

Static Code Analysis

Jakub Papcun
Jan Svoboda

Honeywell

- Honeywell
- Static Code Analysis
- Use of SCA in Honeywell
- Defect Tracking Integration
- Manual Code Review Integration

- **4 Strategic Business Groups (SBGs)**
 - Aero
 - Automation and Control Solutions (ACS)
 - Performance Materials and Technologies (PMT)
 - Transportation Systems (TS)
- **130 000 employees worldwide**
- **ACS Centre of Technologies (ACT)**
 - Software Excellence Group
 - Implements Best Practices for software development
 - Implements tools necessary for fulfilling Best Practices requirements

- **Analysis of the code without executing the program itself**
- **Various types of SCA**
 - **Type checking**
 - ◆ checks for correct assignment of types of objects
 - **Style checking**
 - ◆ checks the style of the code and its formatting
 - **Program Understanding**
 - ◆ helps user make sense of large codebase and may include refactoring capabilities
 - **Program verification and property checking**
 - ◆ attempts to prove that the code correctly implements the specification of the program
 - **Security review**
 - ◆ uses dataflow analysis for detection of possible code injection
 - **Bug finding**
 - ◆ looks for places in the code where program may behave in a different way from the way intended by developer

- **SCA identifies only “shallow” errors and does not look for problems in design or functionality**
- **3 types of results**
 - **True positives**
 - ◆ real issues which are code errors and should be fixed before releasing
 - **False positives**
 - ◆ issues identified by the analysis but not real threats due to for example architecture of the software
 - **False negatives**
 - ◆ real issues which Static Code Analysis did not identify and are still hidden from the knowledge of the developers
- **Possibility to adjust the Static Code Analysis rules to the context**

Example 1

```
private Map<String, String> paths = new HashMap<String, String>();

public void addPath(String name, String path) {
    paths.put(name, path);
}

private String getNormalizedPath(String name) throws IOException {
    return paths.get(name).toLowerCase();
}
```

Example 1

```
private Map<String, String> paths = new HashMap<String, String>();

public void addPath(String name, String path) {
    paths.put(name, path);
}

private String getNormalizedPath(String name) throws IOException {
    return paths.get(name).toLowerCase();
}
```

Can return null



A `NullPointerException` is thrown in case of an attempt to dereference a `null` value.

Example 2

```
private static void foo(){
    int i = 0;
    String s = null;

    if(i > 0){
        s = "positive";
    }

    if(s.contains("pos")){
        System.out.println(s);
    }
}
```


Example 2

```
private static void foo(){
    int i = 0;
    String s = null;

    if(i > 0){
        s = "positive";
    }

    if(s.contains("pos")){
        System.out.println(s);
    }
}
```

Statement always true



1. Statement is always false and never enters the block

Example 2

```
private static void foo(){  
    int i = 0;  
    String s = null;
```

```
    if(i > 0){
```

```
        s = "positive";
```

```
    }
```

```
    if(s.contains("pos")){
```

```
        System.out.println(s);
```

```
    }
```

```
}
```

Statement always true

s may be null

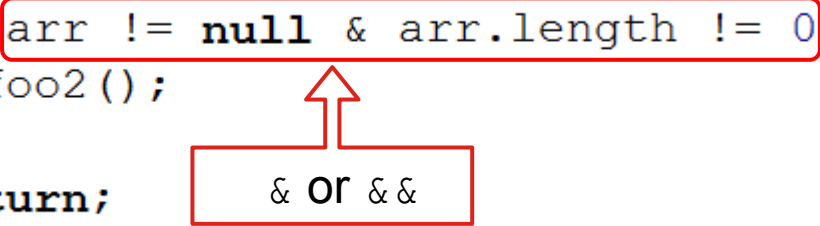
1. Statement is always false and never enters the block
2. s variable may be null and NullPointerException may be thrown

Example 3

```
private static void foo(int arr[]){
    if(arr != null & arr.length != 0){
        foo2();
    }
    return;
}
```

Example 3

```
private static void foo(int arr[]){  
    if(arr != null & arr.length != 0){  
        foo2();  
    }  
    return;  
}
```



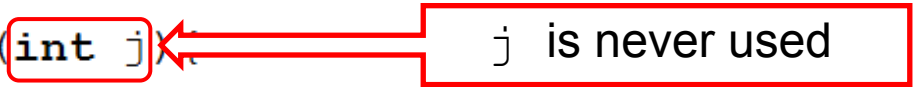
Questionable use of bit operation ‘&’ in expression. Did you mean ‘&&’?

Example 4

```
private static void foo(int j) {
    Integer k;
    switch(k) {
        case 1: System.out.println("k lower than 2."); break;
        case 2: System.out.println("k equals 2."); break;
        case 3: System.out.println("k bigger than 2."); break;
        default: System.out.println("K = " + k);
    }
    return;
}
```

Example 4

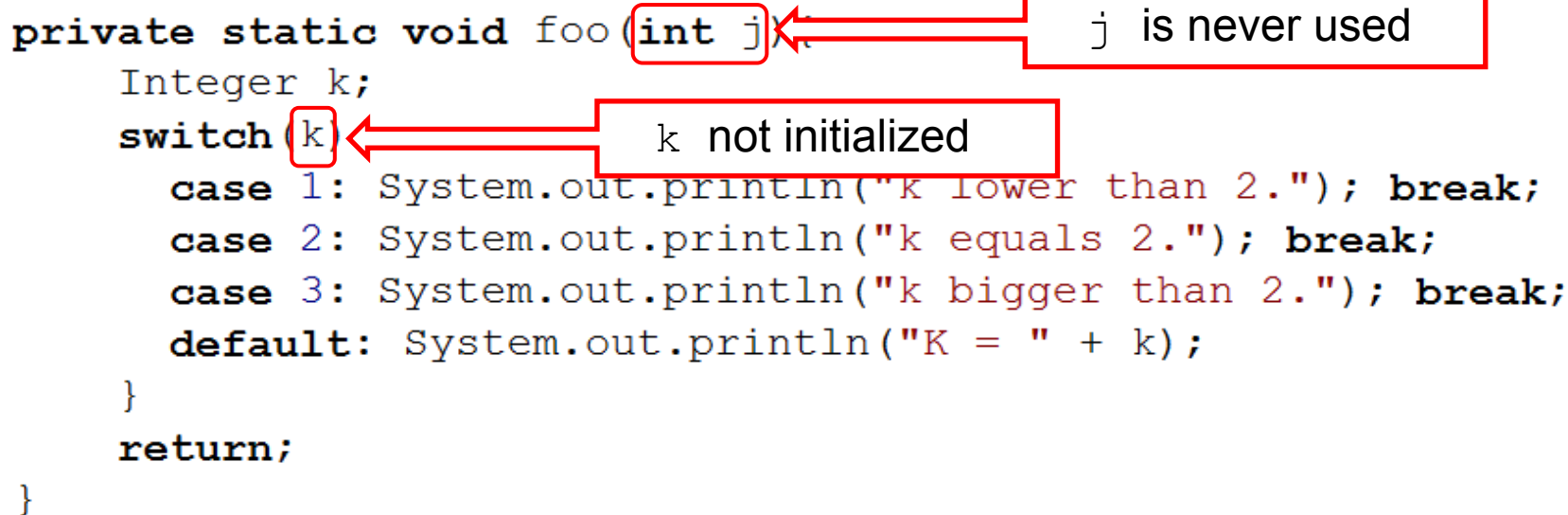
```
private static void foo(int j)  
    Integer k;  
    switch(k) {  
        case 1: System.out.println("k lower than 2."); break;  
        case 2: System.out.println("k equals 2."); break;  
        case 3: System.out.println("k bigger than 2."); break;  
        default: System.out.println("K = " + k);  
    }  
    return;  
}
```



1. j variable is never used and thus redundant

Example 4

```
private static void foo(int j)
    Integer k;
    switch(k)
        case 1: System.out.println("k lower than 2."); break;
        case 2: System.out.println("k equals 2."); break;
        case 3: System.out.println("k bigger than 2."); break;
        default: System.out.println("K = " + k);
    }
    return;
}
```



1. j variable is never used and thus redundant
2. k variable is never initialized and thus unusable

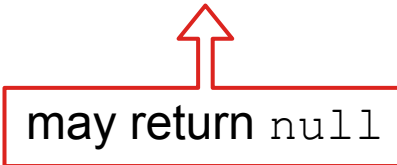
Example 5

```
public void foo() {
    Item item = new Item();
    if(item.getInfo() != null) {
        String info = item.getInfo().trim();
    }
}
```

```
class Item{
    public String getInfo() {
        // Making REST Request
    }
}
```


Example 5

```
public void foo() {  
    Item item = new Item();  
    if(item.getInfo() != null){  
        String info = item.getInfo().trim();  
    }  
}
```



```
class Item{  
    public String getInfo(){  
        // Making REST Request  
    }  
}
```

REST may fail and return null

- **Klocwork**
- **SonarQube**
- **Findbugs**
- **Kiuwan**
- **Others**
 - **Compilers**
 - **IDEs**
 - ◆ **IntelliJ Idea**
 - ◆ **Eclipse**

Capability Maturity Model Integration (CMMI)

- **A set of rules defining the maturity of the company**
- **5 levels**
- **Honeywell achieving level 5**
 - continuous improvement of the processes and evaluation of the results across all the software development disciplines according to collected measurements and metrics
- **ACS Software Development Process (ASDP)**
 - Process used for Software Development across ACS
 - Aims to be compliant with CMMI level 5
 - Static Code Analysis is one of the steps in the Implementation discipline

- **Static Code Analysis tool**
- **Supported languages are C/C++, C#, Java**
- **Identifies code vulnerabilities**
 - Logical errors
 - Security vulnerabilities
 - Coding standards violations
- **Klocwork calculates software metrics such as lines of code, lines of comments, cyclomatic complexity, number of functions/methods**
- **Web Interface and user instance of Klocwork**
 - poor REST API
 - various restrictions on the side of Klocwork query language

ACT_JIRA_Extensions

default

module:+ACT

ISSUES

REPORTS

XREF

VIEWS

MODULES

CONFIGURATION

Search for: build:'build_290' status:+Analyze,+Fix

Search

Sort by: id

SEARCHES

- status:Analyze
- state:Existing
- category:'Suspicious Code Practices'
- code:NPE,NPD
- entity:main
- id:1-100
- severity:Critical
- severity:Error
- status:Analyze severity:1
- state:+New severity:6
- state:+New severity:7
- state:+New severity:4 status:Analyze
- status:+Analyze,+Fix state:Existing
- status:+Analyze,+Fix state:New
- status:+Analyze,+Fix state:Existing severity:1 citedby:e563514

PRINT EDIT: ALL

1 to 25 of 171

#423: Variable 'rs' is always 'null' in this expression.

C:\bamboo-home\xml-data\build-dir\131073\AJE-KA-JOB\1src\main\java\com\honeywell\softcol\jira\plugin\integration\contour\ContourSqlManagerImpl.java:518 | addDocumentTypeToProject
Code: REDUN.NULL | Severity: Review(4) | State: Existing | Status: Fix | Taxonomy: Java | Reference: none | Owner: unowned

#433: Variable 'rs' is always 'null' in this expression.

C:\bamboo-home\xml-data\build-dir\131073\AJE-KA-JOB\1src\main\java\com\honeywell\softcol\jira\plugin\integration\contour\db\ContourCommonDbSqlManager.java:1383 | addDocumentTypeToProject
Code: REDUN.NULL | Severity: Review(4) | State: Existing | Status: Fix | Taxonomy: Java | Reference: none | Owner: unowned

#891: Synchronized method calls another synchronized method 'getItemsForContourProject' with the same lock held

C:\bamboo-home\xml-data\build-dir\131073\AJE-KA-JOB\1src\main\java\com\honeywell\softcol\jira\plugin\integration\contour\ContourManagerImpl.java:450 | getContourItems
Code: SYNCH.NESTED | Severity: Review(4) | State: Existing | Status: Fix | Taxonomy: Java | Reference: none | Owner: unowned

#892: Synchronized method calls another synchronized method 'getDocumentTypeByIdByName' with the same lock held

C:\bamboo-home\xml-data\build-dir\131073\AJE-KA-JOB\1src\main\java\com\honeywell\softcol\jira\plugin\integration\contour\ContourManagerImpl.java:646 | createContourProjectItemLink
Code: SYNCH.NESTED | Severity: Review(4) | State: Existing | Status: Analyze | Taxonomy: Java | Reference: none | Owner: unowned

#893: Synchronized method calls another synchronized method 'getChangeRequestForIssue' with the same lock held

C:\bamboo-home\xml-data\build-dir\131073\AJE-KA-JOB\1src\main\java\com\honeywell\softcol\jira\plugin\integration\contour\ContourManagerImpl.java:648 | createContourProjectItemLink
Code: SYNCH.NESTED | Severity: Review(4) | State: Existing | Status: Analyze | Taxonomy: Java | Reference: none | Owner: unowned

#894: Synchronized method calls another synchronized method 'createChangeRequestAssociation' with the same lock held

C:\bamboo-home\xml-data\build-dir\131073\AJE-KA-JOB\1src\main\java\com\honeywell\softcol\jira\plugin\integration\contour\ContourManagerImpl.java:656 | createContourProjectItemLink
Code: SYNCH.NESTED | Severity: Review(4) | State: Existing | Status: Analyze | Taxonomy: Java | Reference: none | Owner: unowned

Projects About Help E563514

ACT_JIRA_Extensions

default

module:+ACT

ISSUES REPORTS XREF VIEWS MODULES CONFIGURATION

Back to List

Previous

Next

Variable 'rs' is always 'null' in this expression.



ID	423
CODE	REDUN.NULL
NAME	Usage of variable instead of null constant
LOCATION	ContourSqlManagerImpl.java: 518
BUILD	build_290
SEVERITY	Review (4)
OWNER	*no owner*
STATE	Existing
STATUS	Fix View History

TRACEBACK

There is no traceback for this defect.

```
...\\build-dir\\131073\\AJE-KA-JOB\\src\\main\\java\\com\\honeywell\\softco\\jira\\plugin\\integration\\contour\\ContourSqlManagerImpl.java (build_290) 0 of 0
400      * @param appLink
401      * @return see AJE-408
402      */
403      @Override
404      public int addDocumentTypeToProject(int projectId, int documentTypeId, int organizationId, final ApplicationLink appLink) {
405          log.debug("Executing addDocumentTypeToProject()");
406          int rowsAffected = 0; // if there is no organization return null
407          Connection conn = connect(appLink);
408          PreparedStatement stmt = null;
409          ResultSet rs = null;
410          try {...}
411          catch (Exception e) {...}
412          finally {
413              closeStmtRsConn(stmt, rs, conn);
414          }
415          return rowsAffected;
416      }
417
418      @Override
419      public Integer getDocumentTypeIdByKey(String documentTypeKey, final ApplicationLink appLink) {...}
420
421      @Override
422      public String getDocumentTypeKeyById(Integer documentTypeId, final ApplicationLink appLink) {...}
423
424      @Override
425      public int getAssociationCountForChangeRequest(int changeRequestId, final ApplicationLink appLink) / ...
```

- **JIRA**
- **Crucible/FishEye**
- **Reasons**
 - Klocwork is not very user friendly
 - Results of SCA serve as input parameters for processes
 - ◆ These processes have majority of required data in other tools
 - Simplifying the processes
 - Delegation of control over SCA results

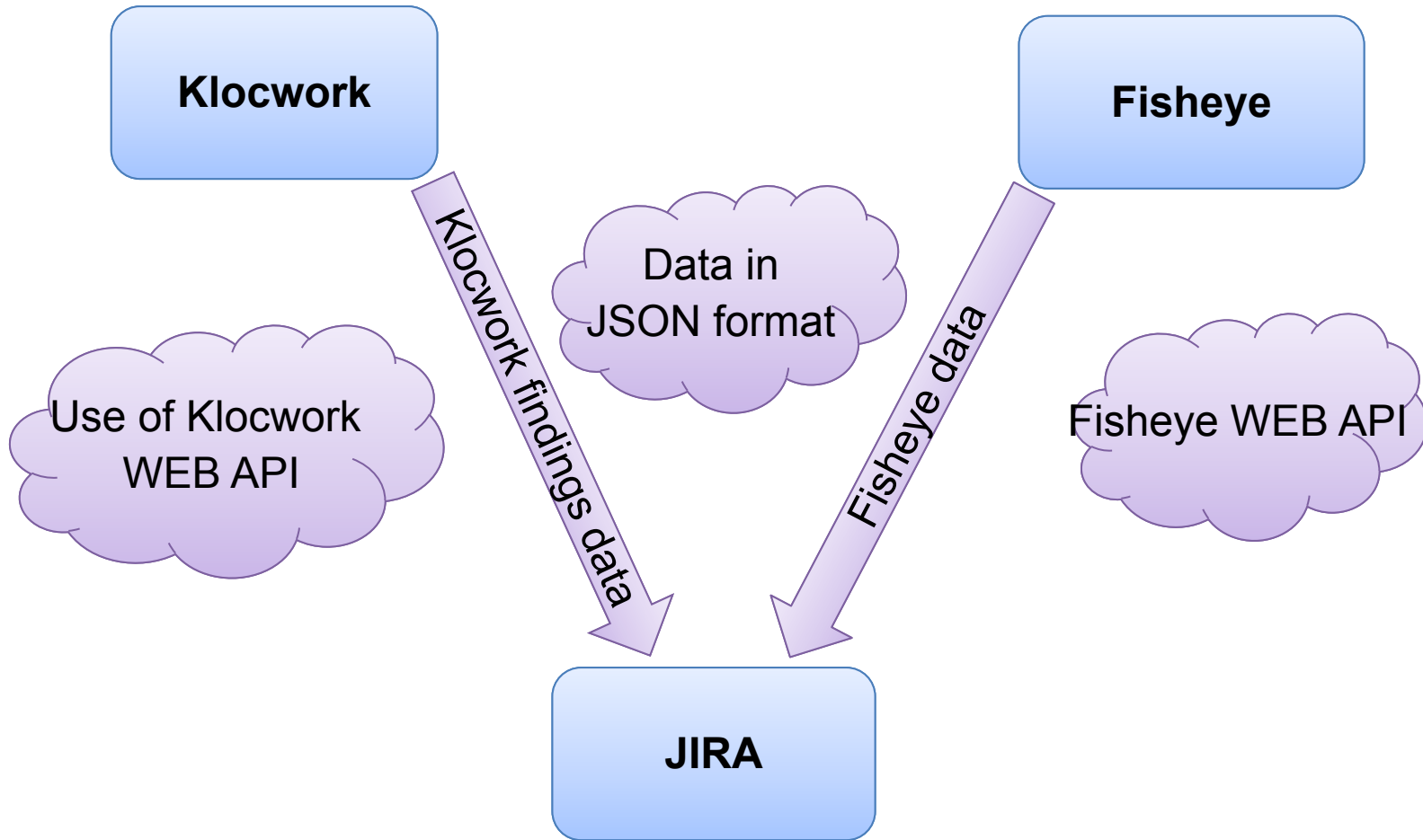
Defect Tracking Integration

Honeywell

- **Honeywell ACS uses JIRA**
- **Issue and Project tracking software**
 - tracks time spent on resolving issues, progress of the project and various metrics connected to software development
 - one of the main tools used across Honeywell ACS
- **Issue**
 - a software bug (defect), a task, a helpdesk ticket, a new feature etc.
- **Easily extensible with wide variety of extensibility options**
- **Great integration capabilities with other Atlassian products**

- **Fisheye**
- **Web Interface for read only access to SVN**
- **Visualization and reporting capabilities regarding source code**
- **Searching capabilities according to commits, comments, people etc.**
- **Great WEB API for communication between tools**

JIRA – Klocwork – Fisheye Integration Overview



Goals

- Development cycle time reduction
- Code quality improvement
- Build basis for creating new more sophisticated metrics combining defect/issue tracking and static code analysis
- Set basis for future development of integration and possible defect prediction

Create JIRA defect from Klocwork

- A simple button in Klocwork Finding
- Creates a JIRA defect with as much information as possible
- Stores the data about defect resolving in JIRA
- Not able to get line numbers
- Not able to get specific faulty code

Create a list of Klocwork findings for specific JIRA issue

- Integrates data from Fisheye with data from Klocwork and creates a list of Klocwork Findings that were introduced into the code as part of implementation of some Issue
- Higher Code Quality
- Lower Development Cycle Time
- Information about the quality of the feature

Activity

All Comments Work Log History **Klocwork** Activity Commits Reopenings History Builds

THERE ARE CRITICAL KLOCWORK FINDINGS

3 Critical	0 Error	1 Warning	6 Review
------------	---------	-----------	----------

Summary	Severity	File	Status	State
Null pointer dereference of 'getApplicationLink()' where null comes from constant	Critical (1)	ImportContourItemsResource.java	Analyze	Existing
Private method 'getApplicationLink' is unused.	Warning (3)	ConcurrentContourProjectManagerImpl.java	Analyze	Existing
Comparing strings "" and 'applicationLinkId' with ==	Review (4)	ConcurrentContourProjectManagerImpl.java	Analyze	Existing
Comparing strings "" and 'username' using equals(), instead of length() == 0	Review (4)	ImportContourItemsResource.java	Analyze	Existing
Synchronized method calls another synchronized method 'fetchReleasesForContourProject' with the same lock held	Review (4)	ContourManagerImpl.java	Analyze	Existing

- Indicates whether the code is ready for testing and release
- Gives overall status about the code quality written as part of the selected version of the product

Version Release Readiness Report

Version Analyzed	Release 4.1
Latest Klocwork Analysis	build_265 - 26/Mar/14 2:06 AM

THERE ARE UNRESOLVED KLOCWORK FINDINGS WITH CRITICAL OR ERROR SEVERITY

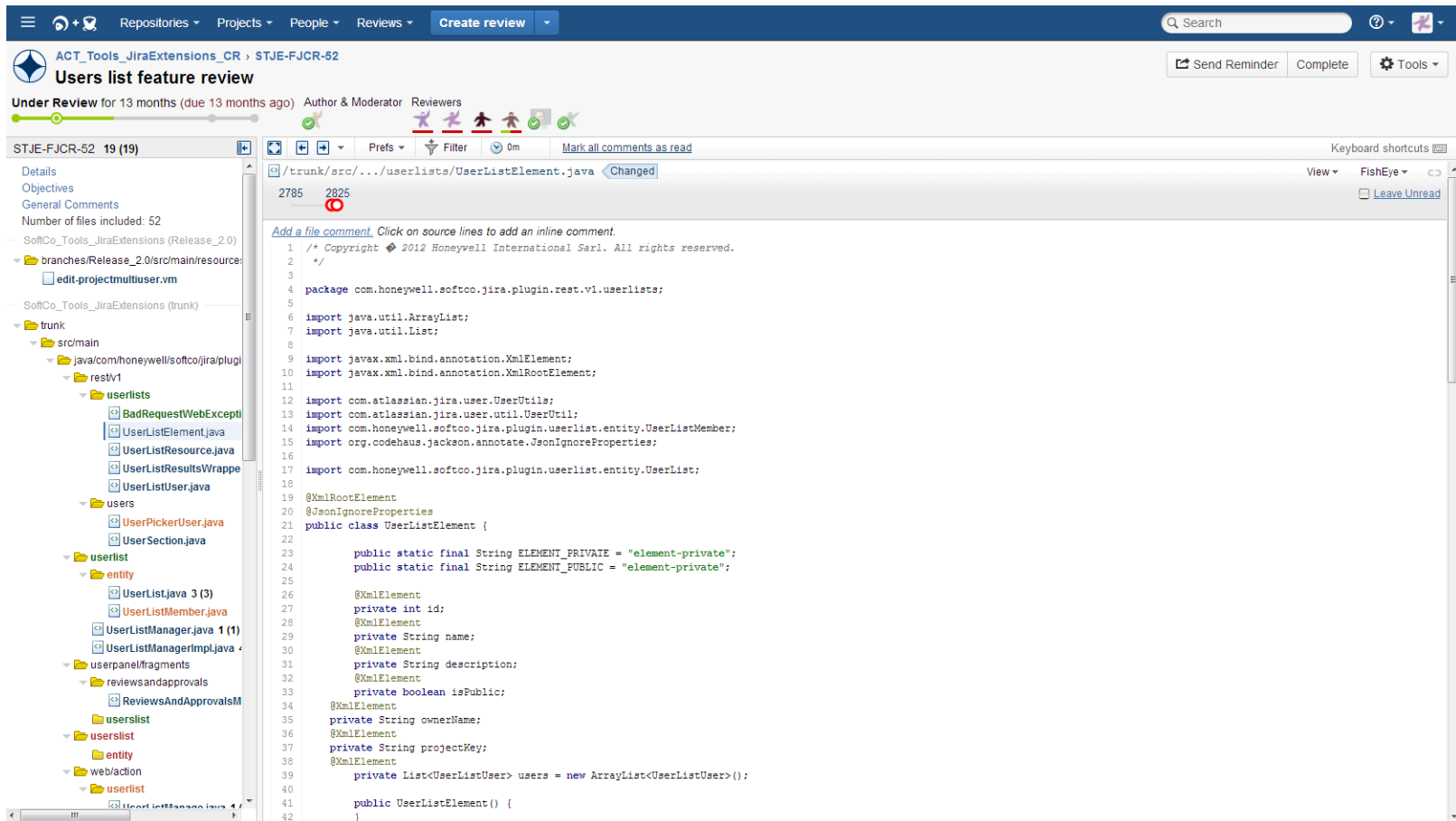
	Critical	Error	Warning	Review
AJE-1266	0	0	0	1
AJE-1145	3	0	1	6
AJE-1144	10	0	1	17
AJE-1263	0	0	0	1
Summary	13	0	2	25

Manual Code Reviews Integration

Honeywell

Manual Code Reviews

- Systematic examination of the source code
- Used to verify the code from various perspectives
- Atlassian Crucible



- Simple rule: cost of a defect rises with the time it is not discovered
 - Apply even for defects found during testing activities
- Solution: try to find defects as soon as possible
 - Apply all available tools/processes
 - SCA
 - Manual Code Reviews
- Not popular activity
 - Need to use two tools for very similar activities
 - Inspecting more complex code can be confusing
 - Human Factor
 - Klocwork User Friendliness

How to improve?

ACT_Tools_JiraExtensions_CR · STJF-FJCR-52
Users list feature review
Under Review for 13 months (due 13 months ago) Author & Moderator Reviewers

STJF-FJCR-52 19 (19)

Details
Objectives
General Comments
Number of files included: 52

SoftCo_Tools_JiraExtensions (Release_2.0)
branches/Release_2.0/src/main/resources
edit/projectMultiuser.vm

SoftCo_Tools_JiraExtensions (trunk)
src/main
java/com/honeywell/softco/jira/plugin
rest/v1
userlists
BadRequestWebExceptionHandler
UserListElement.java
UserListResource.java
UserListResultsWrapper
UserListUser.java
users
@XmlElement
UserPickerUser.java
userlist
entity
UserListJava 3 (3)
UserListMember.java
UserListManager.java 1 (1)
UserListManagerImpl.java
userpanel/fragments
reviewsandapprovals
ReviewsAndApprovalsImpl
userslist
userslist
entity
webaction
userlist
@XmlElement

```
2785 2825  
Add a file comment. Click on source lines to add an inline comment.  
1 /* Copyright © 2012 Honeywell International Ser1. All rights reserved.  
2 */  
3  
4 package com.honeywell.softco.jira.plugin.rest.v1.userlists;  
5  
6 import java.util.ArrayList;  
7 import java.util.List;  
8  
9 import javax.xml.bind.annotation.XmlElement;  
10 import javax.xml.bind.annotation.XmlRootElement;  
11  
12 import com.atlassian.jira.user.UserUtils;  
13 import com.atlassian.jira.user.util.UserUtil;  
14 import com.honeywell.softco.jira.plugin.userlist.entity.UserListMember;  
15 import org.codehaus.jackson.annotate.JsonIgnoreProperties;  
16  
17 import com.honeywell.softco.jira.plugin.userlist.entity.UserList;  
18  
19 @XmlElement  
20 @XmlRootElement  
21 public class UserListElement {  
22  
23     public static final String ELEMENT_PRIVATE = "element-private";  
24     public static final String ELEMENT_PUBLIC = "element-public";  
25  
26     @XmlElement  
27     private int id;  
28     @XmlElement  
29     private String name;  
30     @XmlElement  
31     private String description;  
32     @XmlElement  
33     private boolean isPublic;  
34     @XmlElement  
35     private String ownerName;  
36     @XmlElement  
37     private String projectKey;  
38     @XmlElement  
39     private List<UserListUser> users = new ArrayList<UserListUser>();  
40     public UserListElement() {  
41  
42
```

Student | IDNES.cz - zpravy | My Dash - ACS | JIRA Extensions - | [AIE-126] Default | Buttons | Control | Atlassian User Int | Effectively Comb | Change or delete | Klocwork Insight |

https://acklocworkhoneywell.com/review/insight-review.html#issueDetails_gotooffset=1_problemId=433_project=SoftCo_JIRA_Extensions.searchquery=status%253A%252BAalyze%252C%252BFixsortcx

Projects

SoftCo_JIRA_Extensions "default" module=SoftCo

ISSUES	REPORTS	XREF	VIEWS	MODULES	CONFIGURATION
--------	---------	------	-------	---------	---------------

Back to List Previous Next

C:\bamboo-home\mi-data\build-dir131073\AIE-KA-JOB1\src\main\java\com\honeywell\softco\jira\plugin\integration\contour\ContourCommonDocManager.java (build_209)

432 public int getDocumentTypeIdByDate(String name) [...]
911
912 @Override
913 public int getChangeRequestForIssue(int documentTypeId, int contourProjectId, String issueKey) [...]
951
952 @Override
953 public Long getRelationshipTypeIdByName(String name) [...]
959
960 @Override
991 public int getOrganizationID(String name) [...]
1020
1021 @Override
1022 public boolean isDocumentTypeVisible(int projectId, int documentTypeId, int organizationId) [...]
1062
1063 @Override
1064 public int addDocumentTypeToProject(int projectId, int documentTypeId, int organizationId) {
1065 log.debug("Executing addDocumentTypeToProject()");
1066 int rowsAffected = 0; // if there is no organization return null
1067 Connection conn = sqlManager.connect();
1068 PreparedStatement stmt = null;
1069 ResultSet rs = null;
1070 try {
1071 catch (Exception e) {
1072 finally {
1073 sqlManager.closeStatement(stmt, rs, conn);
1091
1092 return rowsAffected;
1093 }
1094 }
1095 /**
1096 * @param contourItemId
1097 * @return Returns map of IssueProxy Id and Association Id
1098 */
1099 @Override
1099 @Override
1100 public Map<Integer, Integer> getChangeRequestId(int contourItemId, String jiraIssueKey) [...]
1137
1138 @Override
1139 public int getChangeRequestAssociationCount(int changeRequestId) [...]
1166
1167 @Override
1168 public List<ContourFilter> getFiltersForProject(long projectId) [...]
1209
1210 @Override
1211 public Map<Integer, ContourItem> getMultipleContourItems(Collection<Integer> cItemIds) [...]
1275
1276 private String getItemIdString(Collection<Integer> cItemIds) [...]
1278
1285 private String createPossibleItemsString() [...]

Variable 'rs' is always 'null' in this expression.

ID 433
CODE REDUN_NULL
NAME Usage of variable instead of null constant
LOCATION ContourCommonDocManager.java 1090
BUILD build_209
SEVERITY Review (4)
OWNER "no owner"
STATE Existing
STATUS Fix View History
COMMENT
LAST UPDATE 06-19-2013 16:41 by E910689 Save changes
TRACEBACK
There is no traceback for this defect.

How to improve?

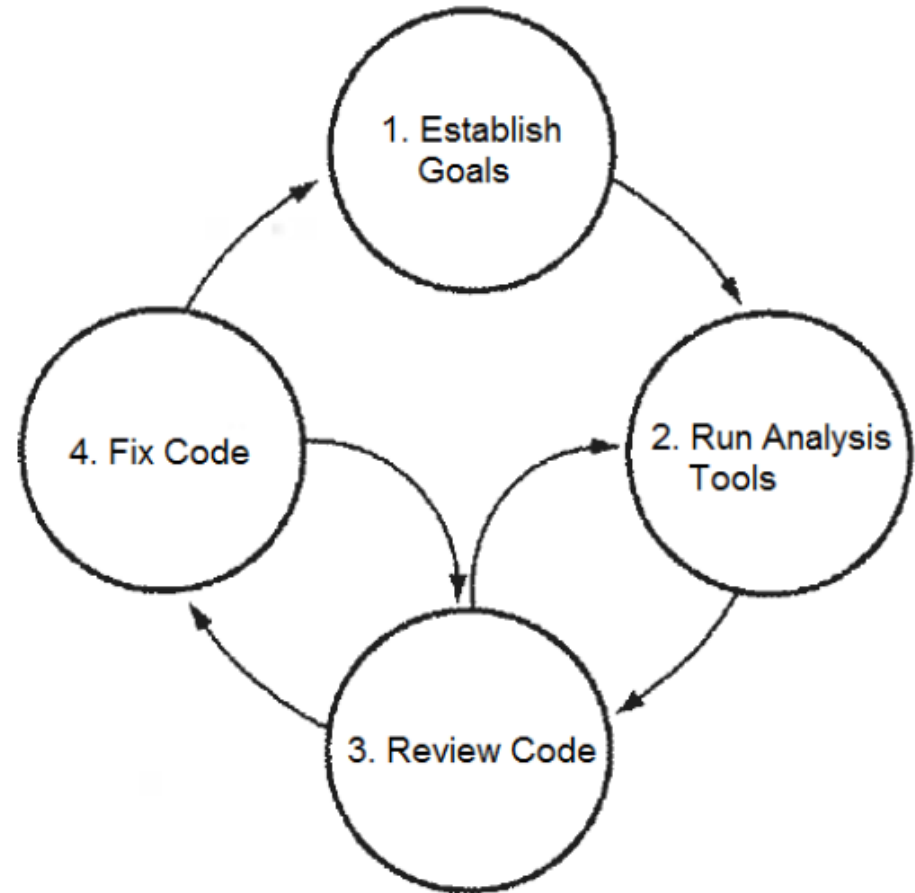
The image displays a code review interface. At the top, a pull request titled "Users list feature review" is shown, with a "Send Reminder" button and a "Tools" menu. Below this, the code editor shows a Java file named "UserListElement.java" with line numbers 1 through 42. A red circle highlights line 2785, which contains the code: `public UserListElement() {`. A large white arrow points from this line to the right, towards a detailed view of the issue.

The detailed view shows the issue details for ID 433, titled "Variable 'rs' is always 'null' in this expression." The issue is located in the file "ContourCommonDoSqlManager.java" at line 1092. The code snippet shown is:

```
try {
    rs = sqlManager.getConnection();
    PreparedStatement stmt = null;
    ResultSet rs = null;
    try {
        catch (Exception e) { ... }
    finally {
        sqlManager.closeStatement(stmt, rs, conn);
    }
    return rowsAffected;
}
```

The issue description states: "Variable 'rs' is always 'null' in this expression." The code snippet shows a try-catch block where the variable 'rs' is assigned a value from 'sqlManager.getConnection()' inside the try block, but it is used in a 'ResultSet' object before the try block ends, which explains why it is always null.

- Both of the tools looks similar
 - Why not integrate them?
- Benefits:
 - Time saving
 - User Friendliness
 - Process Enforcement
- Features
 - Source File View
 - File Tags
 - Analysis Overview panel
 - Hot Spot Review



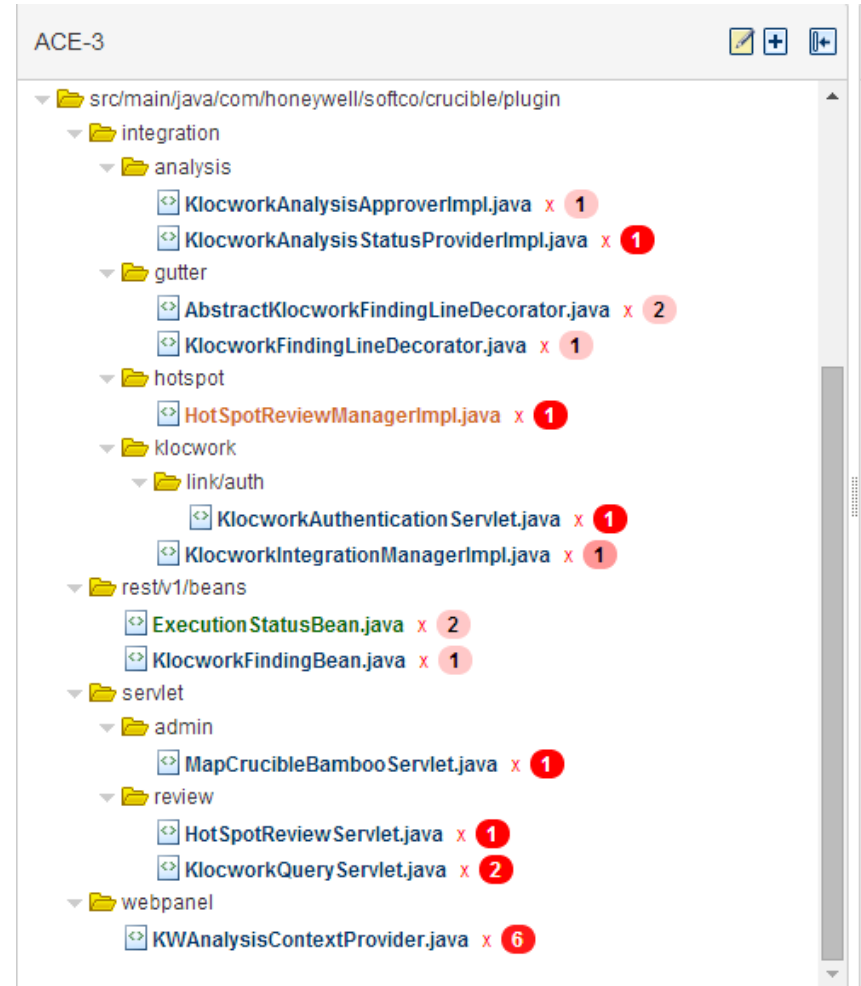
- Displays Klocwork findings directly in Crucible source code view

Null pointer dereference where null comes from constant
Null pointer dereference of 'status.getEntity()' where null comes from constant
Status: Analyze
Severity: Critical

```
    } else {  
        KlocworkAnalysisResult status = klocworkAnalysisStatusProvider.getAnalysisStatus(review.getPermaId().getId());  
        if (status == null || status.getAnalysisStatus().equals(AnalysisStatus.NOT_RUN)) {  
            result.putAll(getNotRunParams(data));  
        } else {  
            switch (status.getAnalysisStatus()) {  
                case FAILED:  
                    result.putAll(getFailedParams(data, status.getEntity()); //TODO  
                    break;  
                case DONE:  
                    result.putAll(getDoneParams(data, status.getEntity());  
                    break;  
                case IN_PROGRESS:  
                    result.putAll(getInProgressParams(data, status.getEntity());
```

File Tags

- Differentiate files under review based on the number and severity of Klocwork findings



Analysis Overview panel

- Displays statistic data about Klocwork findings in the review

Klocwork Analysis

Findings by Severity



Findings by Type



Top three files

- [KWAnalysisContextProvider.java](#)
- [ExecutionStatusBean.java](#)
- [KlocworkQueryServlet.java](#)

Total: 6 (5 Critical 1 Warning)
Total: 2 (2 Review)
Total: 2 (2 Critical)

Hot Spot Review

- „Hot Spot“ = parts of the code satisfying some condition

Add Content to Review ACE-3

Severity: Critical, Error, Warning, Review

Status: Analyze, Ignore, Not a Problem, Fix

State: New, Existing, Fixed

Number of Files 8

Edit Details Abandon Review Start Review Done

Thank You

Honeywell

Q&A

Honeywell

www.honeywell.com

