SZZ Revisited

Chadd Williams
Pacific University
Jaime Spacco
Colgate University
Interesting commits

- Commits that fix bugs
  - SZZ highly accurate for finding these

- Commits that induce bugs
  - SZZ identifies these as well
  - Less well studied
  - How accurate are the identified fix-inducing commits?
    - How often do they introduce bugs?
  - Can we characterize changes that induce fixes?
What do FIs look like?

- Large changes?
- Initial addition of code?
- Multiple bug-fixes per line?
- Are FIs edited more than other lines?
State of the art for identifying bug fixes

- SZZ
  - Sliwerski, Zimmerman, Zeller
  - Improved upon by Kim et al.
- Heuristic method to locate bug fix commits
  - commit meta-data
    - regular expressions
  - bug database
    - bug numbers
    - committer / bug assignee
Identifying bug-fix and fix-inducing commits

*image borrowed from original SZZ paper*
Annotation graphs

- CVS Annotate
  - finds origins of lines using diff
  - not accurate enough

- Annotation graphs
  - ignore some formatting changes and comments
  - doesn’t map through large changes
  - not precise in some cases
Annotation Graphs
Annotation Graphs (2)
Goals:

- Use line number tracking
  - track unique lines across multiple versions
    - Canfora et al., MSR 2007
- Ignore all format changes with DiffJ
  - AST-based source code diff
    - unfortunately, not as detailed as ChangeDistiller
- Validate fix-inducing commits identified by SZZ
Unique line mapping
DiffJ

- AST-based, but turns out to be quite inaccurate
  - doesn’t examine if/for/while
  - handles method signature changes poorly
  - course-grained range of changes
- successfully ignores all comments and formatting changes
How accurate are fix-inducing commits?

- SZZ shows that fix-inducing commits on average are larger than non-fix inducing commits
  - based on number of files changed

- We’d like to:
  - examine changed lines in FIs
  - evaluate accuracy of FIs
Studied Eclipse

- first 35,000 trunk revisions of JDT
- had local copy of bug database
- SZZ identified 5,000 transactions as bug-fix commits
- 3,400 transactions as containing at least one fix-inducing change
Preliminary results

- Chose 25 bug fix commits that were “small”
  - modified 50 total lines
- Manually verified that 43 of the 50 modified lines were involved in a bug fix
  - 23 out of 25 of the bug-fix commits
- Prior work showed SZZ to be highly accurate, so this was not unexpected
More preliminary results

- Of the 43 lines that fixed a bug
  - tracked 33 lines back to the fix-inducing commit identified by SZZ, and found the bug
    - 4 went back farther
    - 6 we couldn’t track due to issues with line number mapping
Some anecdotal observations

- Many bug-fix commits modify if statements
  - fix-inducing commits adds:
    - if \((x \,||\, y)\)
  - bug-fix
    - if \((x \,&&\, y)\)
Also saw:

- fix-inducing adds a method call:
  - new Foo(a, b, null)
- bug-fix:
  - new Foo(a, b, c)
- probably a response to a non-local change
  - true fix-inducing change probably changed the body of Foo()
Conclusions

- Must verify many, many more bug-fix and fix-inducing commits
- Our line mapping implementation was not accurate enough
  - highly accurate for small changes
  - trying unique statement mapping instead
  - DiffJ did a poor job of detecting method signature changes
Thank you!
Key assumption

- Fix-inducing commit immediately precedes the bug report
  - Bugs are reported quickly (in terms of edits) after they are introduced
  - otherwise we risk losing things in a large diff hunk
notes on ancestor picture

- Large ancestor set picture
- Even if we can’t track everything, we can track enough of the lines to get into the general area
  - pick the right chunk?
  - anchoring a couple of lines can be useful
    - context, maybe they are small edits
Idea:

- Try to track unique lines across many versions of software
  
  - based on Canfora et al’s work at MSR ‘07
Better understand fix-inducing commits

- Lots of focus on bug-fix commits
  - comparatively little of fix-inducing commits
- We need to know what lines were changed
- Would like to characterize the types of changes
- Track line numbers back to their origin
  - or until we lose them
- want to run DiffJ to know what kinds of changes take place in fix-inducing commits