Analyzing Student Debugging Practices and Project Outcomes

Rifat Sabbir Mansur, Ayaan M. Kazerouni, Stephen H. Edwards, and Clifford A. Shaffer

Department of Computer Science, Virginia Tech

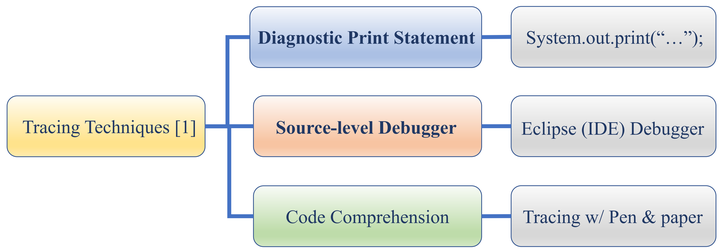
Identifying Debugging Behaviors in Intermediate Programmers

- Intermediate programmers often spend a lot of time debugging
- In a post-CS2 Data Structures and Algorithms course, we used IDE clickstream data to analyze detailed debugging behavior
- We hypothesize that there are differing debugging behaviors exhibited, and that differing behaviors lead to differing project outcomes

Research Questions

- To what *extent* is a particular debugging technique being used?
- Does it matter *when* in the project lifecycle that debugging takes place?
- Can a particular *type* of debugging technique lead to better project score?

Different Debugging Techniques



We focus on two debugging techniques:
1. Diagnostic Print Statements and
2. Source-level Debugger

Diagnostic Print Statement (DPS) Classifier

We want to identify those print statements that the students use for debugging purposes *i.e.* DPS; this is not a trivial process.

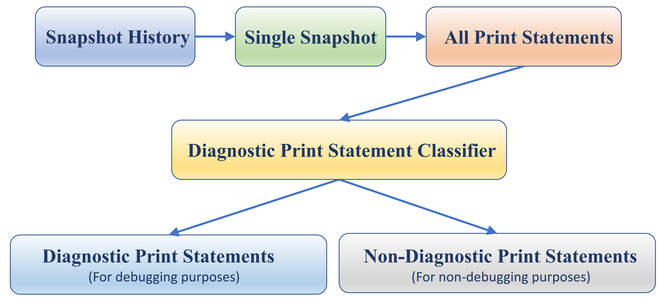
1. Exclude Commented Print Statements
2. Exclude Trivial (Delimiter) Print Statements
3. Exclude Project Specific (Required) Statements

DPS Examples

```
System.out.print(tempValue);
System.out.print("Success!");
```

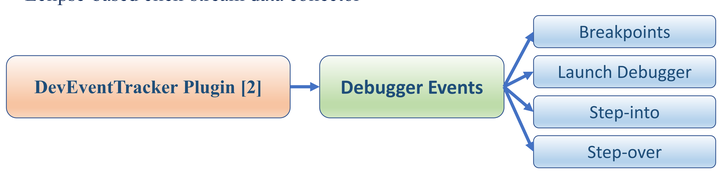
Extracting Debugging Behavior

1. Extracting DPS from Code Snapshots



2. Extracting Eclipse Debugger Events via DevEventTracker* [2]

* Eclipse-based click-stream data collector



Findings

Distribution of Different Debugging Techniques

- **87.21%** of students used the DPS
- **75%** of students used the Eclipse Debugger.
- Most students use both the DPS and the Eclipse Debugger.
- Debugging early and often showed a weak positive correlation with project performance.

$$corr\ coeff = \mathbf{0.19} \quad p - value < \mathbf{0.001}$$

Preliminary Evaluation using DPS Classifier

- **12** sample projects (3 samples from 4 different Projects)
- Print Statements: Total 1467* $(\mu = 122,\ \sigma = 89)$
- DPS: Total 611* $(\mu = 51,\ \sigma = 54)$

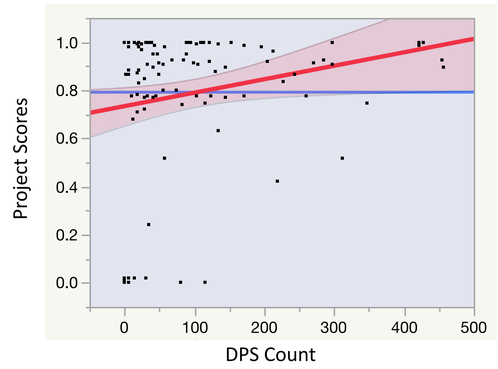
(* for all the intermediate snapshots)

Accuracy: 100%

Therefore, this classifier works well on this dataset.

Relationship with Project Score

1. Project Score vs DPS



$$p - value = 0.0347$$

$p - value < \alpha$ (0.05)
Hence, there is significant evidence that project scores correlate to DPS count.

$$R^2 = 0.0456$$

Therefore, 4.56% variability in the project score can be explained by DPS count.

Figure: Project Score vs DPS Count (for Project 1)

2. Project Score vs Debugger Events

- More Debugger Events (step over, step into*)* → lower Project Score
  (The same student performed better in another project)
- Step over: p-value = **0.039** and Step into: p-value = **0.005**

Therefore, students tend to get lower Project Scores when they spend too much time on the **same bug**.

Key Results

- Students tend to perform better on the project when debugging takes place earlier in the overall project life-cycle.
- There is weak yet statistically significant evidence that both DPS and Debugger Events correlate to overall Project Score.
- Only 4.56% variability in Project Score can be explained by overall DPS count

Future Work

- We plan to focus on individual debugging sessions to find if one type of debugging technique is more effective than another.
- We plan to find out how the students verify that the bug is fixed, such as manual checking, writing new test-cases, and/or by submitting the project for evaluation.

References

[1] Murphy, Laurie, et al. "Debugging: the good, the bad, and the quirky--a qualitative analysis of novices' strategies." ACM SIGCSE Bulletin. Vol. 40. No. 1. ACM, 2008.

[2] Kazerouni, Ayaan M., et al. "DevEventTracker: Tracking development events to assess incremental development and procrastination." Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education. ACM, 2017.

This work is funded in part by NSF grants DUE-1245334 and DUE-1432008