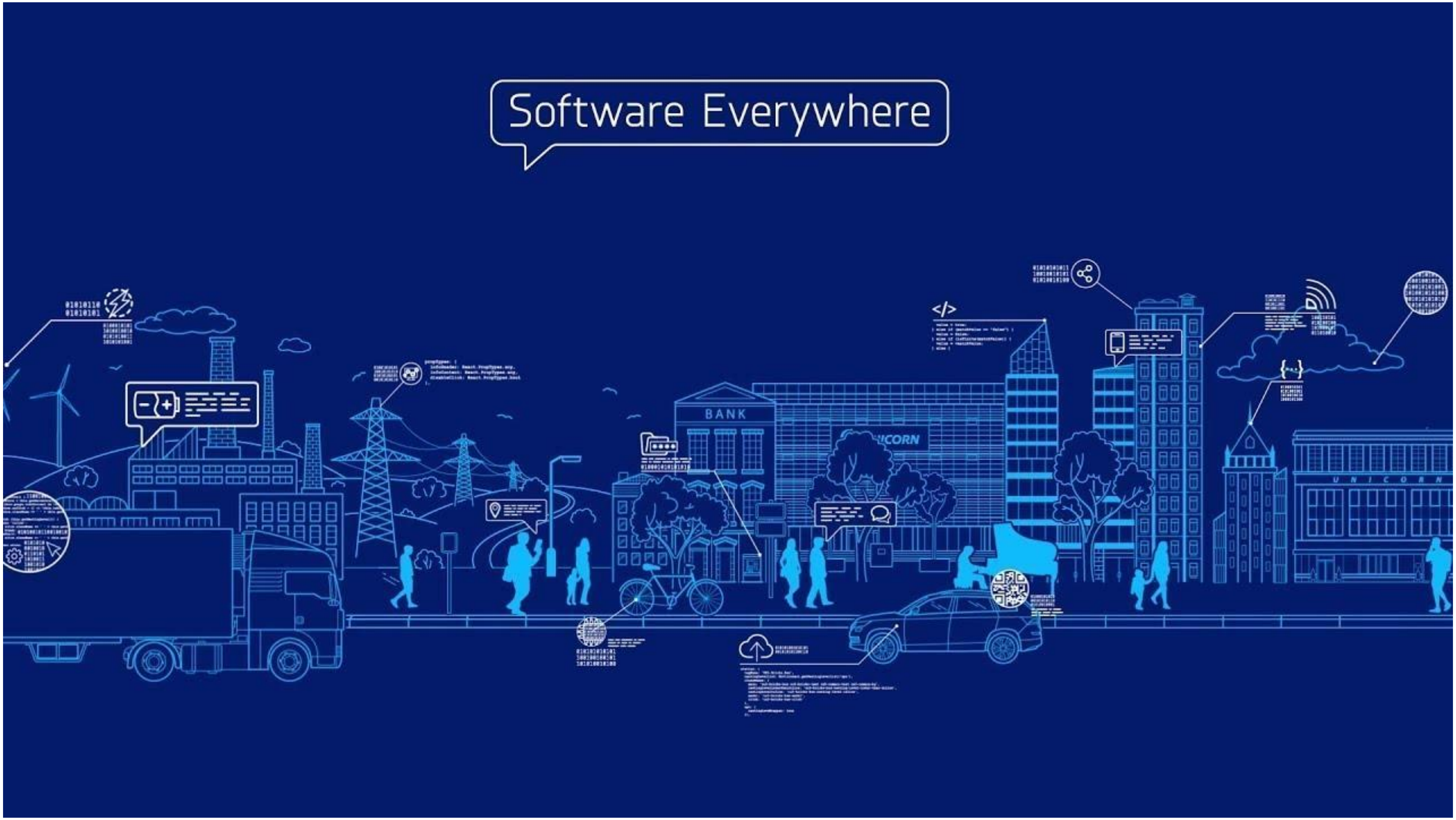
HEINZ NIXDORF INSTITUT
UNIVERSITÄT PADERBORN

Integration of Multiple Static Analysis Tools in a Single Interface

- G. S. Varma

- Supervisors:
- Prof. Dr. Eric Bodden
- Dr.-Ing. Ben Hermann


Software Everywhere
BANK
UNICORN
U N I C O R N

- “ $1.1 Trillion in Assets Affected by Software Bugs in **2016** “

  - Software Fail Watch Annual Report, Tricentis

Static Code Analysis

- It helps in prevention of bugs.
- It examines code without execution.

- Detects Vulnerabilities :

  - Injections

  - Cross Site Scripting (XSS)

  - Buffer Overflow, and Dead Code etc

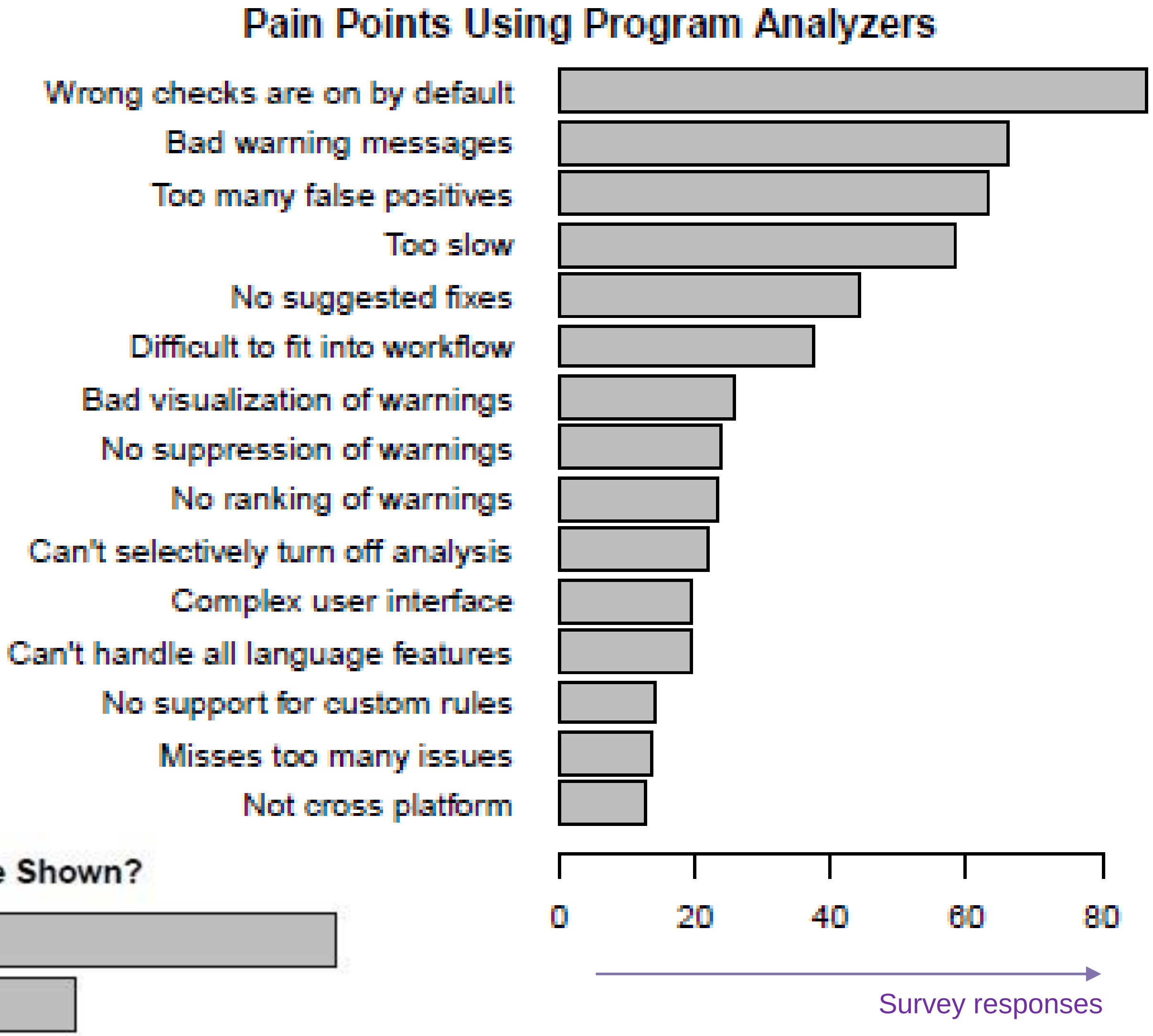
Static Code Analysis

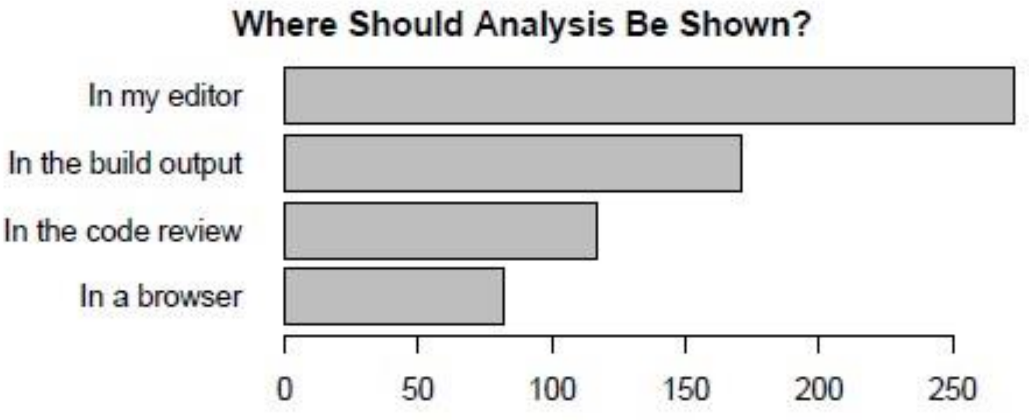
- Tools :
  - IDE Notifications
  - IDE tools
  - Dedicated tools
  - Linters
  - CLI tools

Static Code Analysis

Usability Issues

- Christakis et. al.
- Brittany et. al.
  - Tool output
  - Result understandability





Multiple Tools

- Developers use multiple static analysis tools each having own coverage.

- Research trends:

- Prioritise the bug warning alerts

(Lori et. al.)

- Merges 3 tools for Java to show warnings

(Na Meng et. al.)

Multiple Tools

- Tricorder
  - ReviewBot
  - Separate bug coverage by separate tool
  - Evaluation: Summative – Click rates

(Caitlin et. al.)

- Parfait
  - Scalability (easy , expensive analysis)
  - Precision (bug track – real, no, potential)

(Cristina et. al.)

But USABILITY is not addressed…

Problem Statement

- How to integrate the results of multiple static analysis tools

  in a unified user interface?

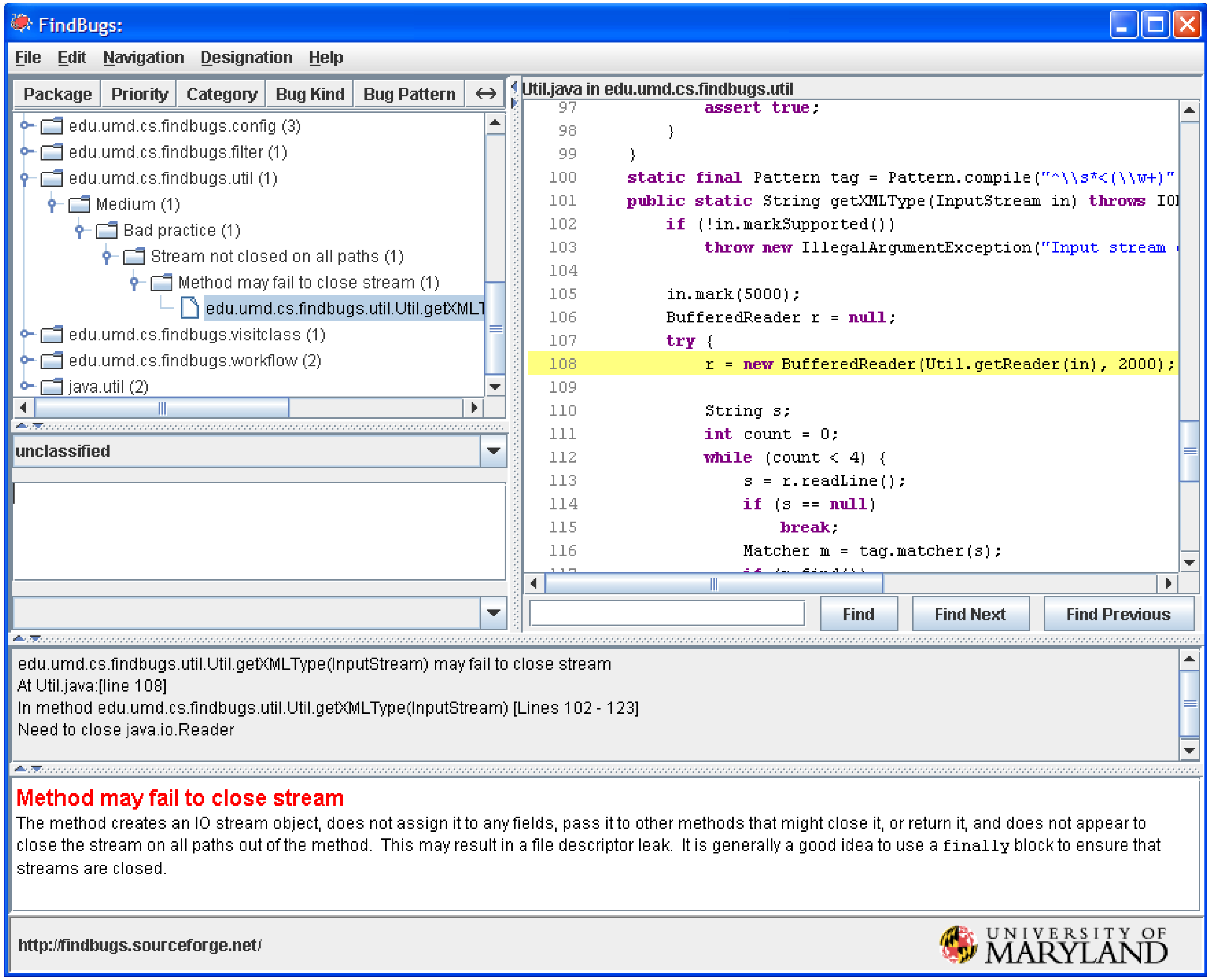
  - 3 Research Questions

Research Question 1

- How to display results of the same codebase from different analysis tools?

What Current Tools do? - RQ 1

- FindBugs


FindBugs:
File Edit Navigation Designation Help
Package Priority Category Bug Kind Bug Pattern
Find Find Next Find Previous

edu.umd.cs.findbugs.config (3)
edu.umd.cs.findbugs.filter (1)
edu.umd.cs.findbugs.util (1)
Medium (1)
Bad practice (1)
Stream not closed on all paths (1)
Method may fail to close stream (1)
edu.umd.cs.findbugs.util.Util.getXMLT
edu.umd.cs.findbugs.visitclass (1)
edu.umd.cs.findbugs.workflow (2)
java.util (2)

unclassified

Util.java in edu.umd.cs.findbugs.util

```
 97                 assert true;
 98             }
 99         }
100         static final Pattern tag = Pattern.compile("^\\s*<(\\w+)"
101         public static String getXMLType(InputStream in) throws IOI
102             if (!in.markSupported())
103                 throw new IllegalArgumentException("Input stream
104 
105             in.mark(5000);
106             BufferedReader r = null;
107             try {
108                 r = new BufferedReader(Util.getReader(in), 2000);
109 
110                 String s;
111                 int count = 0;
112                 while (count < 4) {
113                     s = r.readLine();
114                     if (s == null)
115                         break;
116                     Matcher m = tag.matcher(s);
```

edu.umd.cs.findbugs.util.Util.getXMLType(InputStream) may fail to close stream
At Util.java:[line 108]
In method edu.umd.cs.findbugs.util.Util.getXMLType(InputStream) [Lines 102 - 123]
Need to close java.io.Reader

Method may fail to close stream

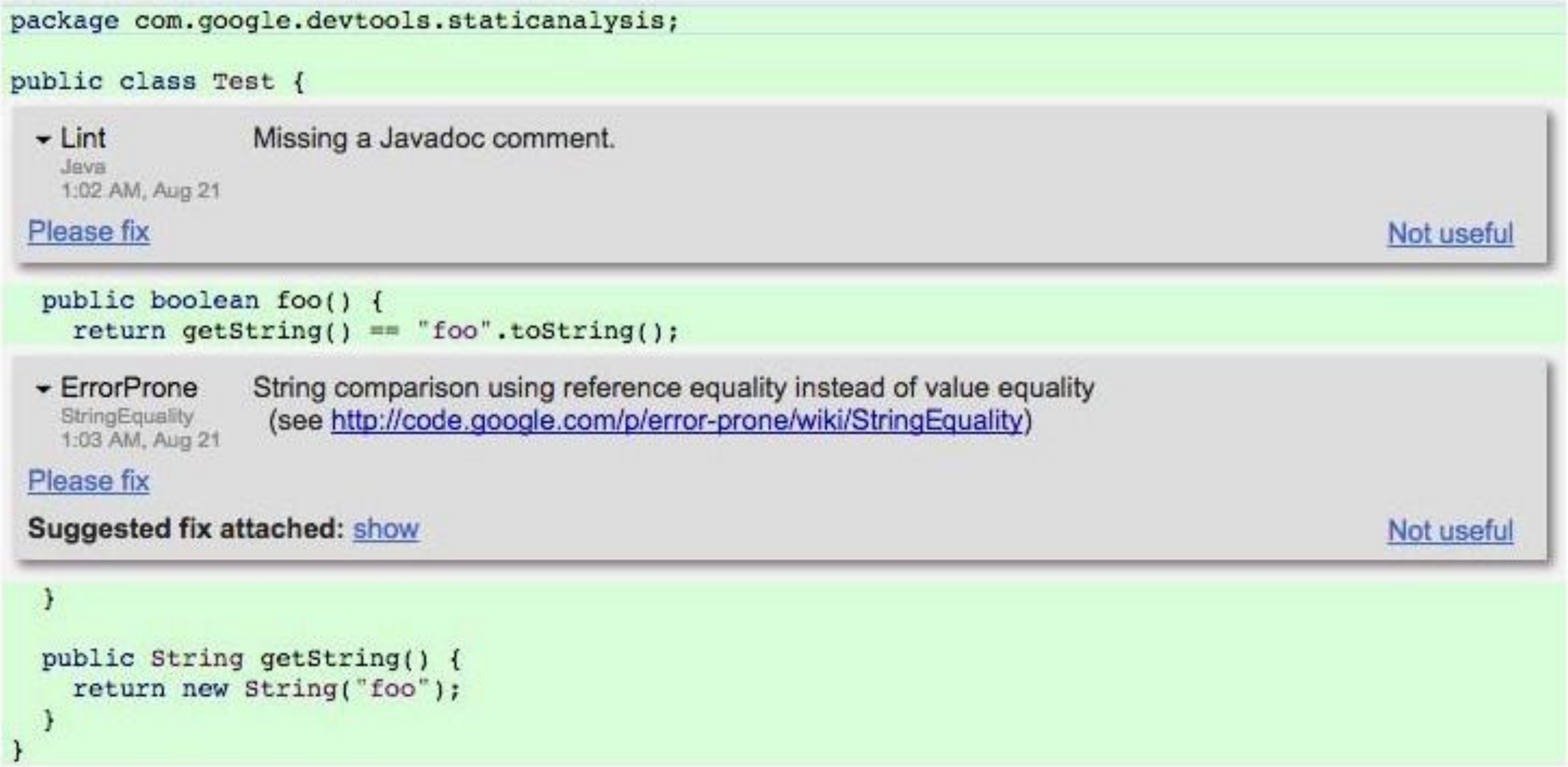
The method creates an IO stream object, does not assign it to any fields, pass it to other methods that might close it, or return it, and does not appear to close the stream on all paths out of the method. This may result in a file descriptor leak. It is generally a good idea to use a `finally` block to ensure that streams are closed.

http://findbugs.sourceforge.net/

UNIVERSITY OF MARYLAND

What Current Tools do? - RQ 1

- Tricorder



```
package com.google.devtools.staticanalysis;

public class Test {
```

Missing a Javadoc comment.

```
  public boolean foo() {
    return getString() == "foo".toString();
```

String comparison using reference equality instead of value equality
(see http://code.google.com/p/error-prone/wiki/StringEquality)

Suggested fix attached: show

```
  }

  public String getString() {
    return new String("foo");
  }
}
```

Research Question 2

- What feedback works to know that the bug fixing is on-going?

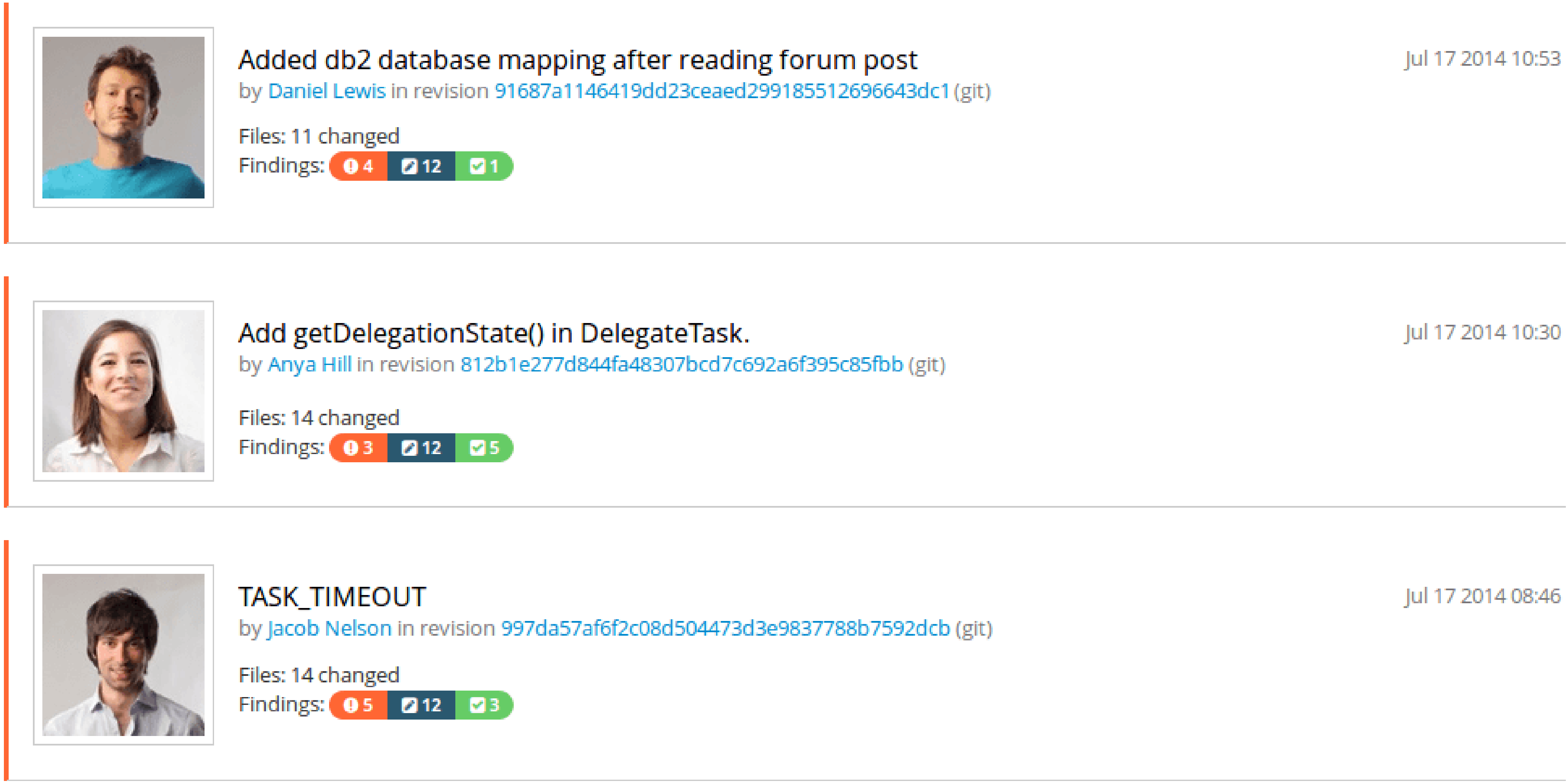
- What current tools do?
  - Traditional approach – Nightly Builds

Research Question 3

- How to carry traceability of bug fixing?

What Current Tools do? - RQ 3

- TeamScale



Added db2 database mapping after reading forum post

by Daniel Lewis in revision 91687a1146419dd23ceaed299185512696643dc1 (git)

Jul 17 2014 10:53

Files: 11 changed

Findings: ❗4 ✎12 ☑1

Add getDelegationState() in DelegateTask.

by Anya Hill in revision 812b1e277d844fa48307bcd7c692a6f395c85fbb (git)

Jul 17 2014 10:30

Files: 14 changed

Findings: ❗3 ✎12 ☑5

TASK_TIMEOUT

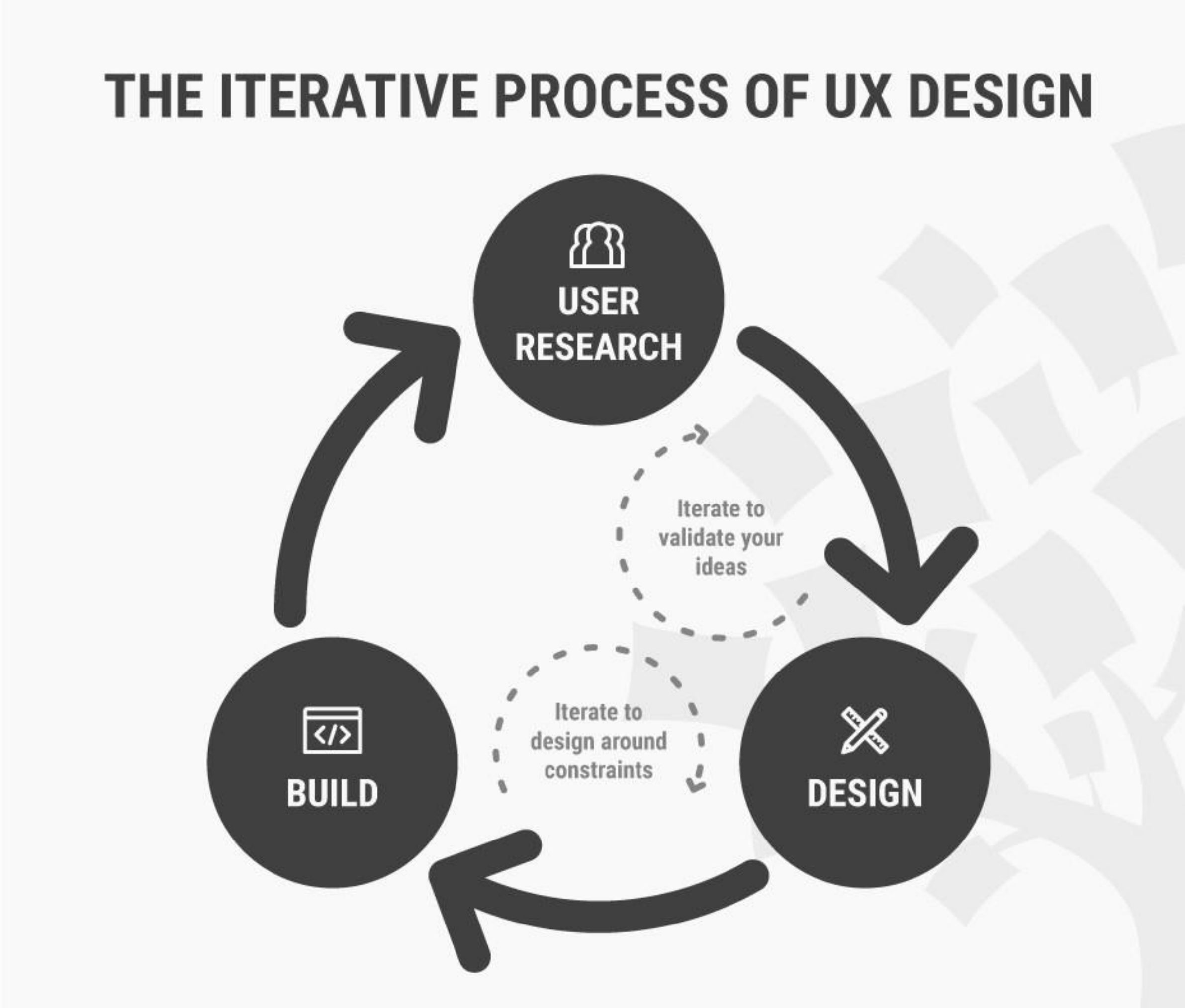
by Jacob Nelson in revision 997da57af6f2c08d504473d3e9837788b7592dcb (git)

Jul 17 2014 08:46

Files: 14 changed

Findings: ❗5 ✎12 ☑3

Our Approaches


THE ITERATIVE PROCESS OF UX DESIGN
USER RESEARCH
DESIGN
BUILD
Iterate to validate your ideas
Iterate to design around constraints

Our Approaches

- Software Engineering disciplines:
  - Complex datasets
  - Compiler reporting
  - Continuous integration
  - Refactoring tools
  - Issue tracker
  - Stack Overflow
  - Gamification
  - Usability Engineering

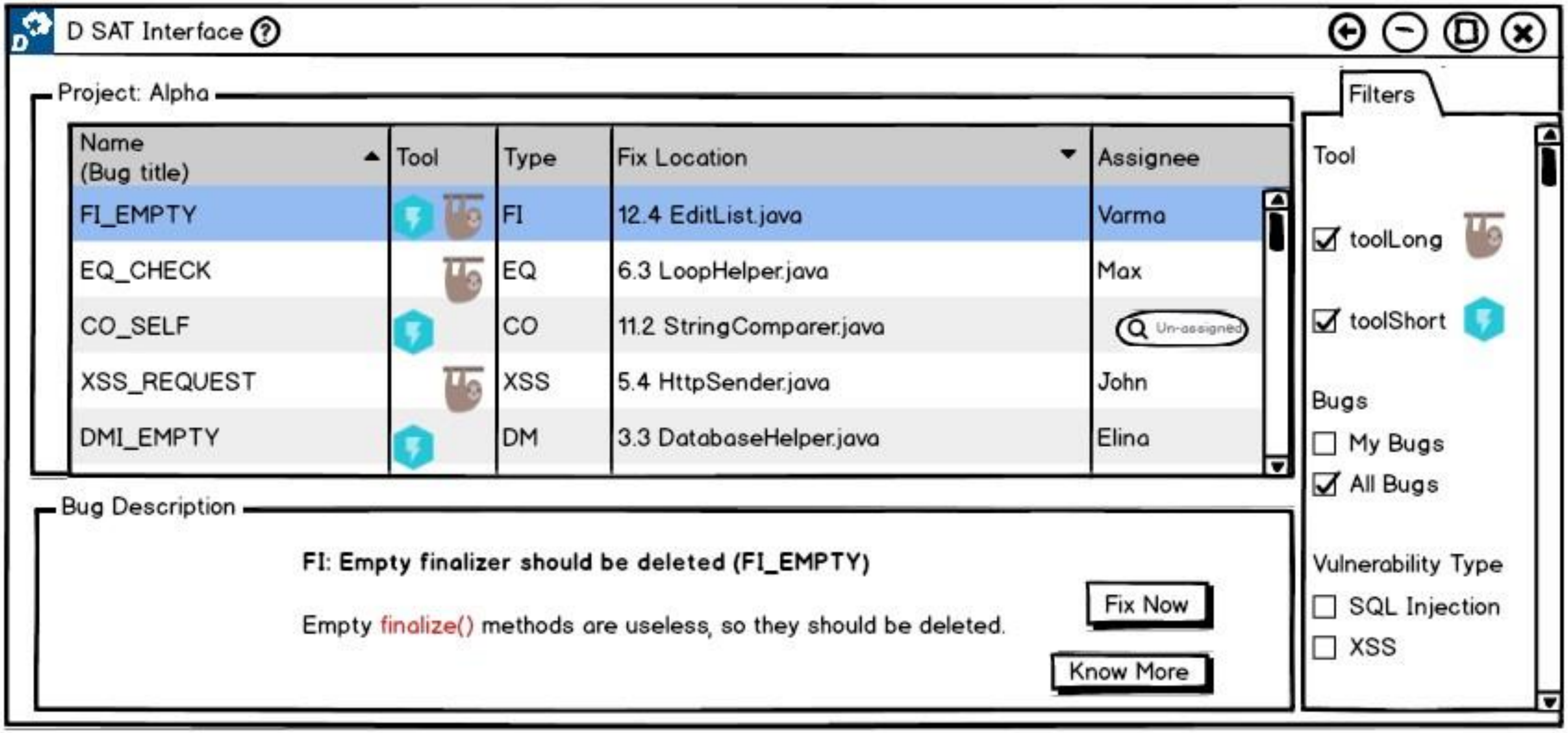
Our Approaches

- Complex datasets:
  - Dix et. al. - complex grouping and linking of datasets for Spreadsheets application

    Design lesson : extensibility of columns
- Issue tracker
  - Baysal et. al. :
    - ❖ Information overload
    - ❖ Expressiveness

Example: RQ 1

- Prototype 1



D SAT Interface

Project: Alpha

Name (Bug title) ▲	Tool	Type	Fix Location ▼	Assignee
FI_EMPTY	⬢ 🦥	FI	12.4 EditList.java	Varma
EQ_CHECK	🦥	EQ	6.3 LoopHelper.java	Max
CO_SELF	⬢	CO	11.2 StringComparer.java	🔍 Un-assigned
XSS_REQUEST	🦥	XSS	5.4 HttpSender.java	John
DMI_EMPTY	⬢	DM	3.3 DatabaseHelper.java	Elina

Bug Description

FI: Empty finalizer should be deleted (FI_EMPTY)

Empty finalize() methods are useless, so they should be deleted.

Filters

Tool

☑ toolLong 🦥

☑ toolShort ⬢

Bugs

☐ My Bugs

☑ All Bugs

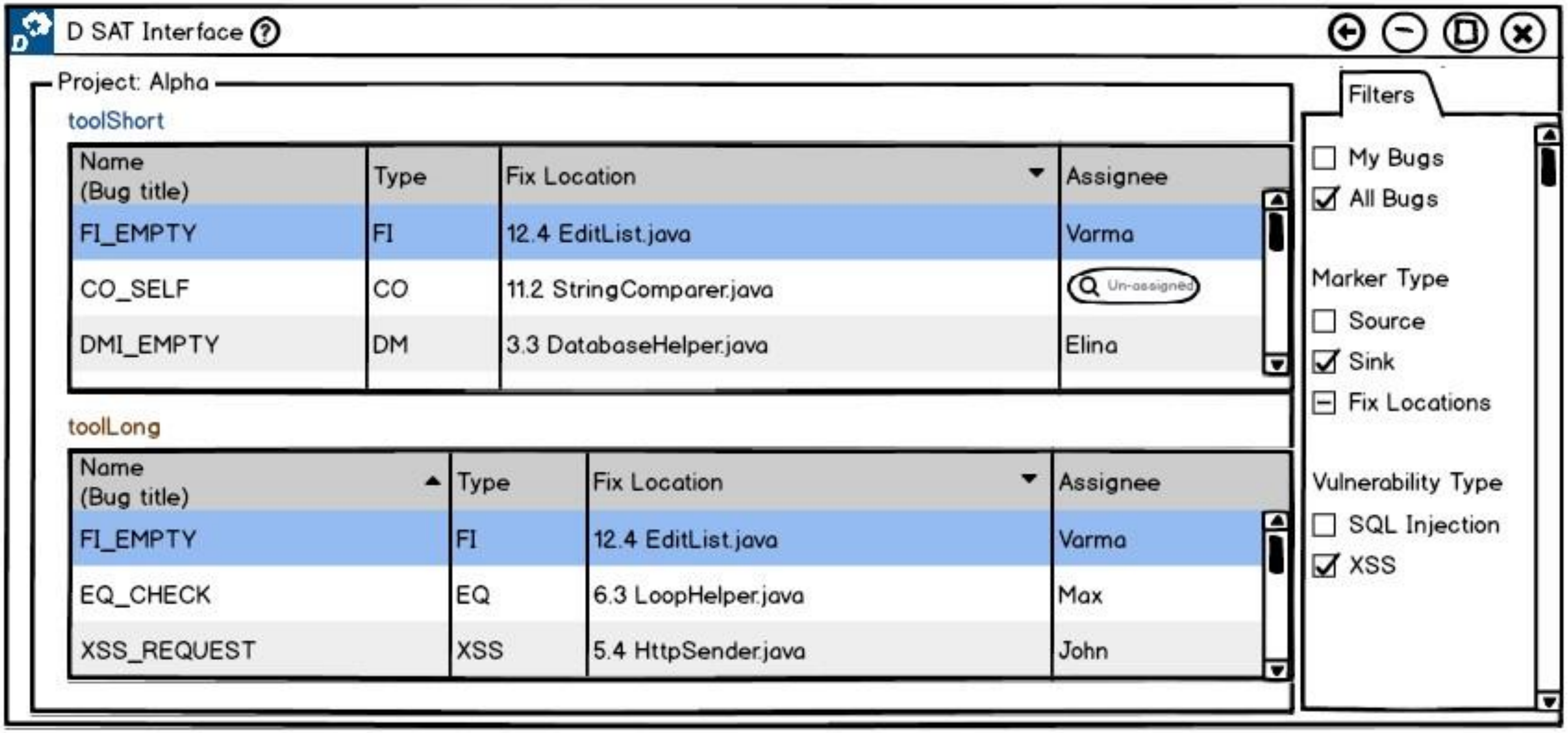
Vulnerability Type

☐ SQL Injection

☐ XSS

Example: RQ 1

- Prototype 2



Project: Alpha

toolShort

Name (Bug title)	Type	Fix Location	Assignee
FI_EMPTY	FI	12.4 EditList.java	Varma
CO_SELF	CO	11.2 StringComparer.java	Un-assigned
DMI_EMPTY	DM	3.3 DatabaseHelper.java	Elina

toolLong

Name (Bug title)	Type	Fix Location	Assignee
FI_EMPTY	FI	12.4 EditList.java	Varma
EQ_CHECK	EQ	6.3 LoopHelper.java	Max
XSS_REQUEST	XSS	5.4 HttpSender.java	John

Filters

□ My Bugs
☑ All Bugs

Marker Type

□ Source
☑ Sink
⊟ Fix Locations

Vulnerability Type

□ SQL Injection
☑ XSS

Evaluation

- Experimental Design
  - Recruit Test Users
  - Order of evaluation altered
  - Perform Tasks
    - Example: Find a bug which is reported in common by available tools.

Evaluation – Usability Inspection Methods

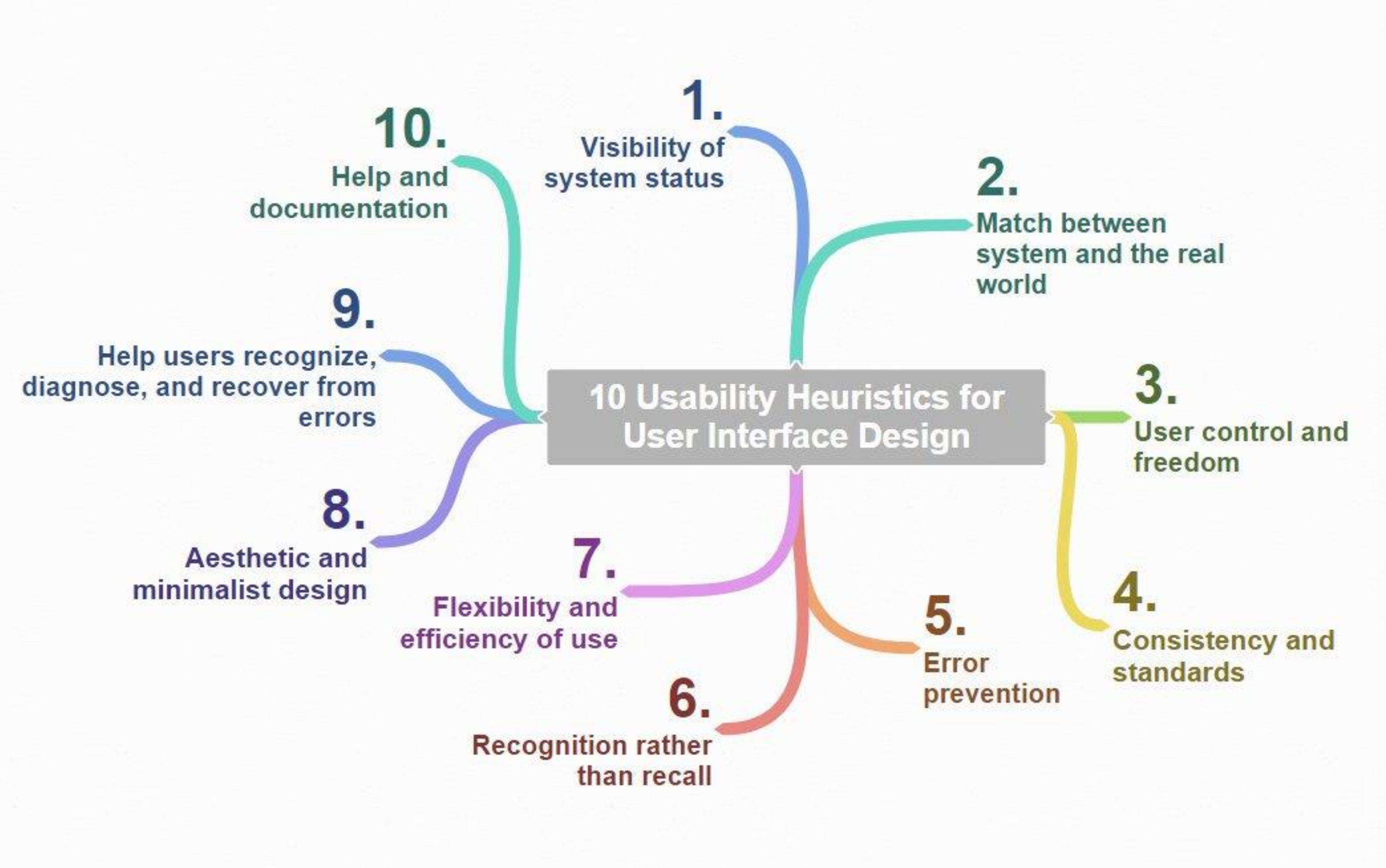
- Cognitive Walkthrough

For each step to a predefined task, the following aspects are analysed.

- Will the user try and achieve the right outcome?
- Will the user notice that the correct action is available to them?
- Will the user associate the correct action with the outcome they expect to achieve?
- If the correct action is performed; will the user see that progress is being made towards their intended outcome?

Evaluation – Usability Inspection Methods

- Heuristic Evaluation



10 Usability Heuristics for User Interface Design
1. Visibility of system status
2. Match between system and the real world
3. User control and freedom
4. Consistency and standards
5. Error prevention
6. Recognition rather than recall
7. Flexibility and efficiency of use
8. Aesthetic and minimalist design
9. Help users recognize, diagnose, and recover from errors
10. Help and documentation

Evaluation – Usability Inspection Methods

- Heuristic Evaluation

Each problem w.r.t. a heuristic is rated accordingly; 0 – 4

0 - do not agree this is a usability problem

1 - cosmetic problem

2 - minor usability problem

3 - major usability problem (important to fix)

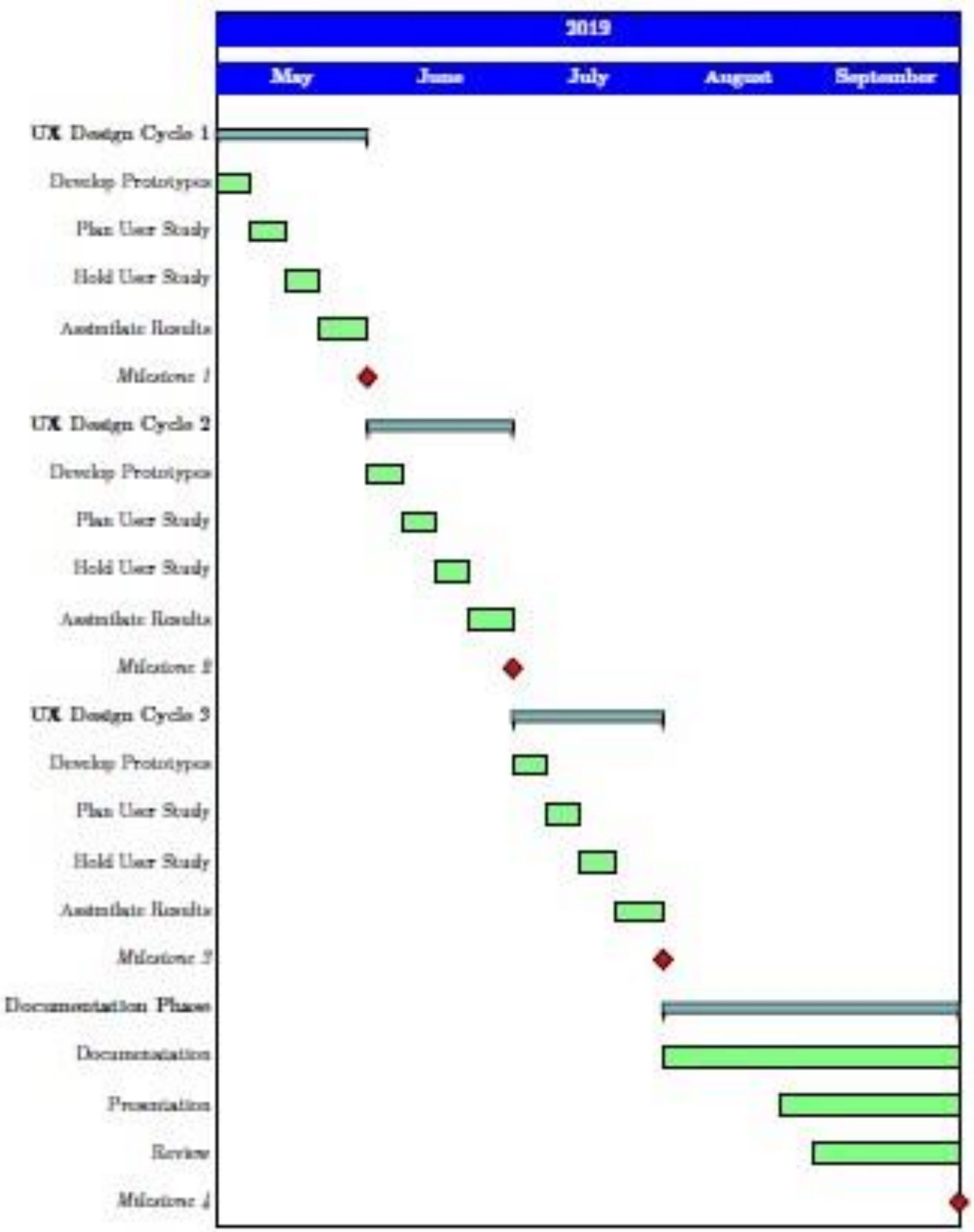
4 - usability catastrophe (imperative to fix)

Comparative Study

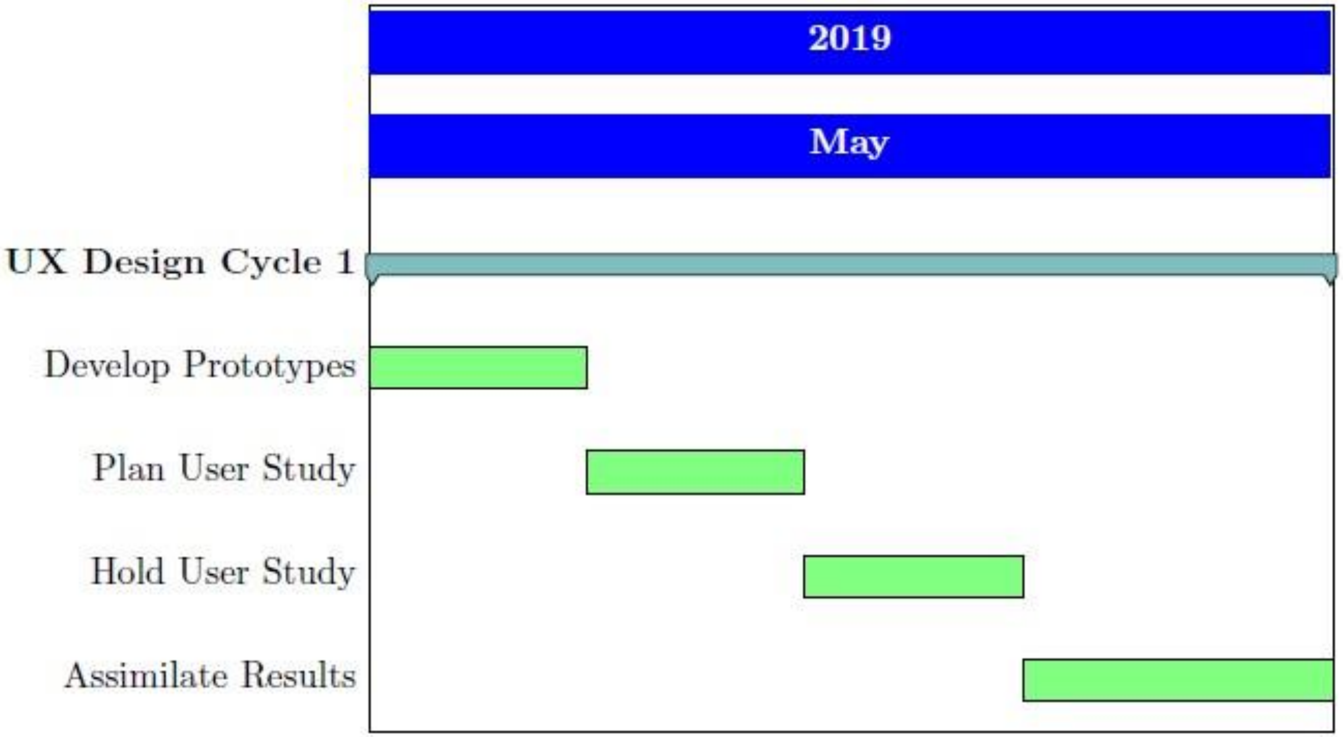
Time Plan

- Official Time: 5 Months

- Milestones: 4

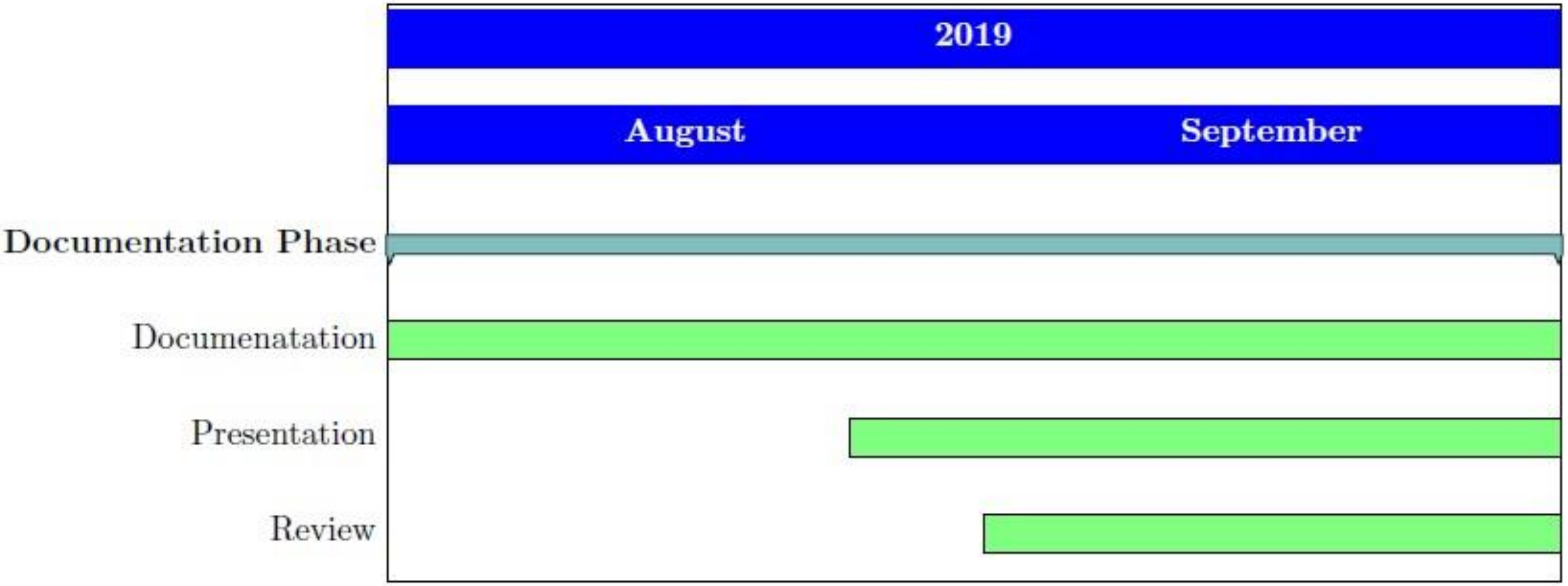


Milestones 1 2 3



Similarly in June and July ...

Milestone 4


2019
August
September
Documentation Phase
Documenatation
Presentation
Review

Summary

- Importance of Static Analysis tools
- Usage of Multiple Static Analysis tools

- Need for a single user interface for multiple tools

- This Thesis work follows UX Design Cycle to achieve usable prototypes focussing on research questions such as,
  - How to display results of the same codebase from different analysis tools?
  - What feedback works to know that the bug fixing is on-going?
  - How to carry traceability of bug fixing?