

REMI: Defect Prediction for Efficient API Testing

ESEC/FSE 2015, Industrial track

September 3, 2015

Mijung Kim*, Jaechang Nam*, Jaehyuk Yeon⁺, Soonhwang Choi⁺, and Sunghun Kim*

*Department of Computer Science and Engineering, HKUST

⁺Software R&D Center, Samsung Electronics CO., LTD

Motivation

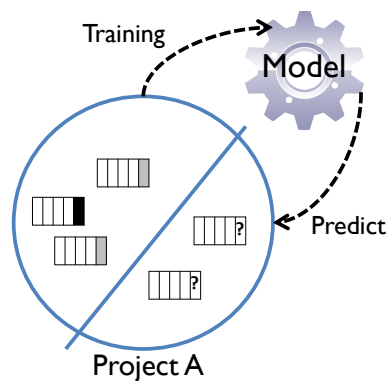
- Cost-intensive software quality assurance (QA) tasks at Samsung
 - Creating test cases for APIs
 - Testing APIs
- How to prioritize risky APIs for efficient API testing?

Goal

- Apply software defect prediction for the efficient API testing.

3

Software Defect Prediction



▬: Metric value

▬▬▬▬▬: Buggy-labeled instance

▬▬▬▬▬: Clean-labeled instance

▬▬▬▬?: Unlabeled instance

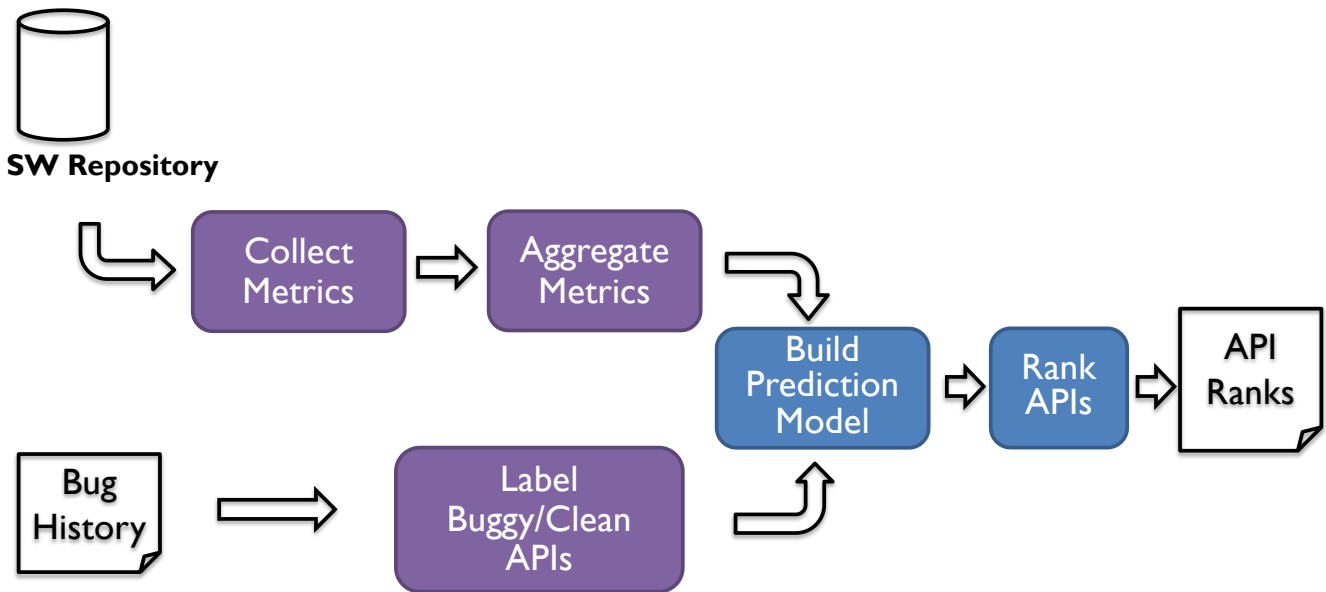
Related Work

Munson@TSE'92, Basili@TSE'95, Menzies@TSE'07,
Hassan@ICSE'09, Bird@FSE'11, D'ambros@EMSE'12
Lee@FSE'11, ...

4

Approach

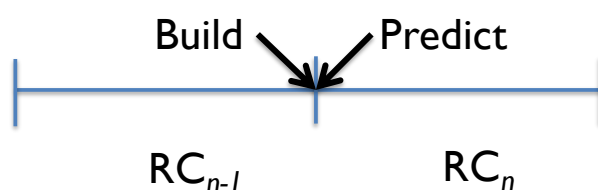
REMI: Risk Evaluation Method for Interface testing



5

Experimental Setup

- Random Forest
- Subject
 - Tizen-wearable
 - Applied REMI for 36 functional packages with about 1100 APIs
 - Release Candidates (RC)
 - RC2 to RC4



With the prediction results, perform more API test activities for the defect-prone APIs.

6

Research Questions

- RQ1
 - How accurately can REMI predict buggy APIs?
- RQ2
 - How useful is REMI for API testing in the actual API development process?

7

RESULT

8

Representative Prediction Results

(RC1 → RC2)

Packages	Depth 0			Depth All		
	Precision	Recall	F-measure	Precision	Recall	F-measure
Package 1	1.000	0.968	0.984	1.000	0.935	0.967
Package 2	0.667	0.154	0.250	0.600	0.462	0.522
Average	0.834	0.561	0.671	0.800	0.699	0.745

9

Results for Test Development Phase

Version	REMI	Resources	Bug Detection Ability		
		Man-Day	API	Test Cases	Bugs Detected
RC2	w/o REMI	7 (M)	70	70	2
	w/ REMI	19.7 (N)	158	158	2
RC3	w/o REMI	4.7 (M)	47	47	0
	w/ REMI	3.25 (N)	26	26	2

M: Modify test cases

N: Create new test cases ← Additional test activity after REMI

10

Results for Test Execution Phase

Version	REMI	Resources	Bug Detection Ability		
		Man-Hour	Test Run	Defected Bugs	Detection Rate
RC2	w/o REMI	2.18	873	6.5	0.74%
	w/ REMI	2.18	873	18	2.06%
RC3	w/o REMI	2.11	845	8.1	0.96%
	w/ REMI	2.11	845	9	1.07%

11

Lessons Learned

- “**The list of risky APIs** provided before conducting QA activities is **helpful** for testers to allocate their testing effort efficiently, especially **with tight time constraints.**”
- “In the process of applying REMI, **overheads** arise during the tool configuration and executions (approximately 1 to 1.5 hours).”
- “It is difficult to collect the **bug information** to label buggy/clean APIs without noise.”

12

Conclusion

- REMI
 - Efficiently manage limited resources for API testing
 - Could identify additional defects by developing new test cases for risky APIs.
- Future work
 - Apply other software projects including open-source API development.

13

Q&A

THANK YOU!

14