Identifying Student Contributions through Automated Team Summaries

Kai Presler-Marshall, Bowdoin College
Sarah Heckman, North Carolina State University
Kathryn T. Stolee, North Carolina State University
Questions? Rude Comments? k.preslermarshall@bowdoin.edu

Motivation

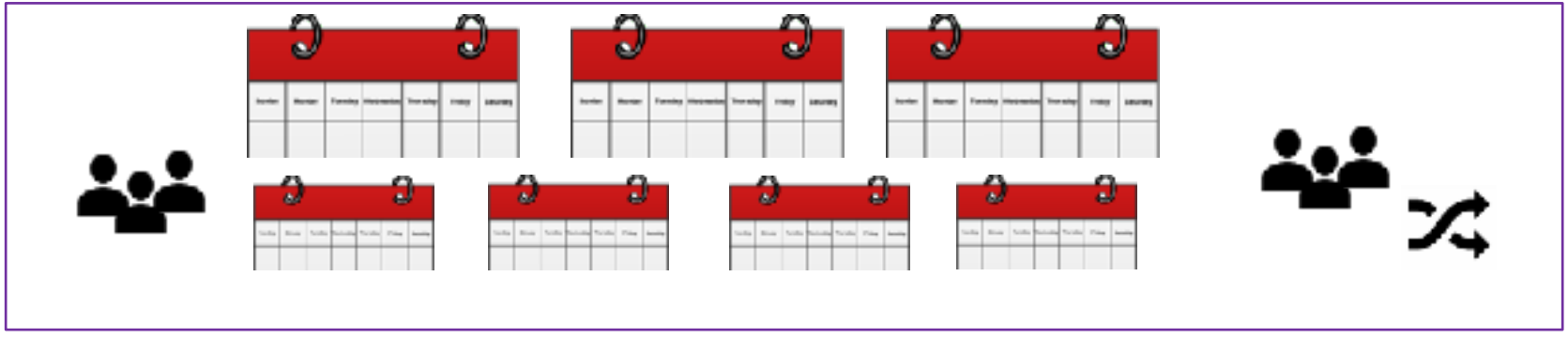
- Teaming is a core element in professional software engineering
- Thus, it's essential for CS and SE programs to teach students how to work in teams
- Some students may be inclined to freeride off of the contributions of their peers, receiving a grade not commensurate to their contributions

Motivation

- Accurately identifying students' contributions to team projects remains an open challenge
- Thus, teaching assistants may struggle to give students consistent & actionable feedback on their contributions
- *Can autogenerated summaries of students' code contributions assist TAs in giving better feedback?*

Course Context

- Sophomore-level Java programming course
- Lecture & projects
- **Associated lab section**
  - Students work on labs in small teams
  - Lab grading is *mostly* automated



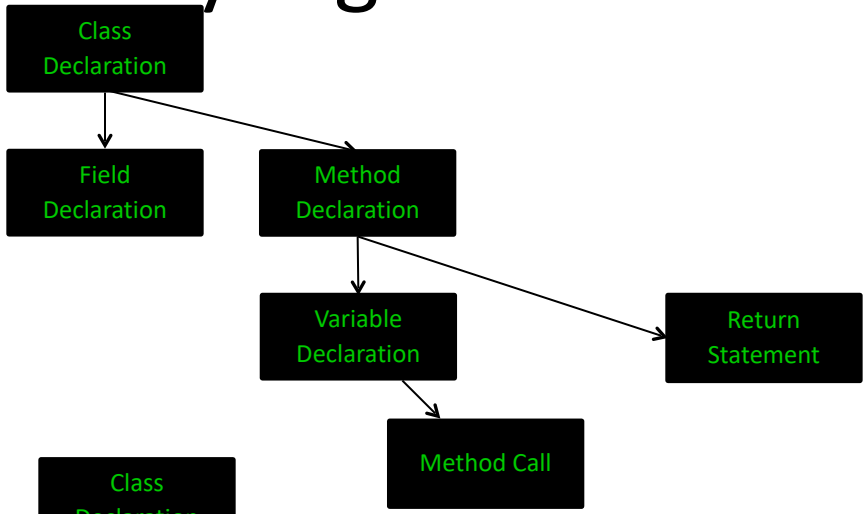
x3

Selected Research Questions

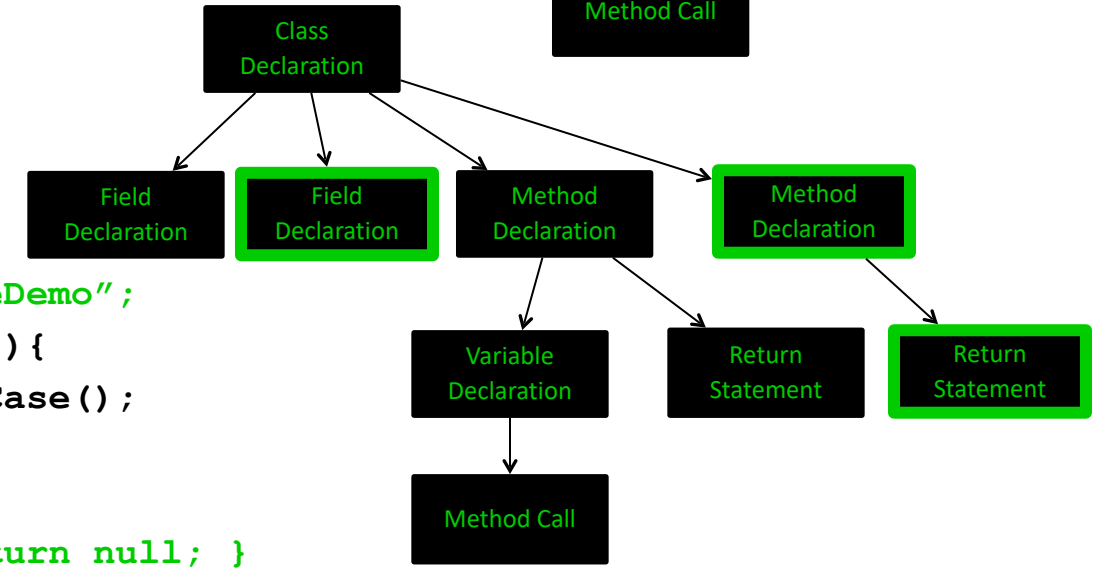
- Can automated summaries of student contributions enable:
  - **RQ1**: Faster grading by TAs (short answer: no)
  - **RQ2**: More consistent grading by TAs?
  - **RQ3**: Less frustrating grading from perspective of TAs?
  - **RQ4**: Better feedback for students?

Contributions Summary Algorithm

```
public class KaiDemoClass {
  private String aField = "demo";
  public String capitaliseAndReturn(){
   String modified = aField.toUpperCase();
   return modified;
  }
}
```




```
public class KaiDemoClass {
  private String aField = "demo";
  private String secondField = "moreDemo";
  public String capitaliseAndReturn(){
   String modified = aField.toUpperCase();
   return modified;
  }
  public Object anotherMethod(){ return null; }
}
```



Study Outline

- I recruited 13 former or current CS TAs
  - 12 of 13 TAs had experience grading team-based projects
- I tasked participants with:
  - Grading projects
  - Considering feedback from their peers
  - Reflecting on the experience

Part 1

Task Number	Click here when you're ready to start this team	Grade Contributions After this timestamp:	Automated Summaries	Repo	Student 'A' Grade	If you gave 'A' less than full credit, provide feedback on what they should do differently next week. If you give full credit, you can still provide feedback!	Why did you give 'A' the grade you did?
Example	☑	9/14/2021 10:20 AM	https://pages.github.ncsu.edu/Grad	https://github.ncsu.edu/GradingStudy/LabRepositoryA/tree/[illegible]	5	You've written some good tests this week, but please make sure that you're more involved with tasks next week	They tested some valid and invalid scenarios, but didn't do as much as their teammates
1	☑	11/16/2021 10:20 AM	https://pages.github.ncsu.edu/Grad	https://github.ncsu.edu/GradingStudy/LabRepository	10		They seemed to make reasonable contribution
2	☑	10/25/2021 7:10 PM	No automated summary available	https://github.ncsu.edu/GradingStudy/LabRepository/tree/[illegible]	10		reasonable contribution on both implementation + testing
3	☑	9/21/2021 10:20 AM	No automated summary available	https://github.ncsu.edu/GradingStudy/LabRepository/tree/[illegible]	10		reasonable contribution on both implementation + testing
4	☑	10/19/2021 2:40 PM	https://pages.github.ncsu.edu/Grad	https://github.ncsu.edu/GradingStudy/LabRepository/tree/[illegible]	0	We would like to see better team contribution next lab, please work for your team to split the tasks	no contribution
5	☑	11/9/2021 10:20 AM	https://pages.github.ncsu.edu/Grad	https://github.ncsu.edu/GradingStudy/LabRepository/tree/[illegible]743bab0e0539d49c07b156cc	10		reasonable contribution on both implementation + testing
6	☑	11/2/2021 10:10 AM	No automated summary available	https://github.ncsu.edu/GradingStudy/LabRepository/tree/[illegible]	5	You've contributed to implementation this week, but please make sure that you're more involved with tasks next week	no enough contribution. Most of changes are javadoc.
[illegible]	☑	[illegible] PM	No automated summary available	https://github.ncsu.edu/GradingStudy/LabRepository/tree/[illegible]	10		good contribution to the implementation + tests

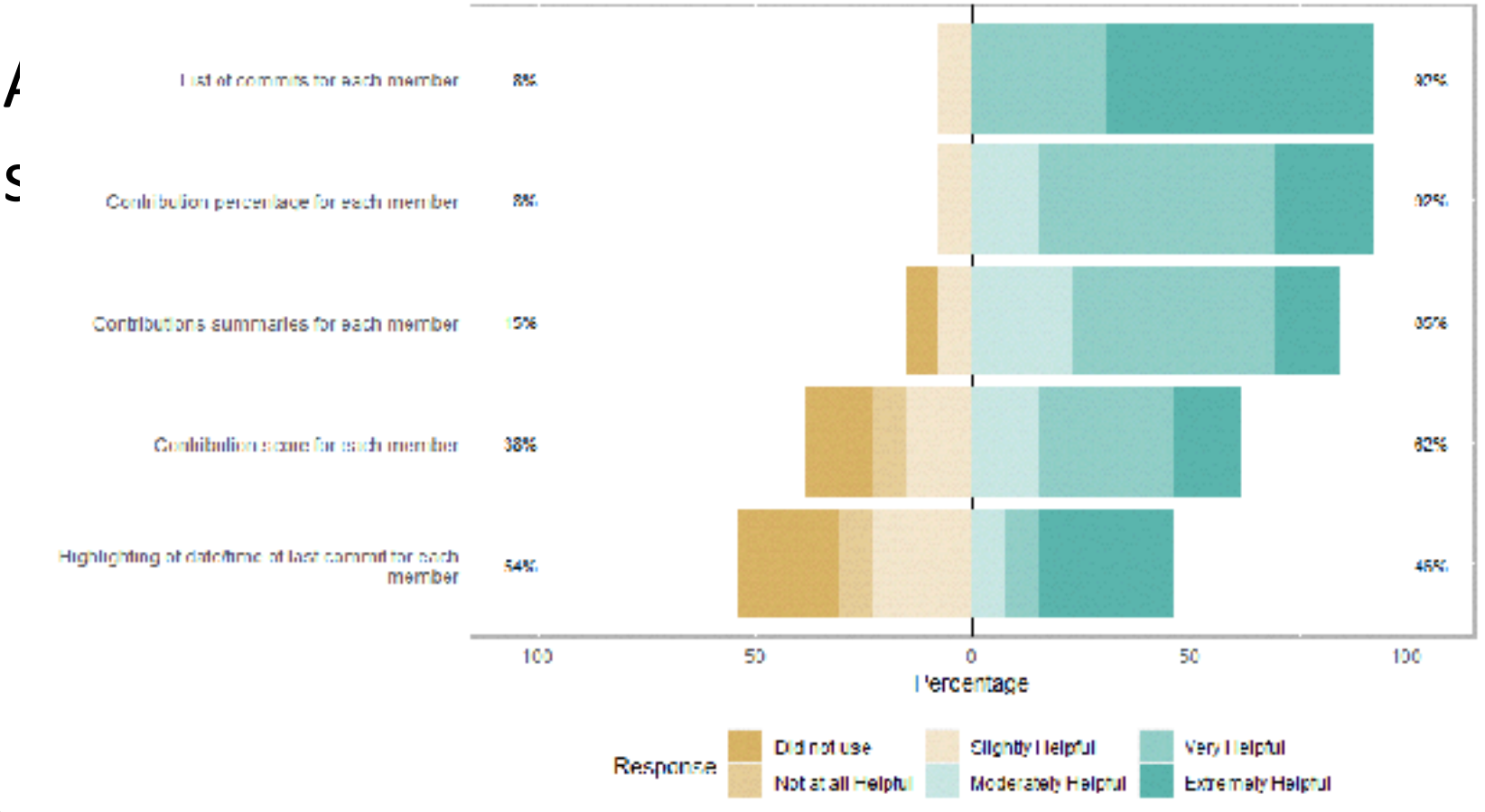
Part 2

Grade 1	Comment 1	Grade 2	Comment 2	I would choose:
0	Try to contribute more by coordinating with your teammates and asking what help is needed.	5	Good job on fixing code and adding tests! Next time see if you can contribute more on the implementation side of things.	Either (no difference)
10	good job in TSM and code contribution	10	Great work both on implementation and testing	Either (no difference)
10	Good job in implementation and testing	10	There is some implementation and testing along with the Javadocs. But it would be better if you did some more implementation.	Comment 2
0	fixing checkstyle is not enough contribution	5	you need to write more tests and implementations rather than fixing typos and generating javadocs	Comment 2
0	It looks like all you mostly did was documentation this week, in the future please try and split the tasks up so everyone has a coding portion.	10	Great work both on implementation and testing.	Comment 1
0	Done nothing	5	Good work on making sure the readme is updated with progress, and generating documentation. However, try to contribute more on the implementation or testing as well.	Comment 2
0	fixing typos and capitalization is not a valid contribution	10	Excellent work on making contributions on the implamentation, testing, and the documentation! Keep up the good work.	Comment 1

RQ2: Grading Consistency

- TAs grade projects much more consistently (*q = .021*) with contributions summaries to assist them
- However, consistency still remains a challenge even with contributions summaries (*α = .609*)

RQ3: Grading Preferences



RQ4: Feedback Quality

- TAs consider feedback from assignments graded with contributions summaries more actionable ($p = .031$)
- TAs provide more partial credit when grading with contributions summaries ($p = .018$)
  - Since we require TAs to provide feedback with partial credit, but not full credit, this shows they can improve the quantity of feedback provided

Discussion & Future Work

- Despite a small sample size & relatively primitive summaries algorithm, a lab study showed value of my contributions summary algorithm
- I am running a follow-on classroom study
  - Same experimental & control groups
  - Do students find feedback more actionable? Do they improve more over the semester?
  - Do we get the same consistency benefits with an entire semester of assignments?

Discussion & Future Work

- There is, of course, a lot more that goes into SE work than just Java code
- How can we efficiently handle other types of (code) contributions? Can language-agnostic AST analysis help with scalability?
- Can we account for all of the other (non-code) contributions to a SE project?
- These are some of the questions I'm hoping to ponder in detail this summer

Summary

- I designed an algorithm to summarise individual students' contributions to team projects, and built it into a tool, AutoVCS
- Through a quantitative lab study, I demonstrated that TAs who use these summaries grade **more consistently**, provide feedback that is possibly **more actionable**, and they **prefer** the grading process