FormField.getPassword()
…
UserPage.ShowError()
```

Approach overview



[1] B. Dit, E. Moritz, and D. Poshyvanyk, "A TraceLab-based Solution for Creating, Conducting, and Sharing Feature Location Experiments,", ICPC2012

[2] https://www.intooitus.com/products/infusion

Scoring

Context Relevance Index

- Accumulating the score of matched modules in IA result

Code smell detection results

Smell	Level	Module	CRI
…	…	…	…
Blob	Method	`LoginPage.login()`	**0.4**
…	…	…	…

Impact analysis results

#1

Relevant modules	Score
`FormField.getPassword()`	0.5
`LoginPage.login()`	0.1
…	

#50

Relevant modules	Score
`UserPage.Reset()`	0.7
`LoginPage.login()`	0.3
…	

EMPIRICAL STUDIES

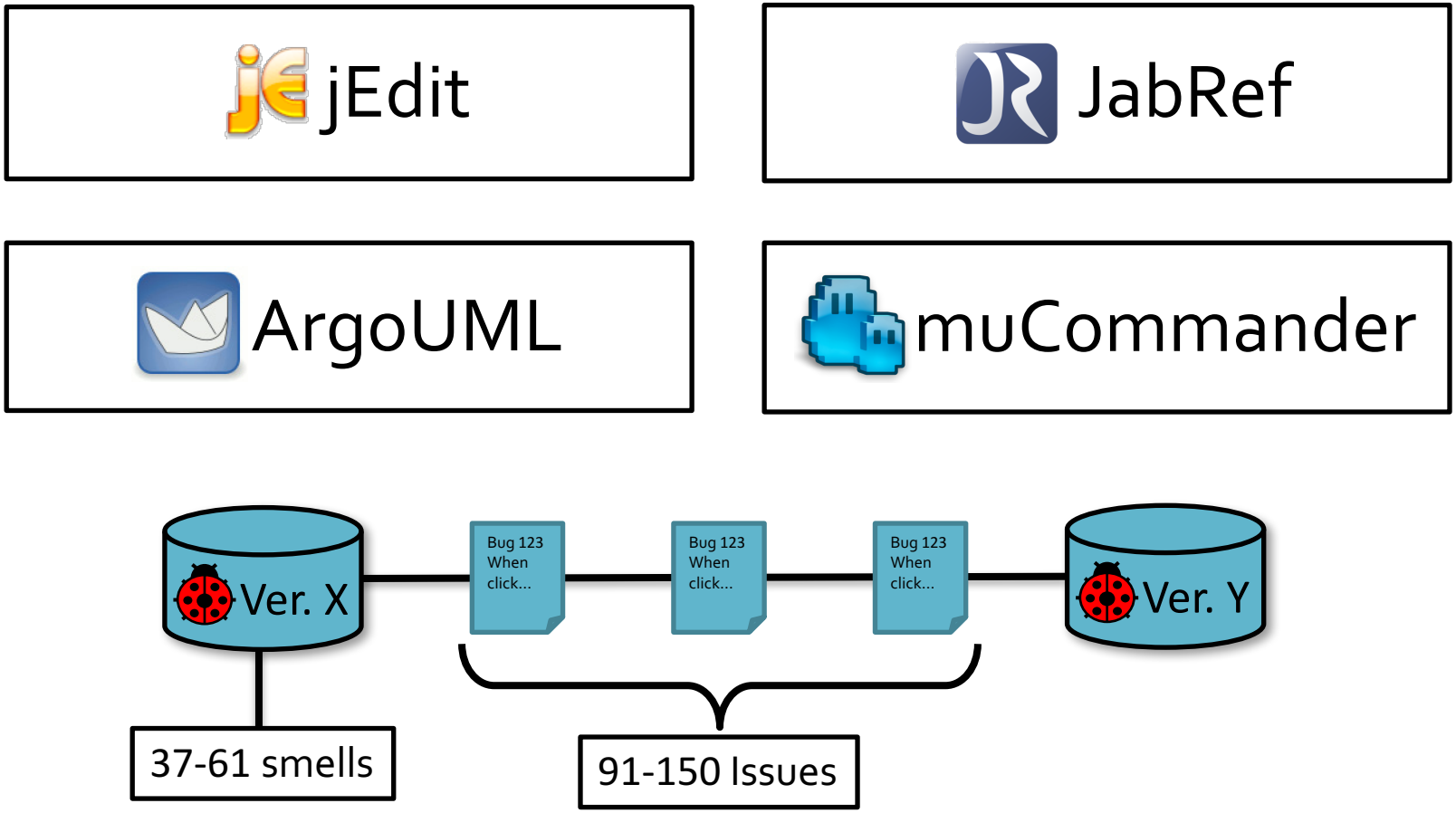
Empirical Studies

RQ2 : Does the accuracy of IA affect quality of the ranking ?

RQ3 : Does Context-based smell prioritization provide more relevant results than the severity-based one?

Subjects

- Use Dit et al.'s benchmark dataset[1]

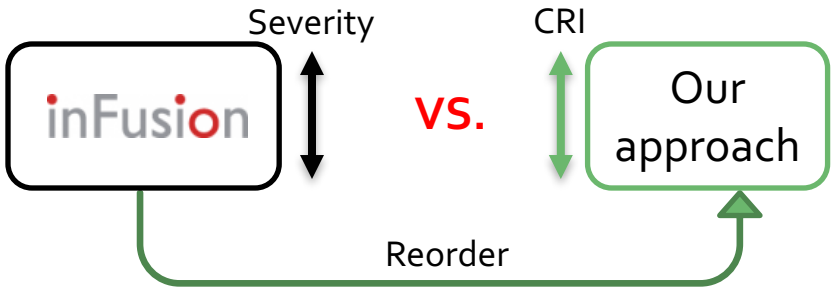


[1] Dit, B., Revelle, M., Gethers, M., and Poshyvanyk, D., "Feature Location in Source Code: A Taxonomy and Survey", JSME2011

Metric

- **nDCG (Normalized Discounted Cumulative Gain)**
  - Metric for evaluating the quality of ranking documents
  - Relevant documents in **higher rank** are **more useful** than the ones in lower rank

- **Calculate nDCG for:**



- **Oracle**
  - Smells occurring in the modules *actually* modified during two releases

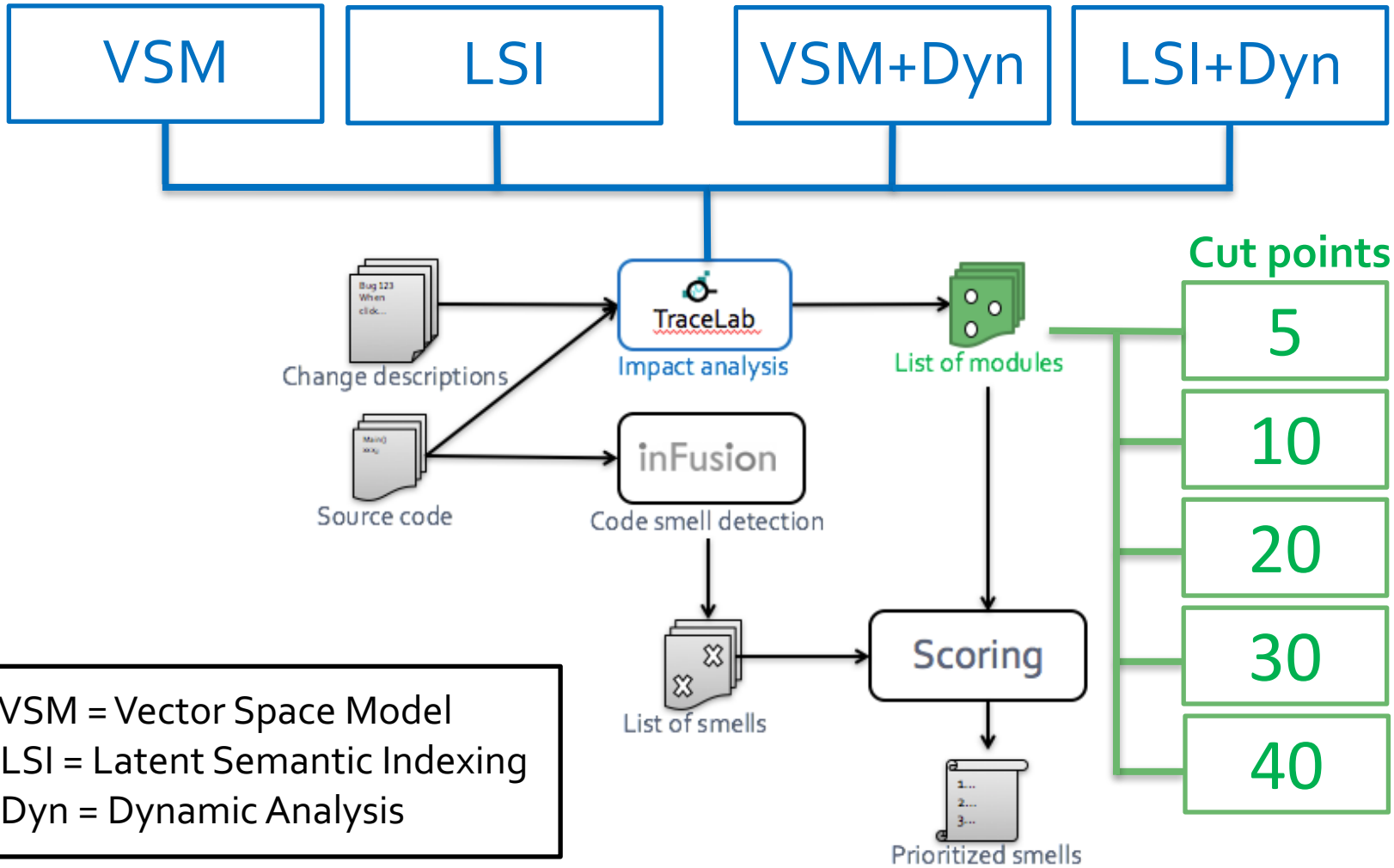
Empirical Studies

RQ2 : Does the accuracy of IA affect quality of the ranking ?

RQ3 : Does Context-based smell prioritization provide more relevant results than the severity-based one?

RQ2

Impact analysis[1]

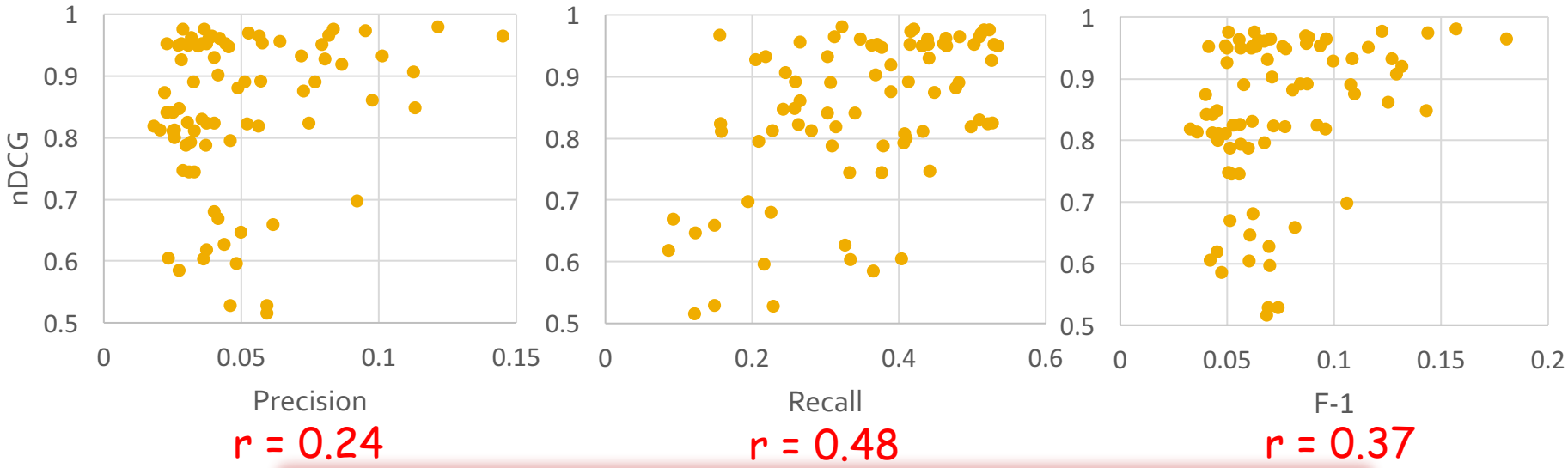


VSM = Vector Space Model
LSI = Latent Semantic Indexing
Dyn = Dynamic Analysis

[1] M. Gethers, B. Dit, H. Kagdi, and D. Poshyvanyk, "Integrated impact analysis for managing software changes," ICSE2012

RQ2

- **RQ2: Does the accuracy of impact analysis affect quality of the ranking?**
- **Spearman's correlation coefficient**
  - Evaluate the association between two variables



Accuracy of IA tends to affect quality of the ranking by our technique

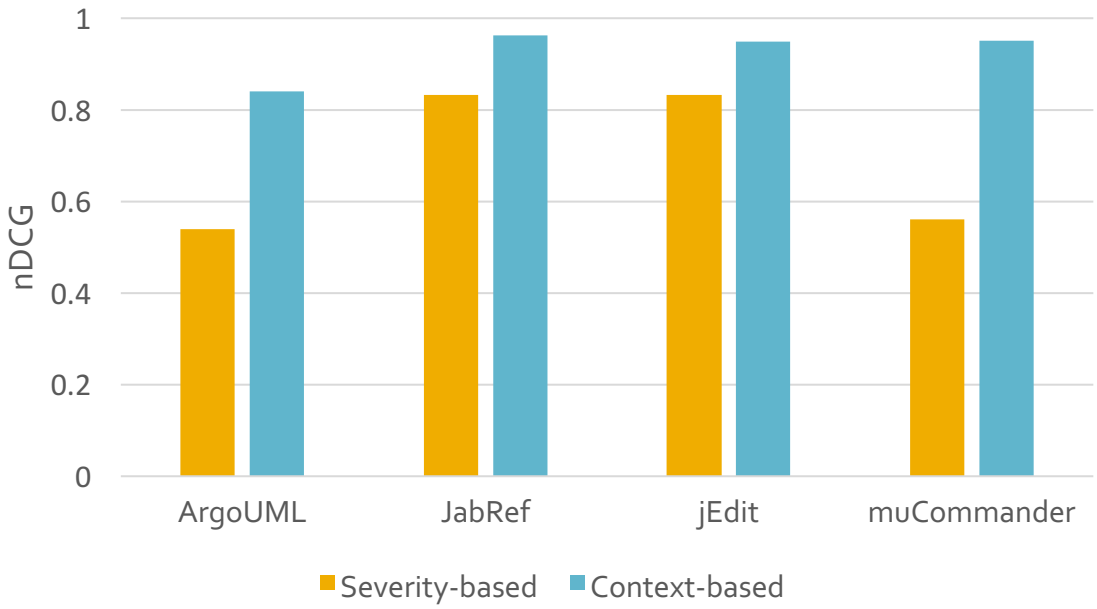
Empirical Studies

RQ2 : Does the accuracy of IA affect quality of the ranking ?

RQ3 : Does Context-based smell prioritization provide more relevant results than the severity-based one?

RQ3

- **RQ3: Does context-based smell prioritization provide more relevant results than the severity-based one?**



YES !

RQ3

Baseline

Rank	Smell Type	Class Name	Severity	#Issues
1	Blob	GeneratorCSharp	8	
2	Blob	GeneratorJava	8	
3	God	FigAssociation	8	5
4	Blob	ParserDisplay	8	1
5	Blob	GeneratorPHP4	7	
6	RPB	FigClassifierRole	7	3
7	Blob	Modeller	7	1
8	SC	Import	6	
9	God	CoreFactoryMDRImpl	5	1
10	RPB	StylePanelFigText	5	

Our approach

Rank	Smell Type	Class Name	CRI	#Issues
1	God	Project	7.90	3
2	God	ProjectBrowser	4.04	7
3	Blob	ProjectBrowser	4.04	7
4	SC	StylePanel	2.43	1
5	God	FigNodeModelElemen	2.18	4
6	God	UMLMutableGraphS	1.54	
7	Blob	GeneratorCSharp	1.04	
8	God	FigEdgeModelIEleme	0.94	3
9	God	ExtensionMechanism	0.91	1
10	God	CoreFactoryMDRImpl	0.80	1

CONCLUSION

Conclusion

Context-based code smells prioritization

Prefactoring

Automated

Accuracy of IA tends to impact the results

More relevant results than severity-based