HEINZ NIXDORF INSTITUT
UNIVERSITÄT PADERBORN

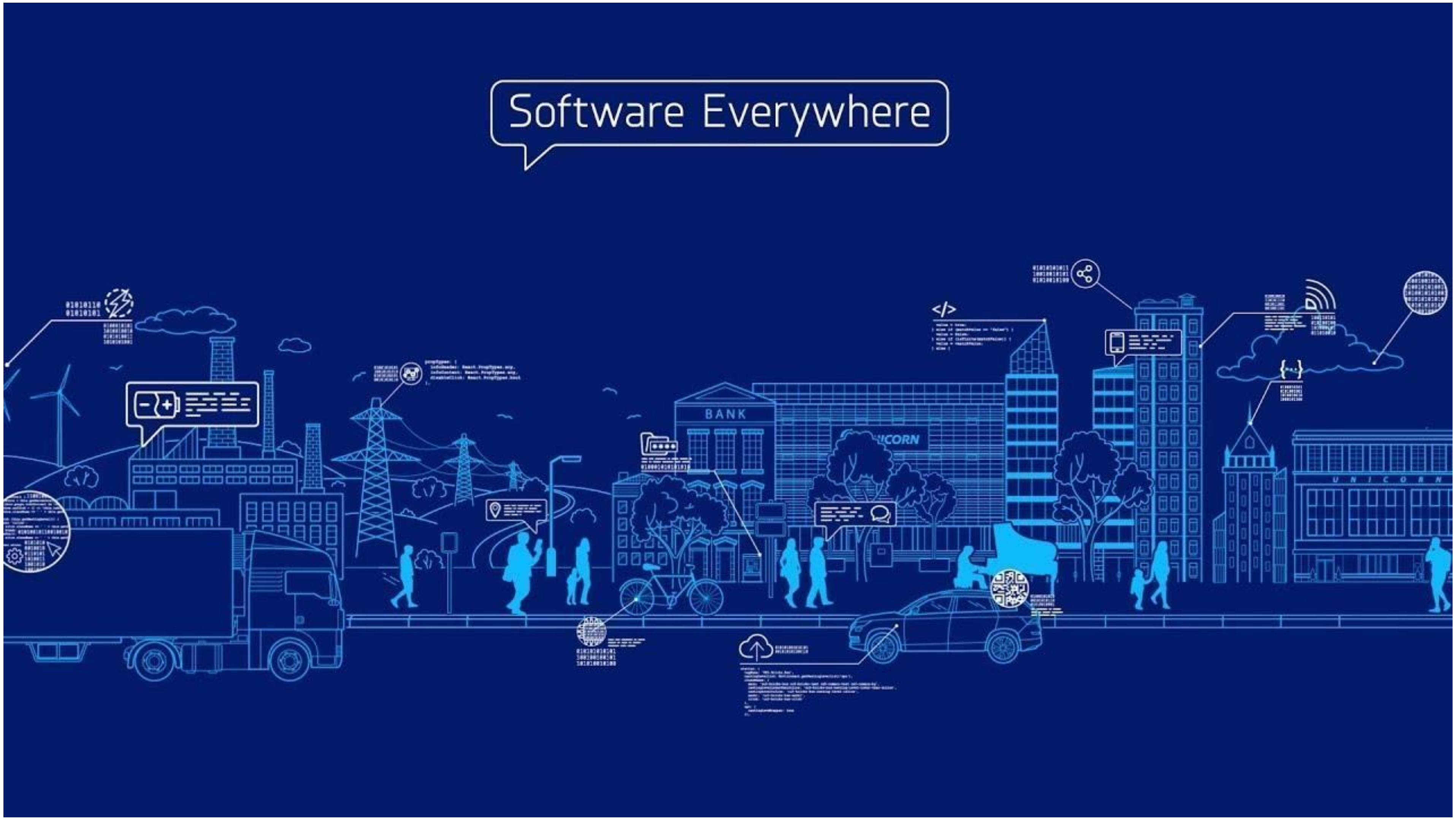
Integration of Multiple Static Analysis Tools in a Single Interface

- G. S. Varma

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

- Formative Study
- Detailed Information – thesis report, online repository*

*https://github.com/gsvarma/MSAT-UI



- https://unicorn.com/en/software-everywhere

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

  - Software Fail Watch Annual Report, Tricentis

❖ https://www.tricentis.com/news/software-fail-watch-says-1-1-trillion-in-assets-affected-by-software-bugs-in-2016/

Static Code Analysis

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

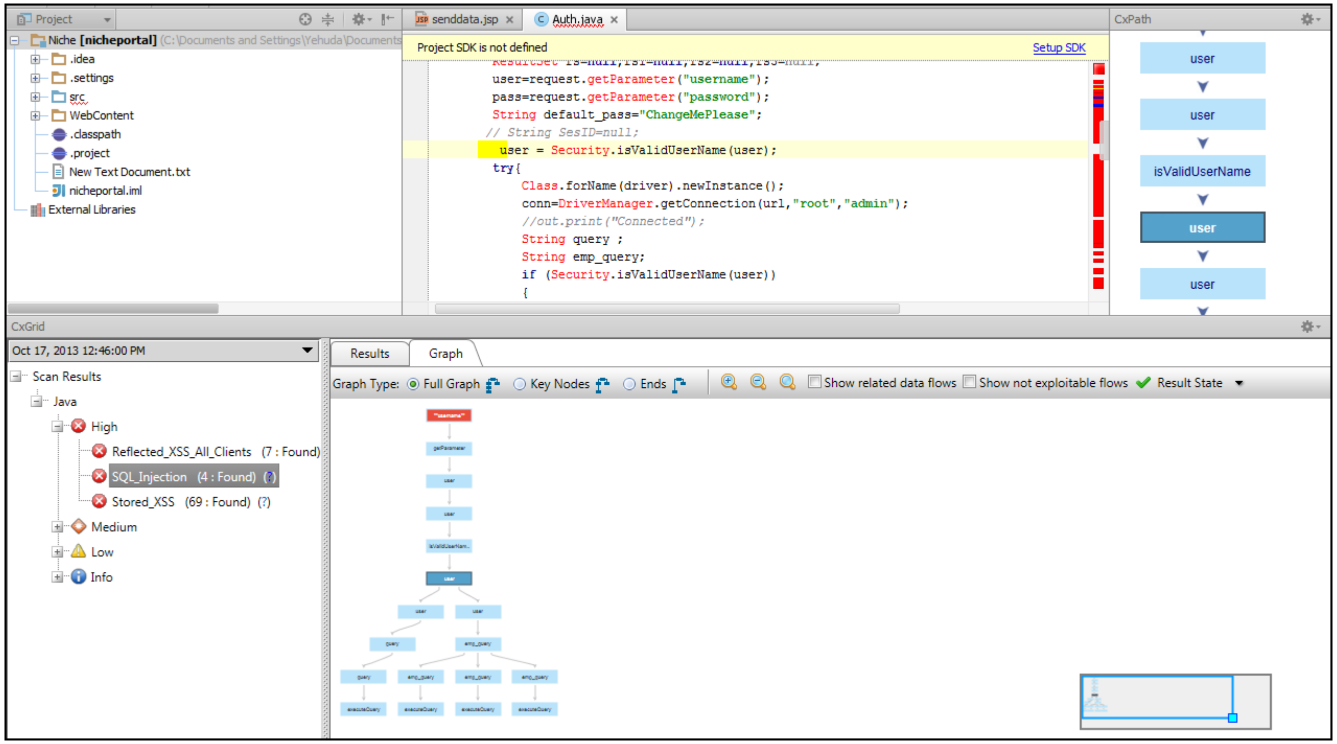
- Detects vulnerabilities :

  - Injections
  - Cross Site Scripting (XSS)
  - Buffer Overflow, and Dead Code etc.

❖ Designing code analyses for Large Software Systems (DECA). url: https://www.hni.uni-paderborn.de/swt/lehre/deca/.

Static Code Analysis

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



Project SDK is not defined

```
    ResultSet rs=null,rs1=null,rs2=null,rs3=null;
    user=request.getParameter("username");
    pass=request.getParameter("password");
    String default_pass="ChangeMePlease";
   // String SesID=null;
     user = Security.isValidUserName(user);
    try{
        Class.forName(driver).newInstance();
        conn=DriverManager.getConnection(url,"root","admin");
        //out.print("Connected");
        String query ;
        String emp_query;
        if (Security.isValidUserName(user))
        {
```

Oct 17, 2013 12:46:00 PM

```
Scan Results
  Java
    High
      Reflected_XSS_All_Clients   (7 : Found)
      SQL_Injection   (4 : Found)  (?)
      Stored_XSS   (69 : Found)  (?)
    Medium
    Low
    Info
```

Graph Type: [✓ Full Graph] Key Nodes Ends

- Checkmarx – Application Security Testing and Static Code Analysis. url: https://www.checkmarx.com/
- CxViewer - Plugins | JetBrains, url: https://plugins.jetbrains.com/plugin/7593-cxviewer
- FindBugs™ - Find Bugs in Java Programs. url: http://findbugs.sourceforge.net/.

Static Code Analysis

- Johnson et al.
  - Tool output
  - Result understandability

Usability Issues

- Christakis et al.
- Habib et al.

❖ Brittany Johnson, Yoonki Song, Emerson Murphy-Hill, and Robert Bowdidge. 2013. Why don't software developers use static analysis tools to find bugs?. In *Proceedings of the 2013 International Conference on Software Engineering* (ICSE '13). IEEE Press, Piscataway, NJ, USA, 672-681.

❖ Maria Christakis and Christian Bird. 2016. What developers want and need from program analysis: an empirical study. In *Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering* (ASE 2016). ACM, New York, NY, USA, 332-343. DOI: https://doi.org/10.1145/2970276.2970347

❖ Habib, A., & Pradel, M. (2018, September). How many of all bugs do we find? a study of static bug detectors. In *ASE* (pp. 317-328).

Multiple Tools

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

- Research trends:

- Prioritise the bug warning alerts

(Flynn et al.)

- Merges 3 tools for Java to show warnings

(Meng et al.)

❖ Lori Flynn, William Snavely, David Svoboda, Nathan VanHoudnos, Richard Qin, Jennifer Burns, David Zubrow, Robert Stoddard, and Guillermo Marce-Santurio. 2018. Prioritizing alerts from multiple static analysis tools, using classification models. In *Proceedings of the 1st International Workshop on Software Qualities and Their Dependencies* (SQUADE '18). ACM, New York, NY, USA, 13-20. DOI: https://doi.org/10.1145/3194095.3194100

❖ N. Meng, Q. Wang, Q. Wu and H. Mei, "An Approach to Merge Results of Multiple Static Analysis Tools (Short Paper)," *2008 The Eighth International Conference on Quality Software*, Oxford, 2008, pp. 169-174.doi: 10.1109/QSIC.2008.30

Multiple Tools

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

(Sadowski et al.)

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

(Cifuentes et al.)

But USABILITY is not addressed…

- Caitlin Sadowski, Jeffrey van Gogh, Ciera Jaspan, Emma Söderberg, and Collin Winter. 2015. Tricorder: building a program analysis ecosystem. In *Proceedings of the 37th International Conference on Software Engineering - Volume 1* (ICSE '15), Vol. 1. IEEE Press, Piscataway, NJ, USA, 598-608.
- Cristina Cifuentes and Bernhard Scholz. 2008. Parfait: designing a scalable bug checker. In *Proceedings of the 2008 workshop on Static analysis* (SAW '08). ACM, New York, NY, USA, 4-11. DOI=http://dx.doi.org/10.1145/1394504.1394505

Problem Statement

- How to integrate the results of multiple static analysis tools

  in a unified user interface?

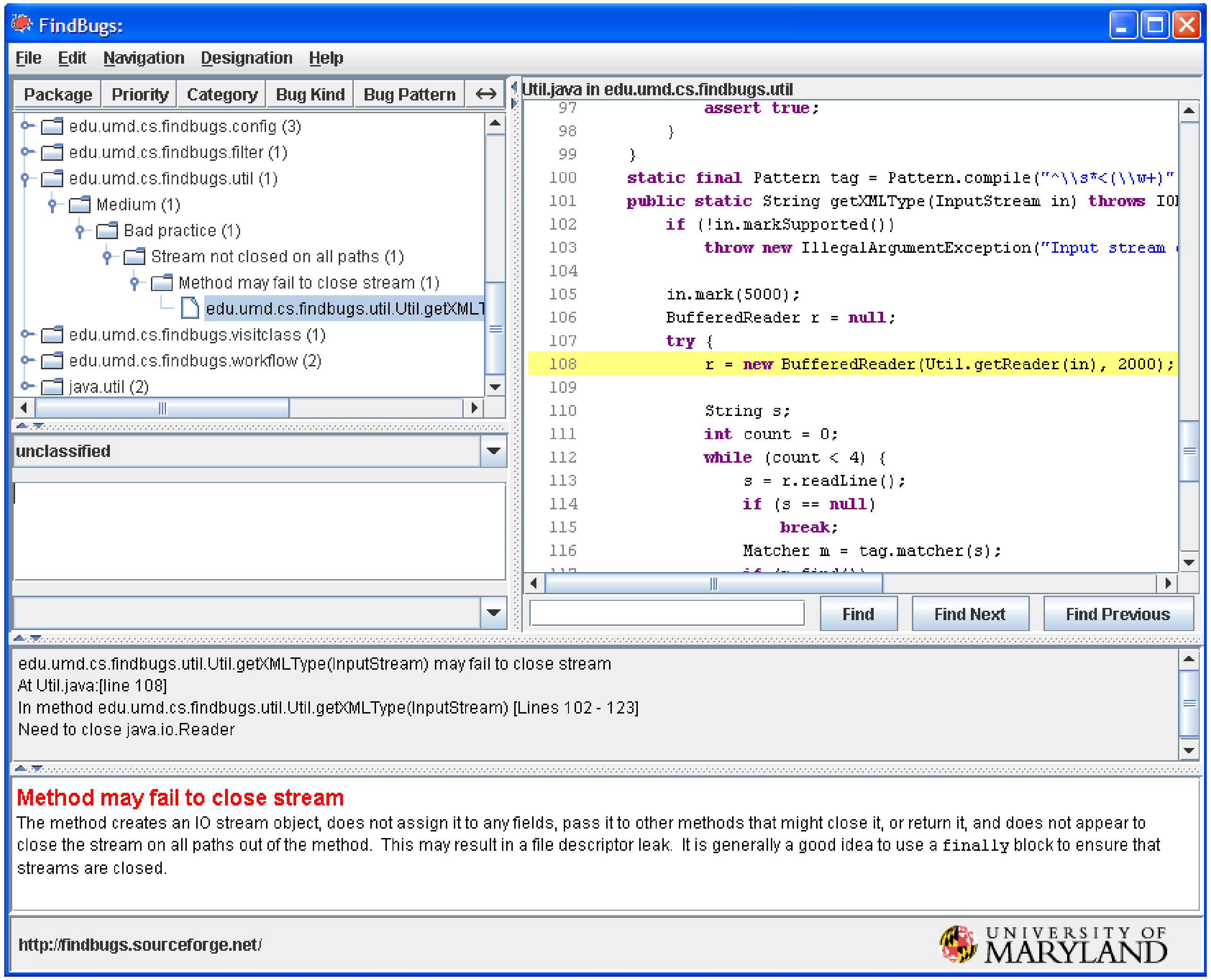
  - 3 Research Questions

Research Question 1

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

What Current Tools do? - RQ 1: .. display results!

- FindBugs


FindBugs:
File Edit Navigation Designation Help
Package Priority Category Bug Kind Bug Pattern
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
Find
Find Next
Find Previous
http://findbugs.sourceforge.net/
UNIVERSITY OF MARYLAND

```
                assert true;
            }
        }
    static final Pattern tag = Pattern.compile("^\\s*<(\\w+)"
    public static String getXMLType(InputStream in) throws IO
        if (!in.markSupported())
            throw new IllegalArgumentException("Input stream 

        in.mark(5000);
        BufferedReader r = null;
        try {
            r = new BufferedReader(Util.getReader(in), 2000);

            String s;
            int count = 0;
            while (count < 4) {
                s = r.readLine();
                if (s == null)
                    break;
                Matcher m = tag.matcher(s);
```

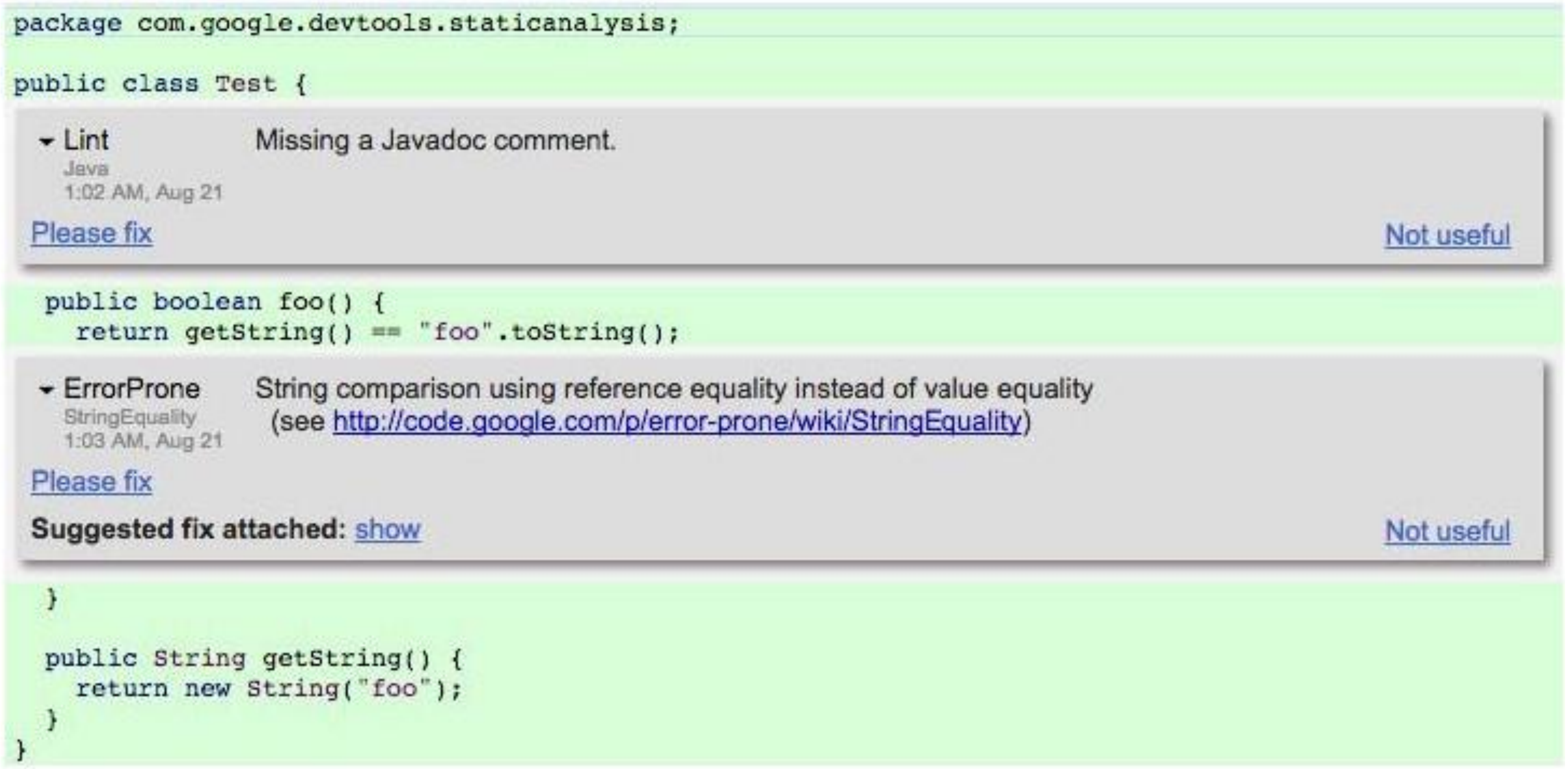
edu.umd.cs.findbugs.util.Util.getXMLType(InputStream) may fail to close stream
At Util.java:[line 108]
In method edu.umd.cs.findbugs.util.Util.getXMLType(InputStream) [Lines 102 - 123]
Need to close java.io.Reader

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.

What Current Tools do? - RQ 1: .. display results!

- Tricorder



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

public class Test {
```

Lint — Missing a Javadoc comment.
Java
1:02 AM, Aug 21
Please fix — Not useful

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

ErrorProne — String comparison using reference equality instead of value equality (see http://code.google.com/p/error-prone/wiki/StringEquality)
StringEquality
1:03 AM, Aug 21
Please fix
Suggested fix attached: show — Not useful

```
  }

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

HEINZ NIXDORF INSTITUT
UNIVERSITÄT PADERBORN

Research Question 2

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

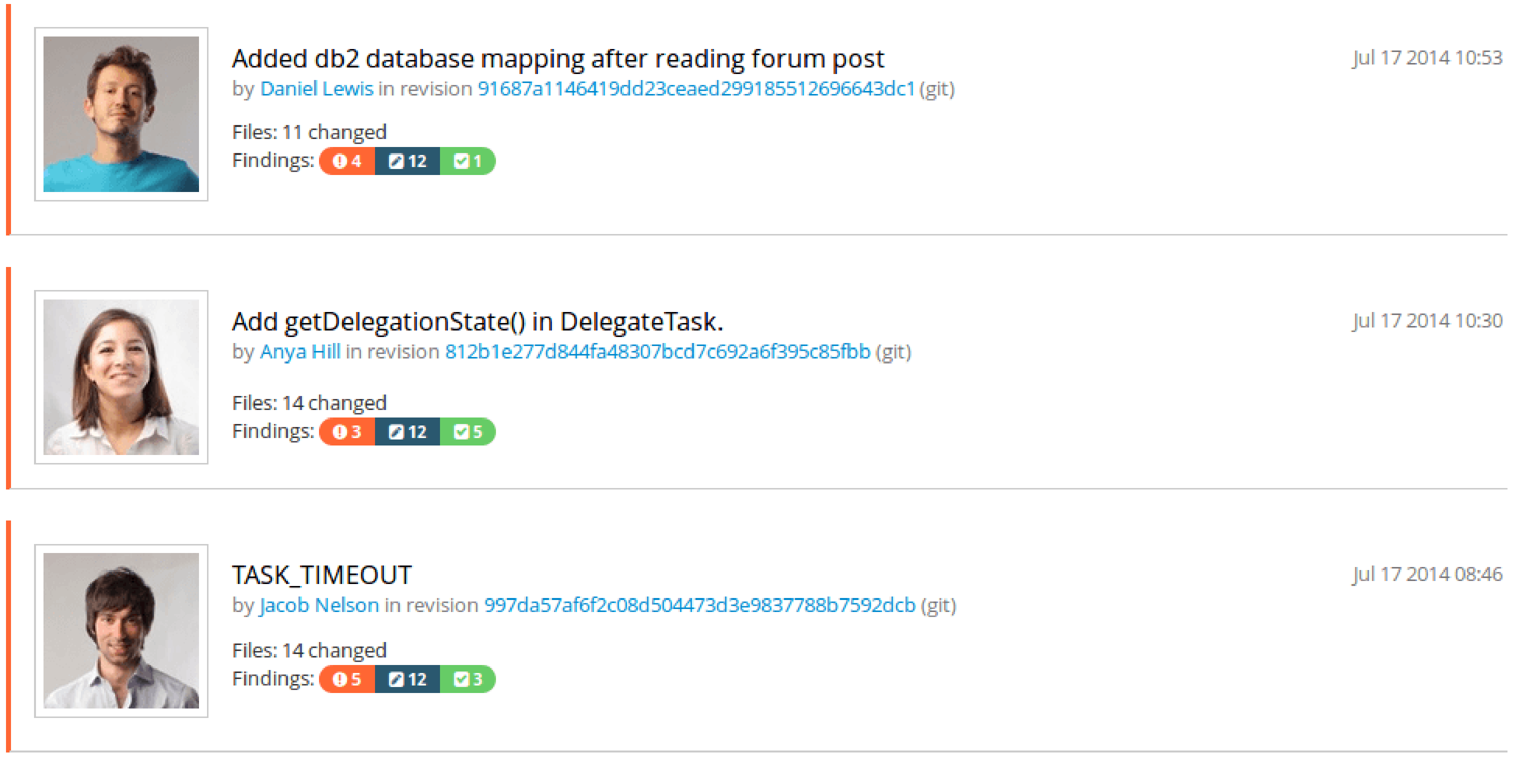
- What current tools do?
  - Traditional approach – Nightly Builds

Research Question 3

- How to carry traceability of bug fixing?

What Current Tools do? - RQ 3: .. traceability!

- 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

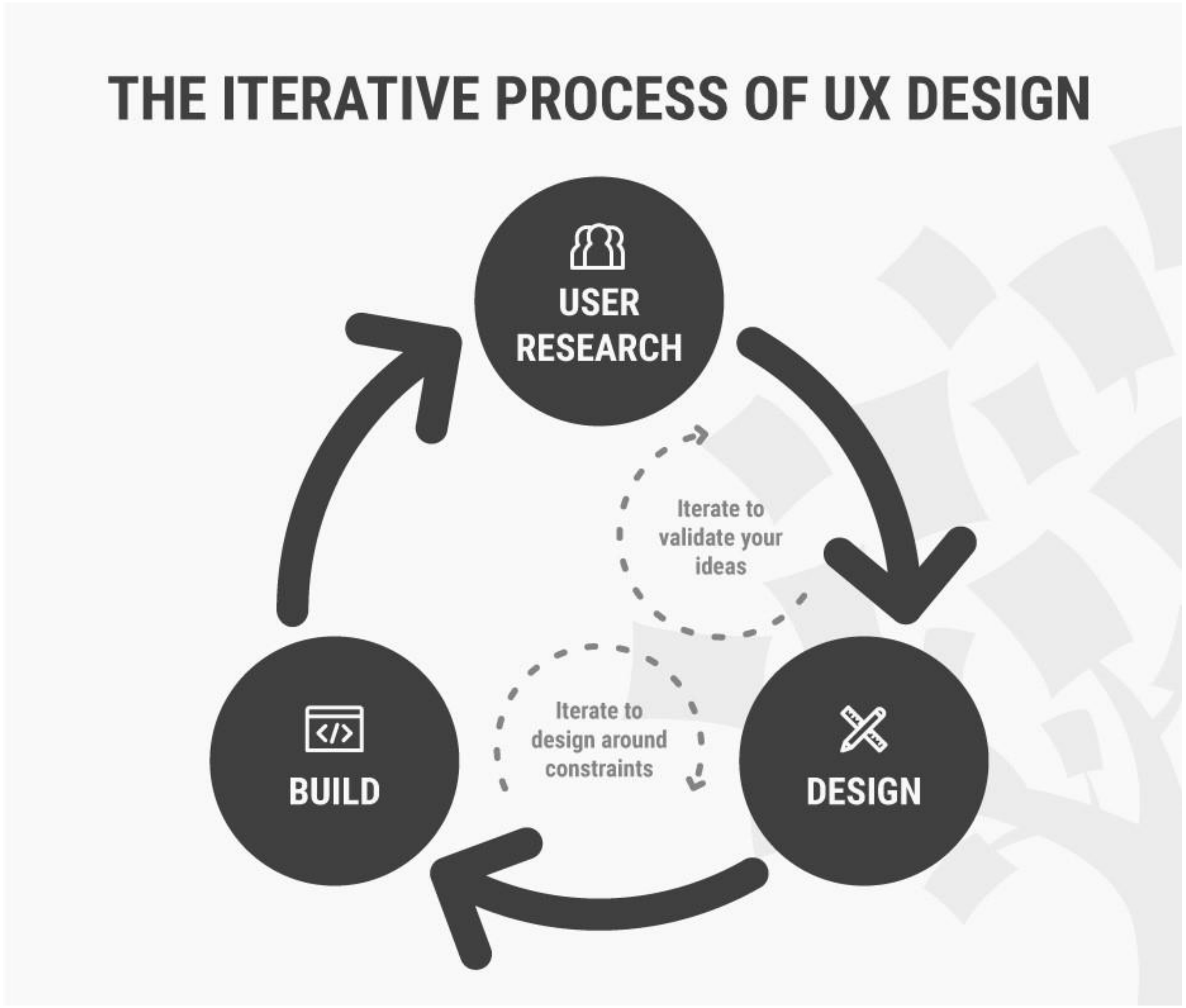
Files: 14 changed
Findings: 3 12 5

TASK_TIMEOUT
by Jacob Nelson in revision 997da57af6f2c08d504473d3e9837788b7592dcb (git)
Jul 17 2014 08:46

Files: 14 changed
Findings: 5 12 3

❖ Teamscale. url: https://www.cqse.eu/en/products/teamscale/features/.

Our Approaches


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

❖ How to Change Your Career from Graphic Design to UX Design. url: https://www.interaction-design.org/literature/article/how-to-change-your-career-fromgraphic-design-to-ux-design.

Our Approaches

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

Our Approaches – research existing scenarios!

- 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

- Alan Dix, Rachel Cowgill, Christina Bashford, Simon McVeigh, and Rupert Ridgewell. 2016. Spreadsheets as User Interfaces. In *Proceedings of the International Working Conference on Advanced Visual Interfaces* (AVI '16), Paolo Buono, Rosa Lanzilotti, and Maristella Matera (Eds.). ACM, New York, NY, USA, 192-195. DOI: https://doi.org/10.1145/2909132.2909271
- Olga Baysal, Reid Holmes, and Michael W. Godfrey. 2014. No issue left behind: reducing information overload in issue tracking. In *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering* (FSE 2014). ACM, New York, NY, USA, 666-677. DOI: https://doi.org/10.1145/2635868.2635887

Evaluation

- Experimental Design
  - Recruit Test Users
  - Order of evaluation altered
  - Likert Scale
  - Usability inspection methods: Cognitive Walkthrough
  - Perform Tasks
    - Example: Find a bug which is reported in common by available tools.

❖ Rensis Likert. “A technique for the measurement of attitudes.” In: Archives of psychology (1932).

Evaluation – Usability Inspection Methods

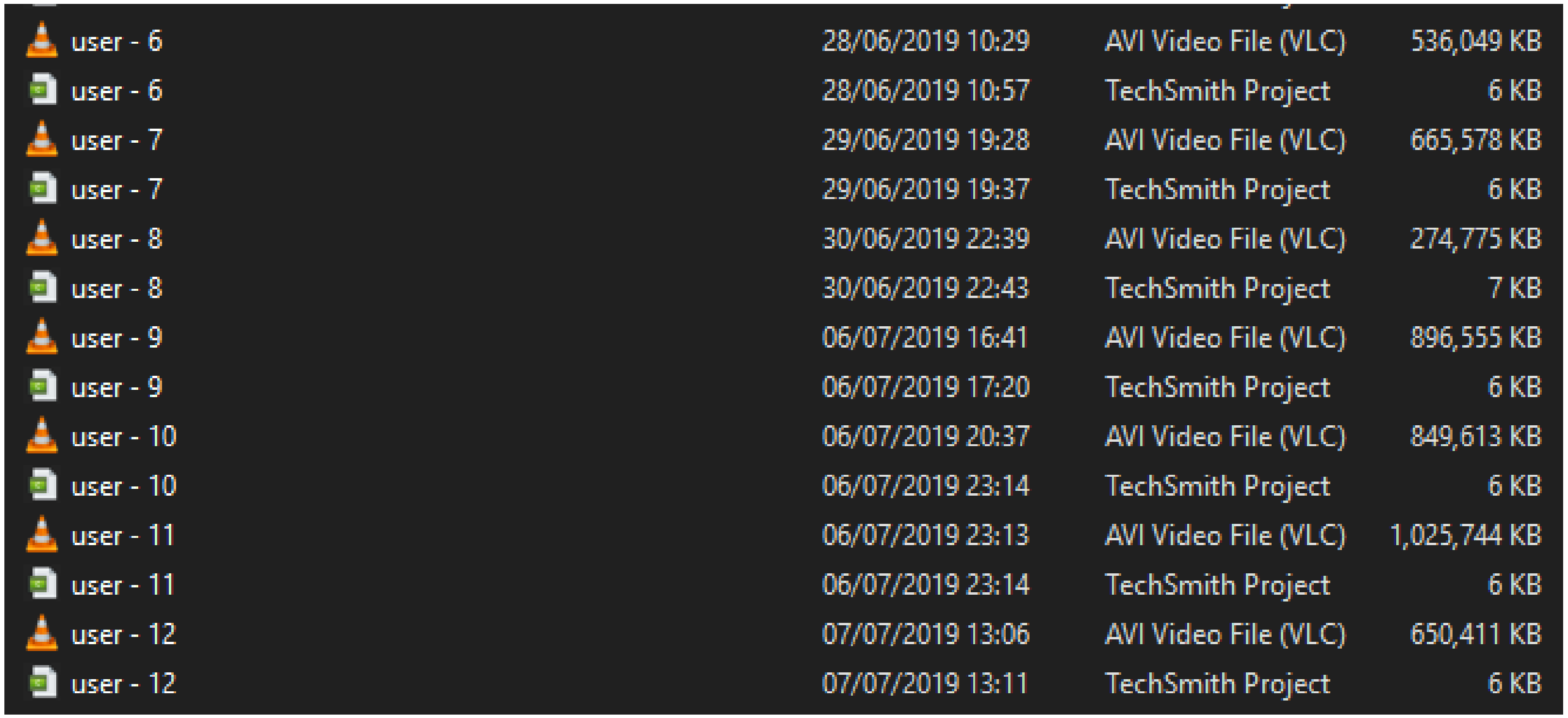
- Cognitive Walkthrough

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?

❖ Jakob Nielsen. “Usability inspection methods”. In: Conference companion on Human factors in computing systems. ACM. 1994, pp. 413–414.

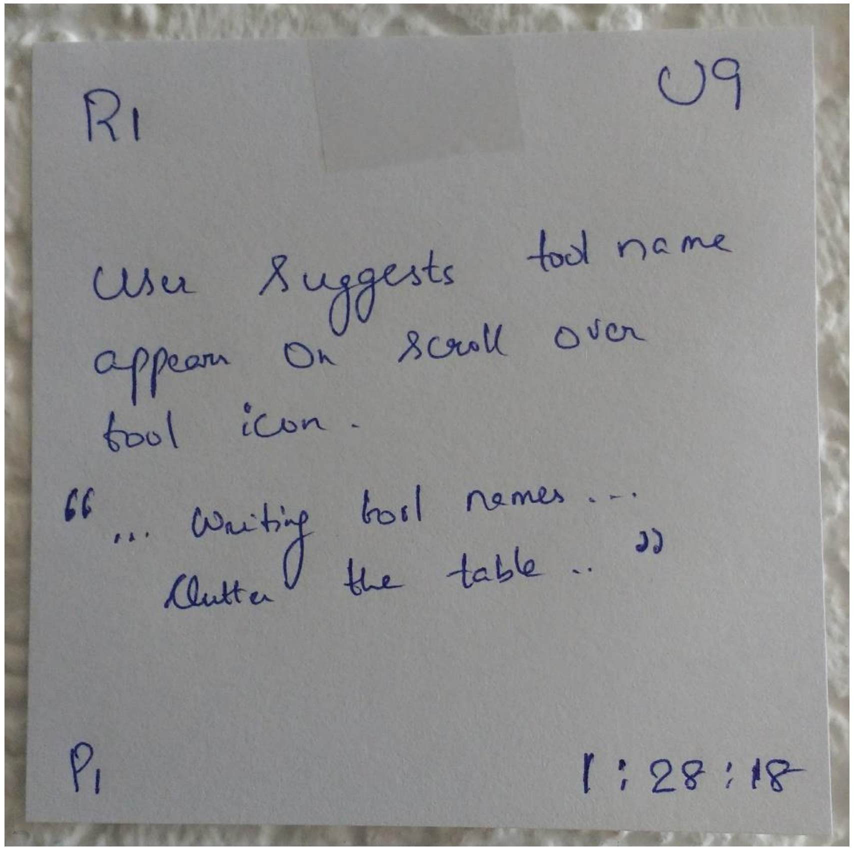
Evaluation - User Study (Example: UX 2 Raw Data)



user - 6	28/06/2019 10:29	AVI Video File (VLC)	536,049 KB
user - 6	28/06/2019 10:57	TechSmith Project	6 KB
user - 7	29/06/2019 19:28	AVI Video File (VLC)	665,578 KB
user - 7	29/06/2019 19:37	TechSmith Project	6 KB
user - 8	30/06/2019 22:39	AVI Video File (VLC)	274,775 KB
user - 8	30/06/2019 22:43	TechSmith Project	7 KB
user - 9	06/07/2019 16:41	AVI Video File (VLC)	896,555 KB
user - 9	06/07/2019 17:20	TechSmith Project	6 KB
user - 10	06/07/2019 20:37	AVI Video File (VLC)	849,613 KB
user - 10	06/07/2019 23:14	TechSmith Project	6 KB
user - 11	06/07/2019 23:13	AVI Video File (VLC)	1,025,744 KB
user - 11	06/07/2019 23:14	TechSmith Project	6 KB
user - 12	07/07/2019 13:06	AVI Video File (VLC)	650,411 KB
user - 12	07/07/2019 13:11	TechSmith Project	6 KB

Evaluation

- Affinity Notes



R1 U9

User suggests tool name appear on scroll over tool icon.

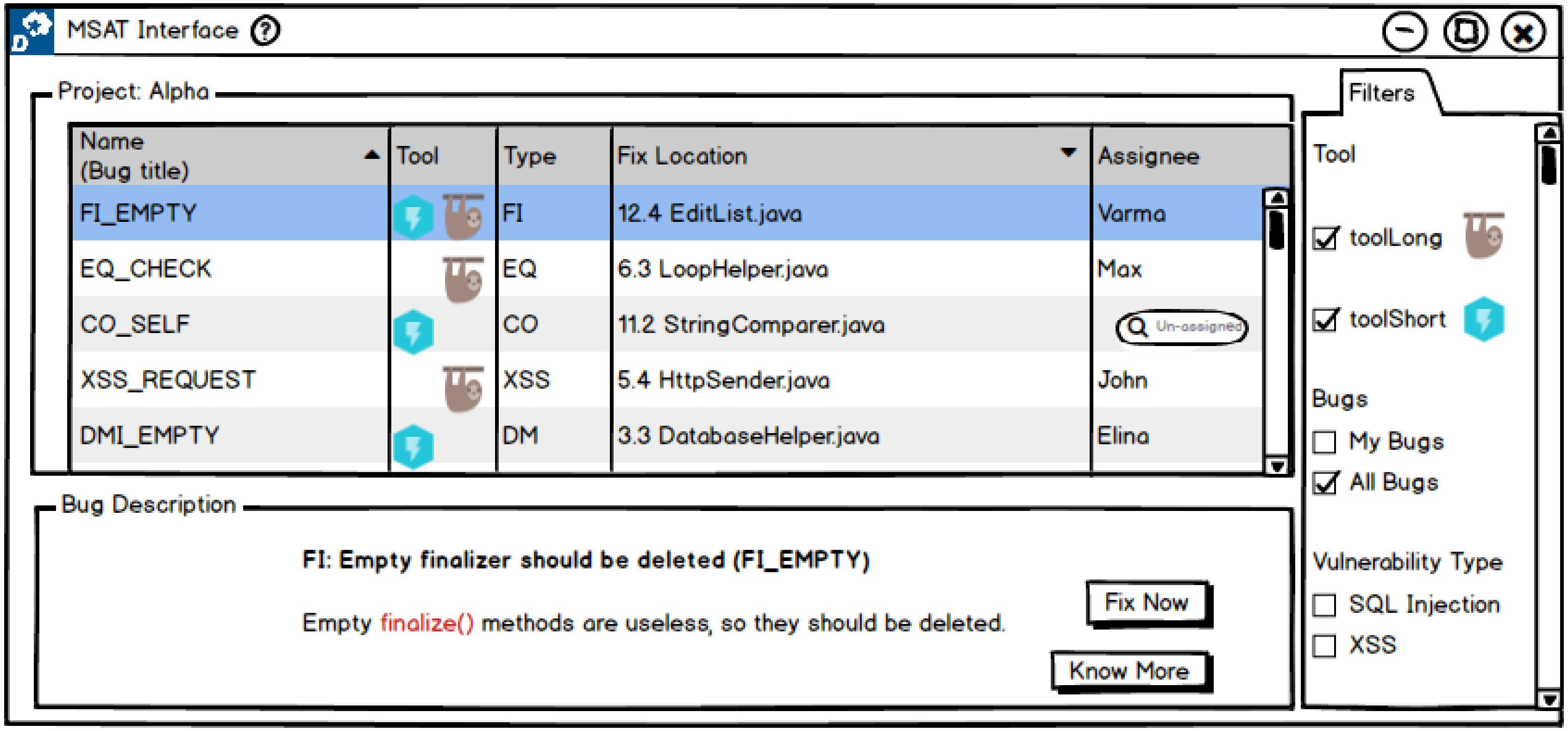
"... Writing tool names ... clutter the table .."

P1 1:28:18

HEINZ NIXDORF INSTITUT
UNIVERSITÄT PADERBORN

UX Design Cycle 1

[RQ 1] Does a separate list or single list help the user to identify the common bug?



Project: Alpha

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

Bug Description

FI: Empty finalizer should be deleted (FI_EMPTY)

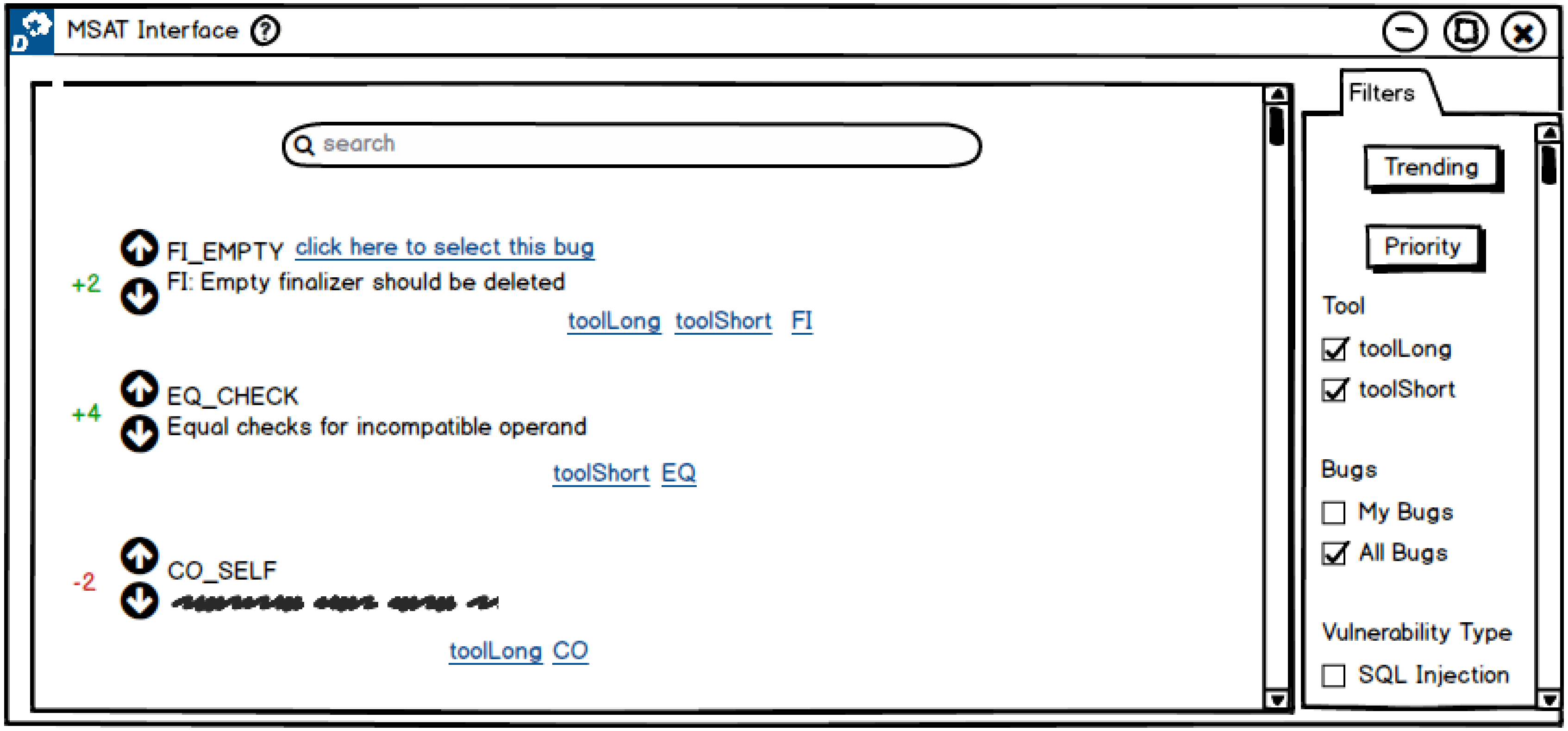
Empty finalize() methods are useless, so they should be deleted.

Tool
- ☑ toolLong
- ☑ toolShort

Bugs
- ☐ My Bugs
- ☑ All Bugs

Vulnerability Type
- ☐ SQL Injection
- ☐ XSS

[RQ 1] Will having tags help in scalability of bugs?



MSAT Interface

search

+2 FI_EMPTY click here to select this bug
FI: Empty finalizer should be deleted
toolLong toolShort FI

+4 EQ_CHECK
Equal checks for incompatible operand
toolShort EQ

-2 CO_SELF
toolLong CO

Filters

Trending

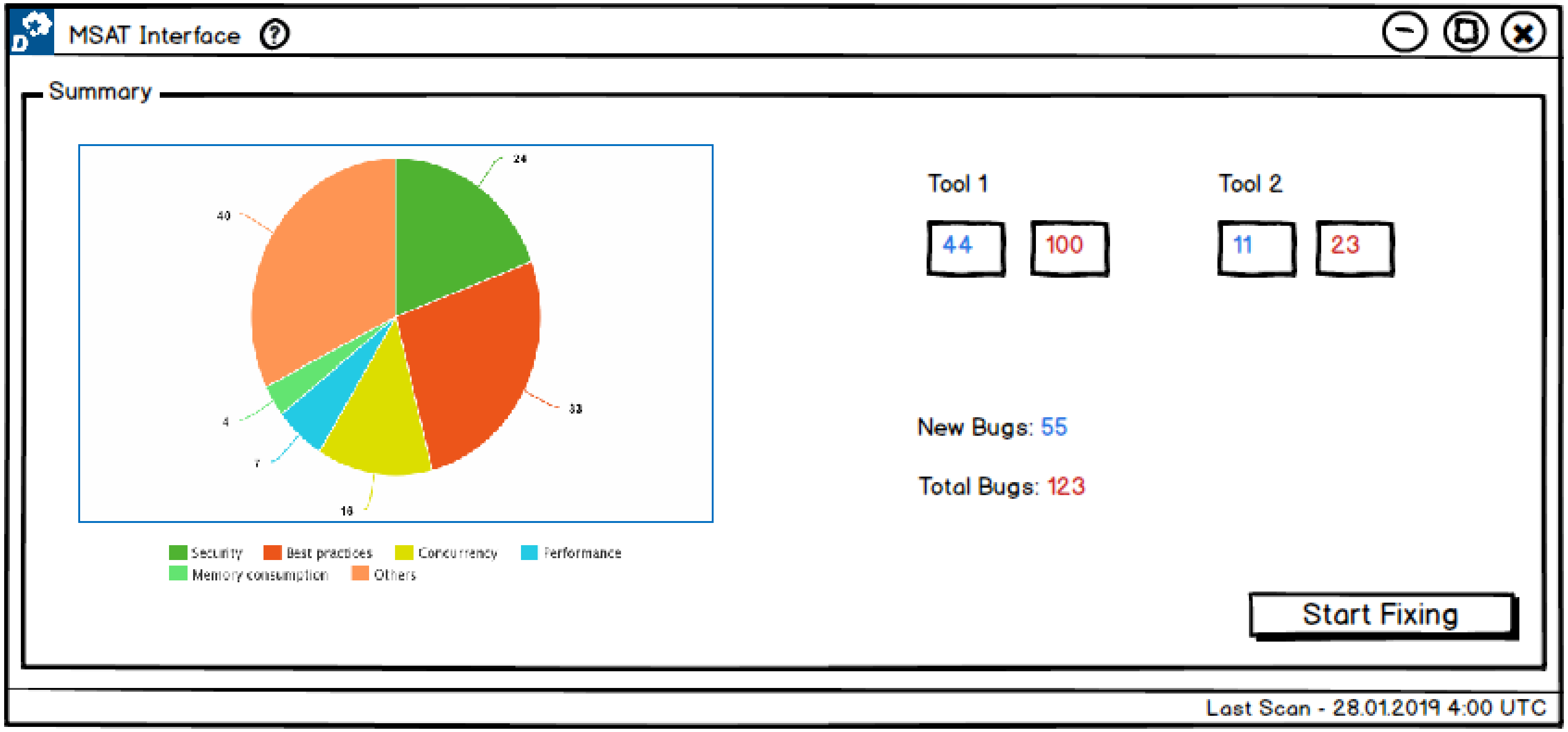
Priority

Tool
✓ toolLong
✓ toolShort

Bugs
☐ My Bugs
✓ All Bugs

Vulnerability Type
☐ SQL Injection

[RQ 1] Does the given statistics screen help the user in understating the analysis results overview?

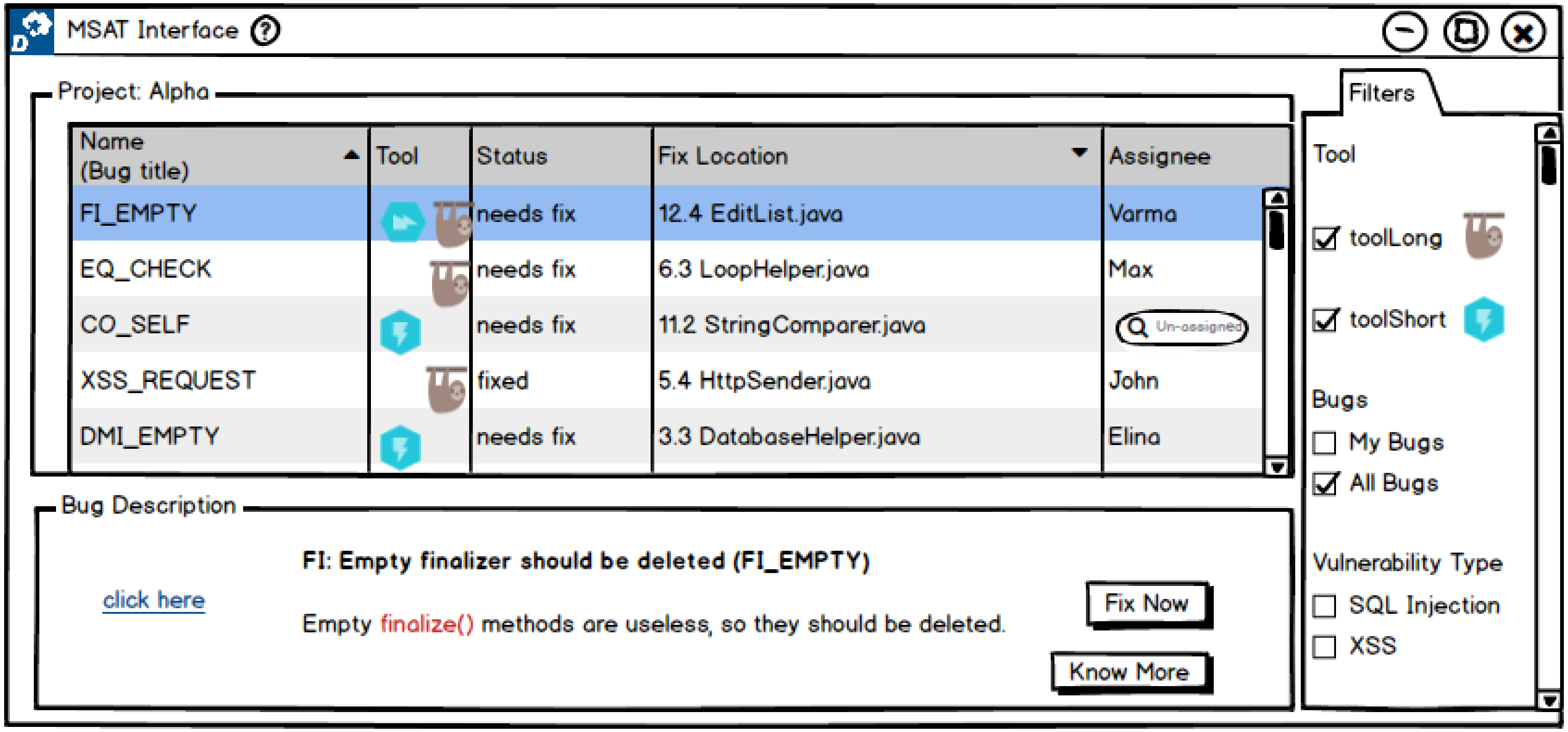


New Bugs: 55

Total Bugs: 123

Last Scan - 28.01.2019 4:00 UTC

[RQ 2] Will the animation (rotation) of icons for tools suffice the feedback required by the user?



MSAT Interface

Project: Alpha

Name (Bug title)	Tool	Status	Fix Location	Assignee
FI_EMPTY		needs fix	12.4 EditList.java	Varma
EQ_CHECK		needs fix	6.3 LoopHelper.java	Max
CO_SELF		needs fix	11.2 StringComparer.java	Un-assigned
XSS_REQUEST		fixed	5.4 HttpSender.java	John
DMI_EMPTY		needs fix	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

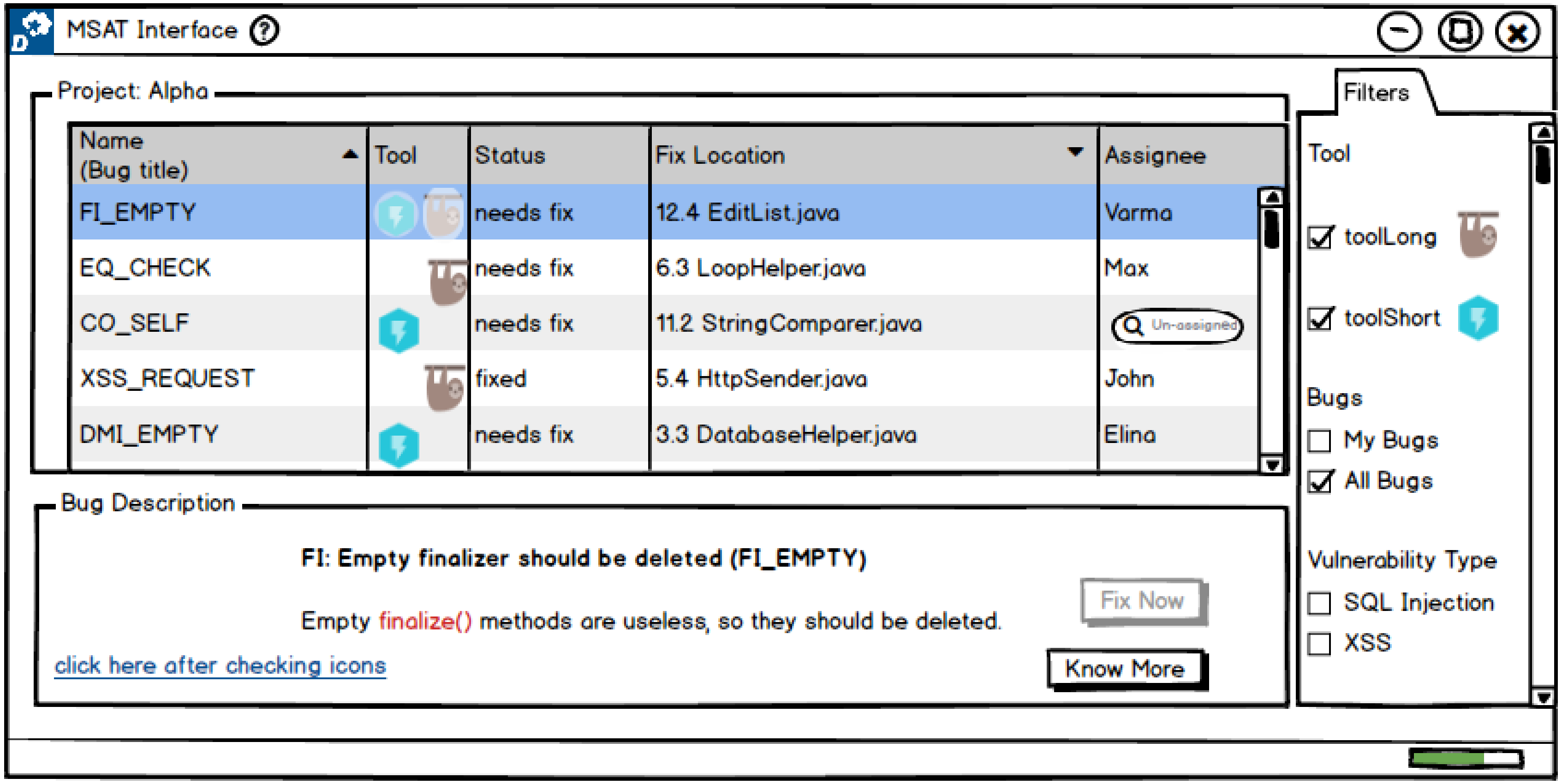
Bugs

- [ ] My Bugs
- [✓] All Bugs

Vulnerability Type

- [ ] SQL Injection
- [ ] XSS

[RQ 2] Will stating the progress of analysis for each tool be better than animation provided as feedback to the user?


MSAT Interface

Project: Alpha

Name (Bug title)	Tool	Status	Fix Location	Assignee
FI_EMPTY		needs fix	12.4 EditList.java	Varma
EQ_CHECK		needs fix	6.3 LoopHelper.java	Max
CO_SELF		needs fix	11.2 StringComparer.java	Un-assigned
XSS_REQUEST		fixed	5.4 HttpSender.java	John
DMI_EMPTY		needs fix	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.

click here after checking icons

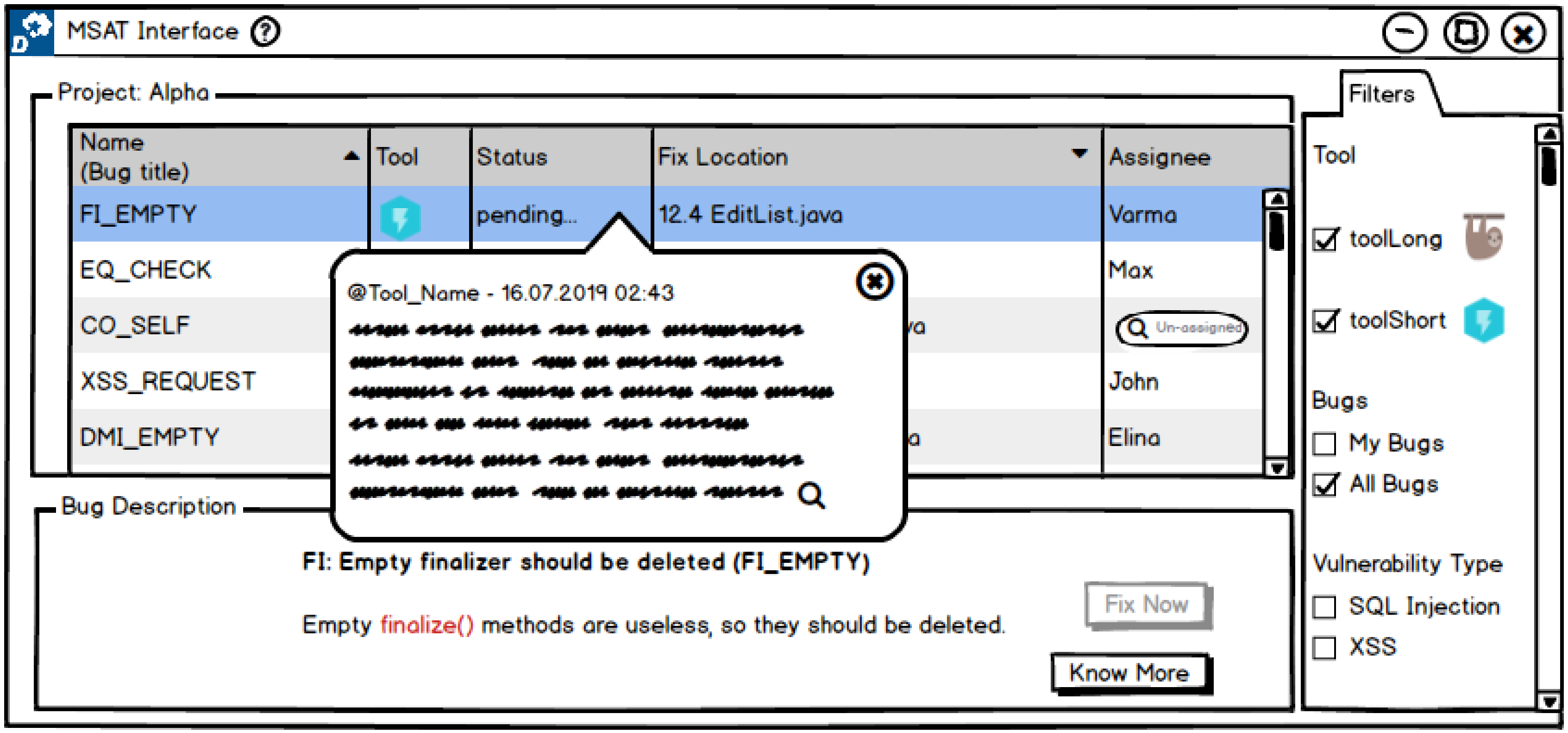
Filters

Tool
- [✓] toolLong
- [✓] toolShort

Bugs
- [ ] My Bugs
- [✓] All Bugs

Vulnerability Type
- [ ] SQL Injection
- [ ] XSS

[RQ 2] Does having more textual information with a popup feedback is required by the user?



MSAT Interface

Project: Alpha

Name (Bug title)	Tool	Status	Fix Location	Assignee
FI_EMPTY	⚡	pending...	12.4 EditList.java	Varma
EQ_CHECK				Max
CO_SELF			va	Un-assigned
XSS_REQUEST				John
DMI_EMPTY			a	Elina

@Tool_Name - 16.07.2019 02:43

Bug Description

FI: Empty finalizer should be deleted (FI_EMPTY)

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

Filters

Tool
- [x] toolLong
- [x] toolShort

Bugs
- [ ] My Bugs
- [x] All Bugs

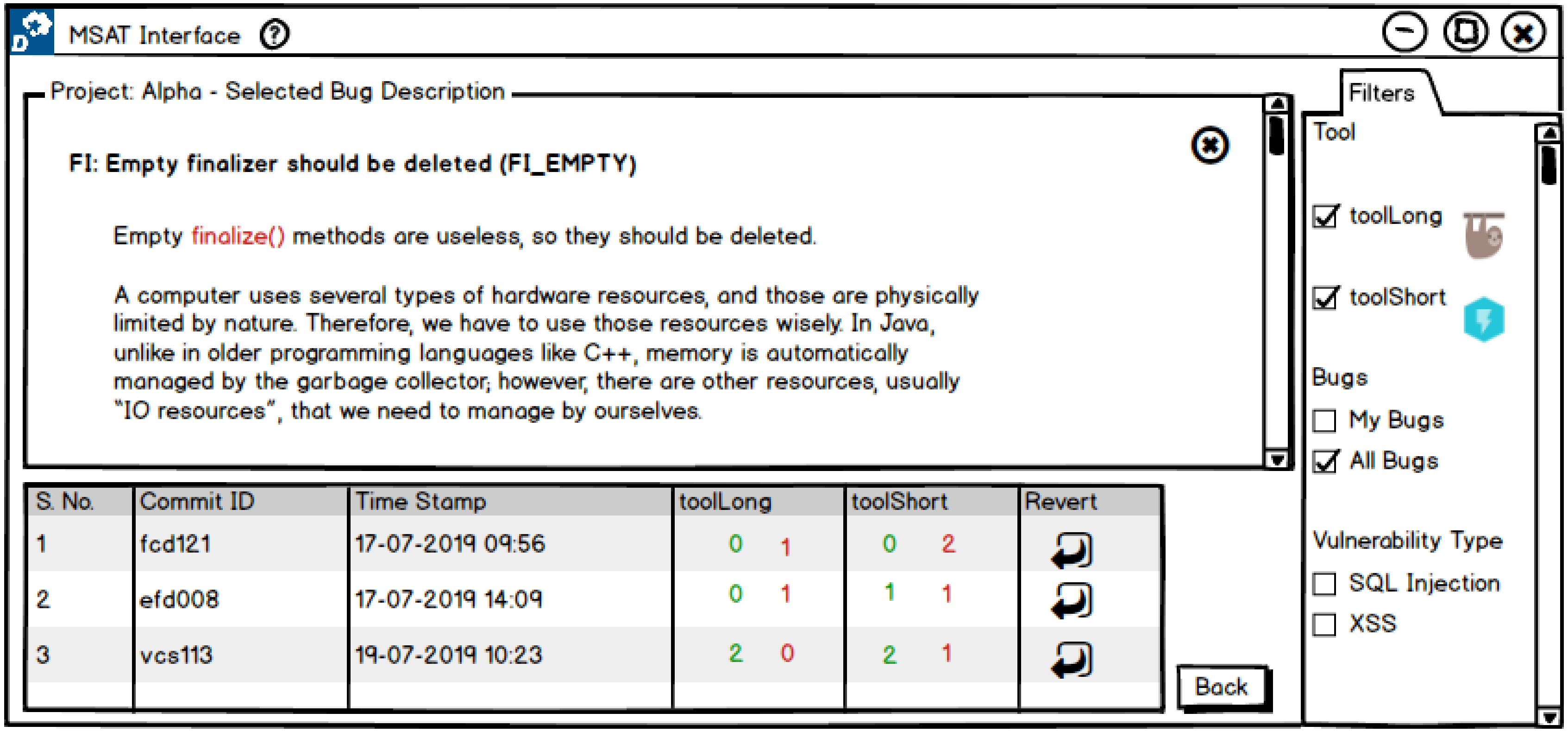
Vulnerability Type
- [ ] SQL Injection
- [ ] XSS

[RQ 2] Do users require multiple feedbacks, i.e., any combination of animated icons, progress bar or pending status popup?

- All the 3 feedback features

- Animated Icons – What bugs are being analysed?
- Progress Bar – How far the bugs got analysed?
- Pending Status – more information on analysis

[RQ 3] Whether the given UI, i.e., previous commits in the process of fixing a bug-finding with numbers determining the adding or removing of other bugs be able to address the scenario from the user perspective?



Project: Alpha - Selected Bug Description

FI: Empty finalizer should be deleted (FI_EMPTY)

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

A computer uses several types of hardware resources, and those are physically limited by nature. Therefore, we have to use those resources wisely. In Java, unlike in older programming languages like C++, memory is automatically managed by the garbage collector; however, there are other resources, usually "IO resources", that we need to manage by ourselves.

S. No.	Commit ID	Time Stamp	toolLong	toolShort	Revert
1	fcd121	17-07-2019 09:56	0 1	0 2	↩
2	efd008	17-07-2019 14:09	0 1	1 1	↩
3	vcs113	19-07-2019 10:23	2 0	2 1	↩

Tool
☑ toolLong
☑ toolShort

Bugs
☐ My Bugs
☑ All Bugs

Vulnerability Type
☐ SQL Injection
☐ XSS

UX 1 – Lessons

- Analysis / Results View ✓

- Improvisations for next UX cycle:

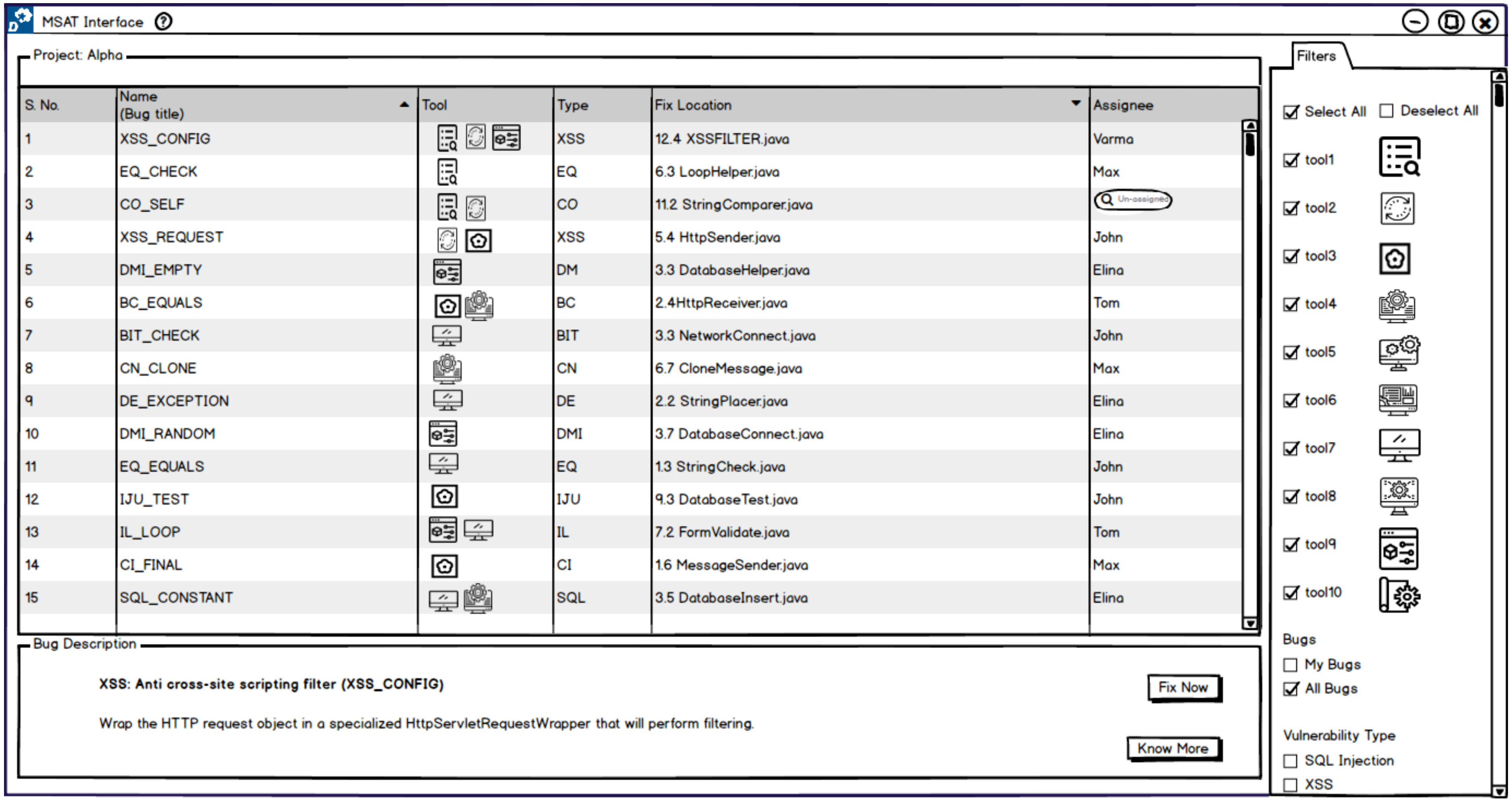
  - Increase code base
  - Volume of bugs (+ scroll)
  - Integrate more tools
  - Code view perspective
  - + new sub RQ’s

UX Design Cycle 2

UX 2

	Analysis View	**Code View**
RQ 1 (display)	• Single List • Separate List • Tags	• Separate Icons • Single Icon
RQ 2 (feedback)	• Animated Icons • Progress Bar • Popup	• Toasts (alerts) • Spinner (status)
RQ 3 (trace)	• Numbers • Adjectives	• Before/After

[RQ 1] From analysis view perspective, does a separate list or single list help the user to identify the common bug?



Project: Alpha

S. No.	Name (Bug title)	Tool	Type	Fix Location	Assignee
1	XSS_CONFIG		XSS	12.4 XSSFILTER.java	Varma
2	EQ_CHECK		EQ	6.3 LoopHelper.java	Max
3	CO_SELF		CO	11.2 StringComparer.java	Un-assigned
4	XSS_REQUEST		XSS	5.4 HttpSender.java	John
5	DMI_EMPTY		DM	3.3 DatabaseHelper.java	Elina
6	BC_EQUALS		BC	2.4HttpReceiver.java	Tom
7	BIT_CHECK		BIT	3.3 NetworkConnect.java	John
8	CN_CLONE		CN	6.7 CloneMessage.java	Max
9	DE_EXCEPTION		DE	2.2 StringPlacer.java	Elina
10	DMI_RANDOM		DMI	3.7 DatabaseConnect.java	Elina
11	EQ_EQUALS		EQ	1.3 StringCheck.java	John
12	IJU_TEST		IJU	9.3 DatabaseTest.java	John
13	IL_LOOP		IL	7.2 FormValidate.java	Tom
14	CI_FINAL		CI	1.6 MessageSender.java	Max
15	SQL_CONSTANT		SQL	3.5 DatabaseInsert.java	Elina

Bug Description

XSS: Anti cross-site scripting filter (XSS_CONFIG)

Wrap the HTTP request object in a specialized HttpServletRequestWrapper that will perform filtering.

Filters

[✓ Select All] Deselect All

[✓ tool1]
[✓ tool2]
[✓ tool3]
[✓ tool4]
[✓ tool5]
[✓ tool6]
[✓ tool7]
[✓ tool8]
[✓ tool9]
[✓ tool10]

Bugs

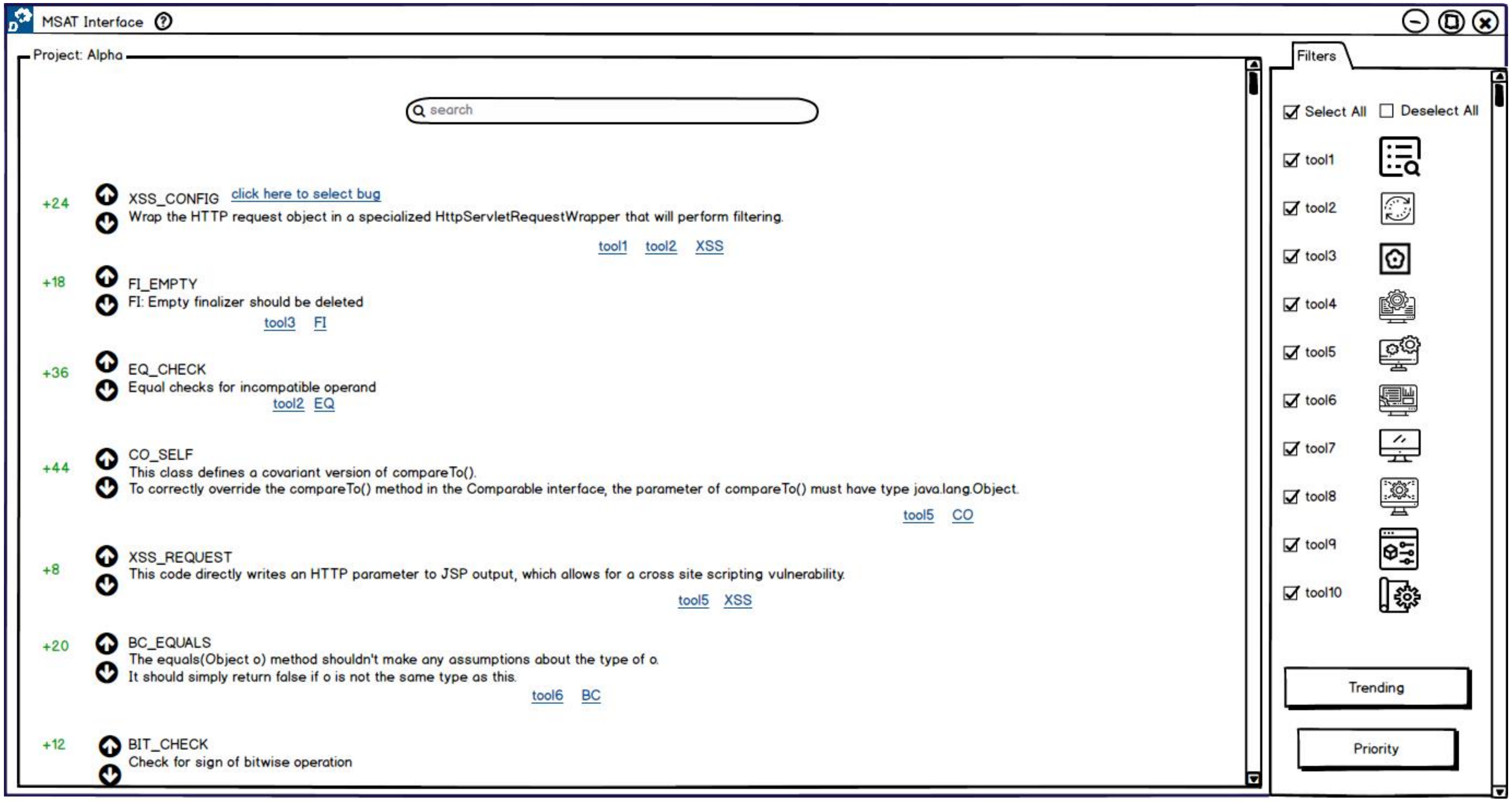
My Bugs
[✓ All Bugs]

Vulnerability Type

SQL Injection
XSS

HEINZ NIXDORF INSTITUT
UNIVERSITÄT PADERBORN

[RQ 1] From analysis view perspective, will tags help in scalability of bug results in comparison to separate list or single list?



+24 XSS_CONFIG click here to select bug
Wrap the HTTP request object in a specialized HttpServletRequestWrapper that will perform filtering.
tool1 tool2 XSS

+18 FI_EMPTY
FI: Empty finalizer should be deleted
tool3 FI

+36 EQ_CHECK
Equal checks for incompatible operand
tool2 EQ

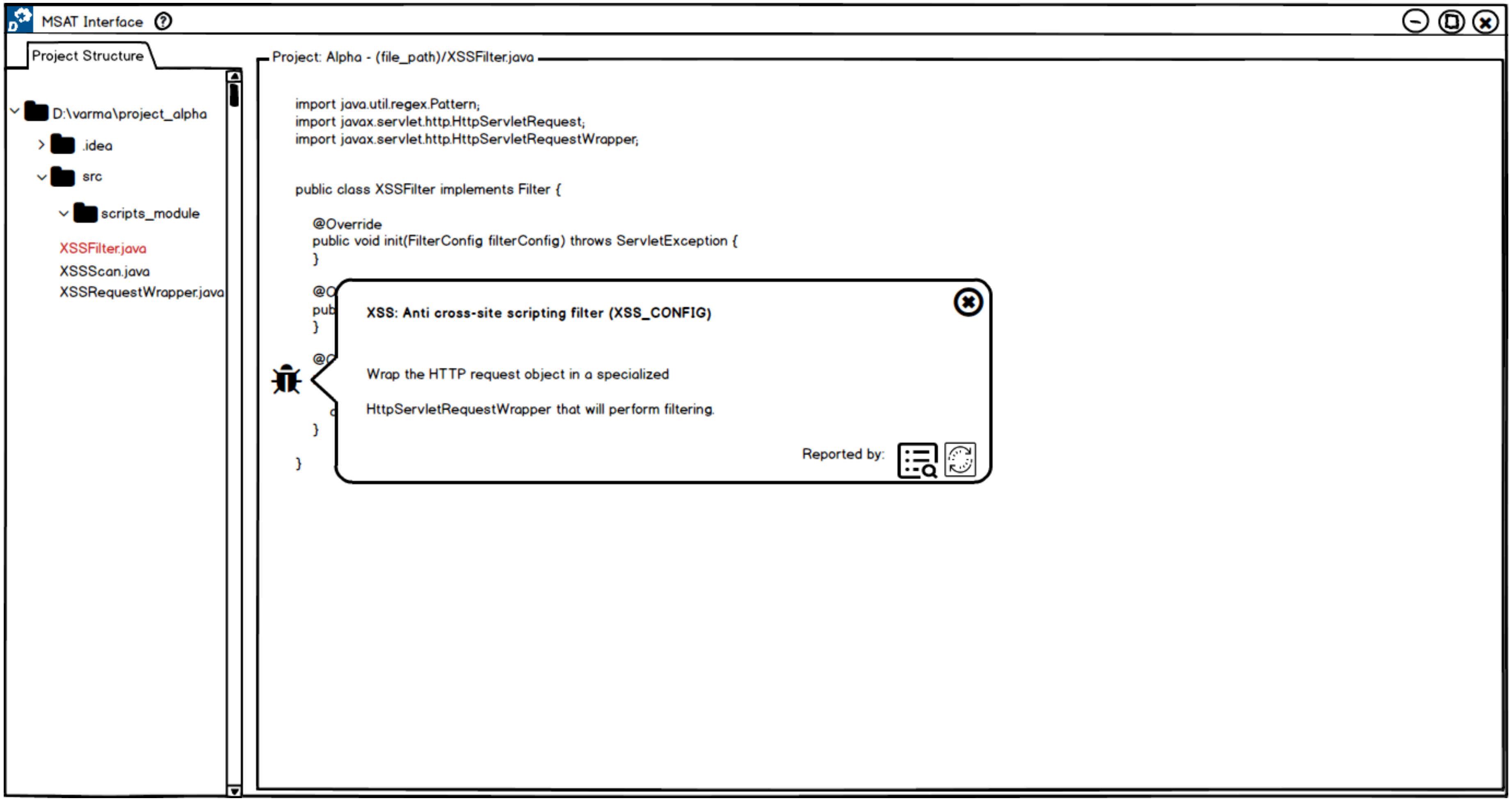
+44 CO_SELF
This class defines a covariant version of compareTo().
To correctly override the compareTo() method in the Comparable interface, the parameter of compareTo() must have type java.lang.Object.
tool5 CO

+8 XSS_REQUEST
This code directly writes an HTTP parameter to JSP output, which allows for a cross site scripting vulnerability.
tool5 XSS

+20 BC_EQUALS
The equals(Object o) method shouldn't make any assumptions about the type of o.
It should simply return false if o is not the same type as this.
tool6 BC

+12 BIT_CHECK
Check for sign of bitwise operation

[RQ 1] From code view perspective, will single icon suffice the showing of different tools icons?



```
import java.util.regex.Pattern;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;

public class XSSFilter implements Filter {

    @Override
    public void init(FilterConfig filterConfig) throws ServletException {
    }

    @O
    pub
    }

    @O

    }

}
```

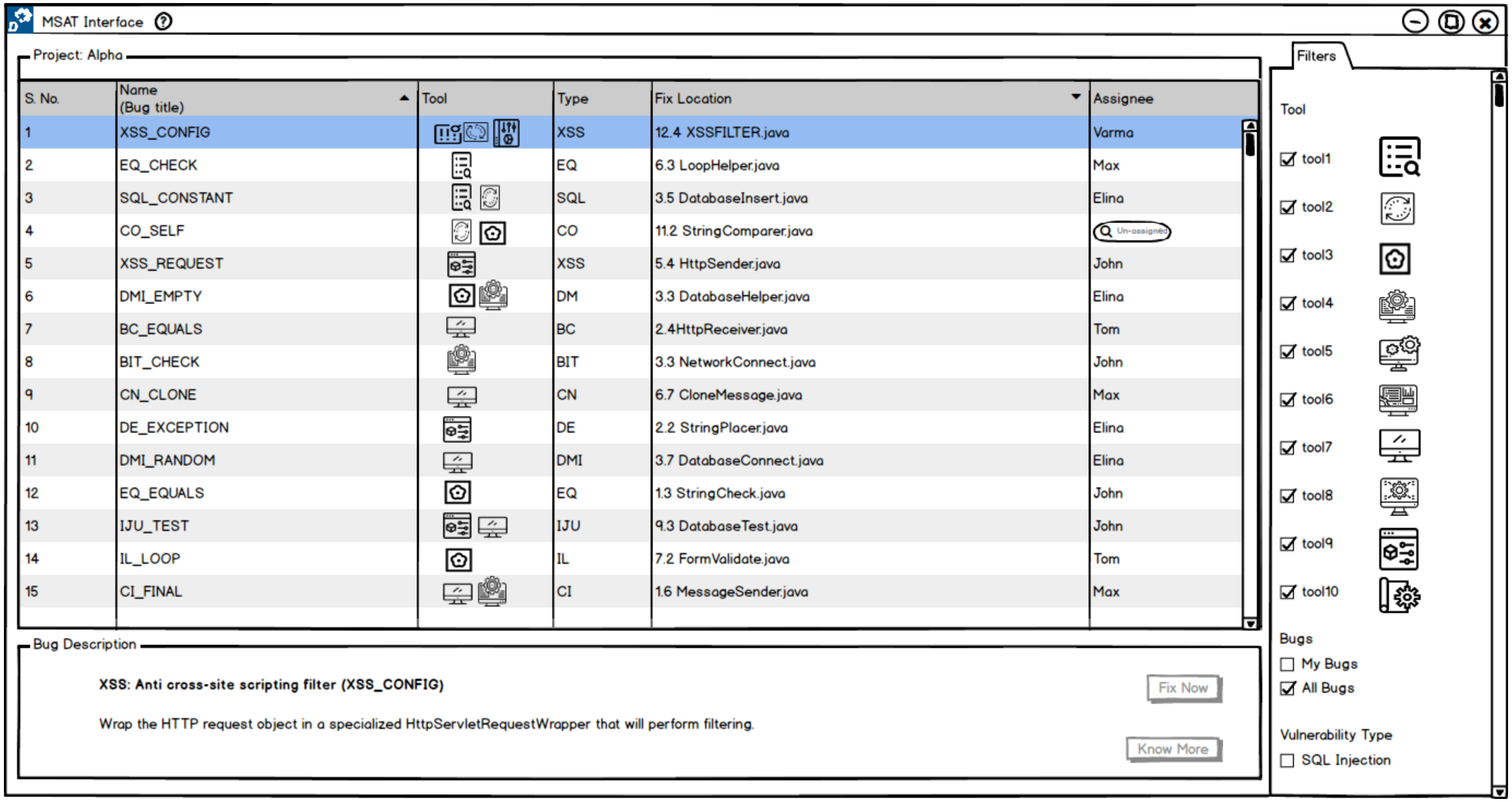
XSS: Anti cross-site scripting filter (XSS_CONFIG)

Wrap the HTTP request object in a specialized

HttpServletRequestWrapper that will perform filtering.

Reported by:

[RQ 2] When submitting the bug for analysis, what feedback does user feel convenient among animation, progress bar or popup?



MSAT Interface

Project: Alpha

S. No.	Name (Bug title)	Tool	Type	Fix Location	Assignee
1	XSS_CONFIG		XSS	12.4 XSSFILTER.java	Varma
2	EQ_CHECK		EQ	6.3 LoopHelper.java	Max
3	SQL_CONSTANT		SQL	3.5 DatabaseInsert.java	Elina
4	CO_SELF		CO	11.2 StringComparer.java	Un-assigned
5	XSS_REQUEST		XSS	5.4 HttpSender.java	John
6	DMI_EMPTY		DM	3.3 DatabaseHelper.java	Elina
7	BC_EQUALS		BC	2.4HttpReceiver.java	Tom
8	BIT_CHECK		BIT	3.3 NetworkConnect.java	John
9	CN_CLONE		CN	6.7 CloneMessage.java	Max
10	DE_EXCEPTION		DE	2.2 StringPlacer.java	Elina
11	DMI_RANDOM		DMI	3.7 DatabaseConnect.java	Elina
12	EQ_EQUALS		EQ	1.3 StringCheck.java	John
13	IJU_TEST		IJU	9.3 DatabaseTest.java	John
14	IL_LOOP		IL	7.2 FormValidate.java	Tom
15	CI_FINAL		CI	1.6 MessageSender.java	Max

Bug Description

XSS: Anti cross-site scripting filter (XSS_CONFIG)

Wrap the HTTP request object in a specialized HttpServletRequestWrapper that will perform filtering.

Filters

Tool

- [✓] tool1
- [✓] tool2
- [✓] tool3
- [✓] tool4
- [✓] tool5
- [✓] tool6
- [✓] tool7
- [✓] tool8
- [✓] tool9
- [✓] tool10

Bugs

- [ ] My Bugs
- [✓] All Bugs

Vulnerability Type

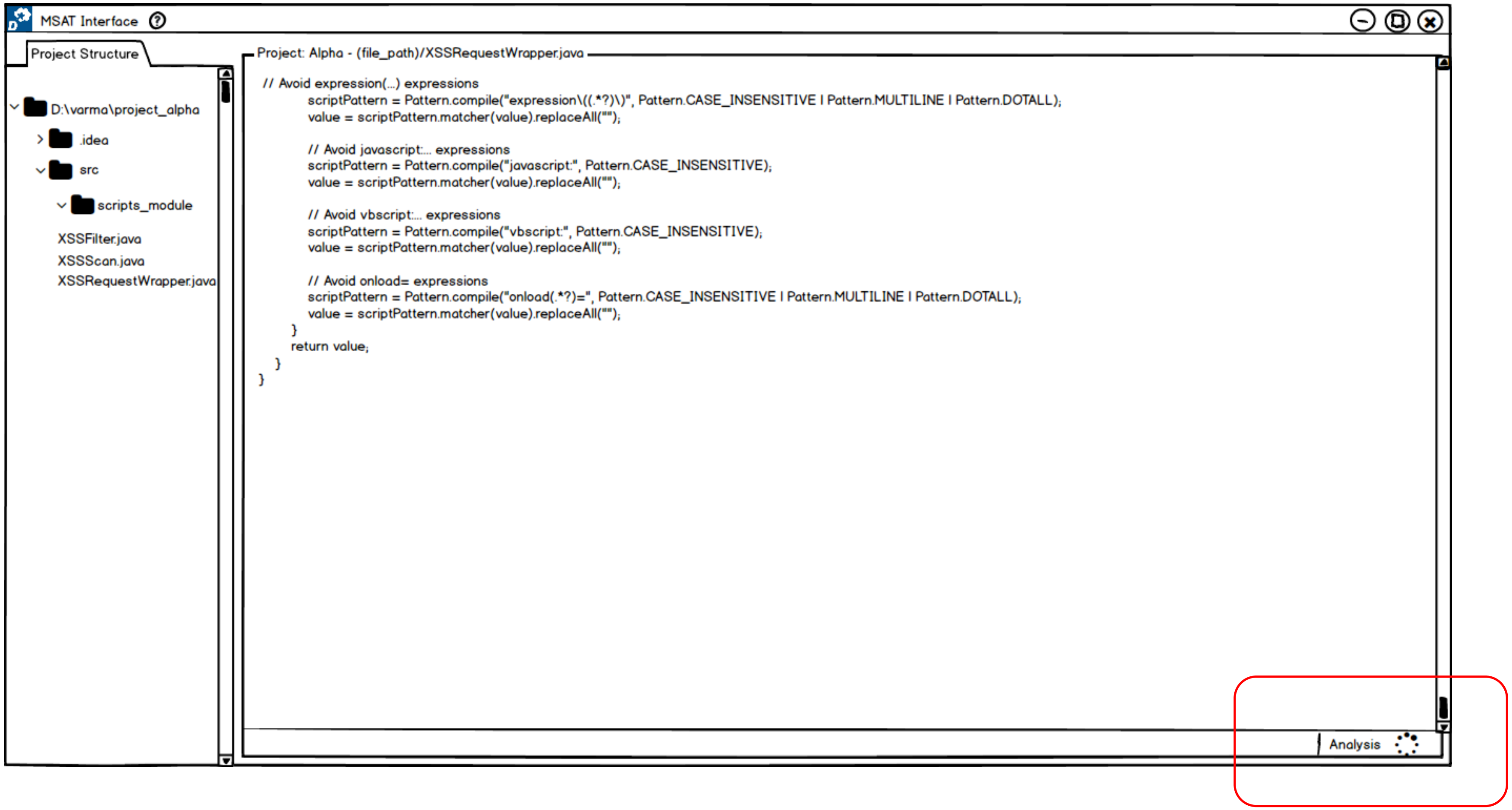
- [ ] SQL Injection

HEINZ NIXDORF INSTITUT
UNIVERSITÄT PADERBORN

[RQ 2] Does a single type of feedback suffice or requires combination?

All 3 solution ideas as each could be depict different understandings.

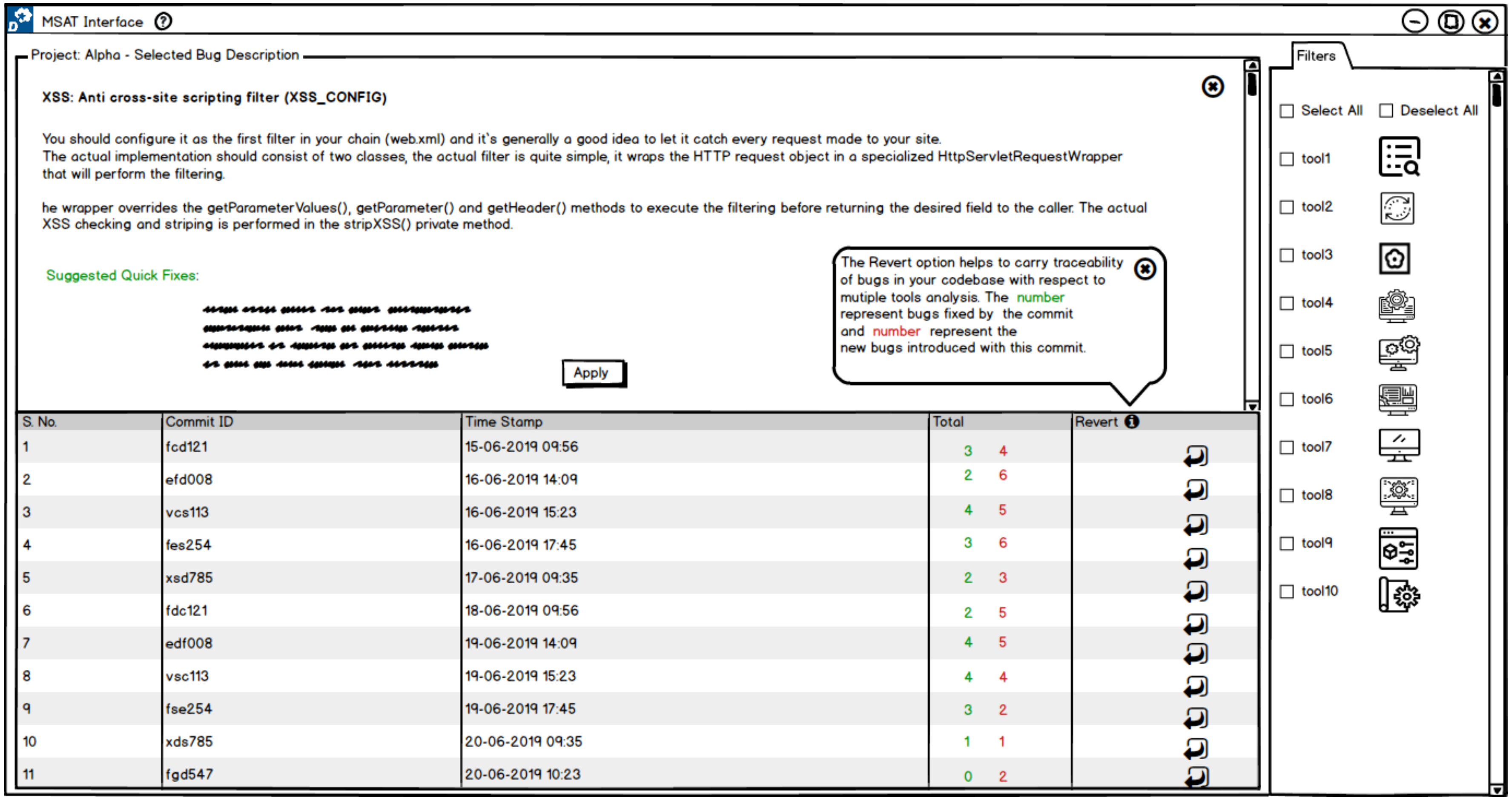
[RQ 2] From code view perspective, i.e., once user fixed a bug and submitted for analysis and then off the analysis results screen, then is popup notifications with analysis progress information better to busy status (spinner)?



```
// Avoid expression(...) expressions
        scriptPattern = Pattern.compile("expression\((.*?)\)", Pattern.CASE_INSENSITIVE | Pattern.MULTILINE | Pattern.DOTALL);
        value = scriptPattern.matcher(value).replaceAll("");

        // Avoid javascript:... expressions
        scriptPattern = Pattern.compile("javascript:", Pattern.CASE_INSENSITIVE);
        value = scriptPattern.matcher(value).replaceAll("");

        // Avoid vbscript:... expressions
        scriptPattern = Pattern.compile("vbscript:", Pattern.CASE_INSENSITIVE);
        value = scriptPattern.matcher(value).replaceAll("");

        // Avoid onload= expressions
        scriptPattern = Pattern.compile("onload(.*?)=", Pattern.CASE_INSENSITIVE | Pattern.MULTILINE | Pattern.DOTALL);
        value = scriptPattern.matcher(value).replaceAll("");
      }
      return value;
   }
}
```

HEINZ NIXDORF INSTITUT
UNIVERSITÄT PADERBORN

[RQ 3] In tracing, will the user need to know the changes made to fix a bug affecting the analysis of other tools?



Project: Alpha - Selected Bug Description

XSS: Anti cross-site scripting filter (XSS_CONFIG)

You should configure it as the first filter in your chain (web.xml) and it's generally a good idea to let it catch every request made to your site.
The actual implementation should consist of two classes, the actual filter is quite simple, it wraps the HTTP request object in a specialized HttpServletRequestWrapper that will perform the filtering.

he wrapper overrides the getParameterValues(), getParameter() and getHeader() methods to execute the filtering before returning the desired field to the caller. The actual XSS checking and striping is performed in the stripXSS() private method.

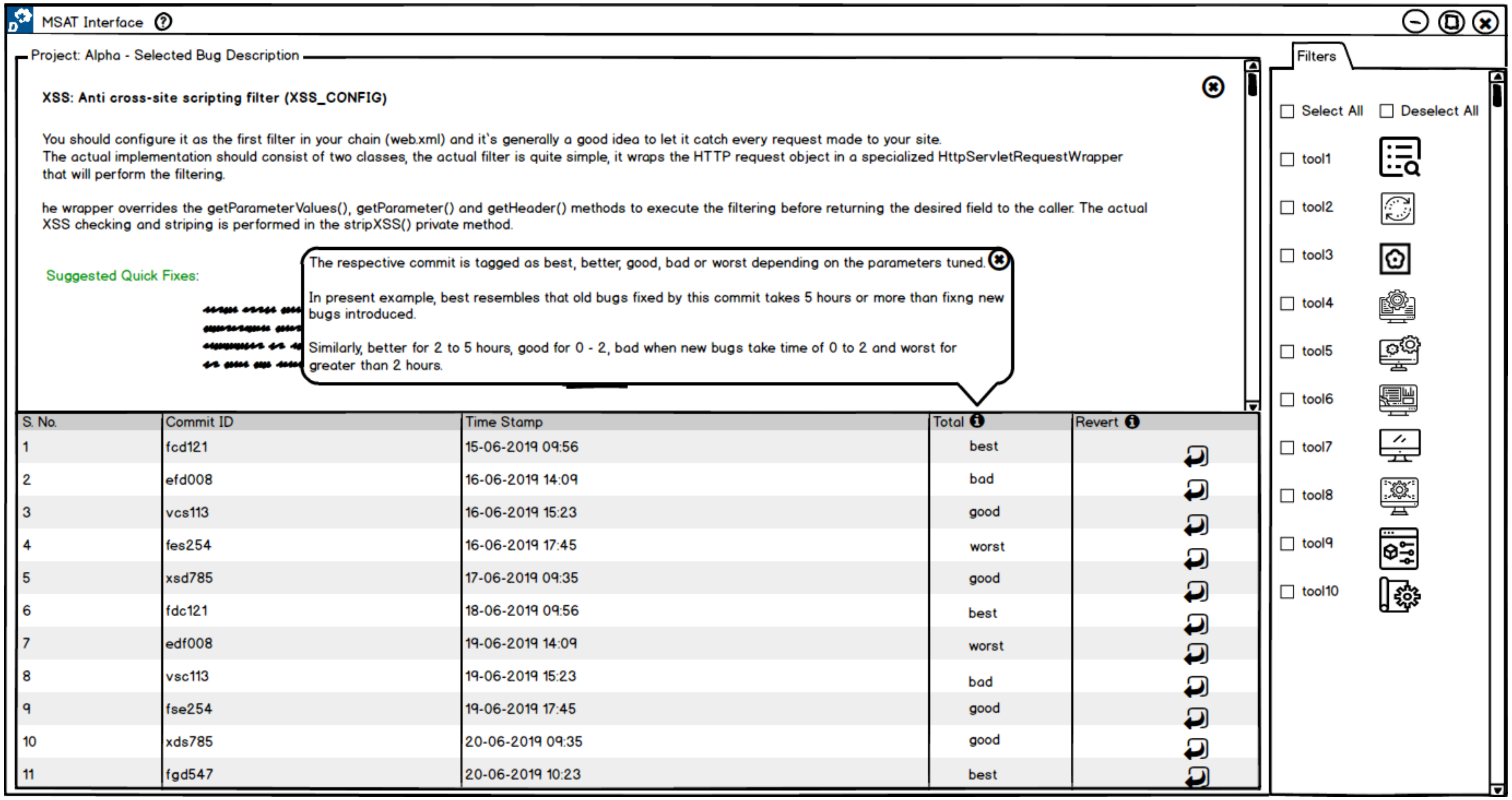
Suggested Quick Fixes:

[illegible]

The Revert option helps to carry traceability of bugs in your codebase with respect to mutiple tools analysis. The number represent bugs fixed by the commit and number represent the new bugs introduced with this commit.

S. No.	Commit ID	Time Stamp	Total		Revert
1	fcd121	15-06-2019 09:56	3	4	↩
2	efd008	16-06-2019 14:09	2	6	↩
3	vcs113	16-06-2019 15:23	4	5	↩
4	fes254	16-06-2019 17:45	3	6	↩
5	xsd785	17-06-2019 09:35	2	3	↩
6	fdc121	18-06-2019 09:56	2	5	↩
7	edf008	19-06-2019 14:09	4	5	↩
8	vsc113	19-06-2019 15:23	4	4	↩
9	fse254	19-06-2019 17:45	3	2	↩
10	xds785	20-06-2019 09:35	1	1	↩
11	fgd547	20-06-2019 10:23	0	2	↩

[RQ 3] Does adjective mapping ease the user to trace the changes made in code in terms of bugs existence?



Project: Alpha - Selected Bug Description

XSS: Anti cross-site scripting filter (XSS_CONFIG)

You should configure it as the first filter in your chain (web.xml) and it`s generally a good idea to let it catch every request made to your site.
The actual implementation should consist of two classes, the actual filter is quite simple, it wraps the HTTP request object in a specialized HttpServletRequestWrapper that will perform the filtering.

he wrapper overrides the getParameterValues(), getParameter() and getHeader() methods to execute the filtering before returning the desired field to the caller. The actual XSS checking and striping is performed in the stripXSS() private method.

Suggested Quick Fixes:

The respective commit is tagged as best, better, good, bad or worst depending on the parameters tuned.

In present example, best resembles that old bugs fixed by this commit takes 5 hours or more than fixng new bugs introduced.

Similarly, better for 2 to 5 hours, good for 0 - 2, bad when new bugs take time of 0 to 2 and worst for greater than 2 hours.

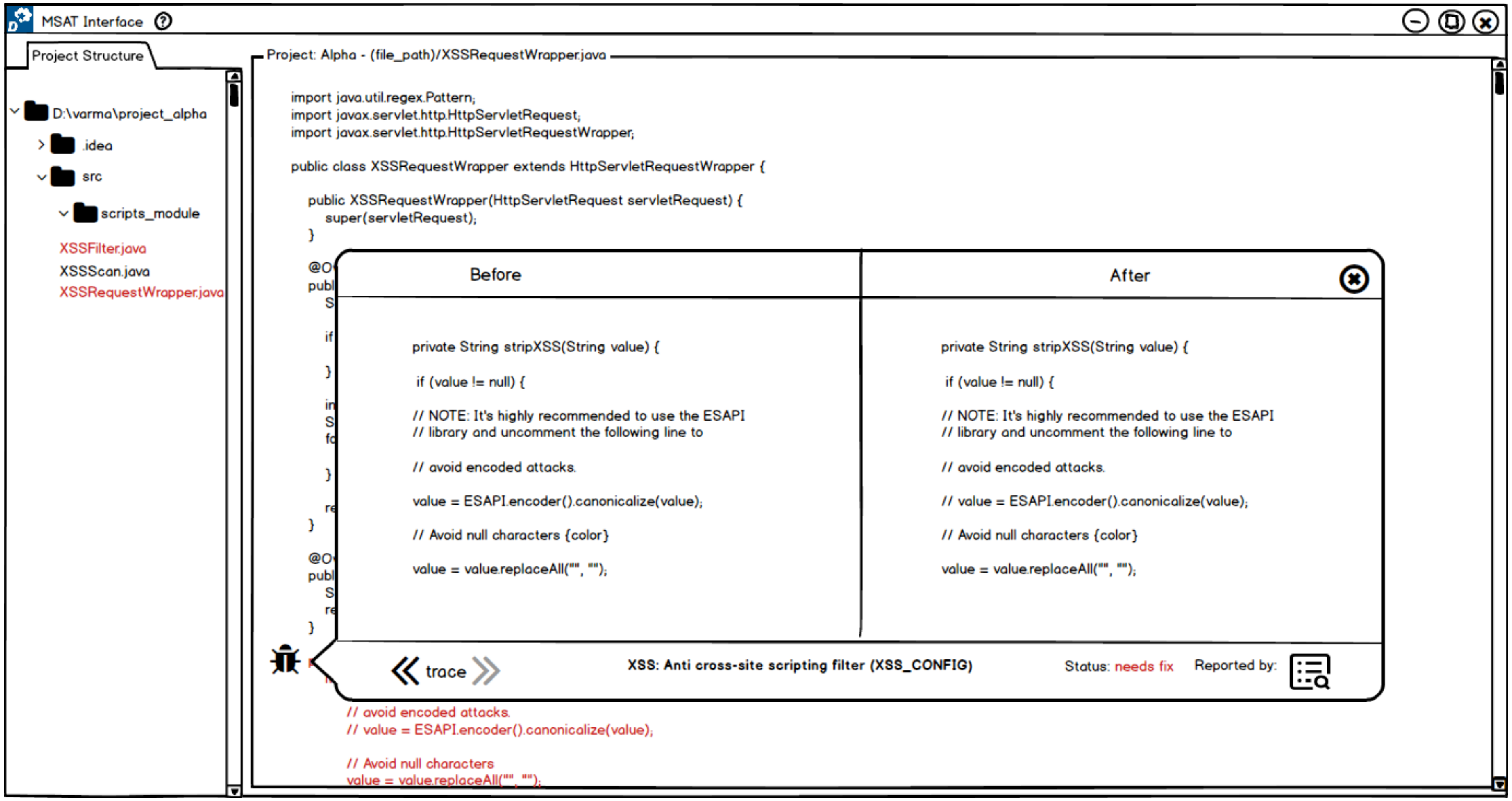
S. No.	Commit ID	Time Stamp	Total	Revert
1	fcd121	15-06-2019 09:56	best	↩
2	efd008	16-06-2019 14:09	bad	↩
3	vcs113	16-06-2019 15:23	good	↩
4	fes254	16-06-2019 17:45	worst	↩
5	xsd785	17-06-2019 09:35	good	↩
6	fdc121	18-06-2019 09:56	best	↩
7	edf008	19-06-2019 14:09	worst	↩
8	vsc113	19-06-2019 15:23	bad	↩
9	fse254	19-06-2019 17:45	good	↩
10	xds785	20-06-2019 09:35	good	↩
11	fgd547	20-06-2019 10:23	best	↩

Filters

☐ Select All ☐ Deselect All

☐ tool1
☐ tool2
☐ tool3
☐ tool4
☐ tool5
☐ tool6
☐ tool7
☐ tool8
☐ tool9
☐ tool10

[RQ 3] From code view perspective, will the bug tool icons with before/after code help understand the user in easing to fix it?



```
import java.util.regex.Pattern;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;

public class XSSRequestWrapper extends HttpServletRequestWrapper {

   public XSSRequestWrapper(HttpServletRequest servletRequest) {
      super(servletRequest);
   }
```

Before:

```
private String stripXSS(String value) {

 if (value != null) {

// NOTE: It's highly recommended to use the ESAPI
// library and uncomment the following line to

// avoid encoded attacks.

value = ESAPI.encoder().canonicalize(value);

// Avoid null characters {color}

value = value.replaceAll("", "");
```

After:

```
private String stripXSS(String value) {

 if (value != null) {

// NOTE: It's highly recommended to use the ESAPI
// library and uncomment the following line to

// avoid encoded attacks.

// value = ESAPI.encoder().canonicalize(value);

// Avoid null characters {color}

value = value.replaceAll("", "");
```

XSS: Anti cross-site scripting filter (XSS_CONFIG)

```
// avoid encoded attacks.
// value = ESAPI.encoder().canonicalize(value);

// Avoid null characters
value = value.replaceAll("", "");
```

UX 2 – Lessons

- Analysis / Results View
- Code View
- UX 1 Scalability

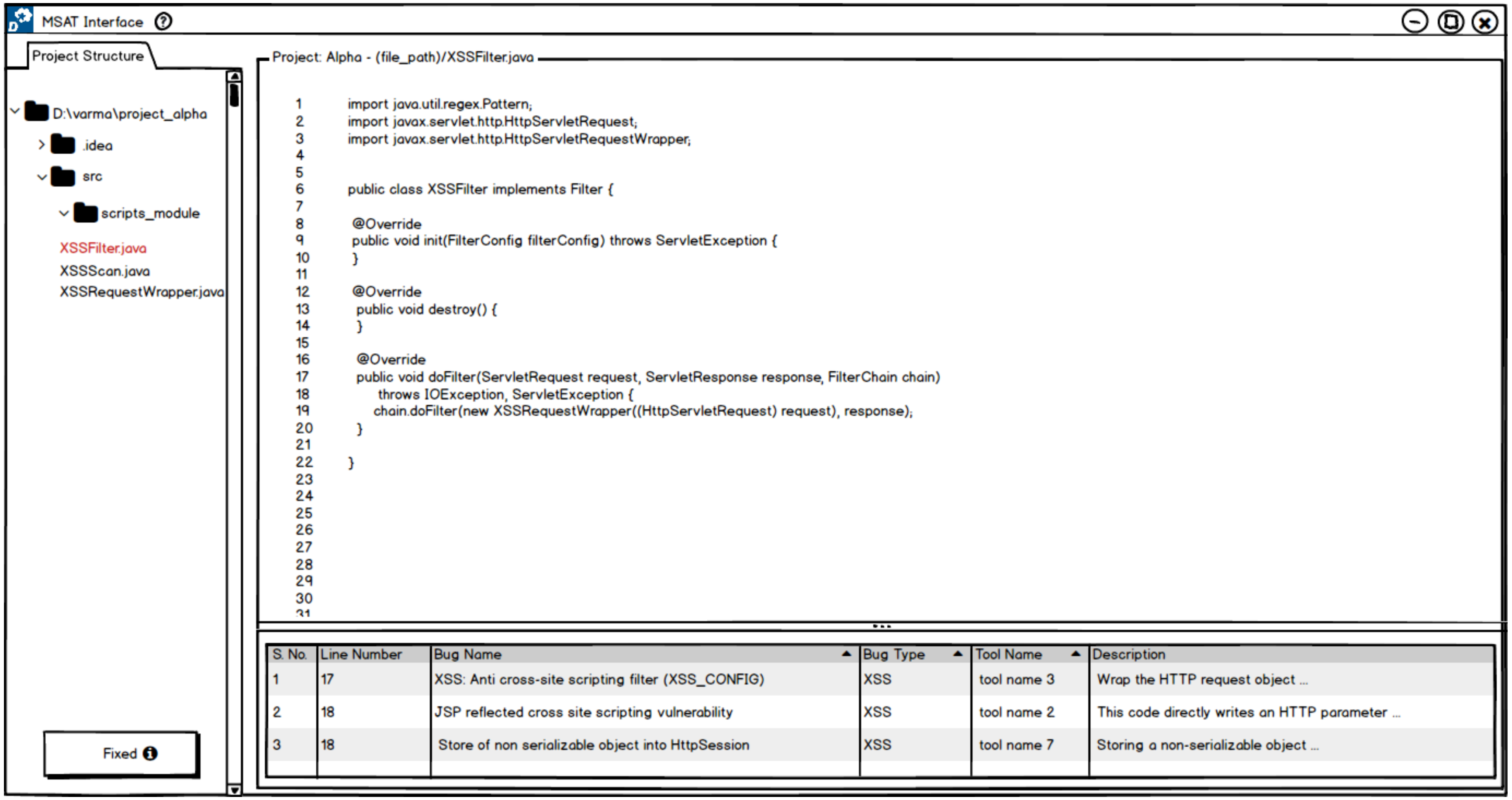
✓

- Improvisations for next UX cycle:

  - UX 2 Scalability
  - + new sub RQ’s

UX Design Cycle 3

[RQ 1]
Do users prefer bug icons or list view for bugs in same file?

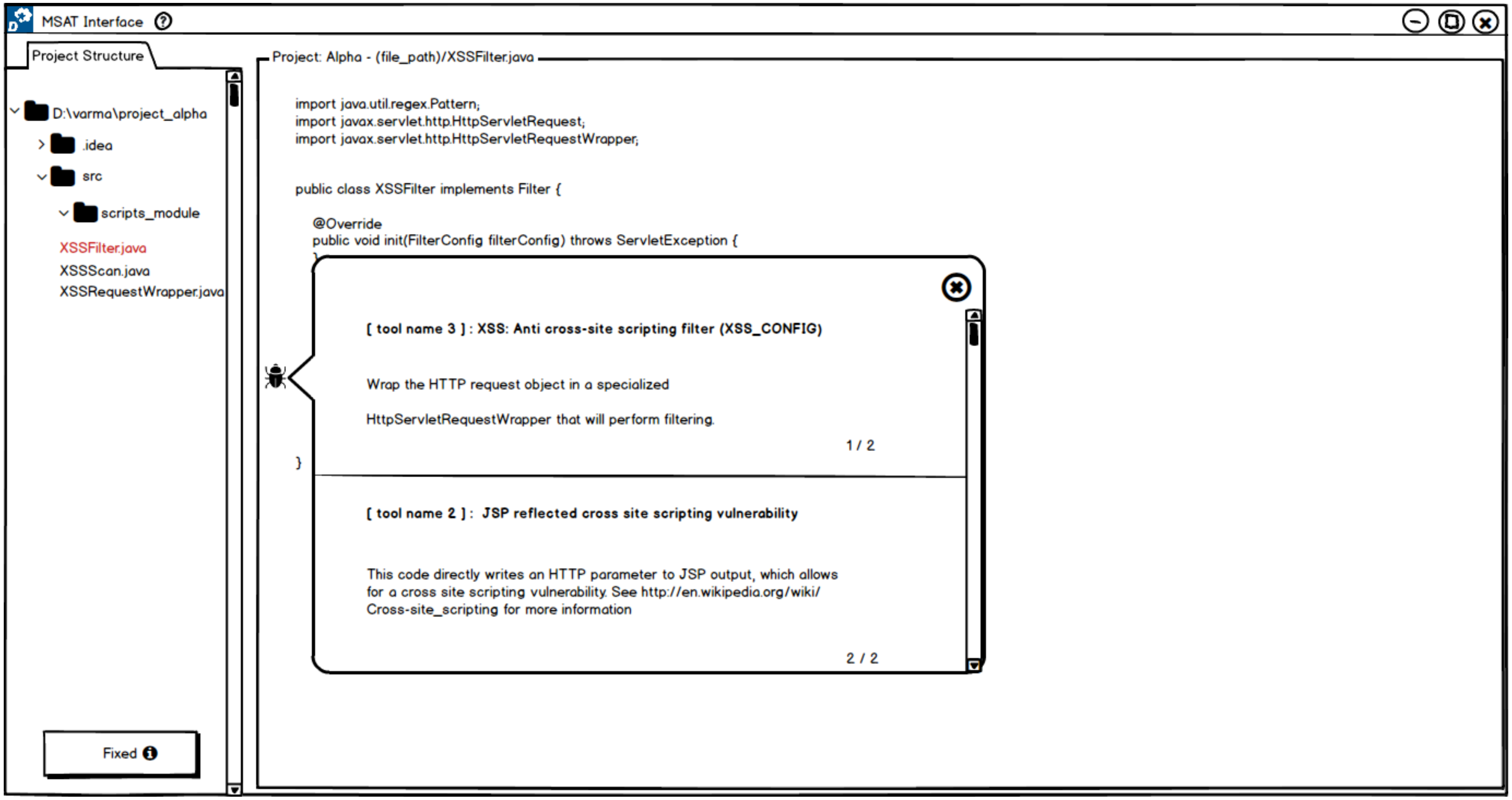


```
import java.util.regex.Pattern;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;


public class XSSFilter implements Filter {

 @Override
 public void init(FilterConfig filterConfig) throws ServletException {
 }

 @Override
  public void destroy() {
  }

  @Override
  public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain)
       throws IOException, ServletException {
      chain.doFilter(new XSSRequestWrapper((HttpServletRequest) request), response);
  }

}
```

S. No.	Line Number	Bug Name	Bug Type	Tool Name	Description
1	17	XSS: Anti cross-site scripting filter (XSS_CONFIG)	XSS	tool name 3	Wrap the HTTP request object ...
2	18	JSP reflected cross site scripting vulnerability	XSS	tool name 2	This code directly writes an HTTP parameter ...
3	18	Store of non serializable object into HttpSession	XSS	tool name 7	Storing a non-serializable object ...

HEINZ NIXDORF INSTITUT
UNIVERSITÄT PADERBORN

[RQ 1]
Do users prefer to see bugs one by one or at once in the context of multiple bugs at the same time?



```
import java.util.regex.Pattern;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;

public class XSSFilter implements Filter {

    @Override
    public void init(FilterConfig filterConfig) throws ServletException {
    }

}
```

[tool name 3] : XSS: Anti cross-site scripting filter (XSS_CONFIG)

Wrap the HTTP request object in a specialized

HttpServletRequestWrapper that will perform filtering.

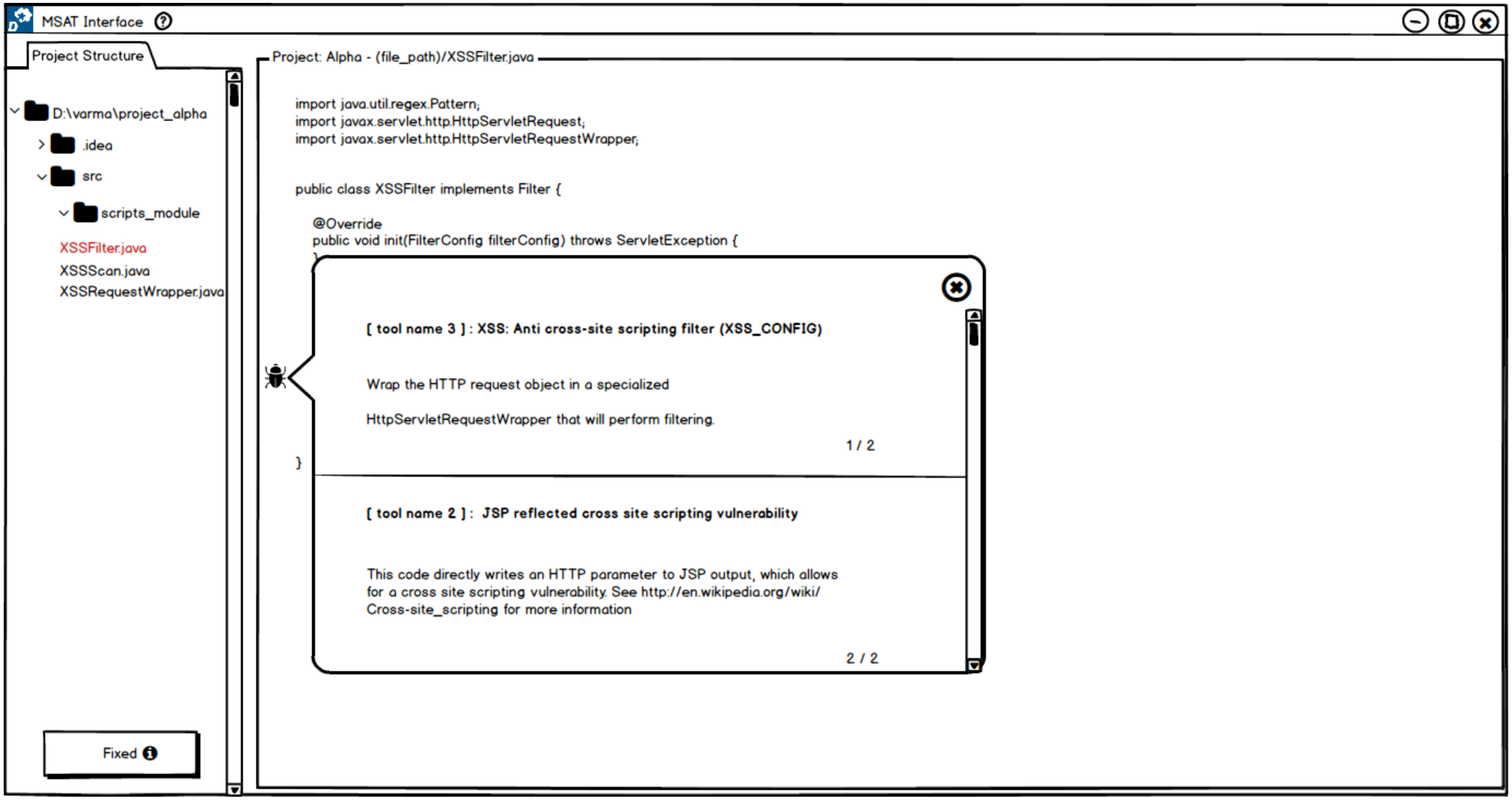
1 / 2

[tool name 2] : JSP reflected cross site scripting vulnerability

This code directly writes an HTTP parameter to JSP output, which allows for a cross site scripting vulnerability. See http://en.wikipedia.org/wiki/Cross-site_scripting for more information

2 / 2

[RQ 1]
Does vertical view help in getting an overview of the presence of multiple bugs over horizontal views?



```
import java.util.regex.Pattern;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;


public class XSSFilter implements Filter {

    @Override
    public void init(FilterConfig filterConfig) throws ServletException {
    }

}
```

[tool name 3] : XSS: Anti cross-site scripting filter (XSS_CONFIG)

Wrap the HTTP request object in a specialized

HttpServletRequestWrapper that will perform filtering.

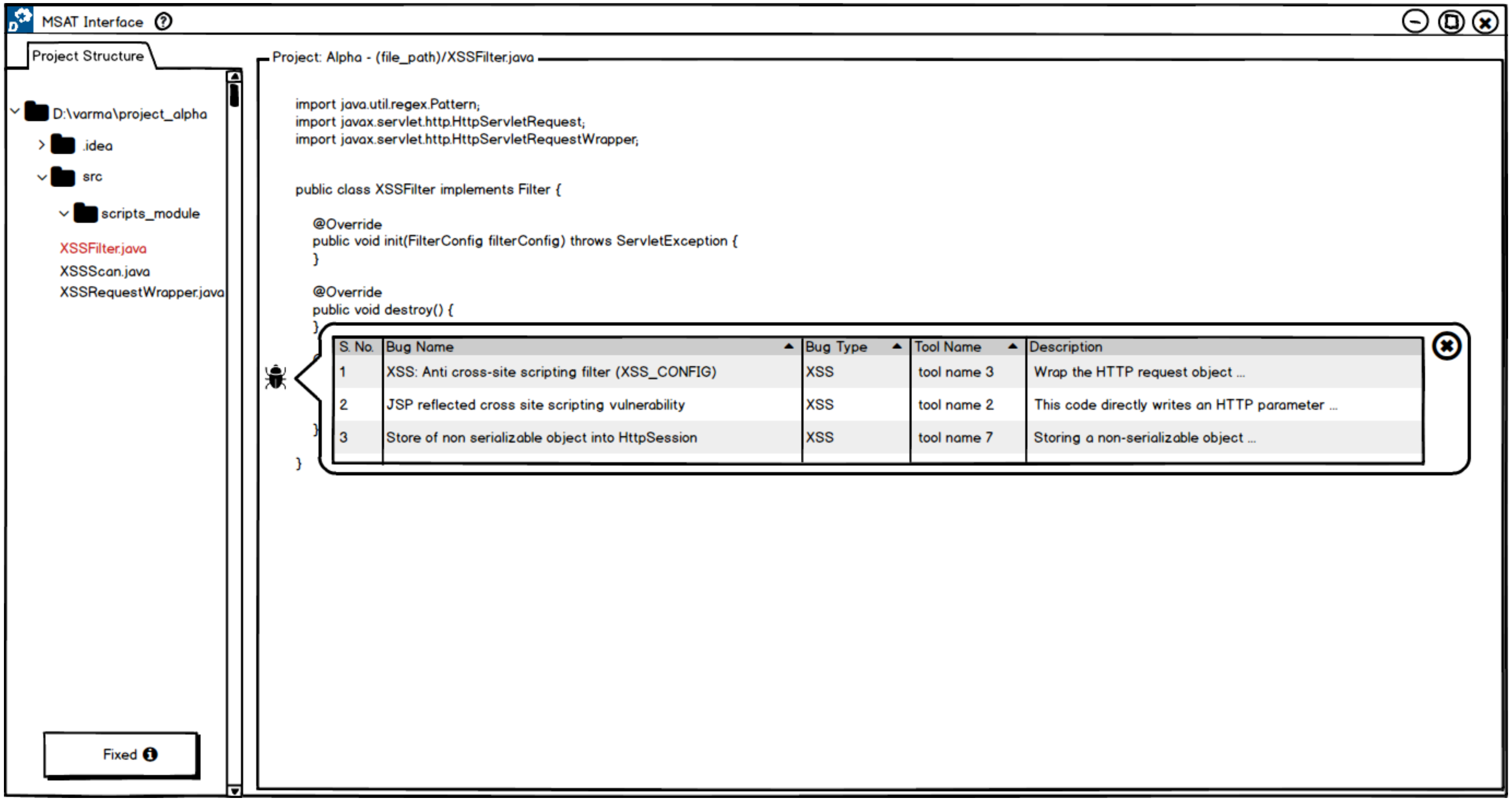
1 / 2

[tool name 2] : JSP reflected cross site scripting vulnerability

This code directly writes an HTTP parameter to JSP output, which allows for a cross site scripting vulnerability. See http://en.wikipedia.org/wiki/Cross-site_scripting for more information

2 / 2

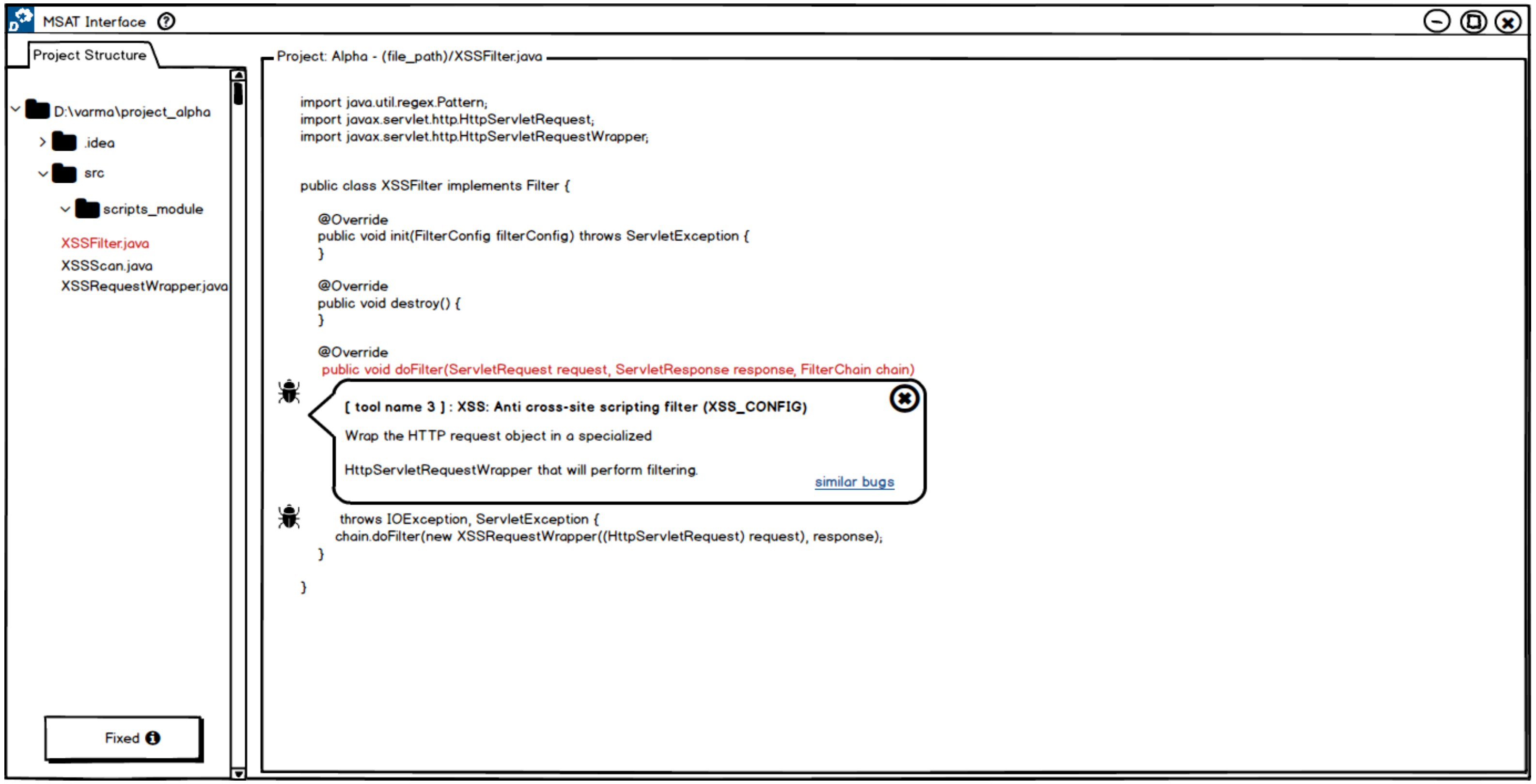
[RQ 1]
Do users prefer for table view over text description shown for multiple bugs at a line of code?



```
import java.util.regex.Pattern;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;


public class XSSFilter implements Filter {

    @Override
    public void init(FilterConfig filterConfig) throws ServletException {
    }

    @Override
    public void destroy() {
    }
    
    }
}
```

S. No.	Bug Name	Bug Type	Tool Name	Description
1	XSS: Anti cross-site scripting filter (XSS_CONFIG)	XSS	tool name 3	Wrap the HTTP request object ...
2	JSP reflected cross site scripting vulnerability	XSS	tool name 2	This code directly writes an HTTP parameter ...
3	Store of non serializable object into HttpSession	XSS	tool name 7	Storing a non-serializable object ...

[RQ 1] In context of same bug identified but with different line numbers, would have ‘similar bugs’ in bug description with on click pops up similar bug description boxes at the identified line or a list at the bottom help user in locating actual line where bug exist?



```
import java.util.regex.Pattern;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;

public class XSSFilter implements Filter {

    @Override
    public void init(FilterConfig filterConfig) throws ServletException {
    }

    @Override
    public void destroy() {
    }

    @Override
    public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain)
      throws IOException, ServletException {
      chain.doFilter(new XSSRequestWrapper((HttpServletRequest) request), response);
    }

}
```

[tool name 3] : XSS: Anti cross-site scripting filter (XSS_CONFIG)

Wrap the HTTP request object in a specialized

HttpServletRequestWrapper that will perform filtering.

- list

[RQ 2] Evaluation Set Up: 5 Feedbacks – MSAT-UI Vs Native UIs

- Three different native UI tools for a single JavaScript project.

  - CLI – ESLint
  - IDE – SonarLint
  - WEB - SonarQube

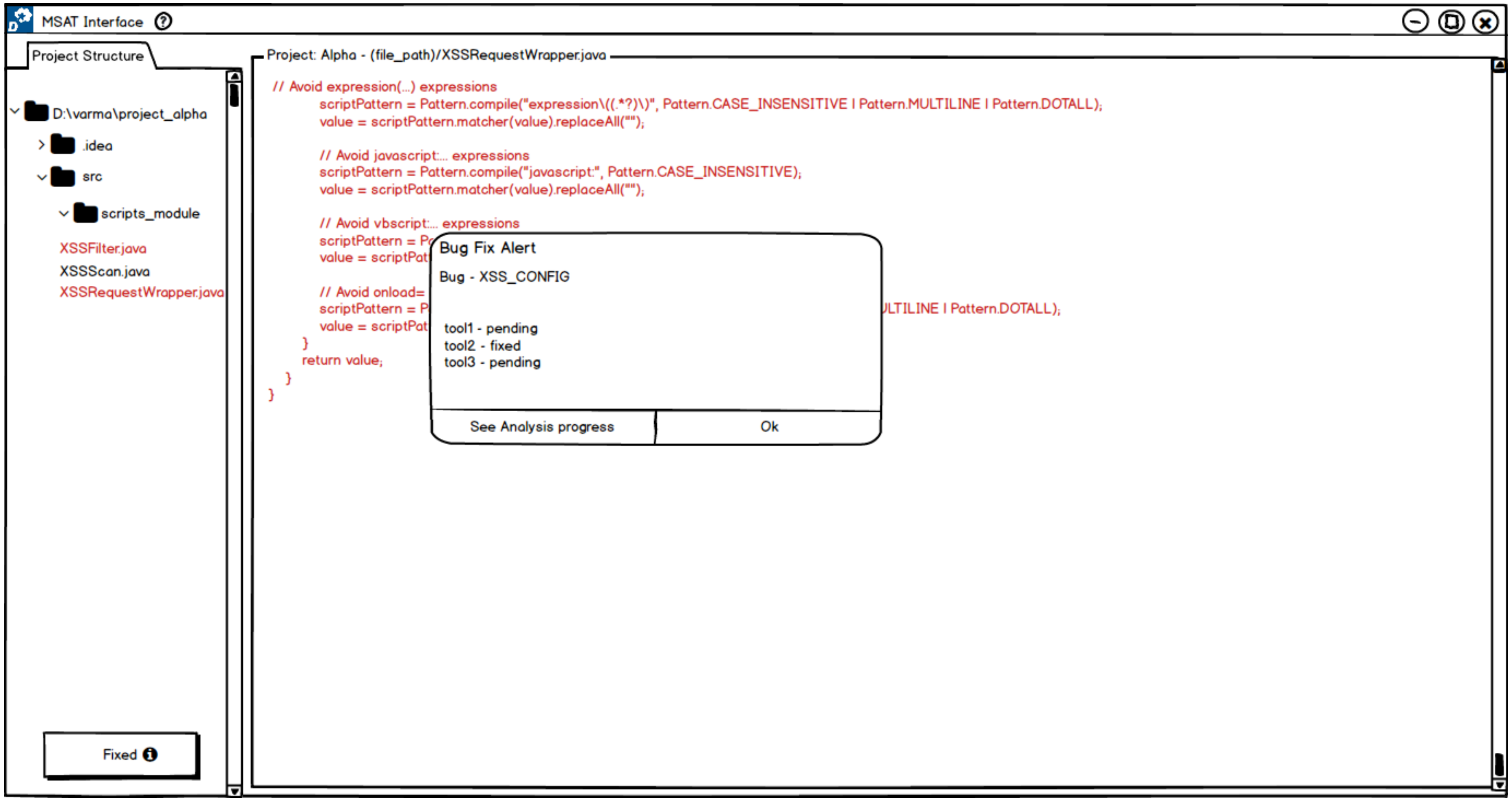
[RQ 2]

How usable are each feedback functionality compared to the scenario of using unified UI to native UIs?

- Animated Icons
- Progress Bar
- Pending Status Popup
- Alerts
- Status

- Almost all users agreed the ideas being novel and hardly present with native UIs.

[RQ 2]
Does alert notification help in fixing more bugs in contrast to its absence in current tools UI?



```
// Avoid expression(...) expressions
        scriptPattern = Pattern.compile("expression\\((.*?)\\)", Pattern.CASE_INSENSITIVE | Pattern.MULTILINE | Pattern.DOTALL);
        value = scriptPattern.matcher(value).replaceAll("");

        // Avoid javascript:... expressions
        scriptPattern = Pattern.compile("javascript:", Pattern.CASE_INSENSITIVE);
        value = scriptPattern.matcher(value).replaceAll("");

        // Avoid vbscript:... expressions
        scriptPattern = P
        value = scriptPat

        // Avoid onload=
        scriptPattern = P                                                  LTILINE | Pattern.DOTALL);
        value = scriptPat
    }
    return value;
  }
}
```

Bug Fix Alert

Bug - XSS_CONFIG

tool1 - pending
tool2 - fixed
tool3 - pending

HEINZ NIXDORF INSTITUT
UNIVERSITÄT PADERBORN

[RQ 2]
Does MSAT UI with five different mechanisms helps in fixing more bugs in comparison to using multiple tools with native user interfaces?

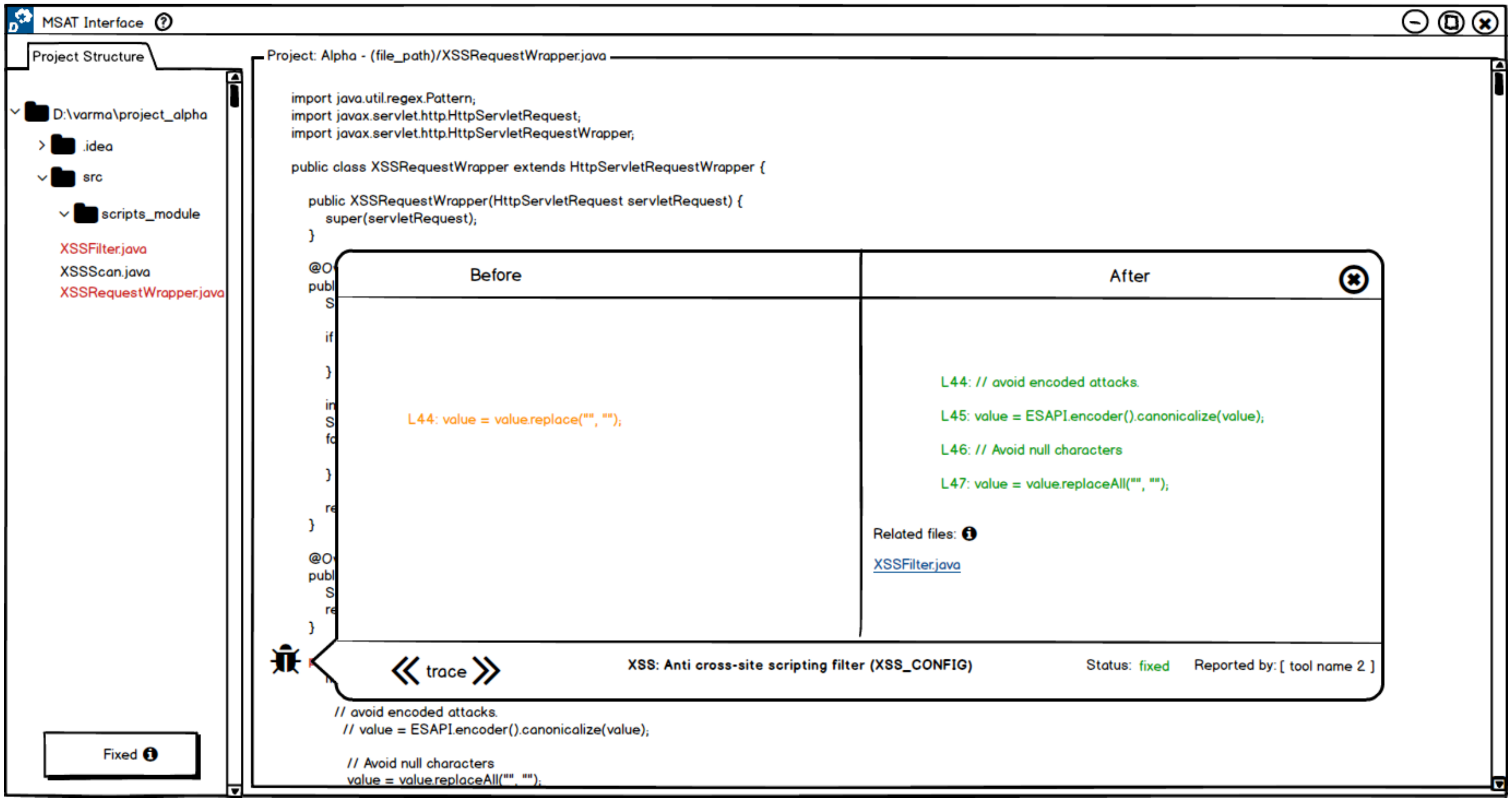
- Alert - when bug fix failed, helps to work on the bug again (state of work flow)
- Status – time for analysing

[RQ 2]

Does MSAT UI with five different mechanisms helps in fixing the bugs in a faster way in comparison to using multiple tools with native user interfaces?

- Visualisations provided by these 5 feedback helps!
- Example: Progress Bar – helps in waiting than making system hang

[RQ 3]
Do users prefer having multiple windows to single window in tracing previous bug fixes in a method?



Project: Alpha - (file_path)/XSSRequestWrapper.java

```
import java.util.regex.Pattern;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;

public class XSSRequestWrapper extends HttpServletRequestWrapper {

  public XSSRequestWrapper(HttpServletRequest servletRequest) {
    super(servletRequest);
  }

  @O
  publ
    S

    if

    }

    in
    S
    fo

    }

    re
  }

  @O
  publ
    S
    re
  }

    // avoid encoded attacks.
      // value = ESAPI.encoder().canonicalize(value);

      // Avoid null characters
      value = value.replaceAll("", "");
```

Before	After

L44: value = value.replace("", "");	L44: // avoid encoded attacks.
	L45: value = ESAPI.encoder().canonicalize(value);
	L46: // Avoid null characters
	L47: value = value.replaceAll("", "");

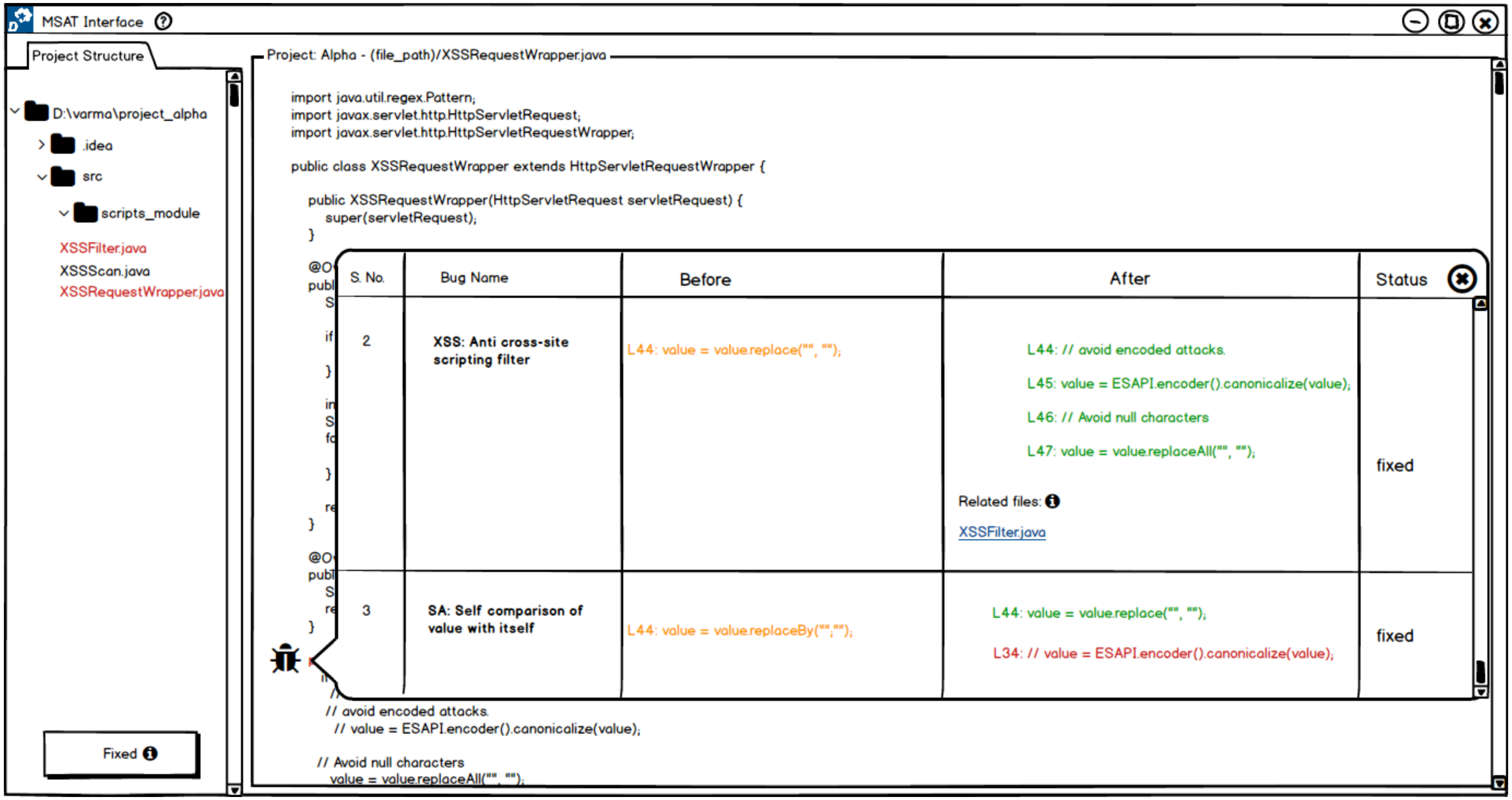
Related files:

XSSFilter.java

trace

XSS: Anti cross-site scripting filter (XSS_CONFIG)   Status: fixed   Reported by: [tool name 2]

[RQ 3] While tracing previous bug fixes in a method, do users prefer a table view to a before/after windows?



Project: Alpha - (file_path)/XSSRequestWrapper.java

```
import java.util.regex.Pattern;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;

public class XSSRequestWrapper extends HttpServletRequestWrapper {

  public XSSRequestWrapper(HttpServletRequest servletRequest) {
     super(servletRequest);
  }

  @O
  publ
     S

     if

     }

     in
     S
     fo

     }

     re
  }

  @O
  publ
     S
     re
  }

     i
      //
     // avoid encoded attacks.
      // value = ESAPI.encoder().canonicalize(value);

    // Avoid null characters
     value = value.replaceAll("", "");
```

S. No.	Bug Name	Before	After	Status
2	**XSS: Anti cross-site scripting filter**	L44: value = value.replace("", "");	L44: // avoid encoded attacks. L45: value = ESAPI.encoder().canonicalize(value); L46: // Avoid null characters L47: value = value.replaceAll("", ""); Related files: XSSFilter.java	fixed

3	**SA: Self comparison of value with itself**	L44: value = value.replaceBy("","");	L44: value = value.replace("", "");
			L34: // value = ESAPI.encoder().canonicalize(value);
				fixed

HEINZ NIXDORF INSTITUT
UNIVERSITÄT PADERBORN

[RQ 3]
Do users be able to keep up in state of workflow as tools scale?

- Yes! proposed solution ideas promised to keep up the scalability.
- However, users preferred ‘table view’ as easy with their consistency model.

Q. Do users prefer having tool names in general?

- Yes!

- Compare tool performance
- Helps to have much information as possible

Limitations

- Number of participants:
  - UX 1 – 5
  - UX 2 – 7
  - UX 3 – 5
- Closed Study
- Priming – order of evaluation

Future Work

Q. Would having tabs help scale the tools visibility with bugs results?

Q. Will the user need to have graphs for separate tools or one graph combining the tools selected?

Q. Does having graphs (example: histograms) at particular part of code with commits related bug fixes help the user to trace?

… many more!

Summary

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

- Need for a single user interface for multiple tools

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