**PROBLEM**

- Many software projects incorporate bug repositories so that developers, testers, and users can report bugs that they have encountered, and call for more useful features or make suggestions for revision.
- This form of bug reporting is asynchronous and loosely organized, due to a project’s reliance on various developers, testers, and users.
- The cost of bug reporters’ searching the repository (to determine if their problem has been reported) can be higher than the cost of creating a new bug report.
- As a result, some reported bugs are not new ones but actually duplicates of existing bugs.
- To avoid the same bug being addressed by multiple bug fixers, it is necessary for a triager to inspect each submitted bug report to determine whether it is a duplicate. However, such inspect efforts can be very expensive or ineffective.

**EXAMPLE[1]**

- **Browser-Closing Bug:**
  - Bug-260331: After closing Firefox, the process is still running. Cannot reopen Firefox after that, unless the previous process is killed manually
  - Bug-239223: [Ghostproc] - [Meta] firefox.exe doesn’t always exit after closing all windows; session specific data retained
- **Document-Contain-No-Data Bug:**
  - Bug-244372: “Document contains no data” message on continuation page of NY Times article

**CHALLENGES**

- **Quantity of existing bug reports:**
  The large number of existing bug reports makes it challenging for the triager to examine all existing bug reports to detect duplication.
  - To address the challenge, the triager can retrieve a small subset of automatically suggested bug reports and compare the new bug report with each retrieved bug report to see whether the new bug report is a duplicate.
- **Quality of the list of suggested duplicate bug reports:**
  - Existing approaches[2] adopt info-retrieval techniques to measure the similarity between bug reports using NL info. Although these approaches already provide some practical help to triagers, there is still a need to improve these approaches due to their low recalls.
  - To address the challenge, both NL info and exec info can be mined for improving detection of duplicate bug reports.

**APPRAOH**

- **Step 1:** Calculate the NL-based Similarities (NL-S) between the new bug report and existing bug reports.
- **Step 2:** Calculate the Exec-info based Similarities (E-S) between the new bug report and existing bug reports.
- **Step 3:** Retrieve potential target reports using the two kinds of similarities based on heuristics.

**ARCHITECTURE of the JDF Tool**

- JDF RCP UI
- JDF Common
- JDF Service

**REFERENCE**

3. JDF. http://sites.google.com/site/asergrp/projects/jazz

Supported in part by 2008 IBM Jazz Innovation Award