Why Don't Software Developers Use Static Analysis Tools To Find Bugs?

Brittany Johnson, Yoonki Song, and Emerson Murphy-Hill and Robert Bowdidge

Presented By : Ranjodh Singh

Outline

- Introduction
- Related Work
- Research Questions
- Methodology
- Results
- Threats to validity
- Conclusion

Introduction

- Software defects or bugs can cost companies significant amount of money.
- Static Analysis Tools can make finding bugs faster and cheaper than manual inspections.
- Still such tools are underused.
- How these tools can be improved.

What are static analysis tools ?

- Static analysis tools are well defined programming rules that provide a means for analyzing code without having to run the code
- **Tools**
  - FindBug, Lint and IntelliJ IDEA
- **Benefits**
  - Faster and cheaper.
  - Reveal errors that do not manifest themselves
  - Enforce coding standards.
  - Ensure higher quality.

Related Work

- Focused on the correctness and functionality of tools.
- **Ayewah and Pugh** claimed that static analysis tools should find bugs as early as possible in the Development cycle, when they are cheap to fix
- **Khoo et al.** examined the user interface and how they can be improved. Developed Path Projection.
- **Heckman and Williams** developed a benchmark-Faultbench to compare and evaluate alert prioritization techniques
- **Laymen et al.** investigated factors developers consider when addressing defects

Research Questions

1. What reasons do developers have for using or not using static analysis tools to find bugs?

2. How well do current static analysis tools fit into the workflows of developers?

3. What improvements do developers want to see being made to static analysis tools?

Methodology

- Conducted semi-structured interviews.
- Script of questions was prepared prior to research.
- Trial interviews done to modify script.
- Manually transcribed the interviews
- Coded the transcriptions to perform qualitative analysis.

Participants

- 16 Professional developers
- 4 Graduate students with previous industry experience
- 3-25 years of development experience
- 2 participants were remotely interviewed

TABLE I
DESCRIPTIVE STATISTICS REPORTED BY PARTICIPANTS.

Participant	Open-source Tools	Closed-source Tools	Local
Abby	FindBugs	IntelliJ	Yes
Adam	CheckStyle, FindBugs, PMD	IntelliJ	Yes
Andy	FindBugs, Lint	Jtest [20]	Yes
Chris	CheckStyle, FindBugs, Lint	Coverity	Yes
Cody	Dehydra	-	Yes
Frank	-	-	Yes
Gordon	Lint, CheckStyle, FindBugs	-	Yes
Jake	FindBugs, Lint	FlexLint, Klocwork Insight [21], Visual Studio [22]	Yes
James	Lint, CheckStyle, FindBugs	Visual Studio	Yes
Jason	Lint, FindBugs	-	Yes
John	CheckStyle, Copy/Paste Detector(CPD), FindBugs, Lint, PMD	CodePro [23]	Yes
Jordan	CheckStyle, FindBugs, PMD	Jtest	Yes
Josh	FindBugs, Lint	Coverity	No
Lee	CheckStyle, FindBugs, Lint	Visual Studio	Yes
Matt	Lint	FlexLint, PyCharm	Yes
Mike	cpplint, Lint	-	Yes
Phil	-	-	Yes
Ray	CheckStyle, FindBugs	-	Yes
Ryan	FindBugs, Lint	Coverity	Yes
Steve	CheckStyle, CPD, FindBugs, Lint	IntelliJ	Yes
Tony	CPD, FindBugs, Lint, Splint,cpplint, PMD, Checkstyle	Coverity	No

Methodology…

Organization of Interviews

- Interviews focused to learn developer's relevant experience with finding defects using static analysis.
- Organized into 3 main parts:
  - Part 1 : Questions and Short Responses
  - Part 2 : Interactive Questions
  - Part 3 : Participatory Design

Part 1 : Questions and Short Responses

- **GOAL :** To get knowledge of their general usage , understanding and opinion of static tools.
- **Questions:**
  - How was your first experience with a static analysis tool?
  - Have you ever used a static analysis tool in a team setting? Was it beneficial and why?
  - Have you ever consciously avoided using a static analysis tool? Why or why not?
  - What in your opinion are the critical characteristics of a good static analysis tool?

Part 2 : Interactive Questions

- **GOAL :** To observe developers actually using a static analysis tool.
- **Questions:**
  - Now that you have run your tool and gotten your feedback, what is your next move(s)?
  - Do you configure the settings of your tool from default? If so, how?
  - Does this static analysis tool aid in assessing what to do about a warning?
  - Do you feel that “quick fixes” or code suggestions would be helpful if they were available?

Part 3 : Participatory Design

- **GOAL:** To get the participants to make design suggestions for improving static analysis tools
- Participatory Design concept was utilized.

Coding Interview Questions

- Manually transcribed each interview.
- Each transcription was coded.
- Coding Categories defined:
  - Tool Output
  - User Input/Customizability
  - Supporting Teamwork
  - Result Understandability
  - Workflows
  - Tool Design

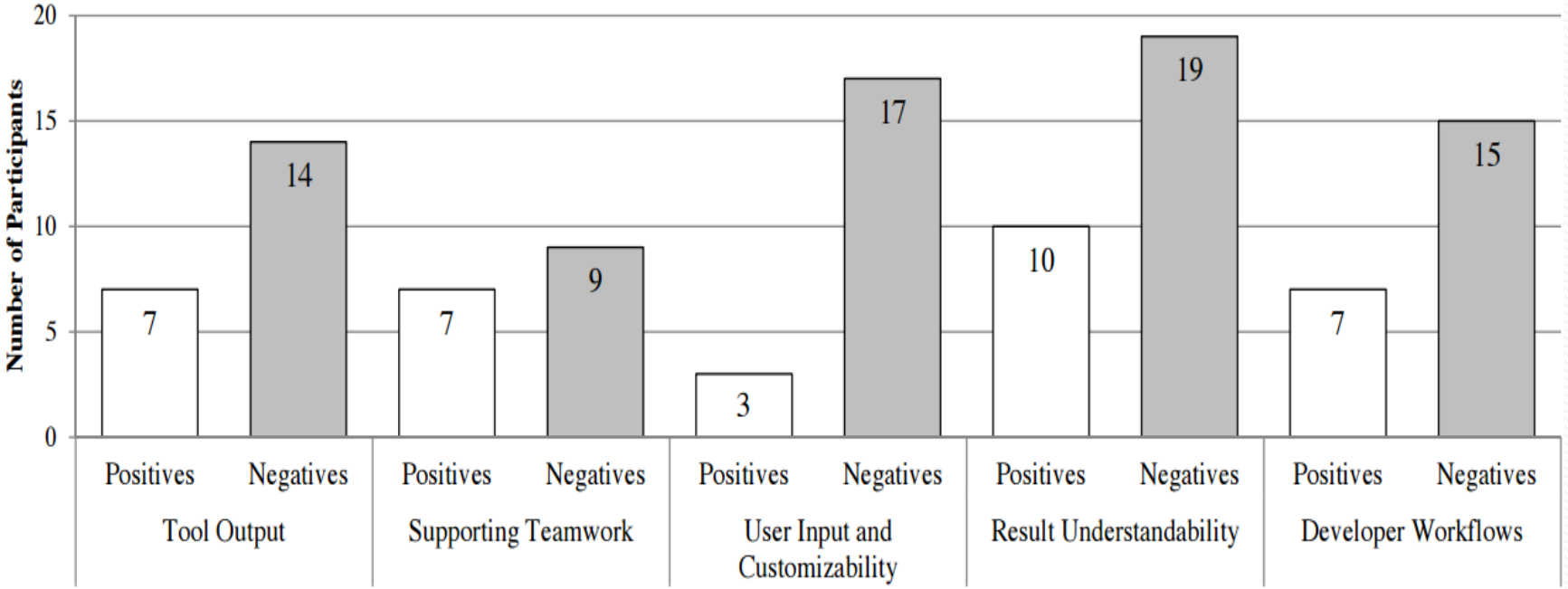
Results

- Research questions were answered by linking the questions to coding categories.

Category	RQ1	RQ2	RQ3
Tool Output	Yes		
User Input/Customizability	Yes		
Supporting Teamwork	Yes		
Result Understandability	Yes		
Workflows	Yes	Yes	
Tool Design			Yes

Positive and Negative Remarks

Each coding category was further divided into positive and negative remarks



Results…

RQ1 : Reasons for use and Underuse

- **Positive Comments:**
  1. Automates the process of finding bugs and saves time and effort.
  2. Developers may use it if it is available in IDE.
  3. To support team development efforts.
  4. Customizability of tool

Results…

RQ1 : Reasons for use and Underuse

- **Negative Comments:**
  - Tool Output:
    1. Number of warnings can be high and are poorly presented.
    2. Bugs are dumped onto the screen with no distinct structure.
    3. 14 out of 20 participants.
  - Collaboration:
    1. Lack or weak support for teamwork.
    2. Can not share the settings with other team members.
    3. Cloud Storage feature of FindBug to share and discuss the bug report but it takes developer out of development environment
    4. 9 out of 20 participants.

Results…

RQ1 : Reasons for use and Underuse

- **Negative Comments:**
  - Customizability:
    1. Difficult to customize the tool to filter the bugs.
    2. Not allowed to record judgements about defect.
    3. 17 out of 20 participants.
  - Result Understandability:
    1. Tool does not enough information to assess what problem is.
    2. 19 out of 20 participants.

Results…

RQ2 : Workflow Integration

- **Positive Comments:**
  1. To fix bugs at the time they are introduced.

- **Negative Comments:**
  1. Disjoint process
  2. Runs slow, even though it is IDE integrable.
  3. Prevent developer from being productive.

Results…

RQ3 : Tool Design

- Most of the proposed designs were for warning notification and quick fix display.
  1. **Quick Fix Design:**
     - Able to preview the fix.
     - Show a diff of code .
     - Three option dialog box.
  2. **Warning Notification and Manipulation Design**
     - Faster and efficient feedback.
     - Do not disrupt the work.
     - Ability to save their judgements.
     - Temporary suppression.
     - Share notification

Results...

RQ3 : Tool Design

3. **Other Design Ideas:**
   - Visual output like pie-style diagram of the project and bugs in it.
   - Heat Map to represent the error location and its severity using colors.

Results…

Threats to Validity

- Less no. of participants
- Interviewed only those developers who have used static analysis tools.
- Some of the developers were not familiar with code .
- Some participants had tool building experience.

Future Work

- Research can be put forward by finding the solution to implement interactive quick fixes.
- Evaluate prototype prepared in this research with large no. developers.
- Factors mentioned in paper can be used to improve the design of tool.

Conclusion

- False positives and overload cause the developer's dissatisfaction.
- Quick Fix may cause developer to be hasty.
- Tool must allow easy customization.
- Allow developer to share settings..

Discussion

- What is the appropriate time to perform automatic static analysis ?
- Do you think tools that report more no. of defects are better ?
- Do you think that static analysis of code is enough to ensure the quality of code ?
- Do you have any opinion to further improve the design of tool ?