Thesis Defense Presentation

Towards an Empirical Model to Identify When Bugs are Introduced …

Gema Rodríguez Pérez

November 21, 2018

Universidad
Rey Juan Carlos

Overview

1. Background
2. Systematic Literature Review
3. Model to Identify Changes that Introduced Bugs
4. Empirical Evaluation of the Model
5. Findings
6. Implications and Recommendations

Overview

1. **Background**
2. Systematic Literature Review
3. Model to Identify Changes that Introduced Bugs
4. Empirical Evaluation of the Model
5. Findings
6. Implications and Recommendations

Understanding the problem

Importance of Bugs

- From 1 to 25 bugs for every 1000 lines inserted.
- Software bugs costs almost $60 billions to the US.

Bug introduction

- It is focuses on identifying which changes introduced bugs.
- In ESEM, it is based on the study of previous bugs.

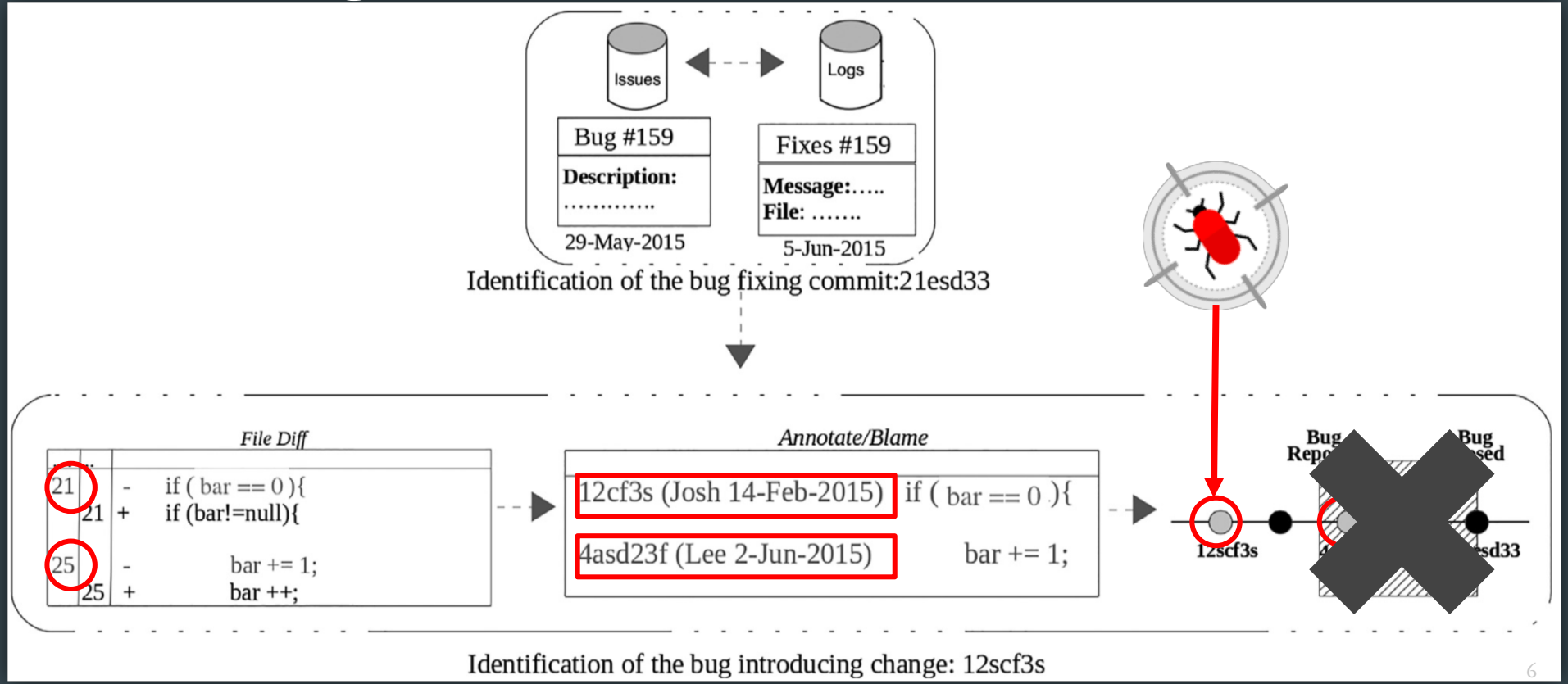
The impact

- Measuring developer performance.
- Measure bug residency time.
- Find bug-prone change patterns.

The Current Assumption:

“A given bug was introduced by the lines of code that were modified to fix it”

The SZZ Algorithm :



Is the assumption fulfilled in these scenarios?

Hyper-V driver failing with dynamic memory due to virtual NUMA

Bug #1305897 reported by Luis Fernandez on 2014-04-10

This bug affects 1 person

6

Affects	Status	Importance	Assigned to	Milestone
OpenStack Compute (nova)	Fix Released	Medium	Alessandro Pilotti	OpenStack Compute (nova) 2015.1.0 "kilo"
Juno	Fix Released	Undecided	Unassigned	OpenStack Compute (nova) 2014.2.4

Also affects project | Also affects distribution/package | Nominate for series

Bug Description

```
Starting with Windows Server 2012, Hyper-V provides the Virtual NUMA
functionality. This option is enabled by default in the VMs depending on
the underlying hardware.

However, it's not compatible with dynamic memory. The Hyper-V driver is
not aware of this constraint and it's not possible to boot new VMs if the
nova.conf parameter 'dynamic_memory_ratio' > 1.
```

Statement I:

“The fact of introducing a bug depends on the definition of bug, and the future work should verify whether the introduction of bug meets a given definition of bug” [Kim et al., 2006]

Statement II:

“It is necessary to develop new theories and mathematical models to increase understanding of software evolution, and to invest in research that tries to bridge the gap between the what of software evolution and the how of software evolution” [Mens et al., 2005]

Thesis' Aim :

To develop a theory that:

1) defines **what is a bug** and,

2) how to identify when this bug was inserted in a software product.

Thesis' Contributions :

1. A SLR on the use of the SZZ algorithm.
2. A quantification of the SZZ algorithm.
3. A theoretical model for identifying bugs-introducing changes.
4. An empirical study to validate the theoretical model.

Overview

1. Background
2. **Systematic Literature Review**
3. Model to Identify Changes that Introduced Bugs
4. Empirical Evaluation of the Model
5. Findings
6. Implications and Recommendations

Reproducibility and Credibility in Empirical Software Engineering:

A case study based on a systematic literature review of the use of the SZZ algorithm:

1. What is the impact of the SZZ algorithm in academia?
2. Are studies that use the SZZ reproducible?
3. Do the publication mention the limitations of SZZ?

Inclusion Criteria :

All published studies written in English that cite either:

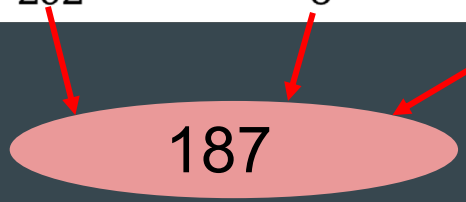
1. The original SZZ publication: “**When do changes induce fixes?**”, or
2. one of the two publications with improved versions:

   “**Automatic Identification of Bug-Introducing Changes**”,

   “**SZZ Revisited: Verifying When Changes Induce Fixes**” .

Exclusion Criteria :

Selection process	#SZZ	#SZZ-1	#SZZ-2
Papers extracted from the databases	788	241	41
Sift based on false alarms	29 removed	10 removed	2 removed
Sift based on not available/English writing	40 removed	4 removed	0 removed
Sift based on duplicates	308 removed	187 removed	32 removed
Full papers considered for review	411	40	7
Removed after reading	149 removed	32 removed	4 removed
Papers accepted to the review	262	8	3



Extracting Data from Papers:

- **Purpose** and **outcome** of the study
- **Venue** and class of publication
- Whether it has a **replication package**
- Whether it has a **detailed description** of the methods and data used
- Whether the **limitations** of the algorithm are mentioned
- Whether the authors use **the improved versions** of the SZZ algorithm.

How is the impact of the SZZ algorithm?

High diversity

Type	# Different	# Publications
Journals	21	42
Conferences & Symposiums	40	102
Workshops	13	13
University theses	20	30

How is the impact of the SZZ algorithm?

High maturity

Type	Name	Rating	# Papers (%)
C	Conf Mining Softw Repositories (MSR)	Class 2 - CORE A	15 (8)
C	Intl Conf Software Eng (ICSE)	Class 1 - CORE A*	12 (6)
C	Intl Conf Soft Maintenance (ICSME)	Class 2 - CORE A	10 (5)
J	Empirical Software Eng (EmSE)	JCR Q1	9 (5)
J	Transactions on Software Eng (TSE)	JCR Q1	9 (5)
C	Intl Symp Emp Soft Eng & Measurement (ESEM)	Class 2 - CORE A	8 (4)
C	Intl Conf Automated Softw Eng (ASE)	Class 2 - CORE A	7 (4)
C	Symp Foundations of Software Eng (FSE)	Class 1 - CORE A*	6 (3)

How is the impact of the SZZ algorithm?

High impact

Purpose
Bug Prediction
Bug Proneness
Bug Detection
Bug Localization

Outcome
Approach / Tools
Empirical Study
Replications
Metrics
DataSets

Are studies that use the SZZ reproducible?

	Package Only	Environment Only	Both	None
YES	19	72	24	72
NO	168	96	163	115

Do the publication mention the limitations of SZZ?

Part	Type
First part	Incomplete mapping [SLR[14]]
	Inaccurate mapping [SLR[15]]
	Systematic bias [SLR[14]]
Second part	Cosmetic changes, comments, blank lines [SLR[16]].
	Added lines in fixing commits [SLR[8]]
	Long fixing commits [SLR[8]]
	Semantic level is weak [SLR[11]]
	Correct changes at the time of being committed [SLR[8]]
	Commit Squashing [17]

Do the publication mention the limitations of SZZ?

	NO-TTV	TTV-1st part only	TTV-2nd part only	Complete TTV
YES	94	44	10	39
NO	93	143	177	148

TTV = Threats to Validity

Drawbacks

Functional

- Many possible BICs
- Only new additions
- Multiple modifications of a line
- Weak semantic level
- Dormant bugs

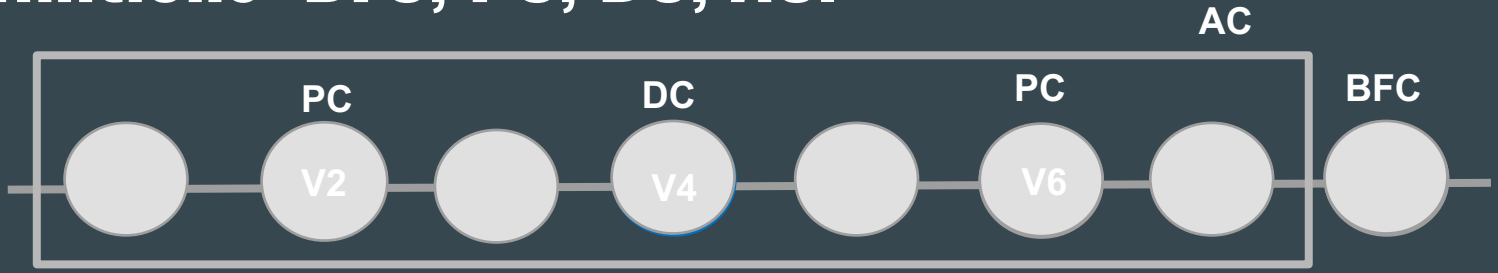
Conceptual

- Changing environment
- Compatibility problems

Overview

1. Background
2. Systematic Literature Review
3. **Model to Identify Changes that Introduced Bugs**
4. Empirical Evaluation of the Model
5. Findings
6. Implications and Recommendations

Definitions: BFC, PC, DC, AC.



V2

```
20  void foo(){
21      if(bar==0){
22          qux=5/bar
23      }
24  }
```

Previous Commit

V4

```
20  void foo(){
21      if(bar==0){
22          qux=5/bar
23      }
24      else
        {
25          bar +=1
26      }
27  }
```

Descendant Commit

V6

```
20  void foo(){
21      if(bar==0){
22          qux=5/bar
23      }
24      else{
25          bar +=1
26      }
27  }
```

Previous Commit

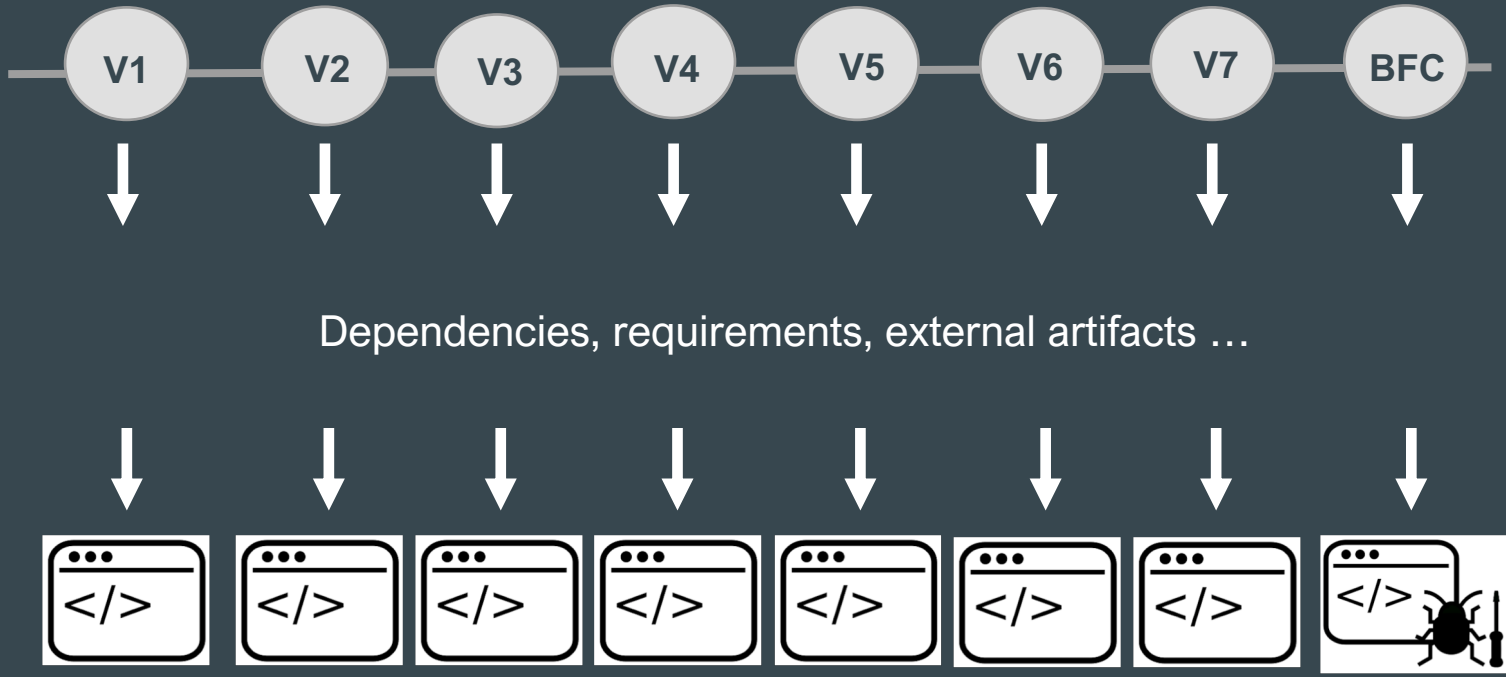
Bug Fixing Commit

```
20  void foo(){
21      if(bar!=0){
22          qux=5/bar
23      }
24      else{
25          bar ++1
26      }
27  }
```

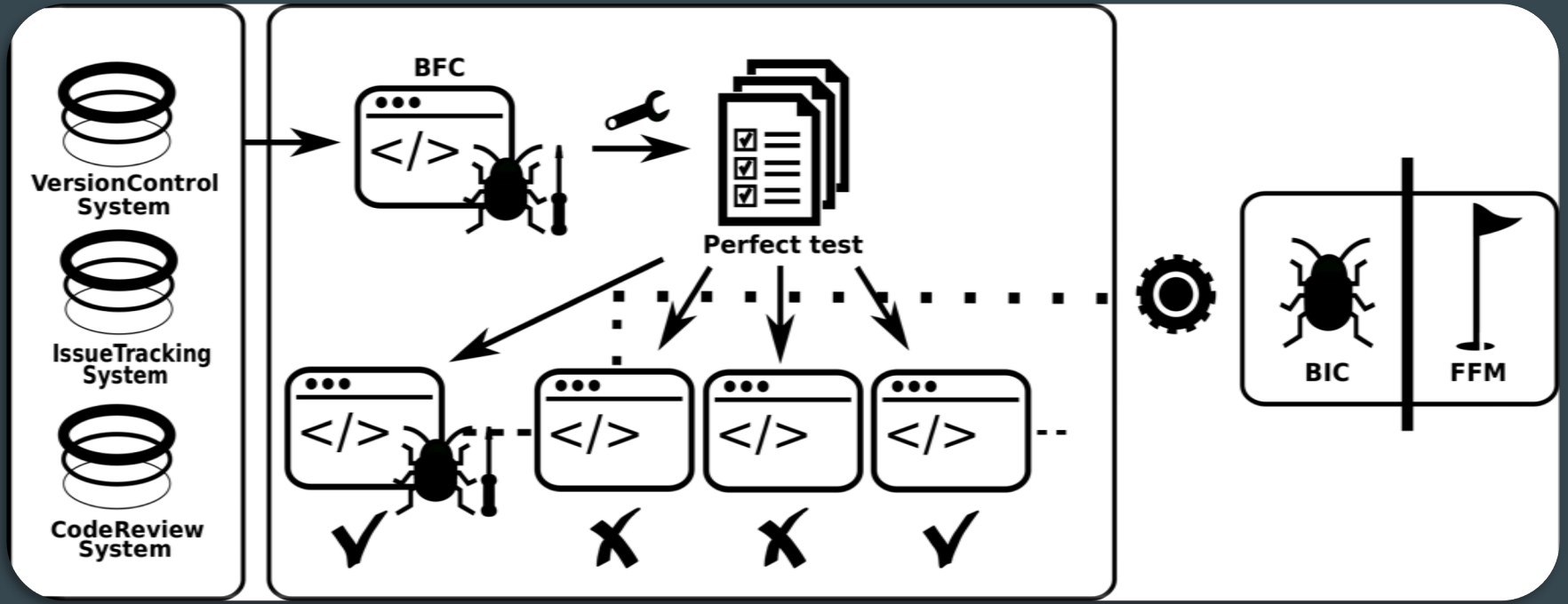
Definitions : Test Signaling a Bug (TSB)

- TSB is a hypothetical test that could be run on any snapshot.
- TSB is a perfect test with 100% of coverage.
- TSB returns TRUE ⟶ No bug in the snapshot
- TSB returns FALSE ⟶ Bug in the snapshot

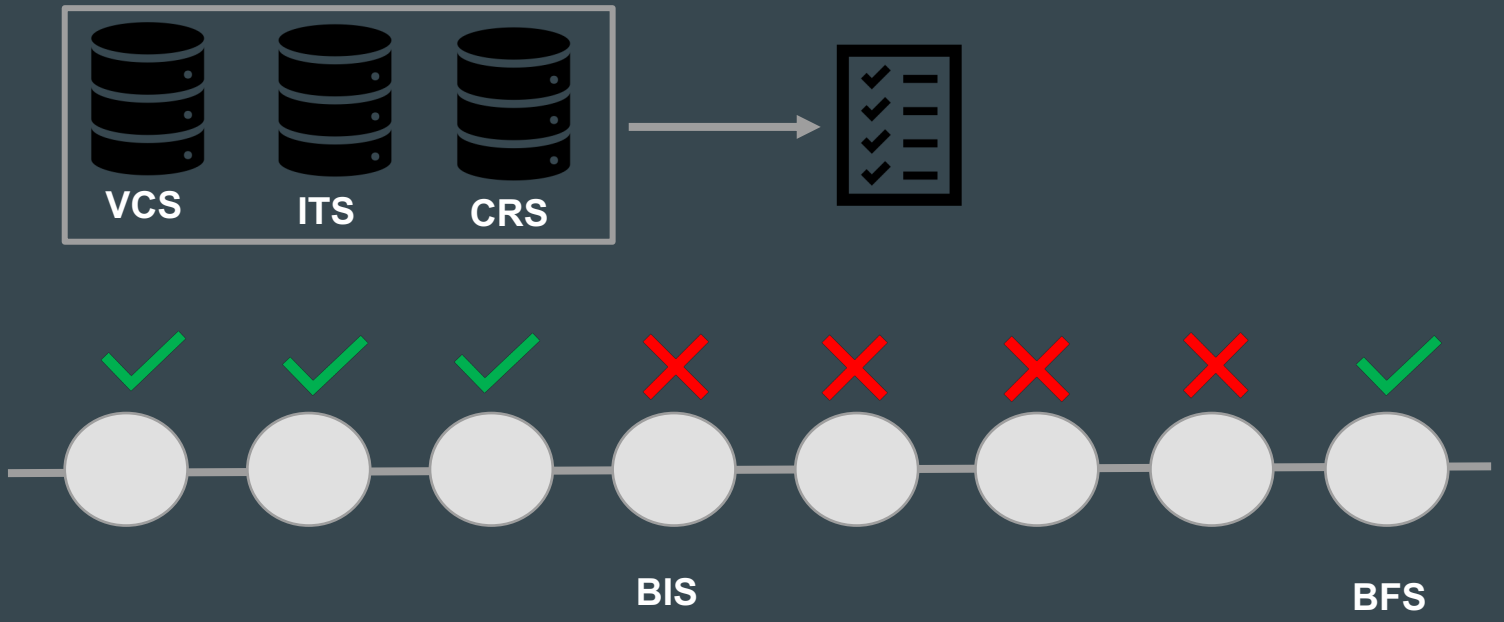
Definitions : Snapshot



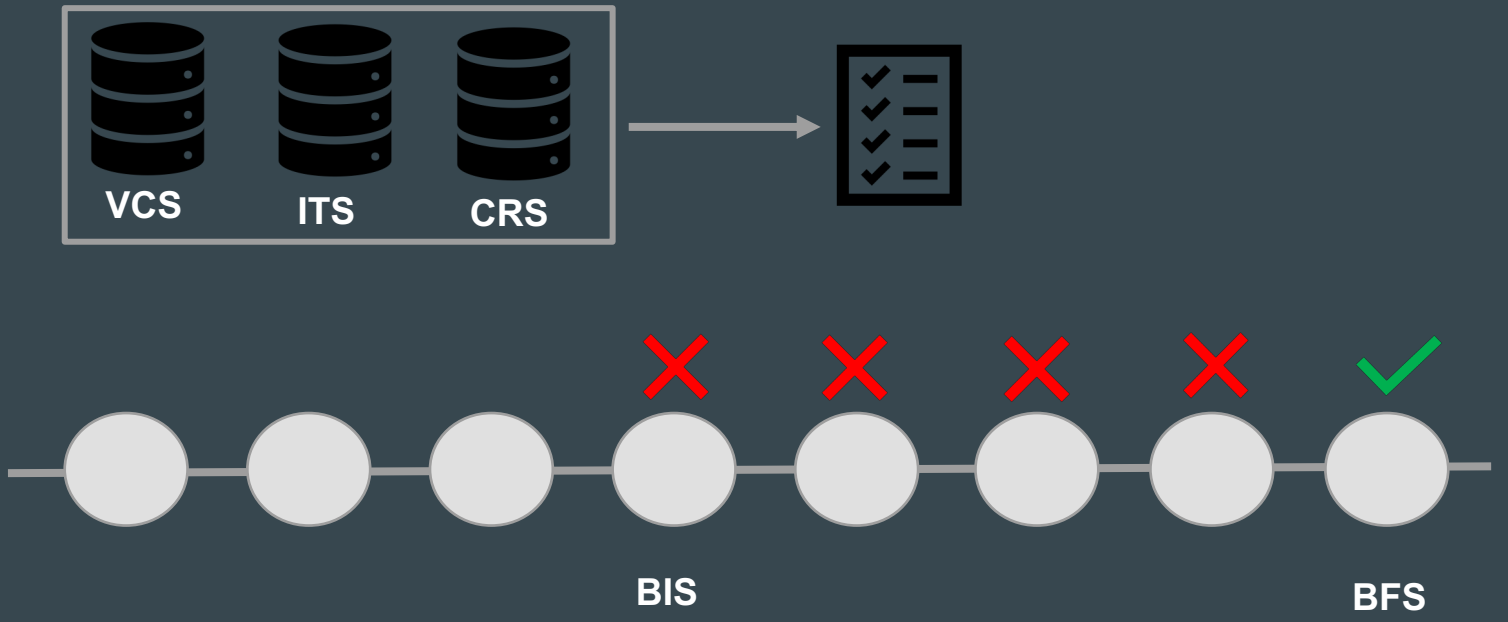
The Empirical Model



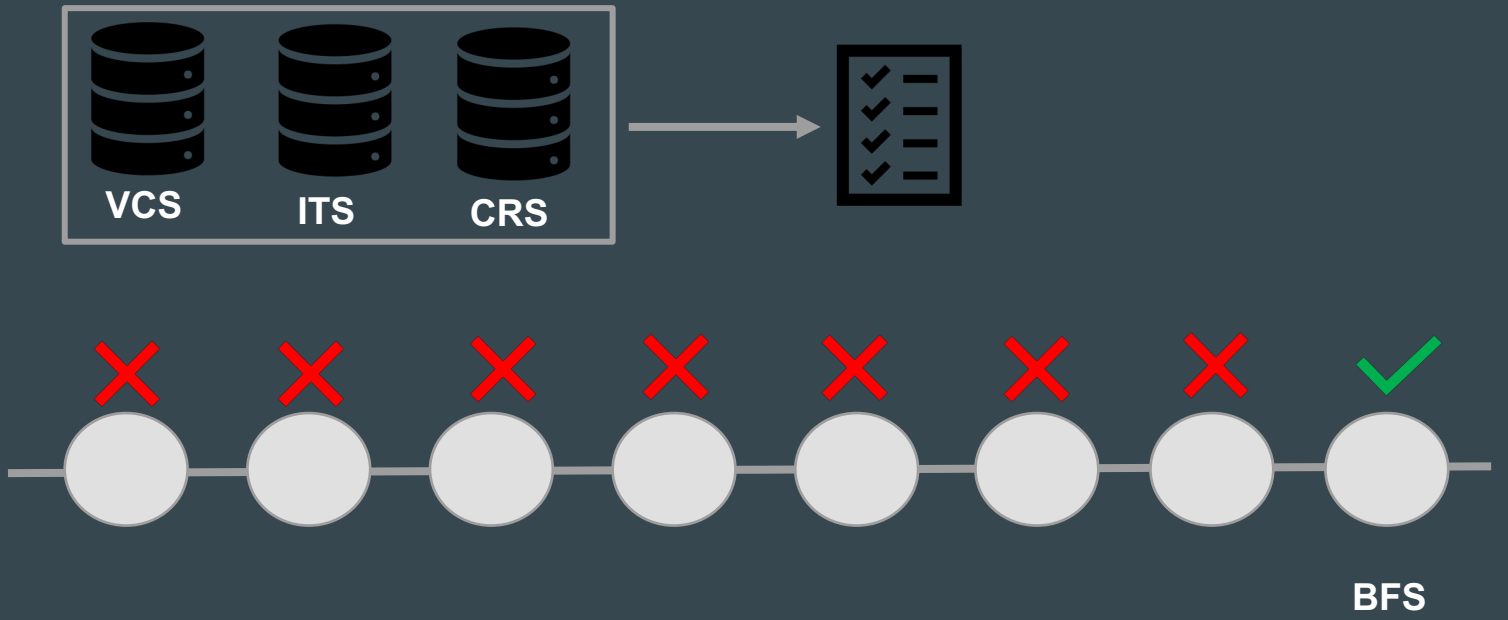
Test Signaling a Bug (TSB) and the BIS



Test Signaling a Bug not runnable



Test Signaling a Bug always fails



Overview

1. Background
2. Systematic Literature Review
3. Model to Identify Changes that Introduced Bugs
4. **Empirical Evaluation of the Model**
5. Findings
6. Implications and Recommendations

Case Studies

NOVA	
216	Companies
32470	Commits
1602	Contributors
4581836	LOCs
1930	Resolved bugs

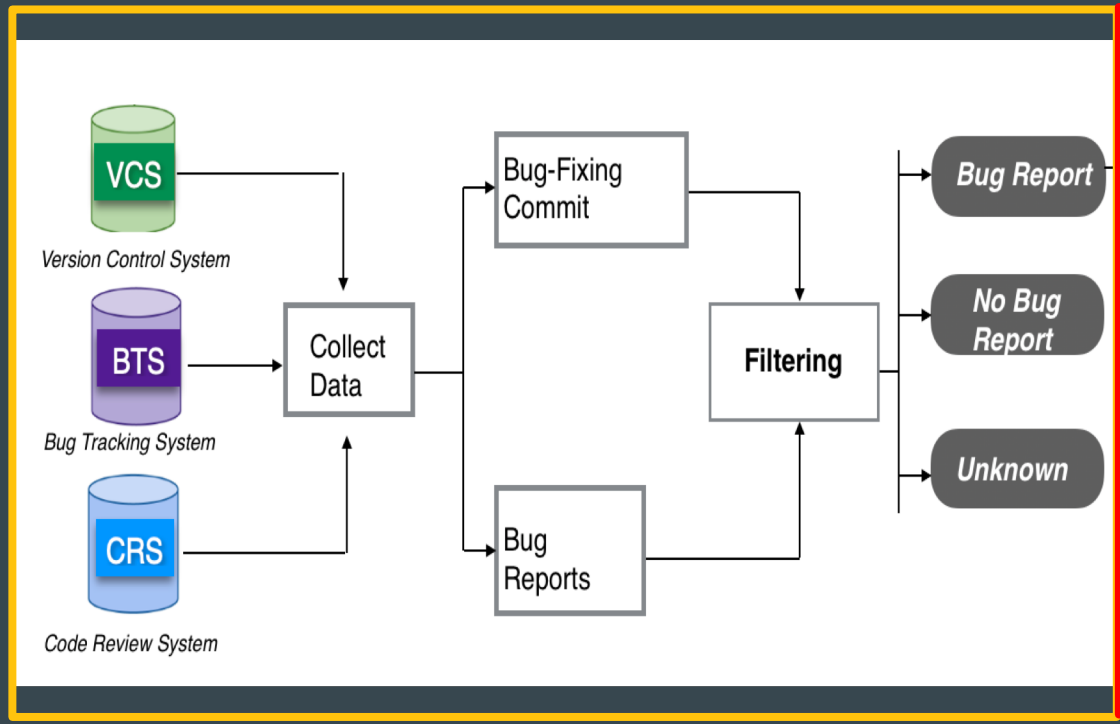
ELASTICSEARCH	
39402	Commits
1032	Contributors
1187732	LOCs
4958	Resolved bugs

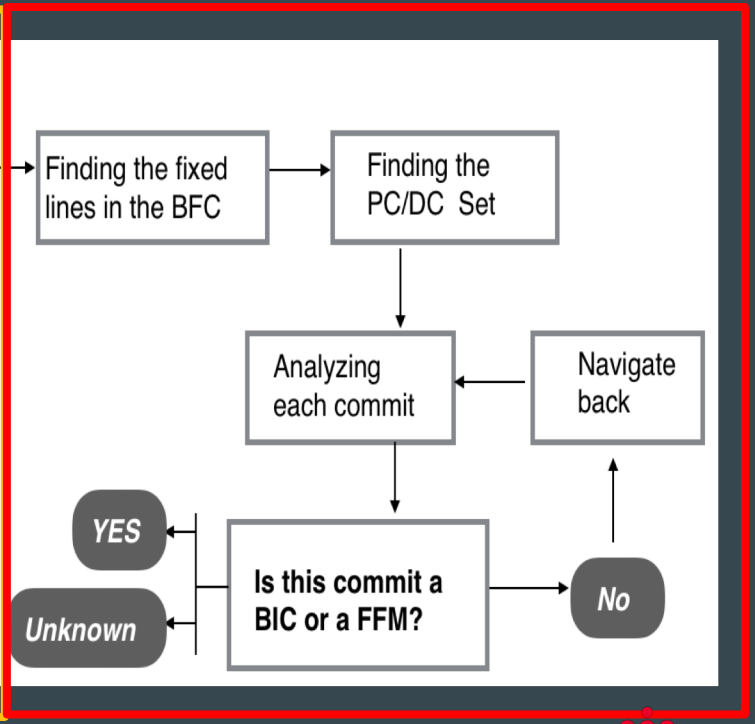
Research questions:

- RQ1: What is the frequency for a BFC being caused by a BIC?
  - RQ1.1: Which reasons can explain that a BFC is not caused by a BIC?
  - RQ1.2: Could the location of a bug be modeled on the BIC and the FFM?

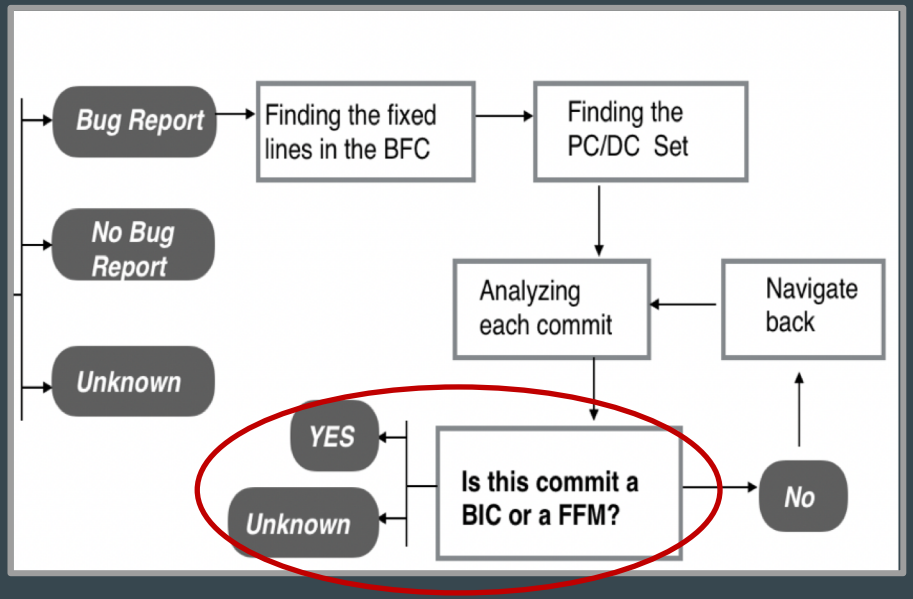
- RQ2: What are the specifications that define the effectiveness of an algorithm used to locate the origin of a bug?
  - RQ2: Which reasons caused that a previous commit was not the BIC?

Methodology





Methodology



GOLDEN SET

1) SZZ algorithm

2) SZZ-1 algorithm

3) SZZ-1E algorithm

Precision **Recall** **F1- Score**

First Step: Filtering

nova list's --tenant flag also requires --all-tenants

Bug #1185290 reported by Aarti Kriplani on 2013-05-29

Rafi Khardalian (rkhardalian) wrote on 2013-06-10:

I disagree Chris, mainly because the structure of the requests and code path should largely be transparent to the user. I'd suggest that specifying --tenant should imply you're doing a query across --all-tenants unless the --tenant specified is the same as what's contained in OS_TENANT_NAME (the unless part is debatable).

a given tenant, it works only when the --all-tenant flag is passed.
For e.g. nova list --tenant 123456 --all-tenants 1
If you leave off the last "--all-tenants 1" flag, you get back an empty response body (with a 200 response)
The "--all-tenants 1" should not be required for the "--tenant" flag to function properly.

Second Step: Identifying the BIC and the FFM

- Finding the lines that fixed the bug:
  - Finding the BFC
  - Finding the lines changed in the BFC
  - Filter out lines that are not code.

Identifying the BIC and the FFM:

- Determine the previous commit
  - Lines modified or removed → Previous commits
  - Lines added → Analize surroundings commits

Identifying the BIC and the FFM:

- Analyze those previous commits to determine the FFM and BIC:
  - The commit inserted the bug ⟶ BIC
  - The commit didn’t insert the bug ⟶ FFM

Overview

1. Background
2. Systematic Literature Review
3. Model to Identify Changes that Introduced Bugs
4. Empirical Evaluation of the Model
5. **Findings**
6. Implications and Recommendations

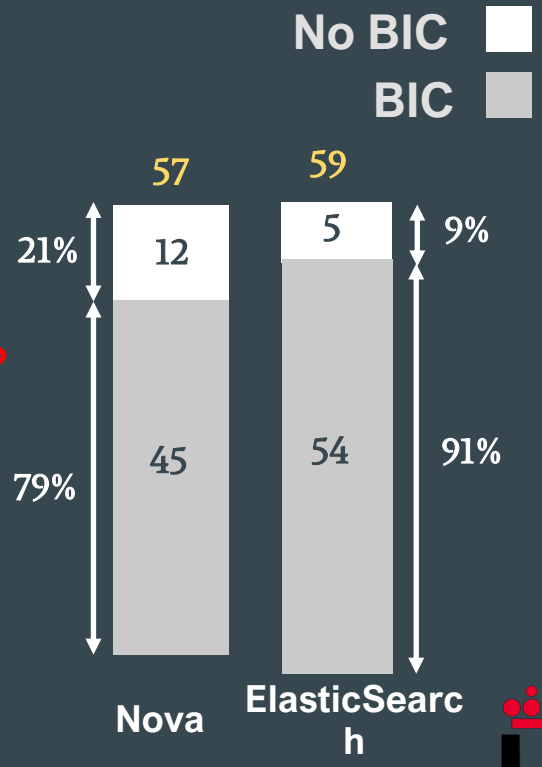
Filtering Results

	Bugs	Other Issues
Nova	57	3
ElasticSearch	59	1
Total	116	4

RQ1 : What is the frequency for a BFC being caused by a BIC?

Reasons of No BIC

	Nova	ElasticSearch
Co-evolution Internal	5 (42%)	2 (40%)
Co-evolution External	2 (17%)	1 (20%)
Compatibility	1 (8%)	0 (0%)
Bug in External API	4 (33%)	2 (40%)



RQ2: What are the specifications that define the effectiveness of an algorithm used to locate the origin of a bug?

	Precision	Recall	F1-Score
Nova SZZ	0.32	0.60	0.42
Nova SZZ-1	0.35	0.58	0.44
Nova SZZ-1E	0.64	0.60	0.66
ElasticSearch SZZ	0.31	0.68	0.43
ElasticSearch SZZ-1	0.32	0.71	0.44
ElasticSearch SZZ-1E	0.42	0.42	0.43

RQ2: Which reasons caused that a previous commit was not the BIC?

1. Variable renaming
2. Equivalent change
3. API changes
4. Obsolete code
5. Refactoring of the BFC

Overview

1. Background
2. Systematic Literature Review
3. Model to Identify Changes that Introduced Bugs
4. Empirical Evaluation of the Model
5. Findings
6. **Implications and Recommendations**

Implications and Recommendations

- Most of the publications are not reporting the limitations of current algorithms to identify bug-introducing commits.
- Studies must be aware of the risk of every assumption used.
- The reproducibility of the studies is discovery limited.
- The correct identification of the origin of the bug can help to improve many areas in SE (bug detection, bug prediction, automatic fix generation …)

Implications and Recommendations

- Bugs are not always introduced in the source code, and this phenomenon should be further investigated.
- A bug has to be contextualized to understand when and how it was inserted.
- The model provides a clear condition to determine if a given algorithm for identifying the change introducing a bug is correct or not when performing the identification.

Thesis' Publications :

1. **Bugtracking: A tool to assist in the identification of bug reports**.
   IFIP International Conference on Open Source Systems, 2016

2. **How much time did it take to notify a Bug? Two case studies: ElasticSearch and Nova.**
   IEEE/ACM 8th Workshop on Emerging Trends in Software Metrics (WETSoM), 2017

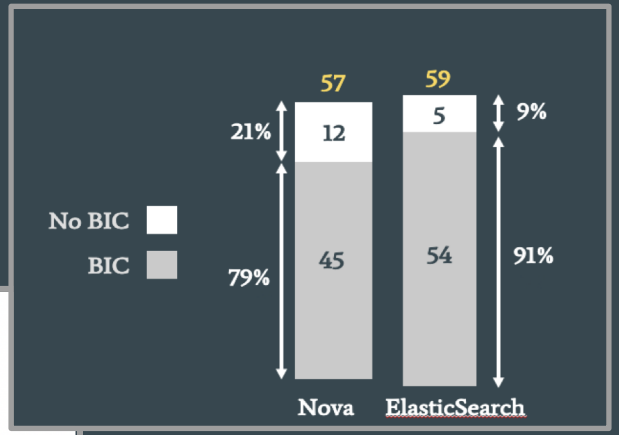
3. **Reproducibility and Credibility in ESE: A Case Study based on a Systematic Literature Review of the use of the SZZ algorithm**.
   Information and Software Technology, 2018.

4. **What if a bug has a Different Origin? Making Sense of Bugs Without an Explicit Bug Introducing Change**.
   12th International Symposium on Empirical Software Engineering and Measurement, 2018

Summary

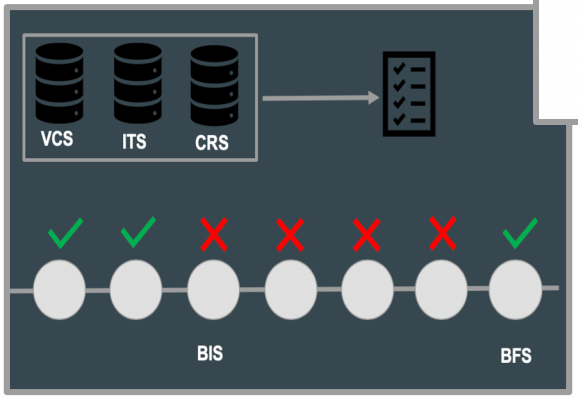
The Current Assumption:

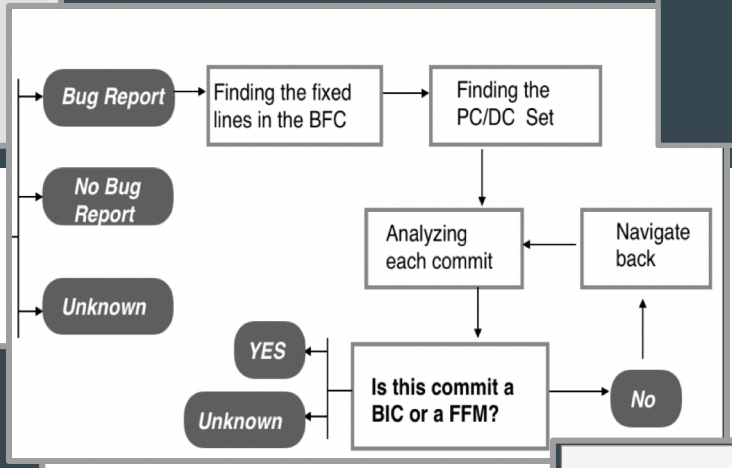
"A given Bug was introduced by the lines of code that were modified to fix it"



The problem

Solution





Evaluation

Findings

Findings

	Precision	Recall	F1-Score
Nova SZZ	0.32	0.60	0.42
Nova SZZ-1	0.35	0.58	0.44
Nova SZZ-1E	0.64	0.60	0.66
ElasticSearch SZZ	0.31	0.68	0.43
ElasticSearch SZZ-1	0.32	0.71	0.44
ElasticSearch SZZ-1E	0.42	0.42	0.43