Lero THE IRISH SOFTWARE RESEARCH CENTRE

Science Foundation Ireland For what's next

University of Limerick
OLLSCOIL LUIMNIGH

Similarity between Feature Enhancement Reports and Modified Code

Joseph Burke, Supervised by: Dr. Jim Buckley, Dr. Nour Ali, Dr. Sebastian Herold

1 Research Problem

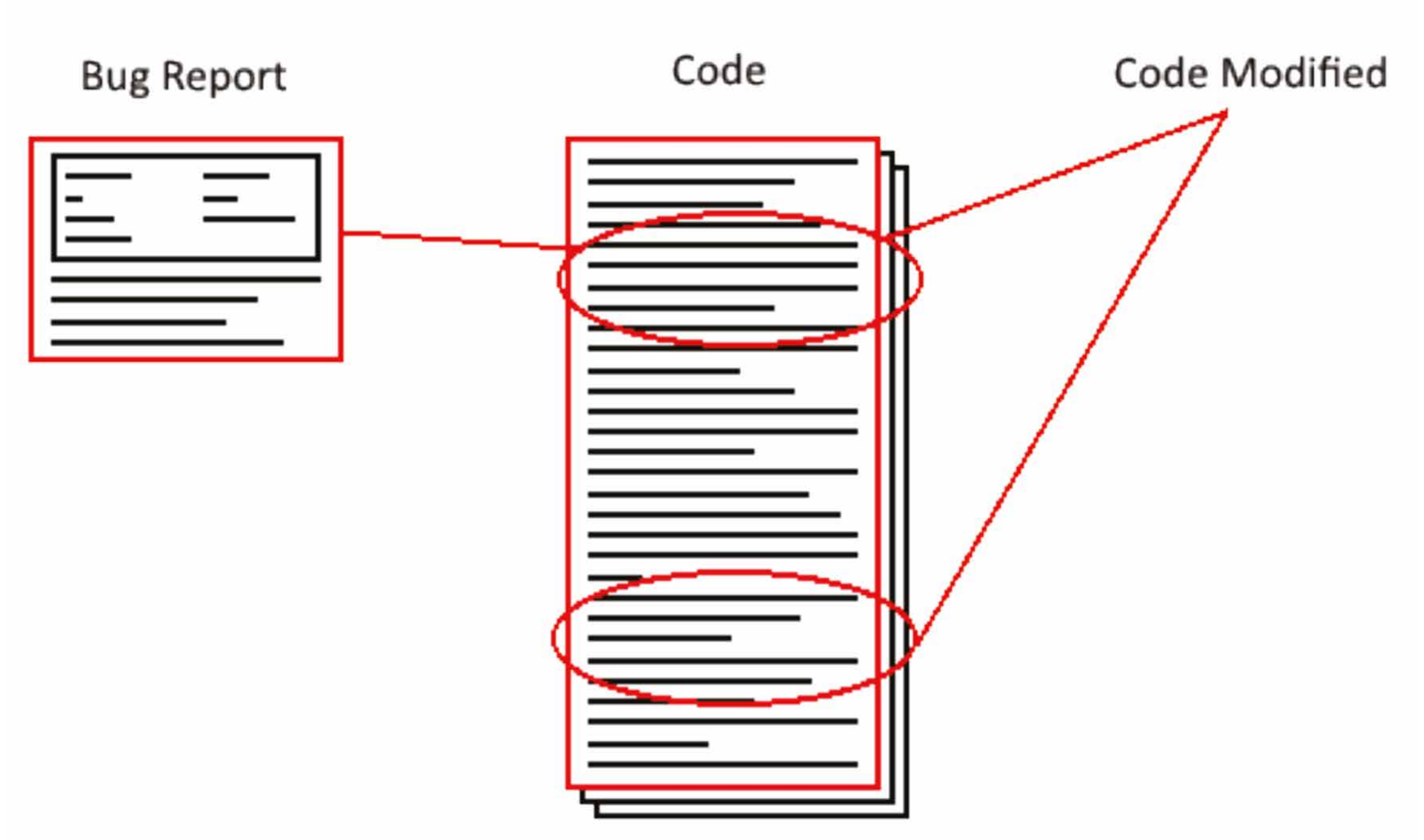
Feature Location is a prevalent component of software maintenance where developers try to map users' specification of system functionality to the source code. In trying to locate features textually there is an assumed correlation between the words used to describe the feature and the words (identifiers and comments) used in the associated source code. Understanding the correlation between the words used in these two vocabularies could greatly improve the results retrieved from query search and textual analysis when trying to locate reusable code for a feature addition, repairing code relevant to a bug and modifying functionality targeted for enhancement.

We automate the comparison of feature enhancement descriptions from bug reports and the code that was subsequently changed as a result of those reports. The study is carried out across five open source projects to find the similarity between the two vocabularies. Our preliminary findings show a high similarity between the feature enhancement vocabulary and class name identifiers in the modified source code.

Research Goals

- Evaluation of the textual link between feature descriptions and located code at method, class, and variable name level and in comments.
- Identify the difference between user and developer reports when comparing the similarity of the reports with the relevant code.
- Provide a tool that allows automatic extraction and analysis of bug descriptions and software repositories that facilitates this comparison.
- Leverage the results of this study and tool towards better feature location.

2 Solution - Examining Code Identifiers

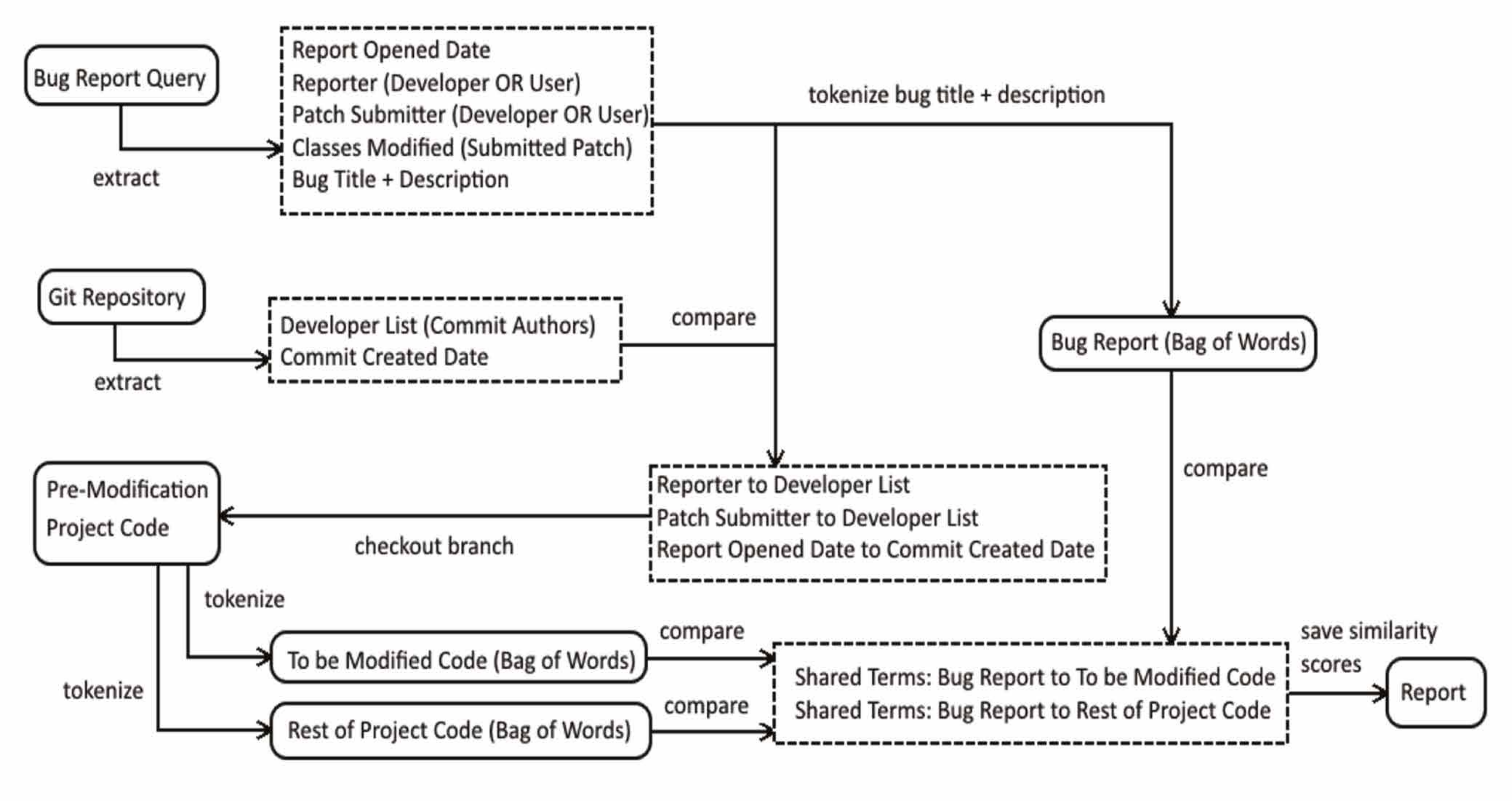


Bug Report and Code Artefacts after Modification

Using our bug-to-code comparison tool we automate the comparison of a large set of enhancement descriptions for bug repository systems. These descriptions are compared to the words used in the source code that subsequently changed to examine the correlation between textual documents and code. Four different types of identifiers are examined under the bug-to-code comparison.

- Class name - terms used in the name of the modified classes
- Method name - terms used in the method names of the modified class
- Variable names - terms used in the variable names of the modified class
- Comments - terms used in comments contained in the modified class

3 Empirical Design – Bug to Code Analysis


Bug Report Query
extract
Report Opened Date
Reporter (Developer OR User)
Patch Submitter (Developer OR User)
Classes Modified (Submitted Patch)
Bug Title + Description
tokenize bug title + description
Git Repository
extract
Developer List (Commit Authors)
Commit Created Date
compare
Bug Report (Bag of Words)
compare
Pre-Modification Project Code
checkout branch
Reporter to Developer List
Patch Submitter to Developer List
Report Opened Date to Commit Created Date
tokenize
tokenize
To be Modified Code (Bag of Words)
Rest of Project Code (Bag of Words)
compare
compare
Shared Terms: Bug Report to To be Modified Code
Shared Terms: Bug Report to Rest of Project Code
save similarity scores
Report

- Data is gathered from five open source software projects with 10+ years of software evolution and maintenance. 5000+ bugs in data set.
- Reports are divided into user and developer reports to check for a difference in similarity to the code depending on the reporter type.
  - *A developer is defined as someone who has made one or more commits that were accepted into the project.*
- Words from the report and code are tokenized, stemmed, stop list filtered and saved as separate instances in the bag.
- Then they are compared holistically, at class, method, variable and comment level, and by developer / user submission.

4 Research Findings

```
35:====================:205838:=====================:35
Desc unq:58 sim:16% tot:152 sim:41%
Desc Bag:[test, tag, por, good, desir, integrator, detectiv, avaialbl, deal, respond, proper, workbench, work, support , multipl, creat, class, appl, mov, not, monit, reintroduc, window, stuff, multi, interes, upda, concept, shawn, rena, launch, add, predictabl, activ, mak, perspectiv, similari, internal, similar, parallel, mock, rememb, manag, nic, edit, suppor, crea, slick, guid, effect, attach, pro, context, awar, method, chang, perspetiv]

Diff unq:12 sim:75% tot:20 sim:85%
Diff Bag:[monit, window, test, manag, perspectiv, multi, edit, awar, context, interact, us, usag]

Date: Tue Oct 09 18:46:00 BST 2007
Link: https://bugs.eclipse.org/bugs/show_bug.cgi?id=205838
Classes: AbstractUserInteractionMonitor.java, MonitorUiPlugin.java,  IContextAwareWindow.java, ContextEditorManager.java, UiUsageMonitorPlugin.java, MultiWindowMonitorTest.java, MonitorUiPlugin.java, ContextPerspectiveManager.java
```

Example Report to Code Comparison

In a preliminary study of 70 bug reports our findings show a high similarity of 70-77% between bug descriptions and class identifiers from the code to be modified.

Future Work

- Automate a larger set of reports to find similarity between bug reports and code identifiers.
- Configure existing Information Retrieval techniques with new textual correlations to see if feature location accuracy improves.

Ireland's European Structural and Investment Funds Programmes 2014-2020
Co-funded by the Irish Government and the European Union

European Union
European Regional Development Fund

United Technologies Research Center

SOUTHERN Regional Assembly
Promoting our Region