On the Rise and Fall of Simple Stupid Bugs: a Life-Cycle Analysis of SSstuBs

Balázs Mosolygó, Norbert Vándor, Gábor Antal, Dr. Péter Hegedűs
Introduction

- Larger systems, and tight deadlines -> easy to make mistakes
- SSStuBs - ManySSStuBs4J
- Our research questions:
  - **RQ1**: Are SSStuBs more likely to occur in code that is modified by multiple developers?
  - **RQ2**: Are SSStuBs more likely to appear in newly added or modified code blocks?
  - **RQ3**: How long does it take to fix SSStuBs, do authors notice their own mistakes faster?
  - **RQ4**: Can PMD flag SSStuB lines as being error prone?
Approach

Section A
- ManySStubs4J
- Commit source search
- Source commits for each stub

Section B
- pmd flag search
- Git information processing
- Results
Results

- **RQ1**: Are SStuBs more likely to occur in code that is modified by multiple developers?
  - The SStuBs typically appear more in larger chunks of code written by the *same developer*.

- **RQ2**: Are SStuBs more likely to appear in newly added or modified code blocks?
  - Most SStuBs are added in the *same commit as their neighboring lines*.

- **RQ4**: Can PMD flag SStuB lines as error prone?
  - Lines with SStuBs are *not* considered error prone by PMD.
Results – RQ3

- **RQ3**: How long does it take to fix a SStuBs, do authors notice their own mistakes faster?
- SStuB fixes take too long
- Quickly noticed by the same developer who introduced them
Acknowledgement

The presented work was carried out within the SETIT Project (2018-1.2.1-NKP-2018-00004) and supported by the Ministry of Innovation and Technology NRDI Office within the framework of the Artificial Intelligence National Laboratory Program (MILAB).
Thank you for your attention!

Gábor Antal
antal@inf.u-szeged.hu

https://github.com/MBalazs8796/MSR2021_LifeCycle