

A Decision Support Platform for Guiding a Bug Triager for Resolver Recommendation Using Textual and Non-Textual Features

Ashish Sureka, Himanshu kumar Singh, Manjunath Bagewadi,
Abhishek Mitra, Rohit Karanth

Siemens Corporate Research and Technology, India

*ashish@in.abb.com, himanshu.singh@siemens.com, manjunathsept11@gmail.com,
abhishek1447@iiitd.ac.in, rohit.karanth@siemens.com*

December 1, 2015

Table of Contents

1 Introduction

- Bug Triaging at Siemens
- The architectural design

2 Solution

- Bug repository Data
- Bug description pre-processing
- Ensemble Classifier
- Initial results
- Confusion Matrix and Accuracy Results

3 Overview of the Tool

- Layouts
- Visualizations

4 References

Table of Contents

1 Introduction

- Bug Triaging at Siemens
- The architectural design

2 Solution

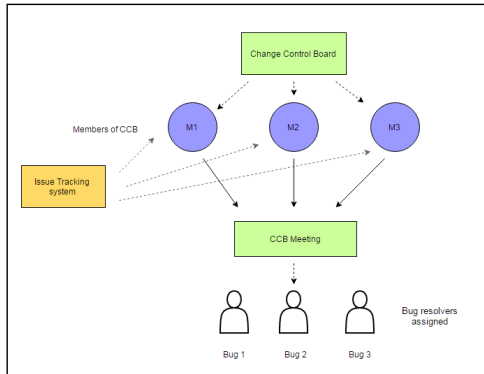
- Bug repository Data
- Bug description pre-processing
- Ensemble Classifier
- Initial results
- Confusion Matrix and Accuracy Results

3 Overview of the Tool

- Layouts
- Visualizations

4 References

Bug Triaging at Siemens



Average time needed to solve a bug

$$= \frac{T_{pre-CCB} * M + T_{CCB}}{N} \quad (1)$$

Challenges in the Bug Triaging process

- ▶ Decisions made by the CCB are knowledge intensive :
 - ▶ prior knowledge about the software system
 - ▶ expertise of the developer
 - ▶ team structure and composition
 - ▶ developer workload
- ▶ Bug resolver assignment is non-trivial in:
 - ▶ large and complex software settings
 - ▶ globally distributed teams
- ▶ Bug Triaging is a Time consuming process. Time equals Money

The architectural design

1

RECOMMENDATION

Who should Fix
this Bug



BUG REPORT

RANK	NAME
1	Abhishek
2	Ashish
3	Himanshu
4	Manjunath
5	Rohit

2

JUSTIFICATION



OLD BUG REPORT



Abhishek



NEW BUG REPORT



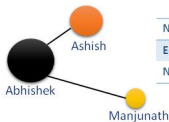
Similarity



TRIAGER & RESOLVER

3

NETWORK



Node Size	Number of Bugs
Edge Distance	Strength of Collaboration
Node Color	Role

4

WORKLOAD & EXPERIENCE

RANK	NAME	WORKLOAD	EXPERIENCE
1	Abhishek	4	12
2	Ashish	10	24
3	Himanshu	7	30
4	Manjunath	1	3
5	Rohit	8	2

Table of Contents

1 Introduction

- Bug Triaging at Siemens
- The architectural design

2 Solution

- Bug repository Data
- Bug description pre-processing
- Ensemble Classifier
- Initial results
- Confusion Matrix and Accuracy Results

3 Overview of the Tool

- Layouts
- Visualizations

4 References

The Data - Two projects

- ▶ Project A :
 - ▶ Image processing based product deployed in CT Scan Machines.
 - ▶ 2 year old and has 772 bugs reported till November 2014.
 - ▶ 345 of them have been solved and validated.
 - ▶ 78 members working.
- ▶ Project B
 - ▶ Its a Ultra-Sound Clinical Workflow Management System and relatively larger than A.
 - ▶ started in 2005 and 17267 bugs reported till October 2014.
 - ▶ 12438 bugs have been resolved.
 - ▶ A total of 253 professionals have worked on the project during the past 9 to 10 years.

Data Format

- ▶ Bug Details :
 - ▶ Bug ID.
 - ▶ Short description of the bug
 - ▶ Long description of the bug.
 - ▶ Component to which the bug might belong.
- ▶ Discussions: Starting from the moment a bug is reported to the time it is closed, all the discussions happening among the team members through emails are recorded.
 - ▶ We extracted the names of all team members belonging to the bug.
 - ▶ A bug can be in one of the 9 states. Submitted , Qualified, In-decision, Deferred , Terminated , In-observation, In-work , Solved , Validated. The Discussion contained the state transitions as well.

Bug description pre-processing

Bug ID	Component	Short description	Long Description
1234	GUI	The d3js social network graph does not refresh with new data	Issue with ABT screen. Whenever I click the Social network button , it should refresh the social network graph and the name of the new developers who have joined the team should be visible

STOP WORDS / SPECIAL CHARS
REMOVAL

Bug ID	Component	Short description	Long Description
1234	GUI	d3js social network graph refresh data	issue screen click Social network button refresh social network graph name developers joined team visible

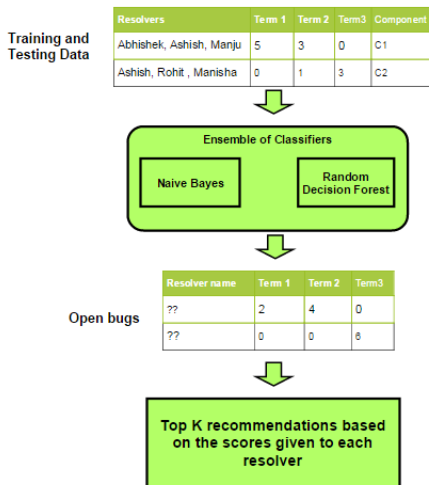
WORDS STEMMING USING PORTER
STEMMER (However Stemmer does not
modify identified keywords)

Bug ID	Component	Short description	Long Description
1234	GUI	d3js social network graph refresh data	issu screen click Social network button refresh social network graph name develop join team visibl

NOISE REMOVAL AND FREQUENCY
CALCULATION

d3js	social	network	graph	refresh	data	issu	screen	click	button	name	deve	lo	join	team	visibl
1	3	3	2	2	1	1	1	1	1	1	1		1	1	1

Ensemble Classifier



Initial Results and Steps taken for improvement

- ▶ The initial results were not very good.
- ▶ Name of the Triagers , test leads and Bug assigners kept on popping up as top recommendations
- ▶ We needed to extract the developers from the Discussion data
- ▶ We had a meeting with Change control board members from Project A and Project B
- ▶ In the discussion with them, We identified 43 valid state transitions, and out of them 22 of the transitions involved team members wearing a "developer's cap"

	Submitted	Qualified	In-Decision	Deferred	Terminated	In-Observation	In-Work	Solved	Validated
Submitted	TRG, TRG	TRG, TRG	TRG, TRG	TRG, TRG	DEV, TST	TRG, TRG	TRG, DEV	-	-
Qualified	-	DEV, DEV	TRG, TRG	TRG, TRG	TRG, TRG	TRG, TRG	TRG, DEV	-	-
In-Decision	-	-	DEV, DEV	DEV, TST	DEV, TST	DEV, TST	TRG, DEV	-	-
Deferred	-	-	TRG, TRG	TRG, TRG	TRG, TRG	TRG, TRG	TRG, TRG	-	-
Terminated	-	-	TST, TST	-	DEV, DEV	-	TST, DEV	-	-
In-Observation	-	-	TRG, TRG	TRG, TRG	DEV, TST	TST, TST	TST, DEV	-	-
In-Work	-	-	DEV, TST	DEV, TST	DEV, TST	DEV, TST	DEV, DEV	DEV, TST	-
Solved	-	-	-	-	-	-	TST, DEV	DEV, TST	TST, TST
Validated	-	-	-	-	-	-	TST, DEV	-	TST, TST

Precision and Recall

Recall@k , Precision@k and F-Measure@k for Project A

	K=1	K=2	K=3	K=4	K=5	K=6	K=7	K=8	K=9	K=10
RECALL	0.160	0.337	0.474	0.547	0.599	0.647	0.688	0.721	0.750	0.774
PRECISION	0.324	0.353	0.324	0.287	0.254	0.232	0.214	0.199	0.186	0.174
F-MEASURE	0.214	0.345	0.385	0.376	0.357	0.342	0.327	0.313	0.298	0.284

Recall@k , Precision@k and F-Measure@k for Project B

	K=1	K=2	K=3	K=4	K=5	K=6	K=7	K=8	K=9	K=10
RECALL	0.242	0.644	0.794	0.819	0.848	0.864	0.875	0.890	0.900	0.905
PRECISION	0.437	0.599	0.493	0.408	0.342	0.292	0.255	0.228	0.206	0.188
F-MEASURE	0.312	0.620	0.609	0.545	0.488	0.436	0.395	0.363	0.335	0.311

Table of Contents

1 Introduction

- Bug Triaging at Siemens
- The architectural design

2 Solution

- Bug repository Data
- Bug description pre-processing
- Ensemble Classifier
- Initial results
- Confusion Matrix and Accuracy Results

3 Overview of the Tool

- Layouts
- Visualizations

4 References

The Tool - Settings

step 1 : Resolvers

step 2 : Components

Step 3 : Training Duration

Step 4 : Include /Exclude

Step 5 : Train Model

All Components

Check the check box to remove the component from training model.

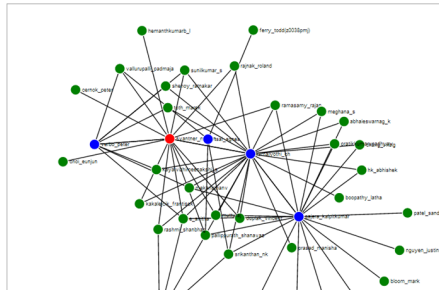
S.No	Component Name	Remove Component?
1	Atlas Valves	<input type="checkbox"/>
2	Atlas_LVA	<input type="checkbox"/>
3	Core CAP Host	<input type="checkbox"/>
4	CoreLVA	<input type="checkbox"/>
5	CT	<input type="checkbox"/>
6	CTS	<input type="checkbox"/>
7	DICOM	<input type="checkbox"/>
8	DICOM Integration	<input type="checkbox"/>

The Tool - Recommendations

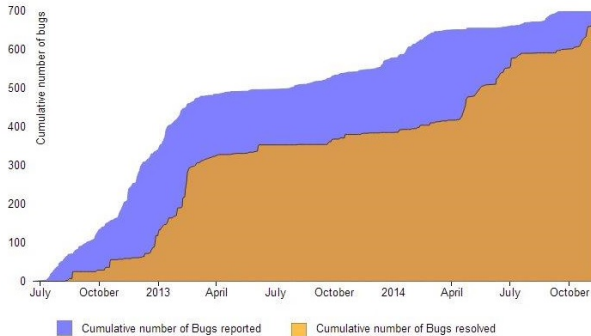
Top Resolvers

Rank	Resolver Name	Confidence Value	Bugs Solved	Current Load	Filter Social network
1	jashin02-memon	22.52	10	0	<input checked="" type="checkbox"/>
2	valish_ghosh	11.72	6	0	<input checked="" type="checkbox"/>
3	gaurav_bansal	10.54	0	0	<input checked="" type="checkbox"/>
4	shankar_gang	8.14	0	0	<input checked="" type="checkbox"/>
5	vishal_gupta	7.64	2	0	<input checked="" type="checkbox"/>
6	gajendra-helphikumar	7.32	39	0	<input checked="" type="checkbox"/>
7	mohammed_ajwad	5.11	2	0	<input checked="" type="checkbox"/>
8	nitin_garg	2.47	2	0	<input checked="" type="checkbox"/>
9	niraj_gupta	1.8	42	1	<input checked="" type="checkbox"/>
10	hitesh_joshi	1.67	42	1	<input checked="" type="checkbox"/>

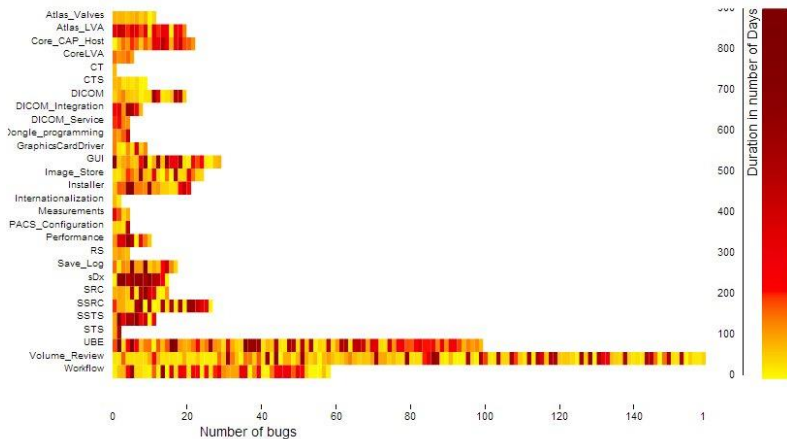
Filter Social Network?



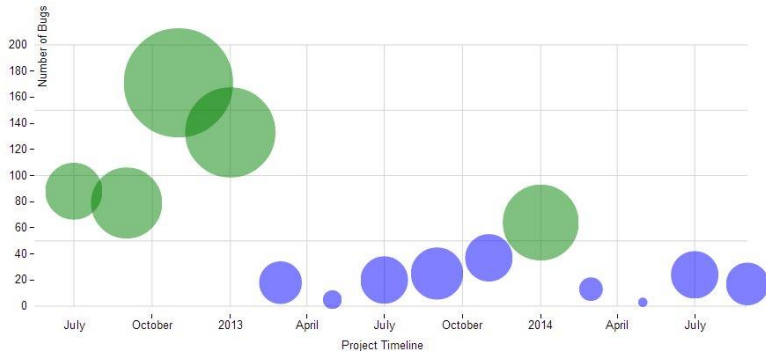
Visualizations - Area Plot



Visualizations - Heat Map

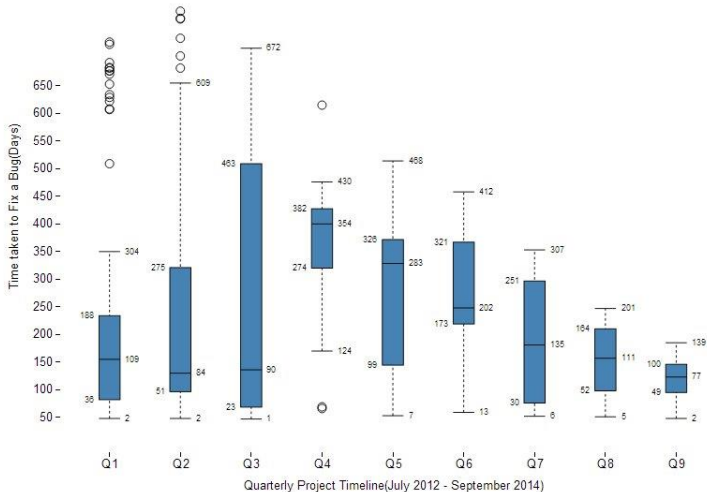


Visualizations - Bubble Chart



- Size of the circle denotes component diversity(More the size, more the components affected)
- More than average developers needed to fix the bug during the Three month period
- Less than average developers needed to fix the bug during the Three month period

Visualizations - Box Plot



Visualizations - Correlation Matrix

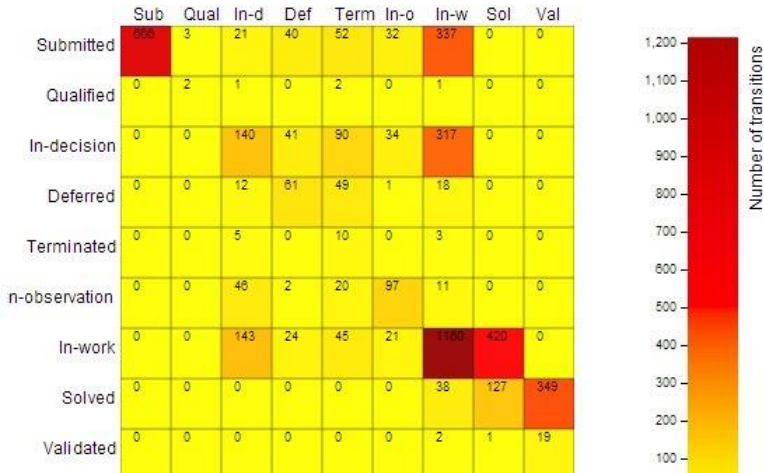


Table of Contents

1 Introduction

- Bug Triaging at Siemens
- The architectural design

2 Solution

- Bug repository Data
- Bug description pre-processing
- Ensemble Classifier
- Initial results
- Confusion Matrix and Accuracy Results

3 Overview of the Tool

- Layouts
- Visualizations

4 References

References I



Gerald Bortis and André van der Hoek.

Porchlight: A tag-based approach to bug triaging.

In *Proceedings of the 2013 International Conference on Software Engineering, ICSE '13*, pages 342–351, 2013.



Xin Xia, D. Lo, Xinyu Wang, and Bo Zhou.

Accurate developer recommendation for bug resolution.

In *Reverse Engineering (WCRE), 2013 20th Working Conference on*, pages 72–81, 2013.



Xihao Xie, Wen Zhang, Ye Yang, and Qing Wang.

Dretom: Developer recommendation based on topic models for bug resolution.

In *Proceedings of the 8th International Conference on Predictive Models in Software Engineering, PROMISE '12*, pages 19–28, 2012.



Wenjin Wu, Wen Zhang, Ye Yang, and Qing Wang.

Drex: Developer recommendation with k-nearest-neighbor search and expertise ranking.

In *Software Engineering Conference (APSEC), 2011 18th Asia Pacific*, pages 389–396, 2011.

References II

-  Ahmed Tamrawi, Tung Thanh Nguyen, Jafar M. Al-Kofahi, and Tien N. Nguyen. Fuzzy set and cache-based approach for bug triaging. In *Proceedings of the 19th ACM SIGSOFT Symposium and the 13th European Conference on Foundations of Software Engineering*, pages 365–375, 2011.
-  Chiara Francalanci and Francesco Merlo. Empirical analysis of the bug fixing process in open source projects. In *Open Source Development, Communities and Quality*, pages 187–196, 2008.
-  Emad Shihab, Akinori Ihara, Yasutaka Kamei, WalidM. Ibrahim, Masao Ohira, Bram Adams, AhmedE. Hassan, and Ken-ichi Matsumoto. Studying re-opened bugs in open source software. In *Empirical Software Engineering*, pages 1005–1042, 2013.
-  S. Lal and A. Sureka. Comparison of seven bug report types: A case-study of google chrome browser project. In *Software Engineering Conference (APSEC), 2012 19th Asia-Pacific*, volume 1, pages 517–526, Dec 2012.

References III



Ashish Sureka, Atul Goyal, and Ayushi Rastogi.

Using social network analysis for mining collaboration data in a defect tracking system for risk and vulnerability analysis.

In *Proceedings of the 4th India Software Engineering Conference, ISEC '11*, pages 195–204, New York, NY, USA, 2011. ACM.

Questions ?