5. CONCLUSION AND FUTURE WORK

Three stages: bug introduction, bug detection, bug fixing. Bugs are not typically introduced and detected by the same developer. SStuBs are not detected earlier than other bugs. Tests are not typically written before detection.

To lower detection time, developers should check (and correct) their code more extensively and/or work with shared code ownership.

Developers should write (more) test code to reduce the number of SStuBs.

Repository owners should be more strict with developers working structured and precisely, to lower the bug-detection time.

Future: Research whether the bug-fixing time will be less if developers consistently follow Git's and repository's standards.

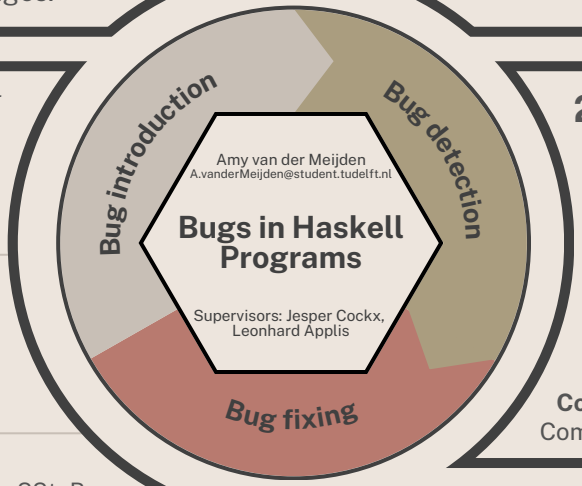
Future: Research whether tests are added after the bug-fixing.

Future: Research non-open-source repositories and other languages.

1. RESEARCH QUESTION AND SUB-QUESTIONS

What are the different stages of bugs in Haskell programs?

RQ1: Which, if any, definition of bugs is most applicable for open-source Haskell repositories?
RQ2: How much time or commits pass between the introduction and detection of bugs? Are tests written before the detection?
RQ3: Are bugs typically detected by the same developer who introduced them?
RQ4: How much time or commits pass between the detection and solving of bugs? Are tests written after the detection?
RQ5: Are the so-called Simple Stupid Bugs detected earlier than other bugs?



4. RESULTS

100 bugs spread over 9 open-source Haskell repositories.

RQ1: Currently the definition is fairly broad. Based on the existing definitions and analysis of these bugs, we deducted a definition that could be applied to open-source Haskell repositories.

RQ2: Median time between bug introduction and detection: 381 days or 332 Commit Cycles. 14 BFCs had a test before detection.

RQ4: Median time between bug detection and fixing: 3 days or 2 Commit Cycles. 25 BFCs added a test in the PR. 61 BFCs did not include a test.

RQ3: 24 bugs with same introducer and detector. Median of 100 Commit Cycles (30% of global median). 19 of 24 part of the fastest 50% of all bug detections.

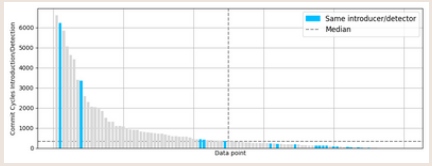


Fig 1: Commit Cycles between bug introduction and detection, same introducer/detector, sorted.

RQ5: 14 of 100 SStuBs, spread over all commits. SStuBs are fixed 4.2 times faster with same introducer. SStuBs occur 2.5 times more often when no test is written.

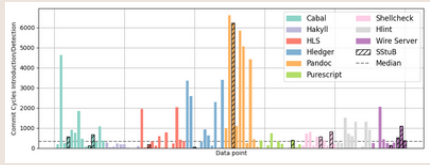


Fig 2: Commit Cycles between bug introduction and detection, SStuBs.

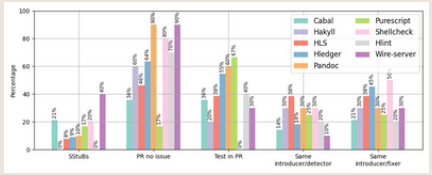


Fig 3: Values per Repository.

Values per Repository
Purescript has the second lowest detection time, most PRs with included test, and least PRs without linked issue.

2. BACKGROUND

Simple Stupid Bug (SStuB): Compile both before and after repair, tedious to manually spot, but only need a simple fix.

Bug-Introducing Commit (BIC): Commit introducing a bug in a program. Identified using Rodríguez-Pérez et al.'s model.

Bug-Fixing Commit (BFC): Commit fixing a bug in a program.

Commit Rate: Number of commits introduced per hour.

Commit Cycle: Metric for normalising time differences.
Commit Cycles = Time difference * Commit Rate

3. METHODOLOGY