Bug Report Analytics

David Lo

School of Information Systems
Singapore Management University

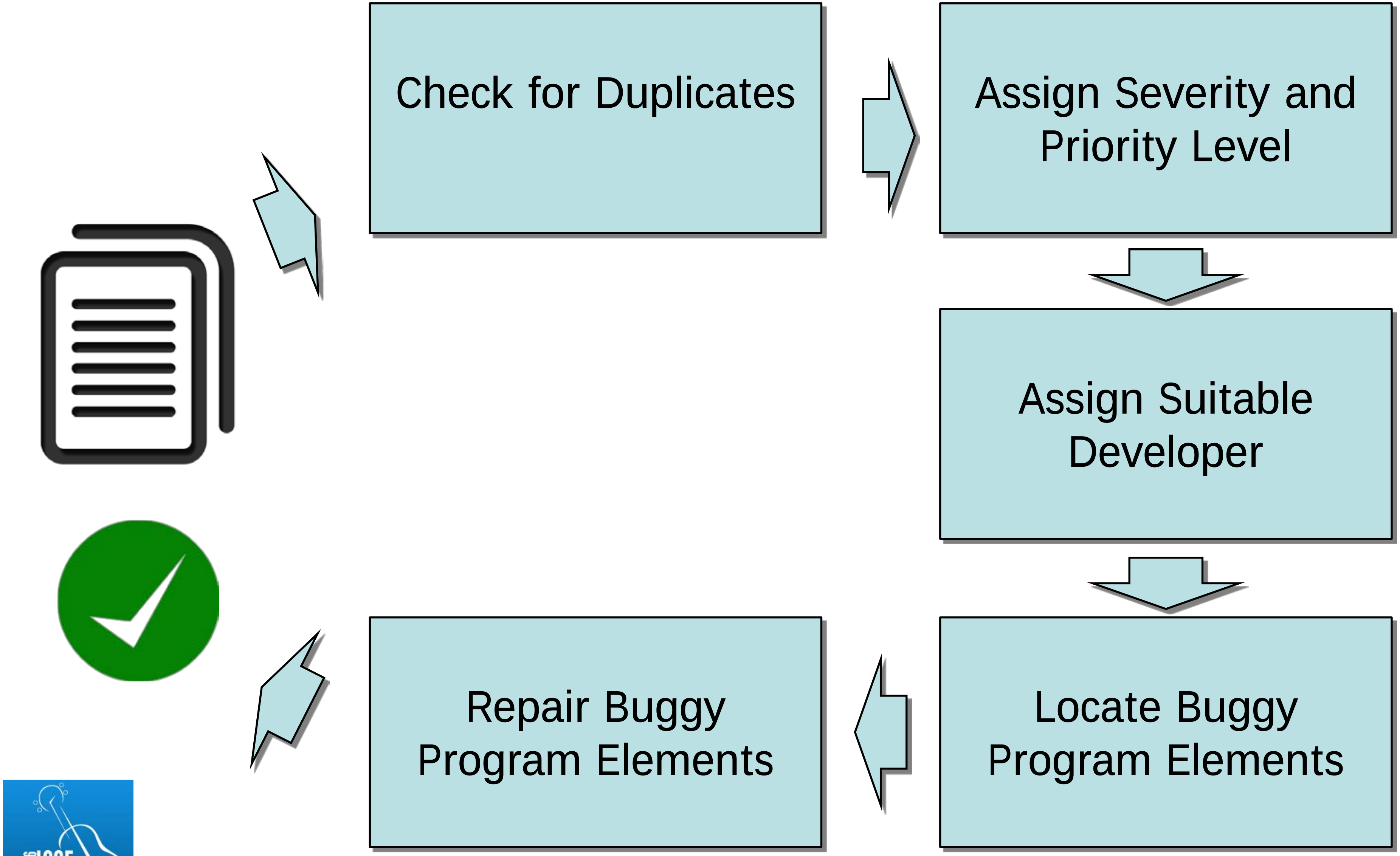
davidlo@smu.edu.sg

38th ACM/IEEE International Conference on Software Engineering, Austin, Texas, USA

Problems with Bug Reports

- Bugs are reported in bug tracking systems
- The number of bug reports are often too many for developers to handle (Anvik et al., ETX 2005)
- Management of bugs is an expensive process (NIST, 2002)

Bug Report Management Process



Check for Duplicates

Assign Severity and Priority Level

Assign Suitable Developer

Locate Buggy Program Elements

Repair Buggy Program Elements

How Analytics Can Help?

Automation

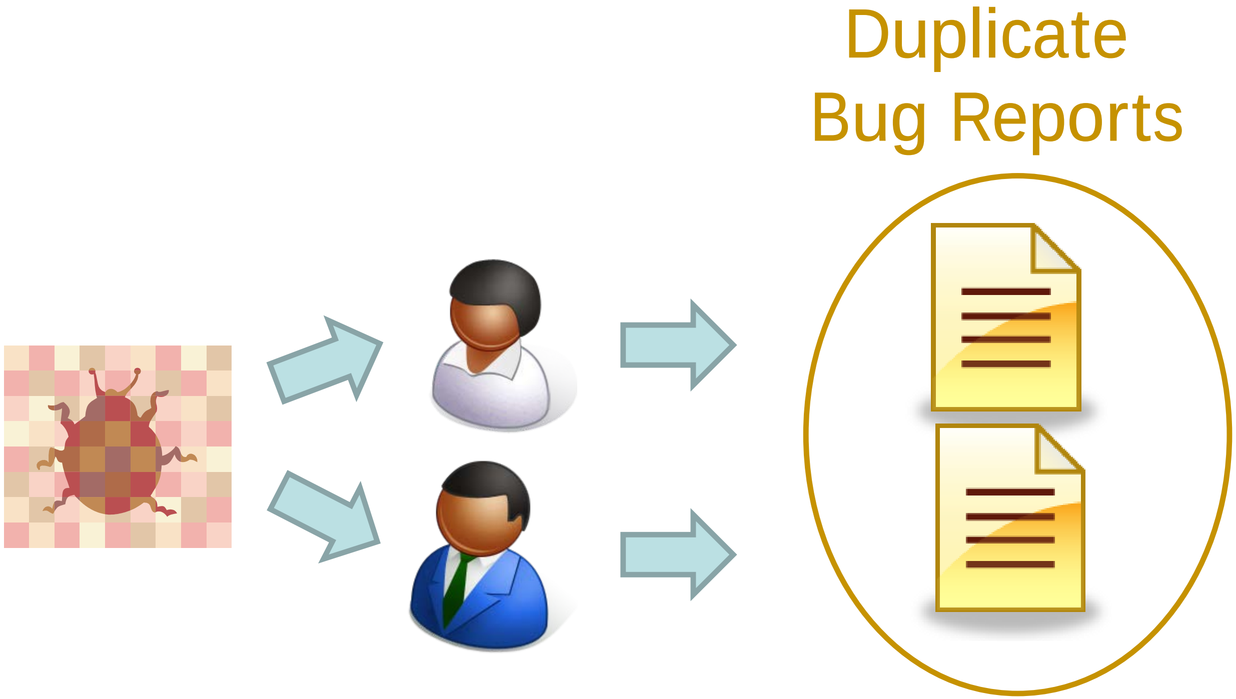
Recommendation

Structure of This Talk

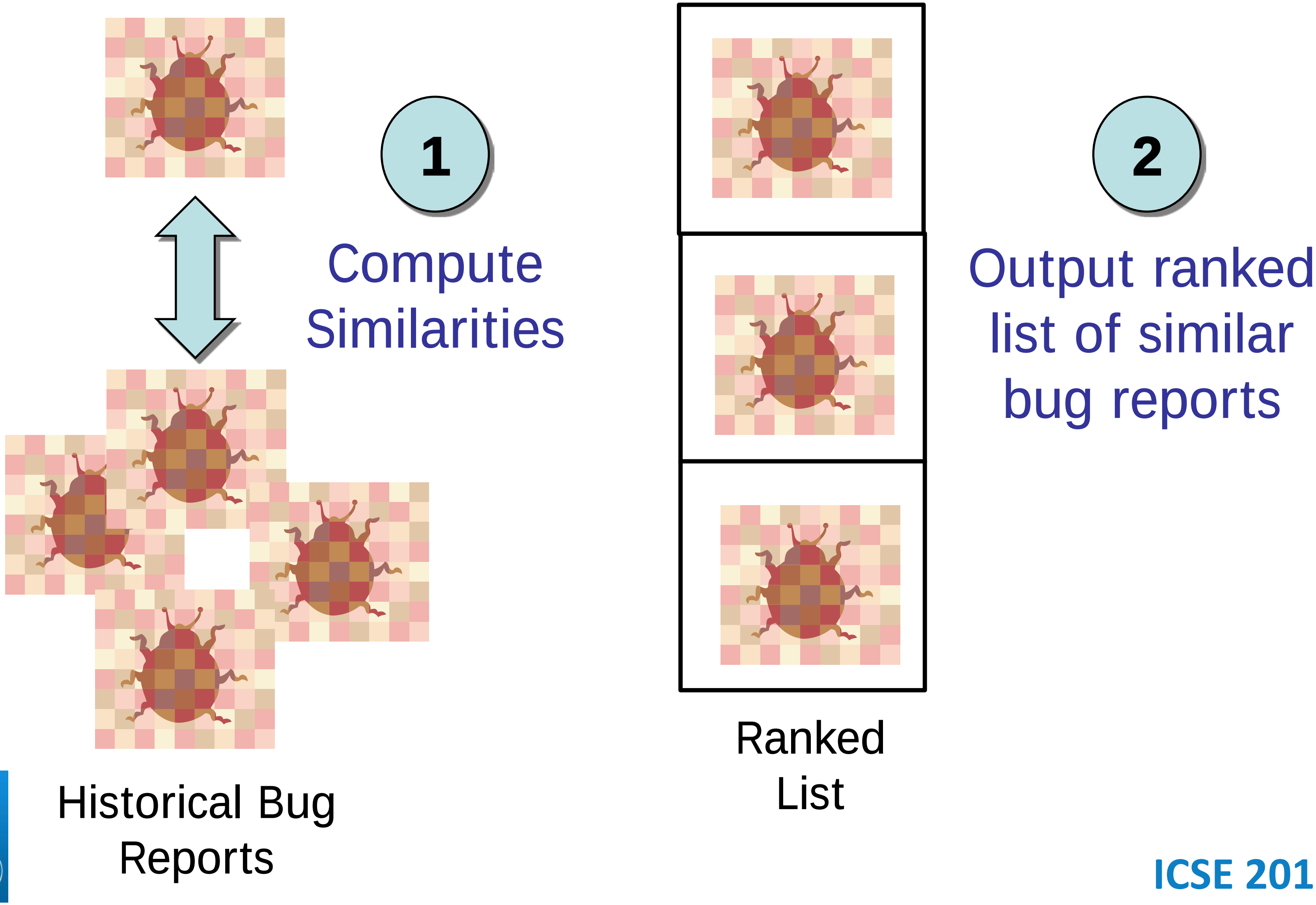
1. Duplicate Bug Report Detection
2. Priority/Severity Prediction
3. Developer Assignment
4. Bug Localization
5. Automated Repair

Duplicate Bug Report Detection

- Bug reporting is inherently a distributed and uncoordinated process.
- Similar people (users, testers) report the same bug in a different reports.



Duplicate Bug Report Detection



1

Compute Similarities

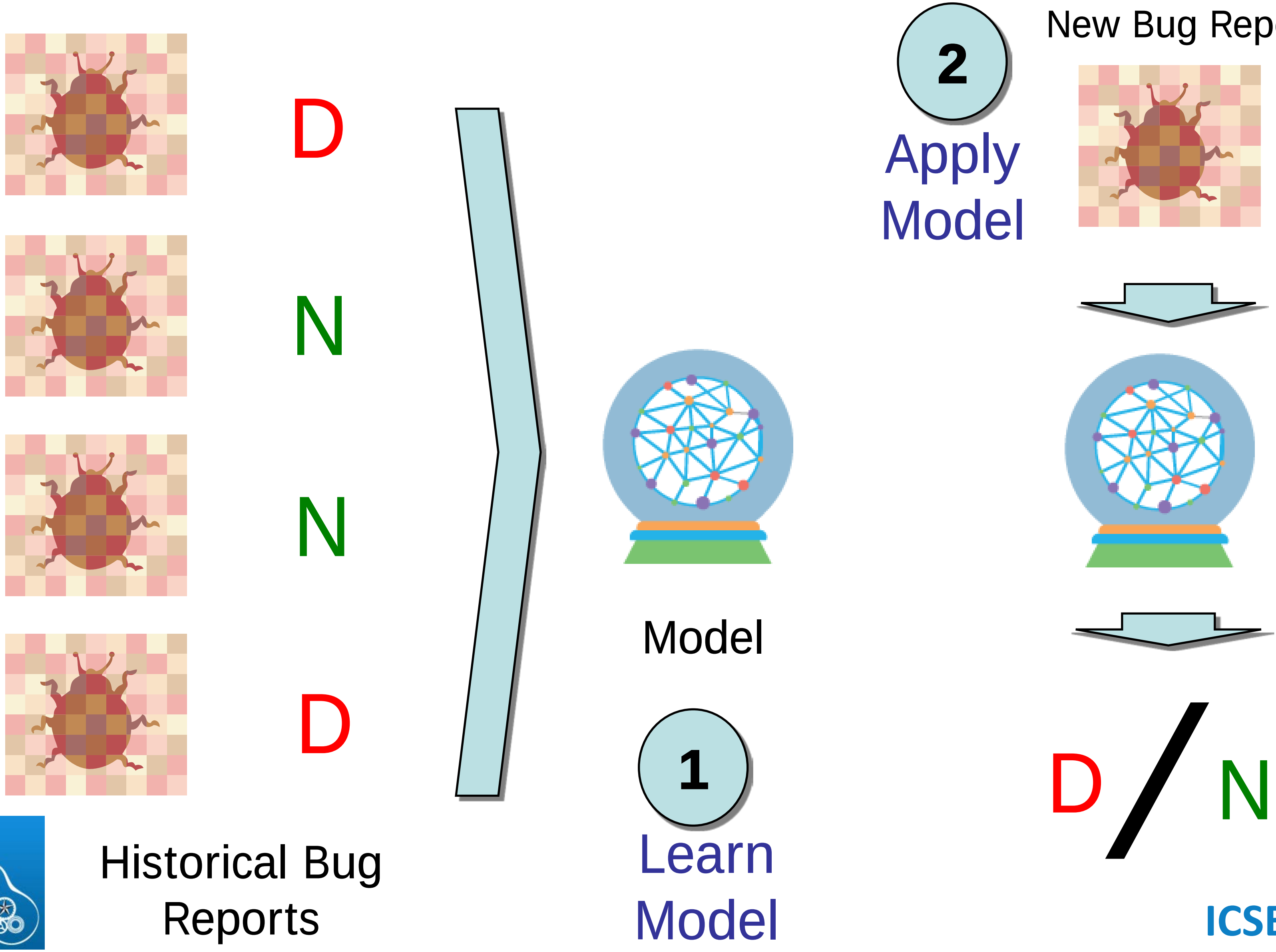
2

Output ranked list of similar bug reports

Historical Bug Reports

Ranked List

Duplicate Bug Report Detection



D

N

N

D

Historical Bug Reports

Model

1

Learn Model

2

Apply Model

New Bug Report

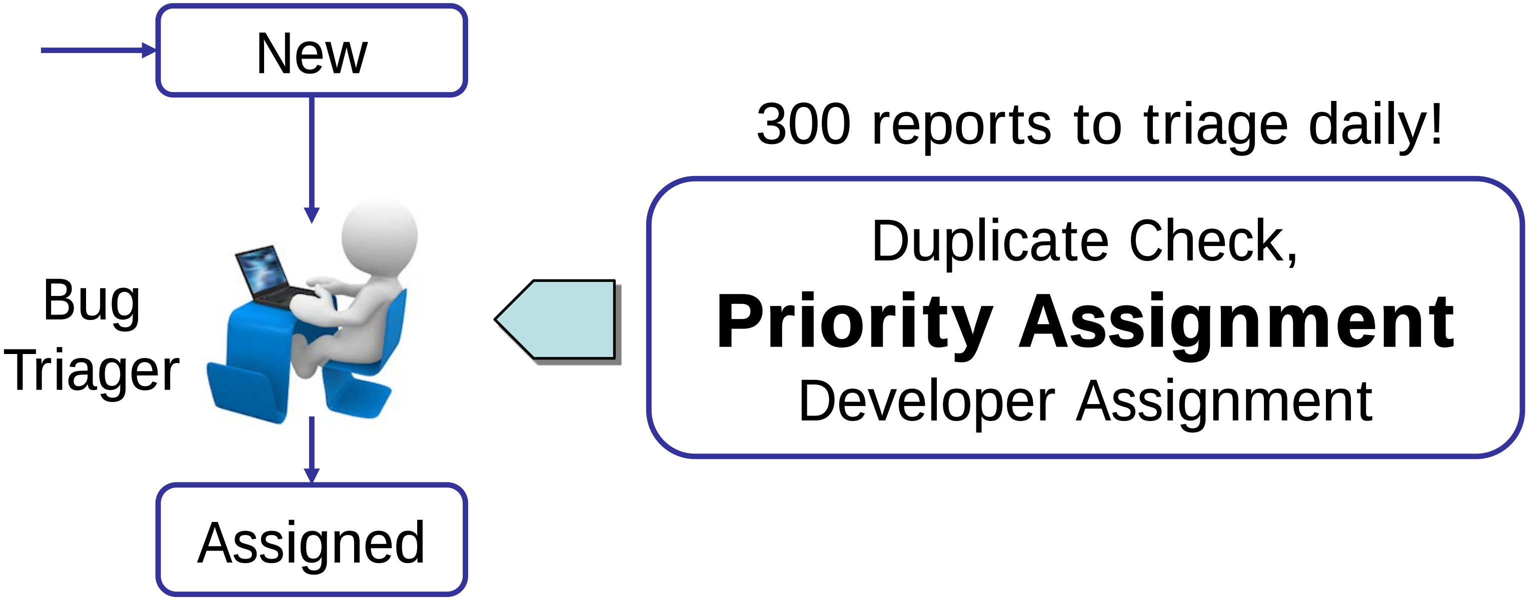
D / N

Duplicate Bug Report Detection

- Similarity Based
  - Anh Tuan Nguyen, Tung Thanh Nguyen, Tien N. Nguyen, David Lo, Chengnian Sun: Duplicate bug report detection with a combination of information retrieval and topic modeling. ASE 2012: 70-79
  - Chengnian Sun, David Lo, Siau-Cheng Khoo, Jing Jiang: Towards more accurate retrieval of duplicate bug reports. ASE 2011: 253-262
  - Chengnian Sun, David Lo, Xiaoyin Wang, Jing Jiang, Siau-Cheng Khoo: A discriminative model approach for accurate duplicate bug report retrieval. ICSE (1) 2010: 45-54
- Classification Based
  - Anahita Alipour, Abram Hindle, Eleni Stroulia: A contextual approach towards more accurate duplicate bug report detection. MSR 2013: 183-192
  - Yuan Tian, Chengnian Sun, David Lo: Improved Duplicate Bug Report Identification. CSMR 2012: 385-390

Severity/Priority Prediction

- Developers have limited time
- Some reports are more important than others
- **Severity** of reports need to be estimated
- Bug reports need to be **prioritized**



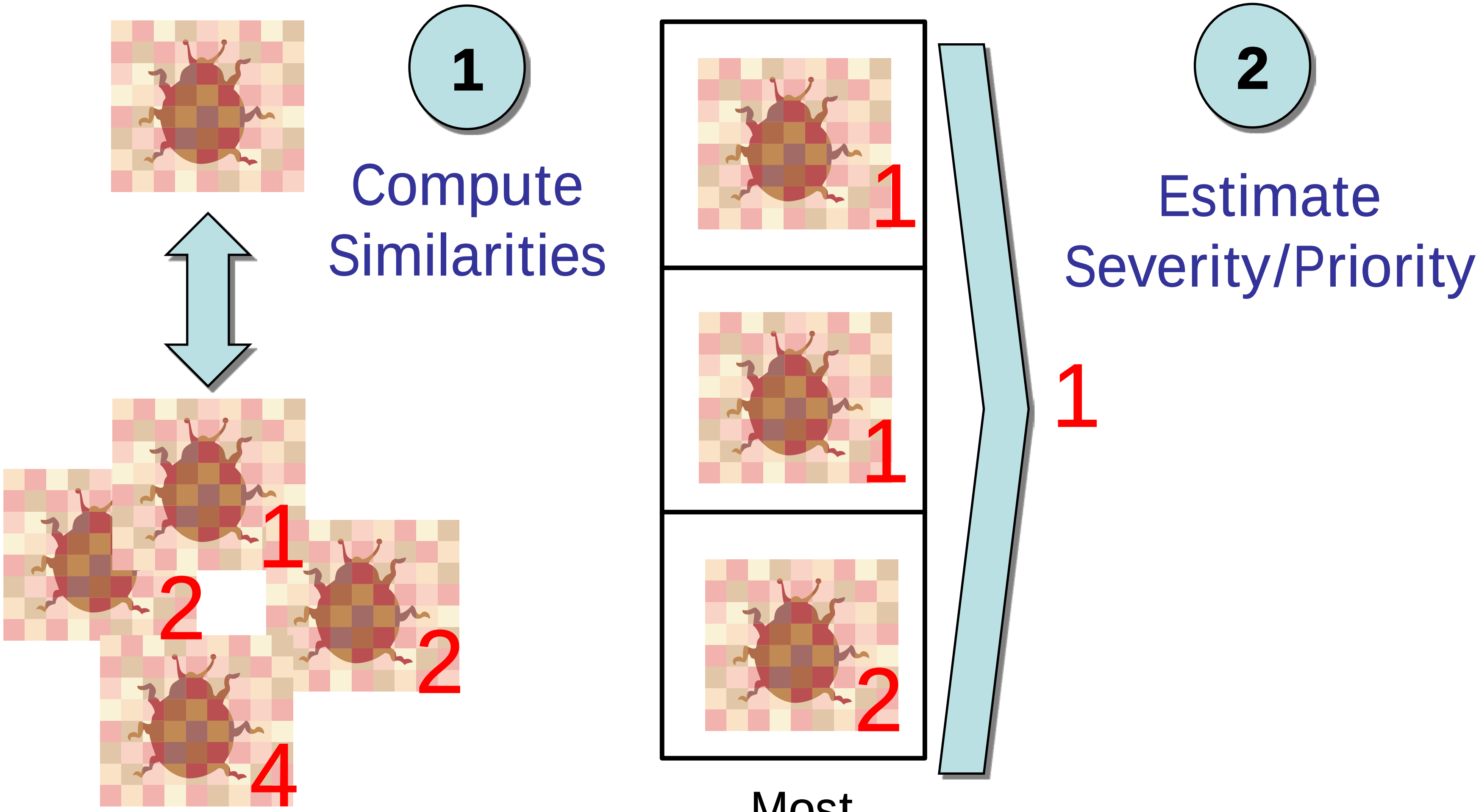
300 reports to triage daily!

Duplicate Check,

Priority Assignment

Developer Assignment

Severity/Priority Prediction



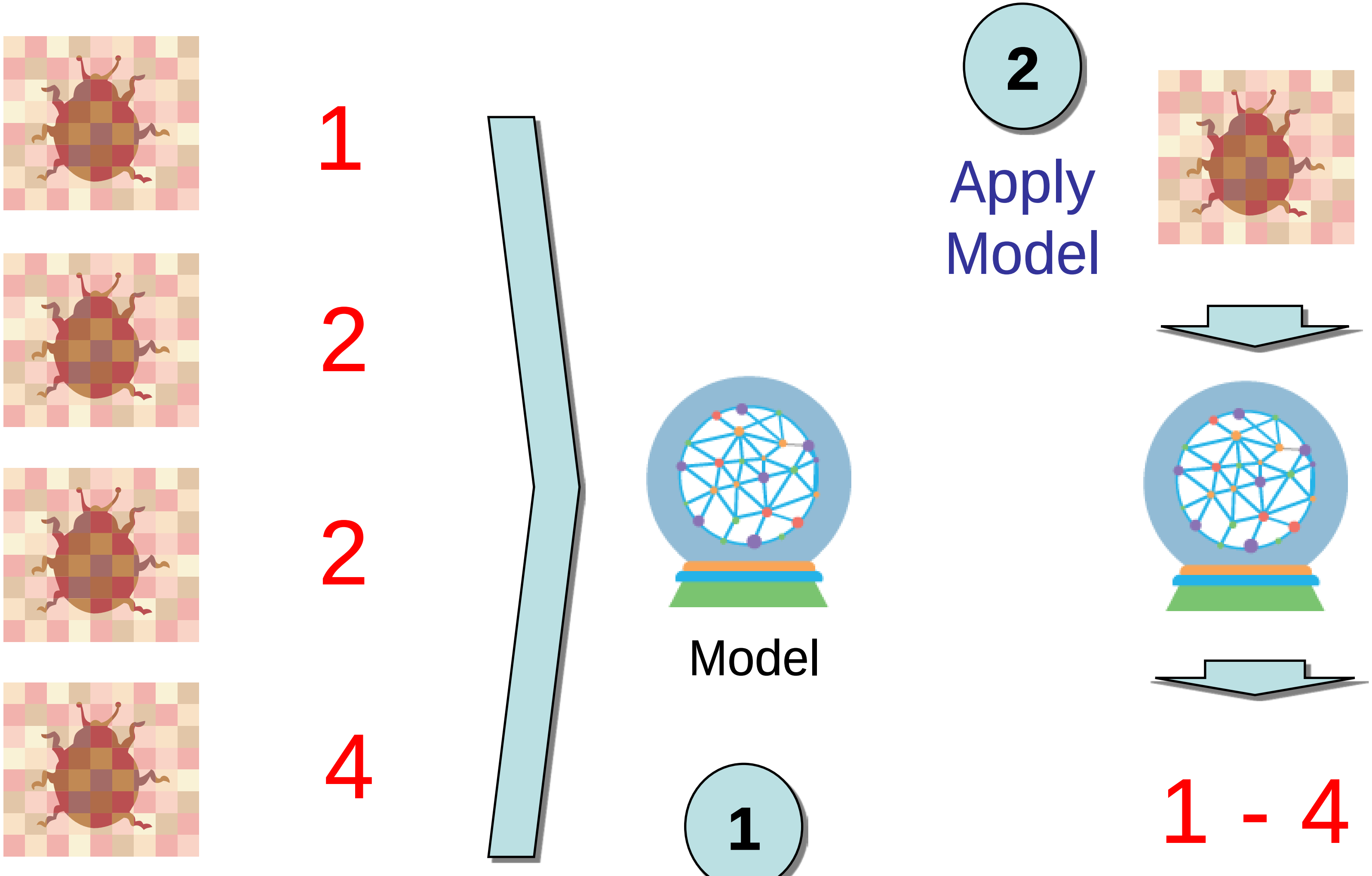
Compute Similarities

Estimate Severity/Priority

Historical Bug Reports

Most Similar Reports

Severity/Priority Prediction



1

2

2

4

Historical Bug Reports

1 Learn Model

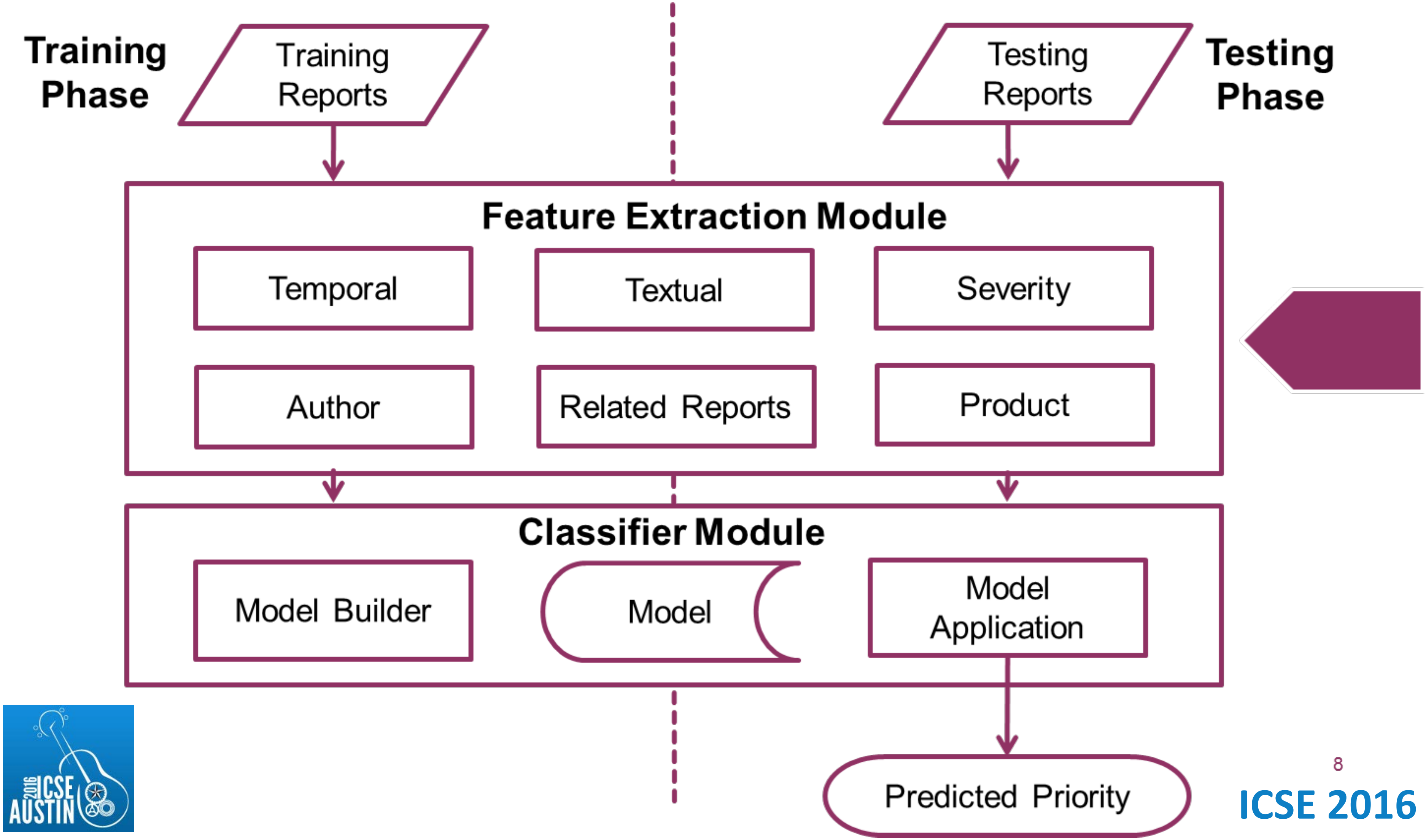
Model

2 Apply Model

1 - 4

Severity/Priority Prediction

(DRONE, Tian et al., 2012)



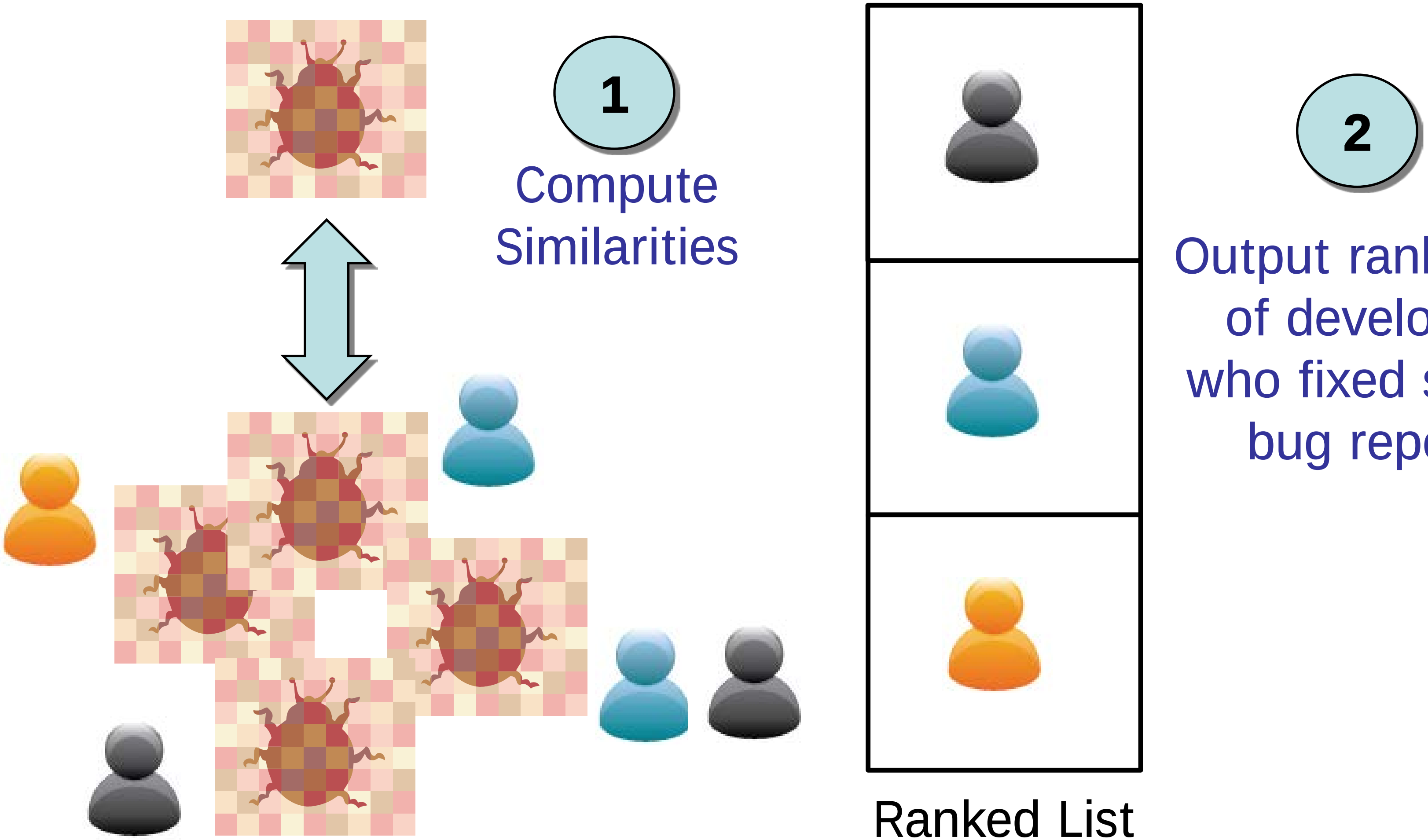
Severity/Priority Prediction

- Severity Prediction
  - Yuan Tian, David Lo, Chengnian Sun: Information Retrieval Based Nearest Neighbor Classification for Fine-Grained Bug Severity Prediction. WCRE 2012: 215-224
  - Tim Menzies, Andrian Marcus: Automated severity assessment of software defect reports. ICSM 2008: 346-355
  - Ahmed Lamkanfi, Serge Demeyer, Quinten David Soetens, Tim Verdonck: Comparing Mining Algorithms for Predicting the Severity of a Reported Bug. CSMR 2011: 249-258
  - Ahmed Lamkanfi, Serge Demeyer, Emanuel Giger, Bart Goethals: Predicting the severity of a reported bug. MSR 2010: 1-10
- Priority Prediction
  - Yuan Tian, David Lo, Xin Xia, Chengnian Sun: Automated prediction of bug report priority using multi-factor analysis. Empirical Software Engineering 20(5): 1354-1383 (2015)

Developer Assignment

- Many projects have a large number of contributors
- Each contributor have different expertise
- How to assign the right contributor to a suitable bug report?

Developer Assignment



1

Compute Similarities

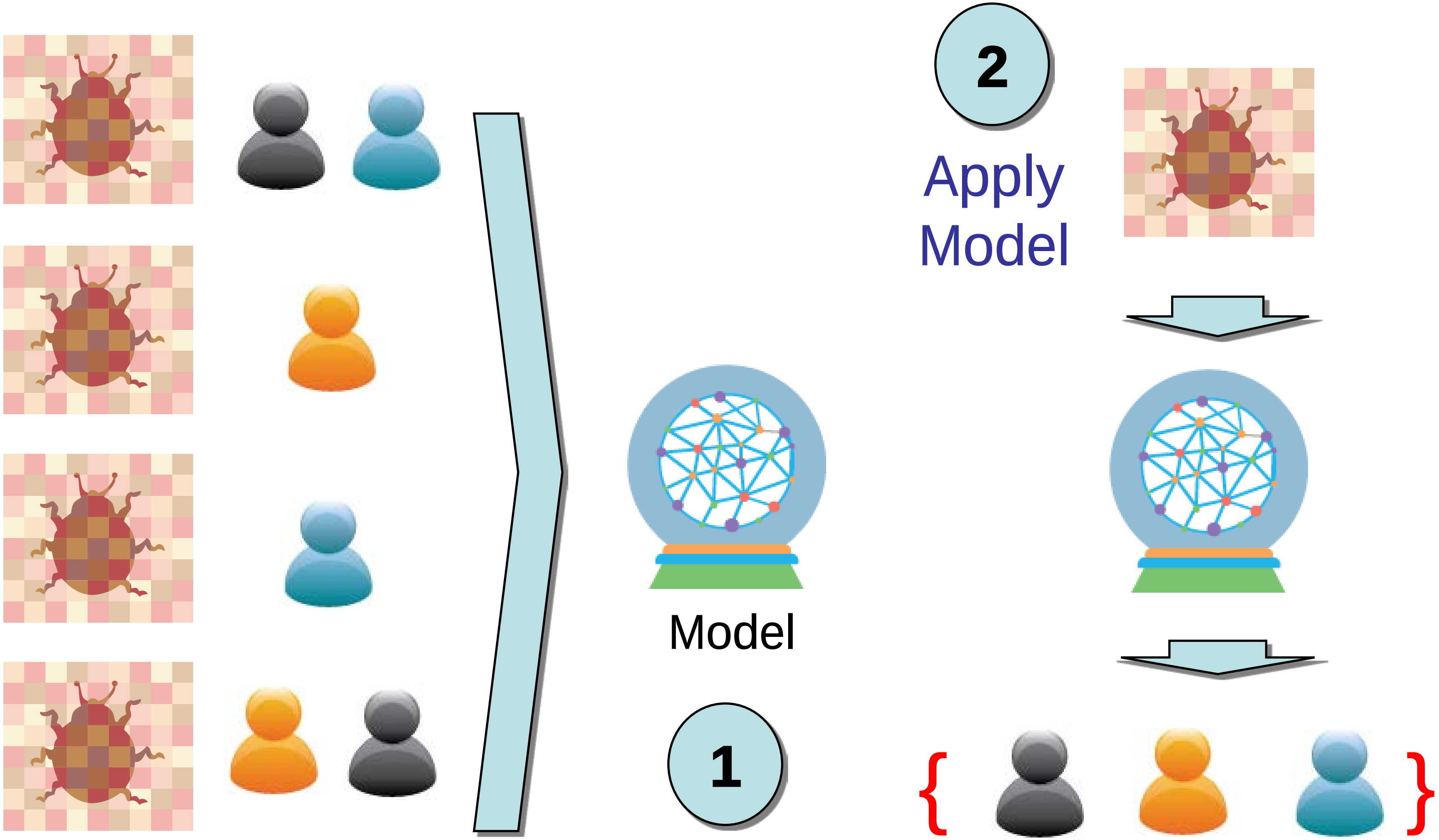
2

Output ranked list of developers who fixed similar bug reports

Ranked List

Historical Bug Reports

Developer Assignment



Historical Bug Reports

Model

1

Learn Model

2

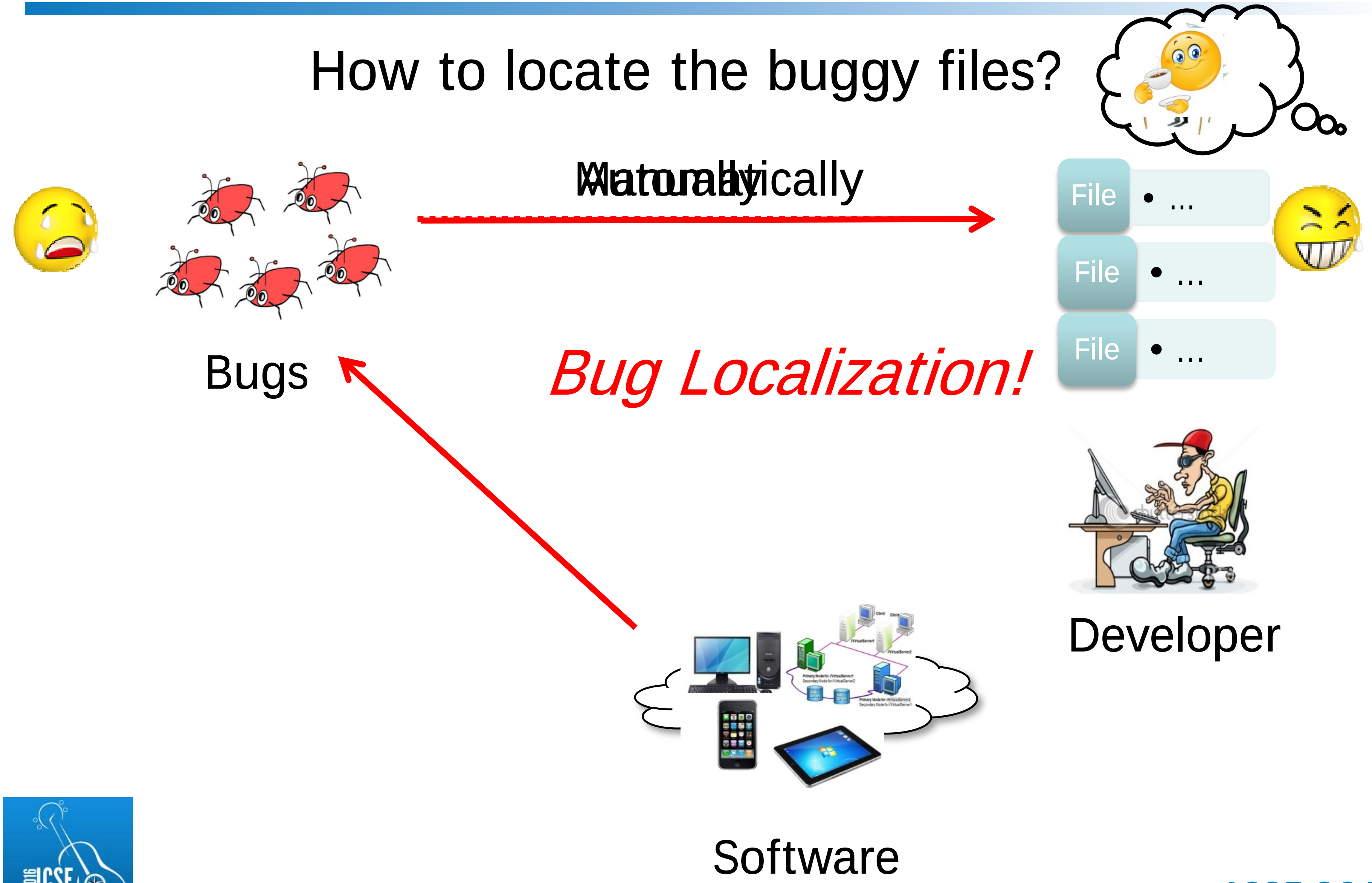
Apply Model

{ }

Developer Assignment

- Similarity Based
  - Xin Xia, David Lo, Ying Ding, Jafar M. Al-Kofahi, Tien N. Nguyen, Xinyu Wang. "Improving Automated Bug Triaging with Specialized Topic Model". IEEE Transactions on Software Engineering (TSE), 26 pages. (to appear)
  - Ahmed Tamrawi, Tung Thanh Nguyen, Jafar M. Al-Kofahi, Tien N. Nguyen: Fuzzy set and cache-based approach for bug triaging. SIGSOFT FSE 2011: 365-375
- Classification Based
  - John Anvik, Lyndon Hiew, Gail C. Murphy: Who should fix this bug? ICSE 2006: 361-370
  - Xin Xia, David Lo, Xinyu Wang, Bo Zhou: Accurate developer recommendation for bug resolution. WCRE 2013: 72-81
  - Jifeng Xuan, He Jiang, Zhilei Ren, Jun Yan, Zhongxuan Luo: Automatic Bug Triage using Semi-Supervised Text Classification. SEKE 2010: 209-214

Bug Localization



How to locate the buggy files?

Manually

Automatically

Bugs

Bug Localization!

File • ...

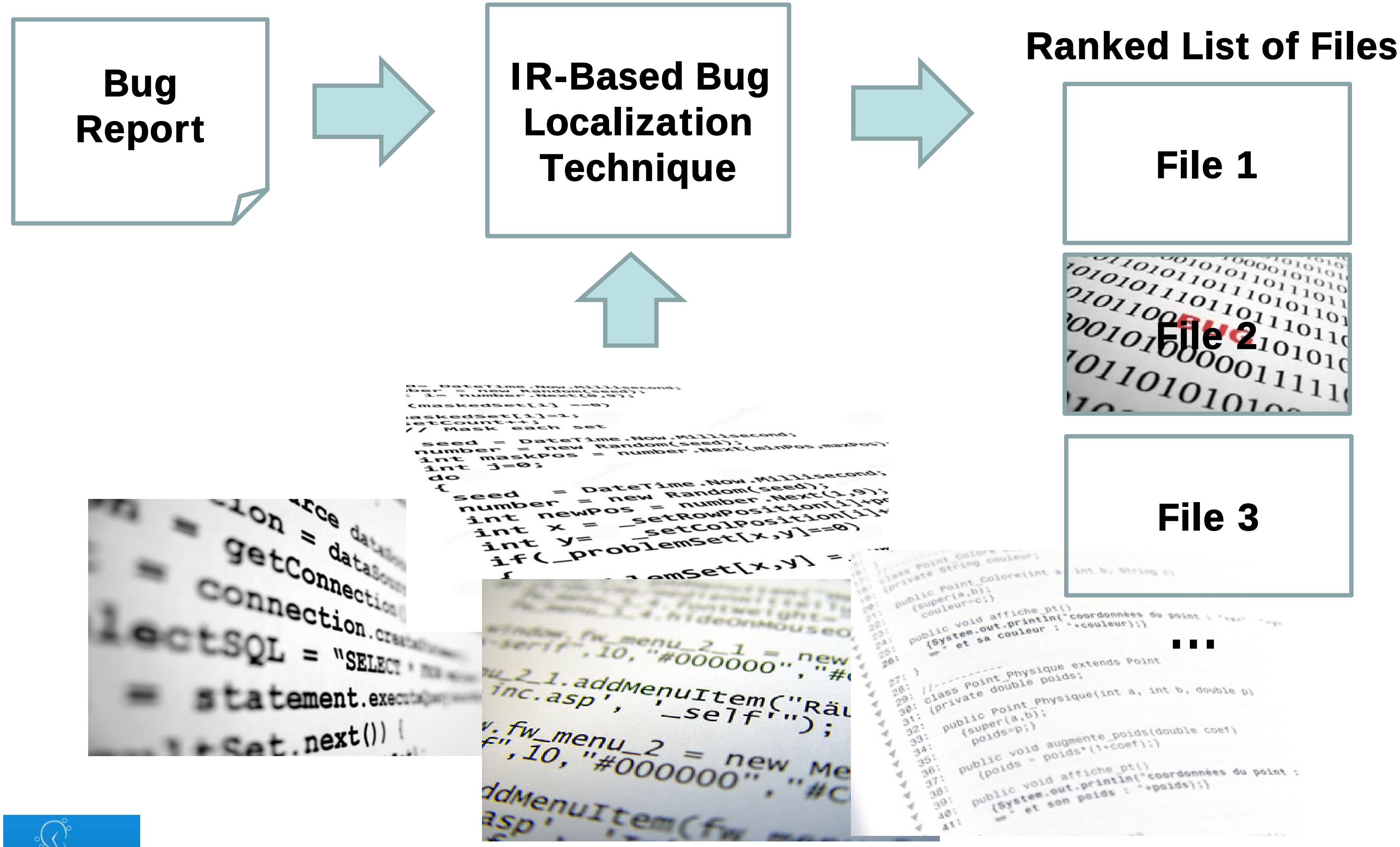
File • ...

File • ...

Developer

Software

IR-Based Bug Localization



Bug Report

IR-Based Bug Localization Technique

Ranked List of Files

File 1

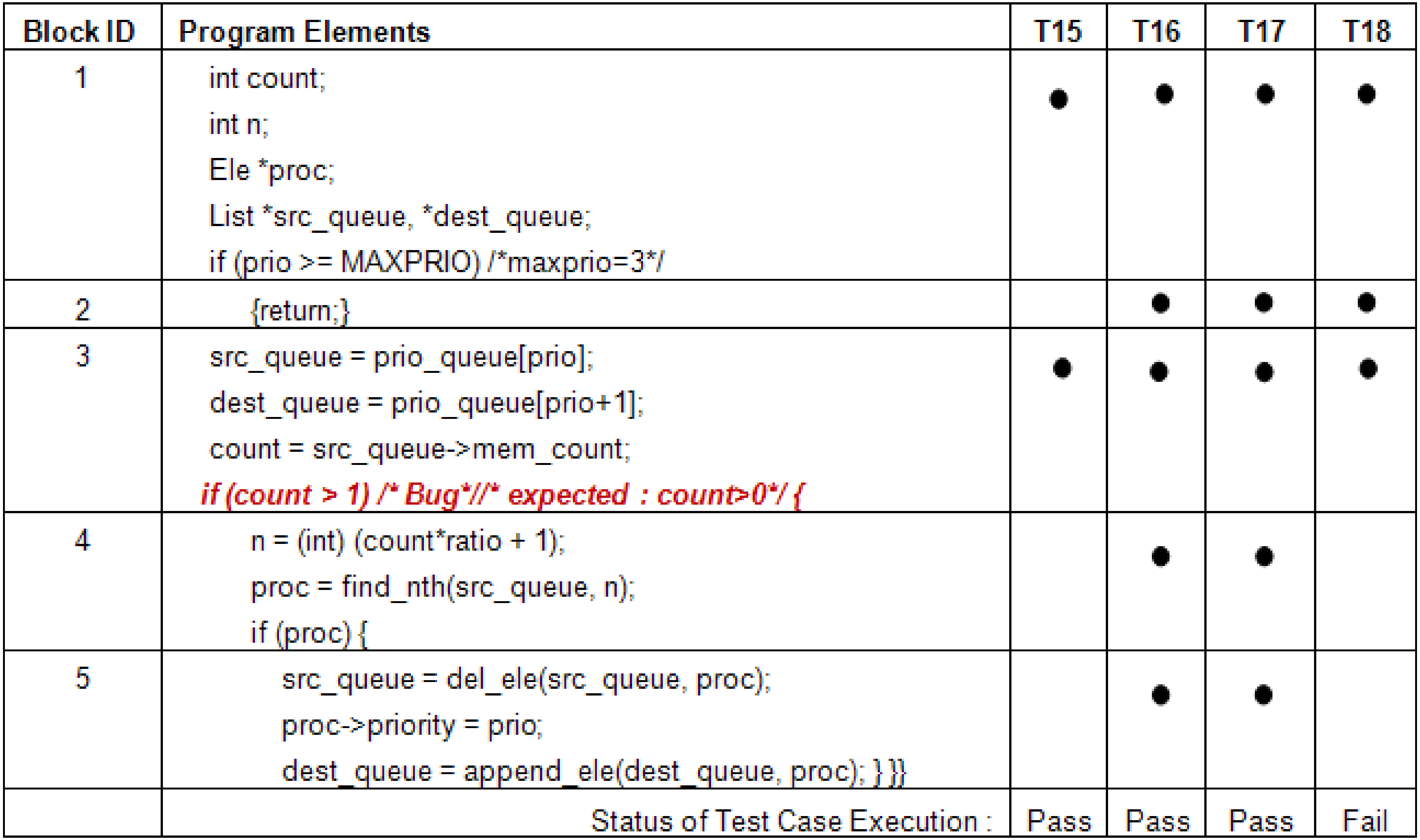
File 2

File 3

…

(Thousands of) Source Code Files

Spectrum-Based Bug Localization



Block ID	Program Elements	T15	T16	T17	T18

1	int count;
	int n;
	Ele *proc;
	List *src_queue, *dest_queue;
	if (prio >= MAXPRIO) /*maxprio=3*/
		●	●	●	●

2	{return;}		●	●	●

3	src_queue = prio_queue[prio];
	dest_queue = prio_queue[prio+1];
	count = src_queue->mem_count;
	if (count > 1) /* Bug*//* expected : count>0*/ {
		●	●	●	●

4	n = (int) (count*ratio + 1);
	proc = find_nth(src_queue, n);
	if (proc) {
			●	●	

5	src_queue = del_ele(src_queue, proc);
	proc->priority = prio;
	dest_queue = append_ele(dest_queue, proc); } }}
			●	●	

	Status of Test Case Execution :	Pass	Pass	Pass	Fail

Spectrum-Based Bug Localization

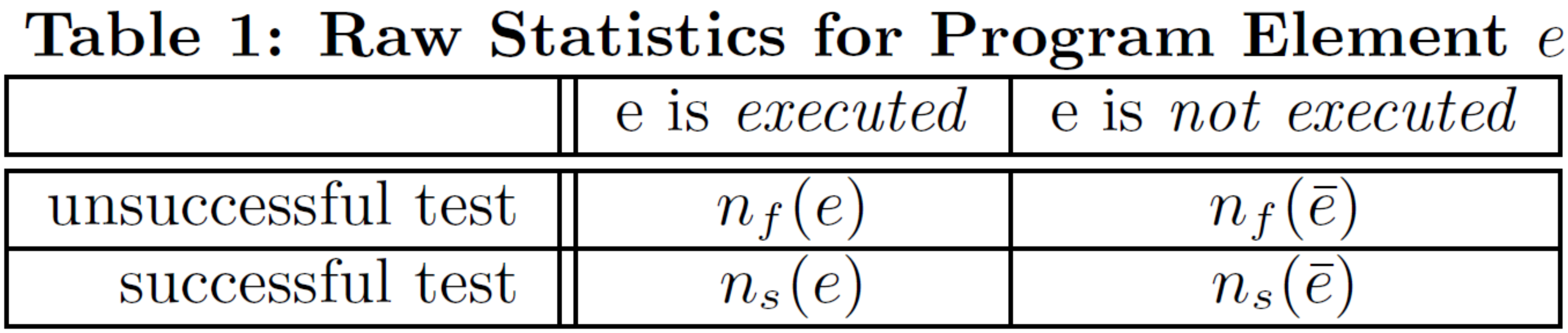
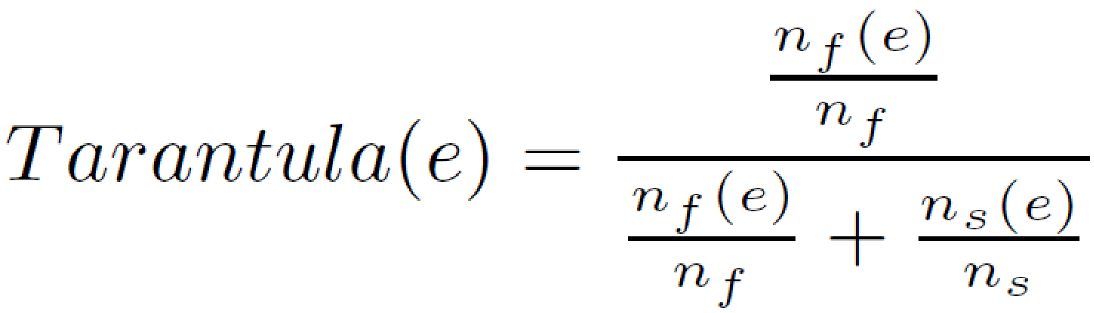


Table 1: Raw Statistics for Program Element e

	e is *executed*	e is *not executed*
unsuccessful test	$n_f(e)$	$n_f(\bar{e})$
successful test	$n_s(e)$	$n_s(\bar{e})$



$$Tarantula(e) = \frac{\frac{n_f(e)}{n_f}}{\frac{n_f(e)}{n_f} + \frac{n_s(e)}{n_s}}$$

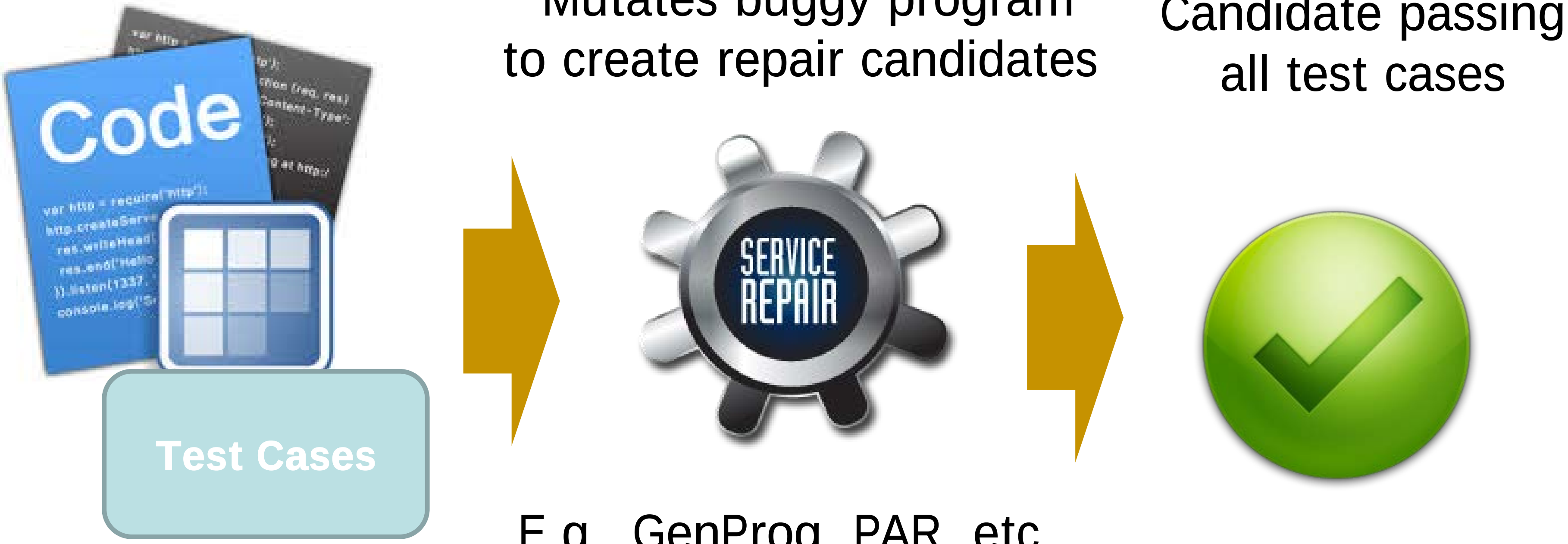
Bug Localization

- IR-Based Bug Localization
  - Shaowei Wang, David Lo, Julia Lawall: Compositional Vector Space Models for Improved Bug Localization. ICSME 2014: 171-180
  - Shaowei Wang, David Lo: Version history, similar report, and structure: putting them together for improved bug localization. ICPC 2014: 53-63
  - Xin Xia, David Lo, Xingen Wang, Chenyi Zhang, Xinyu Wang: Cross-language bug localization. ICPC 2014: 275-278
  - Xin Ye, Razvan C. Bunescu, Chang Liu: Learning to rank relevant files for bug reports using domain knowledge. SIGSOFT FSE 2014: 689-699
  - Jian Zhou, Hongyu Zhang, David Lo: Where should the bugs be fixed? More accurate information retrieval-based bug localization based on bug reports. ICSE 2012: 14-24

Bug Localization

- Spectrum-Based Bug Localization
  - Tien-Duy B. Le, David Lo, Claire Le Goues and Lars Grunske. A Learning-to-Rank Based Fault Localization Approach using Likely Invariants. ISSTA 2016 (to appear)
  - Lucia, David Lo, Lingxiao Jiang, Ferdian Thung, Aditya Budi: Extended comprehensive study of association measures for fault localization. Journal of Software: Evolution and Process 26(2): 172-219 (2014)
  - James A. Jones, Mary Jean Harrold: Empirical evaluation of the tarantula automatic fault-localization technique. ASE 2005: 273-282
- Combination
  - Tien-Duy B. Le, Richard Jayadi Oentaryo, David Lo: Information retrieval and spectrum based bug localization: better together. ESEC/SIGSOFT FSE 2015: 579-590

Automatic Repair

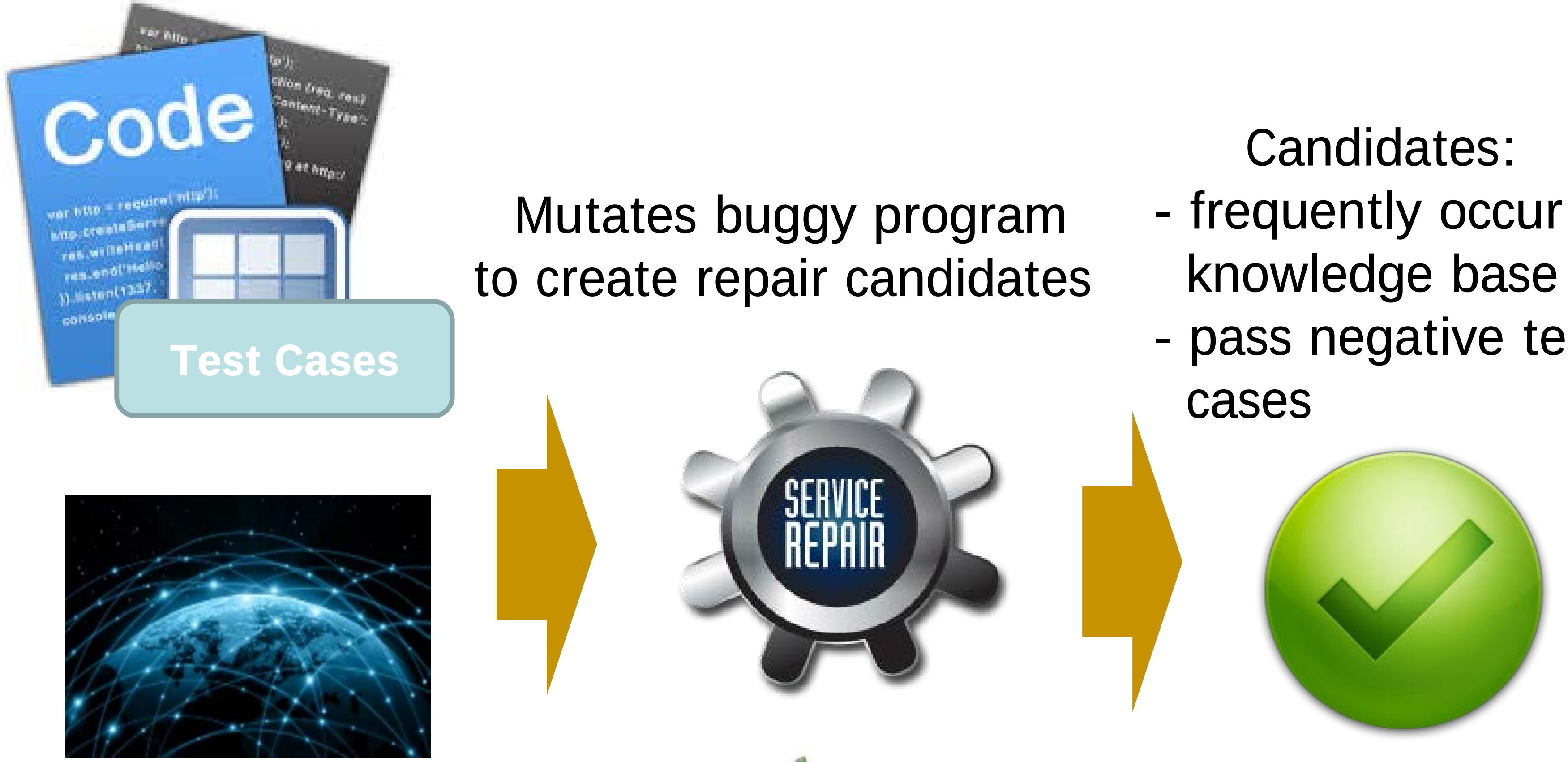


Mutates buggy program to create repair candidates

Candidate passing all test cases

E.g., GenProg, PAR, etc

History Driven Repair (Le et al., SANER’16)



Mutates buggy program to create repair candidates

Candidates:

- frequently occur in knowledge base
- pass negative test cases

Knowledge base: Learned bug fix behaviors from history

👍 Fast

👍 Avoid nonsensical patches

Automatic Repair

- Xuan-Bach D. Le, David Lo, and Claire Le Goues. History Driven Program Repair. 23rd IEEE International Conference on Software Analysis, Evolution, and Reengineering (SANER) 2016
- Xuan-Bach D. Le, Tien-Duy B. Le, David Lo: Should fixing these failures be delegated to automated program repair? ISSRE 2015: 427-437
- Siqi Ma, David Lo, Teng Li, and Robert H. Deng: CDRep: Automatic Repair of Cryptographic-Misuses in Android Applications. AsiaCCS 2016
  Chen Liu, Jinqiu Yang, Lin Tan, Munawar Hafiz: R2Fix: Automatically Generating Bug Fixes from Bug Reports. ICST 2013: 282-291
- Sergey Mechtaev, Jooyong Yi, Abhik Roychoudhury: DirectFix: Looking for Simple Program Repairs. ICSE (1) 2015: 448-458
- Shin Hwei Tan, Abhik Roychoudhury: relifix: Automated Repair of Software Regressions. ICSE (1) 2015: 471-482
- Fan Long, Martin Rinard: Staged program repair with condition synthesis. ESEC/SIGSOFT FSE 2015: 166-178

Future Opportunities on Bug Report Analytics

- Achieve higher accuracy
  - Technical innovation
  - Additional data sources
- AI-Human interaction
  - Incorporating incremental user feedback
- Tool support
  - Integration with standard IDEs/bug trackers
- Field study
  - Deploying bug report analytics techniques live and get feedback

Thank you!

Questions? Comments?

davidlo@smu.edu.sg