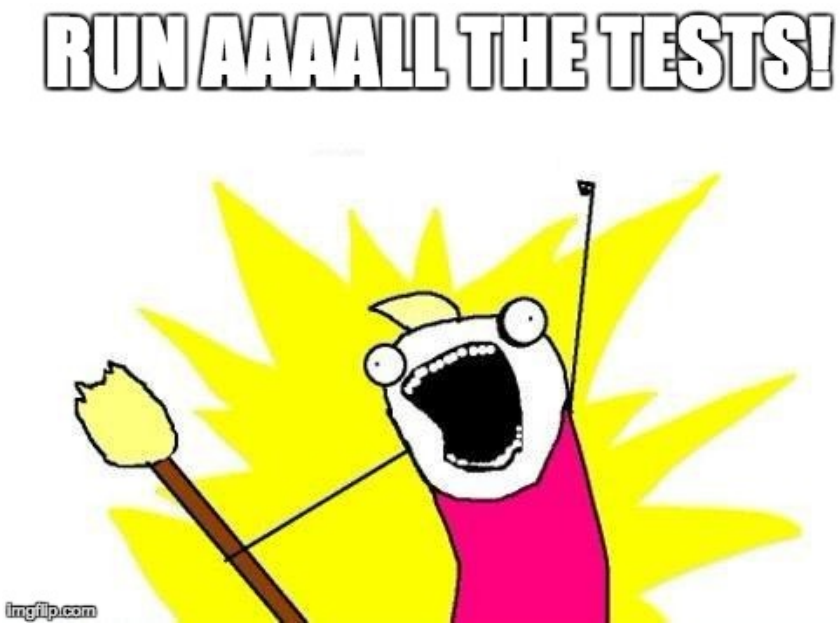
Teaching Software Testing with Automated Feedback

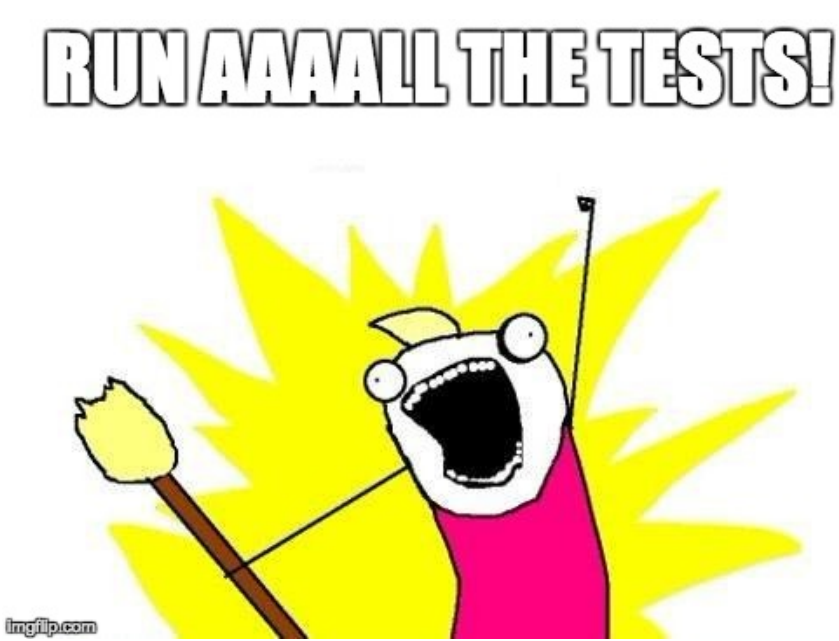
James Perretta and Andrew DeOrio, University of Michigan

ASEE Annual Conference and Exposition, June 2018

How important is it for your students to learn software testing?

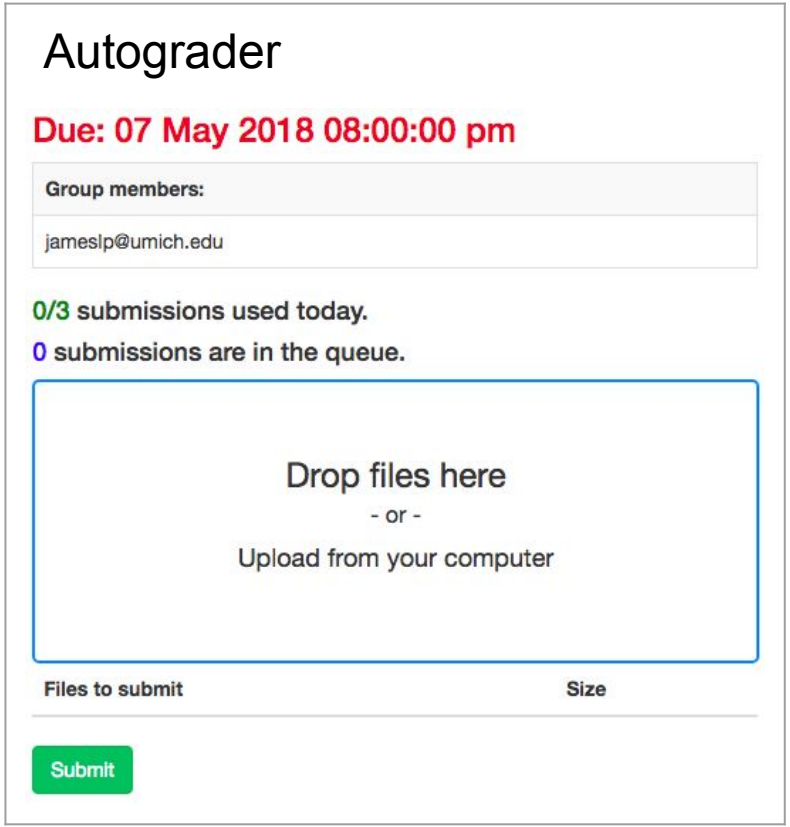


How do your students feel about it?



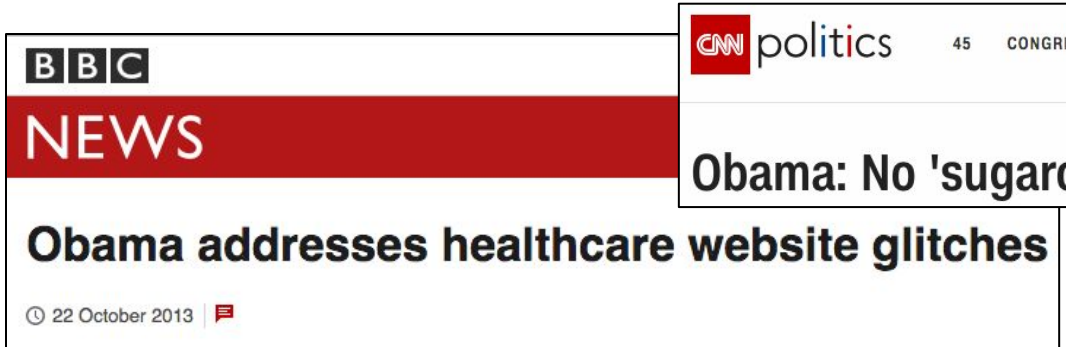
Motivation

- Software testing is important!
  - But little time spent teaching it. (Edwards 2003)
- Testing takes practice.
- Automated grading becoming more common in CS courses.



Software Testing!

- 41% of IT budgets spent on QA and testing. (Hannigan & Walker 2015)
- HealthCare.gov
  - Launched Oct. 1, 2013, standard Web 2.0 app
  - Many users couldn't register, combination of high load and software issues
  - Some applications submitted with missing info

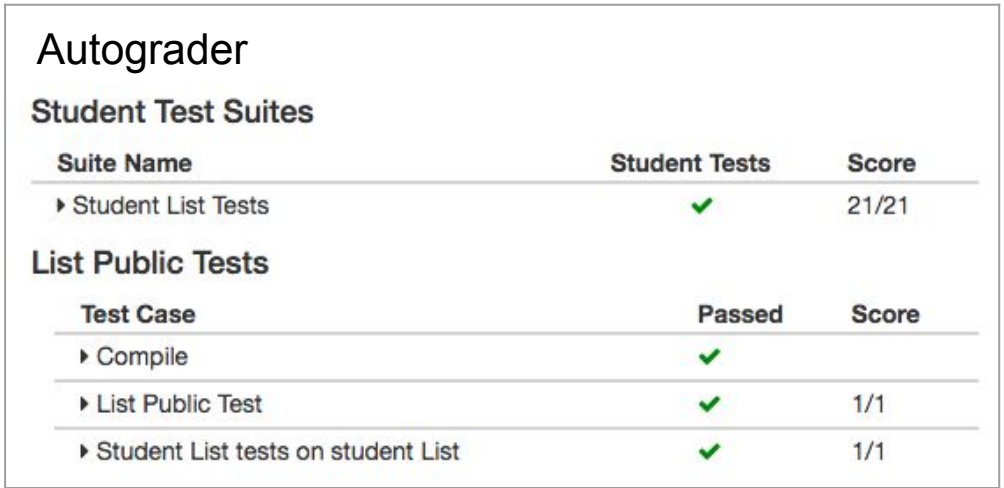


Teaching Software Testing

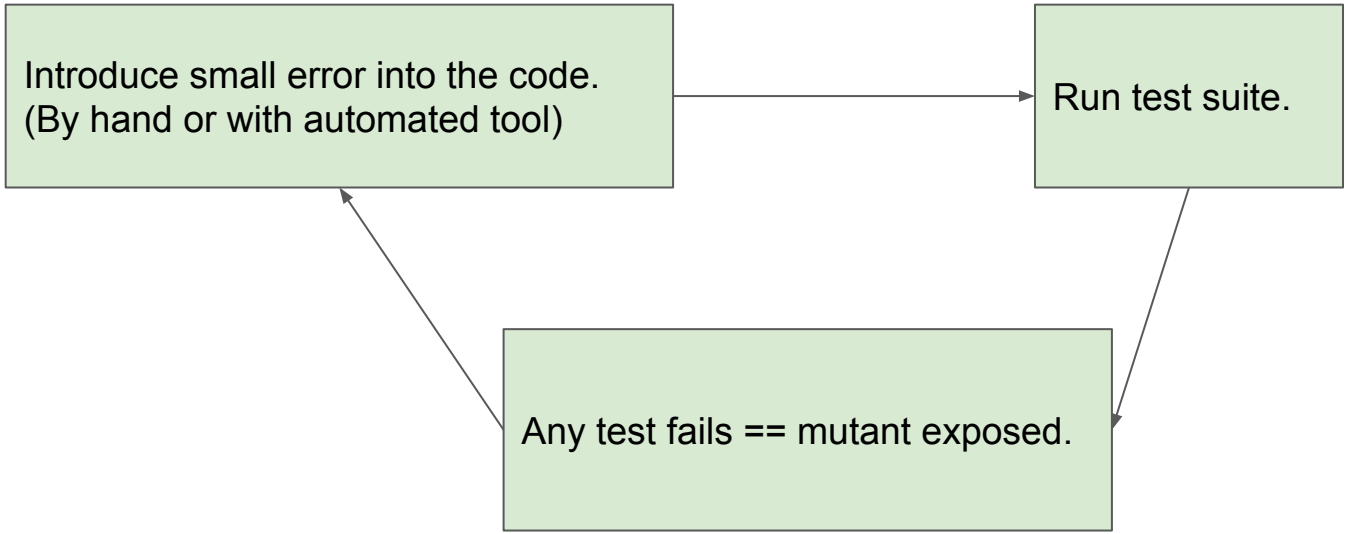
- Process-driven approaches:
  - Test-driven development (Desai et al 2008)
    - Test early, test often
  - SPRAE: Specification, Premeditation, Repeatability, Accountability, Efficiency (Jones & Chatman 2001)
    - Systematic approach to writing tests

Automatically Grading Student Tests

- Gives students immediate feedback on their tests.
- Test quality metrics:
  - Coverage: “What percentage of source code is exercised?”
  - Whether a test suite is free of **false positives**
  - Mutation Testing: “How good are tests at catching real bugs?” (**true positives**)



Mutation Testing



- Mutant: One copy of code with bug added.
- A high-quality test suite should expose more mutants than a low-quality test suite. (Jia & Harman 2010)

Research Questions

- Does automated feedback improve students' ability to write high-quality test cases?
- What type of feedback best encourages student learning of software testing?

Goal: Conduct an experiment to measure the effectiveness of automated feedback policies.

Methods: Course Overview

- Population: 1,556 students over two semesters of a second-semester programming course.
- 3 hrs lecture and 2 hrs lab per week.
- Lecture and lab sections synchronized, students could attend any section and learn same material.
- Both semesters in our study synchronized for content and organization.

Methods: Programming Projects

- 5 programming projects total (we used 3 in our study):
  - Implement one or more abstract data types (ADTs).
  - Writing unit tests for the ADTs.
  - A command-line program using the ADTs.
  - Students could work alone or with a partner

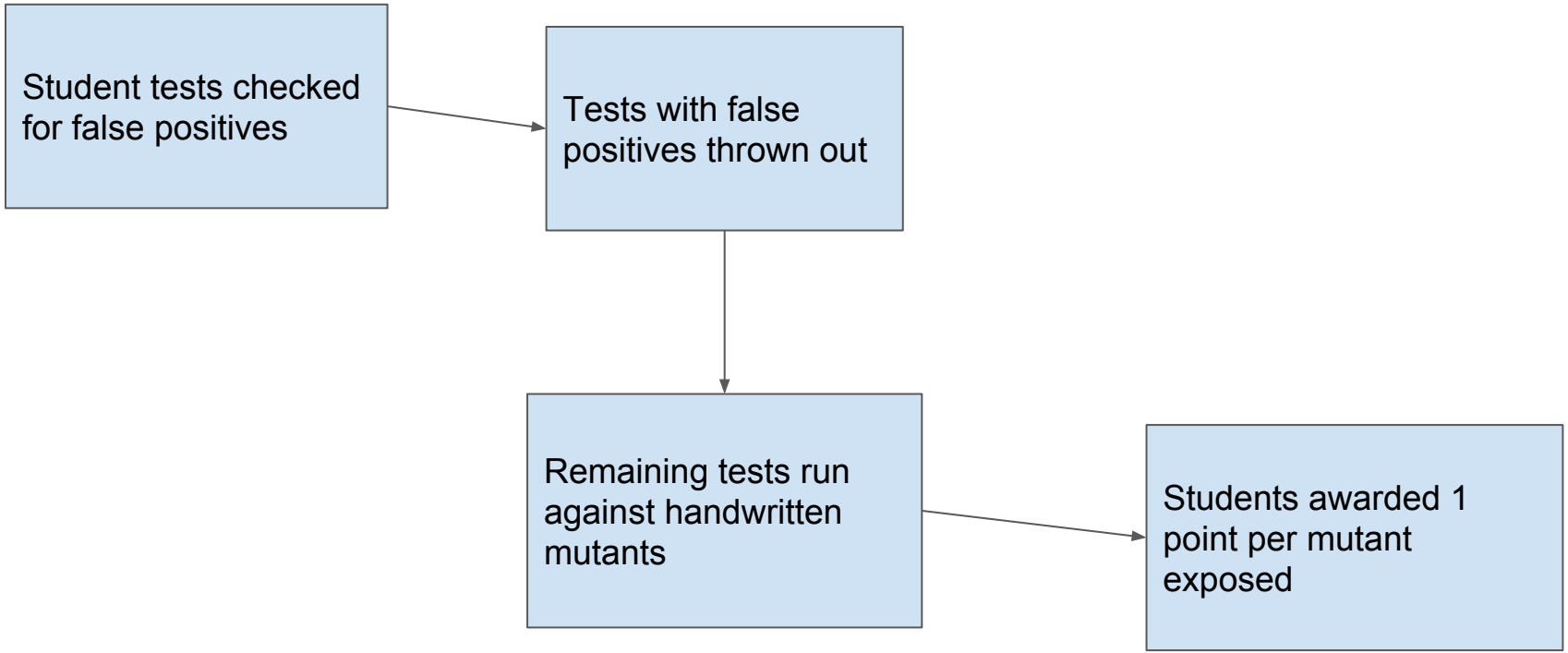
	Project 1	Project 2	Project 3	Project 4	Project 5
Instructor LOC	140	301	595	372	495

Methods: Programming Projects

- 5 programming projects total (we used 3 in our study):
  - Implement one or more abstract data types (ADTs).
  - Writing unit tests for the ADTs.
  - A command-line program using the ADTs.
  - Students could work alone or with a partner

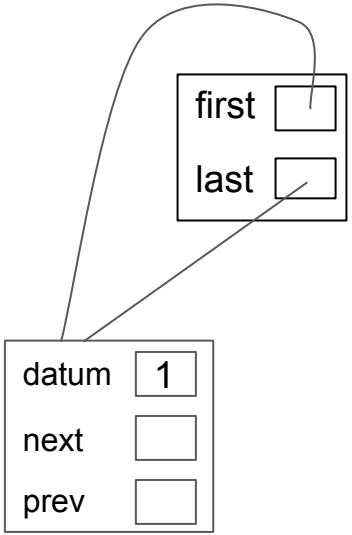
	Project 1	Project 2	Project 3	Project 4	Project 5
Instructor LOC	140	301	595	372	495
Average Student LOC	165	388	857	378	533

Methods: Student Test Evaluation



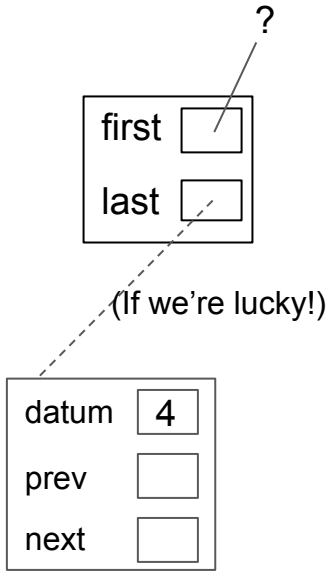
Example: Instructor-written Mutant

```
// CORRECT implementation.
template <typename T>
void List<T>::push_back(const T &datum) {
  Node *np = new Node;
  if (empty()) {
    np->prev = 0;
    first = np;
  } else {
    np->prev = last;
    last->next = np;
  }
  np->next = 0;
  np->datum = datum;
  last = np;
  ++num_nodes;
}
```




```
// BUGGY implementation: Fails if list is empty.
template <typename T>
void List<T>::push_back(const T &datum) {
  Node *np = new Node;

  np->prev = last;
  last->next = np;
  np->next = 0;
  np->datum = datum;
  last = np;
  ++num_nodes;
}
```



Methods: Control Group

- Students enrolled in first semester.
- Same feedback on all three projects

Autograder

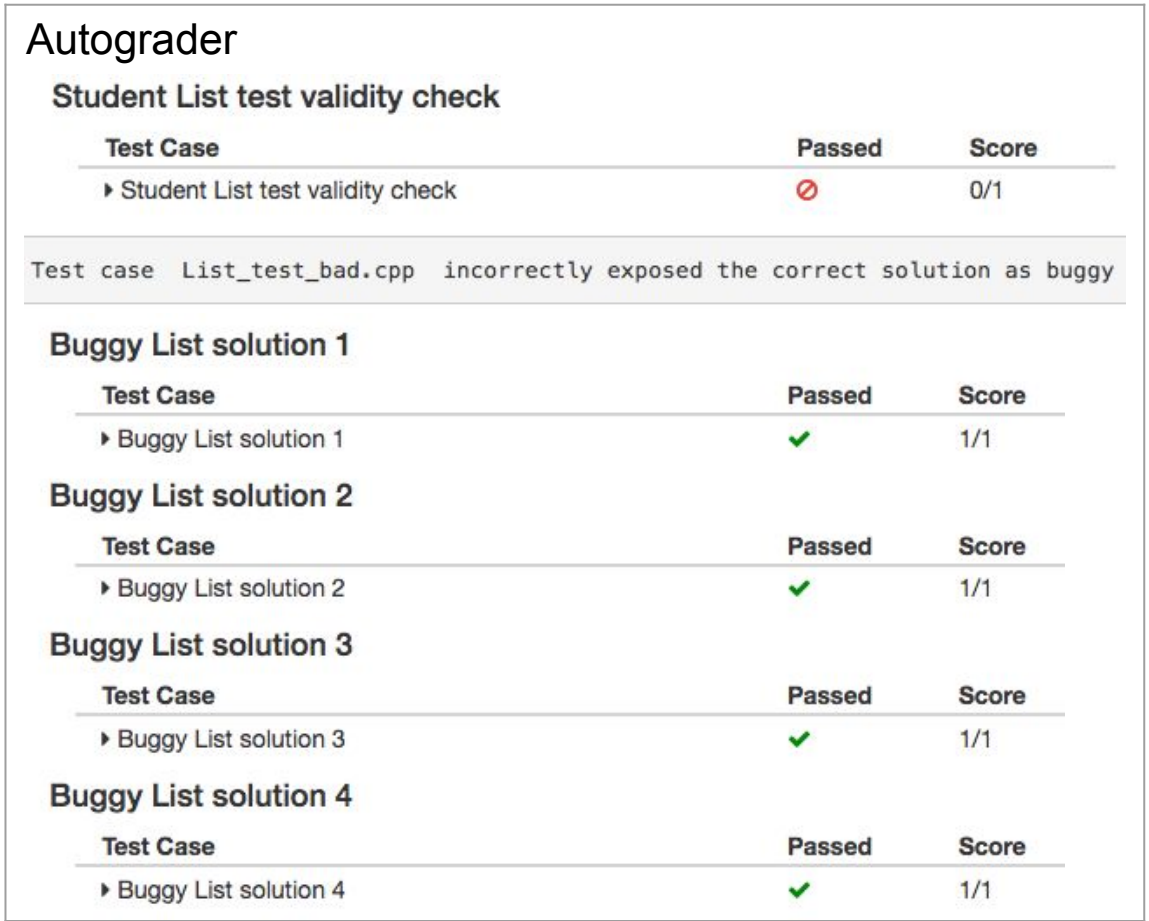
Student List test validity check

Test Case	Passed	Score
▸ Student List test validity check	⊘	0/1

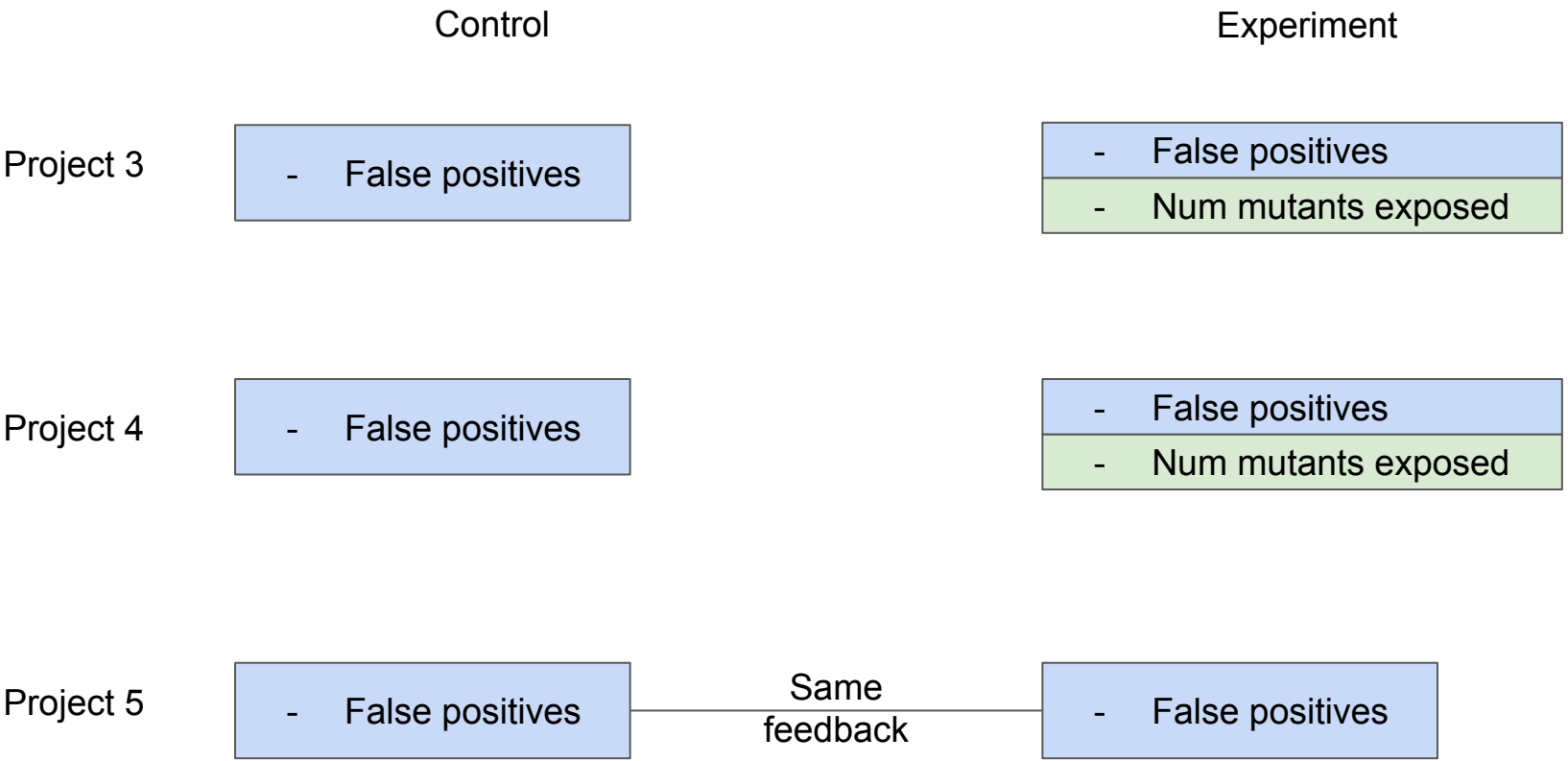
```
Test case  List_test_bad.cpp  incorrectly exposed the correct solution as buggy
```

Methods: Experiment Group

- Students enrolled in second semester.
- Additional feedback on first 2 projects.



Methods: Control & Experiment Groups



Methods: Variables

- Independent variables:
  - Test case feedback type (control and experiment groups)
  - Partnership status
  - GPA (control for this variable)
- Dependent variables:
  - Student test case quality (percentage of mutants exposed)

We used ANOVA to look for significant associations.

Results: Significance

	Project 3				Project 4				Project 5			
	df	Sum Sq.	F	**PR(>F)**	df	Sum Sq.	F	**PR(>F)**	df	Sum Sq.	F	**PR(>F)**
Feedback	1	2.2	40.95	**2.34e-10**	1	3.43	114.92	**1.64e-25**	1	0.46	12.04	**5.44e-04**
Partner	1	3.03	56.32	1.31e-13	1	1.59	53.38	5.45e-13	1	1.24	32.29	1.75e-08
Feedback x Partner	1	0.01	0.11	7.39e-01	1	0.27	8.97	2.81e-03	1	0.14	3.6	5.82e-02
GPA	1	25.91	481.46	3.19e-88	1	11.76	394.25	1.08e-74	1	9.66	251.18	1.36e-50
GPA x Feedback	1	0.02	0.34	5.60e-01	1	0.0	0.12	7.26e-01	1	0.04	1.02	3.14e-01
GPA x Partner	1	0.0	0.0	9.63e-01	1	0.15	4.9	2.71e-02	1	0.0	0.02	8.88e-01
GPA x Feedback x Partner	1	0.0	0.07	7.87e-01	1	0.07	2.4	1.21e-01	1	0.06	1.56	2.11e-01
Residual	1056	56.83			1045	31.17			991	38.12		

Significant association b/w feedback type and test quality on all 3 projects.

Results: Significance

	Project 3				Project 4				Project 5			
	df	Sum Sq.	F	**PR(>F)**	df	Sum Sq.	F	**PR(>F)**	df	Sum Sq.	F	**PR(>F)**
Feedback	1	2.2	40.95	**2.34e-10**	1	3.43	114.92	**1.64e-25**	1	0.46	12.04	**5.44e-04**
Partner	1	3.03	56.32	**1.31e-13**	1	1.59	53.38	**5.45e-13**	1	1.24	32.29	**1.75e-08**
Feedback x Partner	1	0.01	0.11	7.39e-01	1	0.27	8.97	2.81e-03	1	0.14	3.6	5.82e-02
GPA	1	25.91	481.46	3.19e-88	1	11.76	394.25	1.08e-74	1	9.66	251.18	1.36e-50
GPA x Feedback	1	0.02	0.34	5.60e-01	1	0.0	0.12	7.26e-01	1	0.04	1.02	3.14e-01
GPA x Partner	1	0.0	0.0	9.63e-01	1	0.15	4.9	2.71e-02	1	0.0	0.02	8.88e-01
GPA x Feedback x Partner	1	0.0	0.07	7.87e-01	1	0.07	2.4	1.21e-01	1	0.06	1.56	2.11e-01
Residual	1056	56.83			1045	31.17			991	38.12		

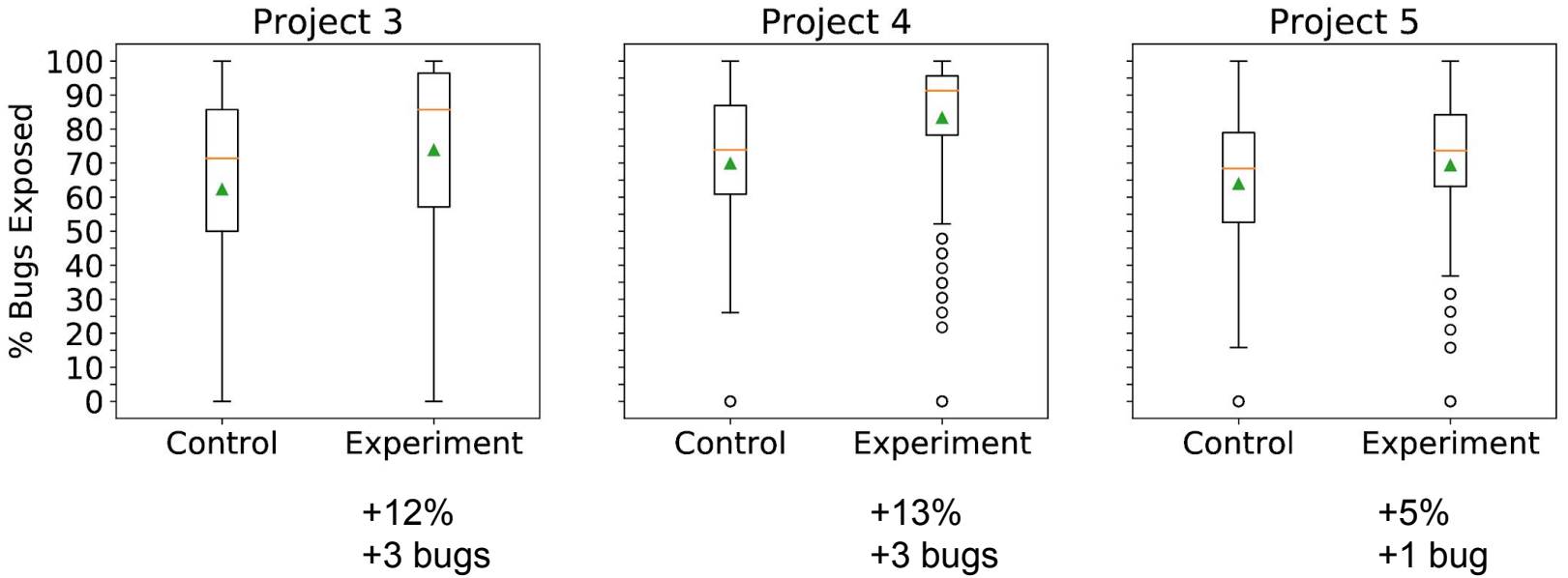
- Significant association b/w partnership status and test quality on all 3 projects.
- Magnitude of association comparable to that of feedback type.

Results: Significance

	Project 3				Project 4				Project 5			
	df	Sum Sq.	F	**PR(>F)**	df	Sum Sq.	F	**PR(>F)**	df	Sum Sq.	F	**PR(>F)**
Feedback	1	2.2	40.95	**2.34e-10**	1	3.43	114.92	**1.64e-25**	1	0.46	12.04	**5.44e-04**
Partner	1	3.03	56.32	**1.31e-13**	1	1.59	53.38	**5.45e-13**	1	1.24	32.29	**1.75e-08**
Feedback x Partner	1	0.01	0.11	7.39e-01	1	0.27	8.97	2.81e-03	1	0.14	3.6	5.82e-02
GPA	1	25.91	481.46	**3.19e-88**	1	11.76	394.25	**1.08e-74**	1	9.66	251.18	**1.36e-50**
GPA x Feedback	1	0.02	0.34	5.60e-01	1	0.0	0.12	7.26e-01	1	0.04	1.02	3.14e-01
GPA x Partner	1	0.0	0.0	9.63e-01	1	0.15	4.9	2.71e-02	1	0.0	0.02	8.88e-01
GPA x Feedback x Partner	1	0.0	0.07	7.87e-01	1	0.07	2.4	1.21e-01	1	0.06	1.56	2.11e-01
Residual	1056	56.83			1045	31.17			991	38.12		

- Control for GPA
- Significant association b/w GPA and test quality on all 3 projects.

Results: Test Case Quality vs. Feedback Type



+12%
+3 bugs

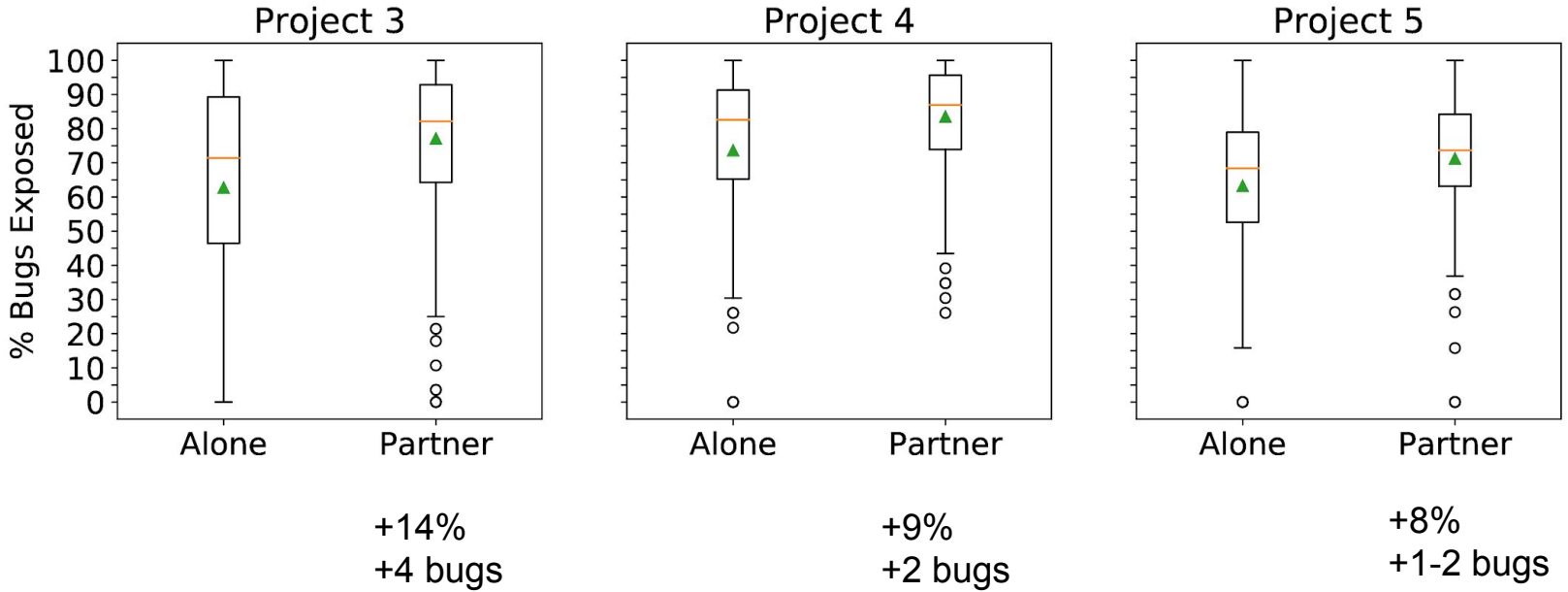
+13%
+3 bugs

+5%
+1 bug

(Additional feedback removed)

All 3 differences in mean are statistically significant.

Results: Test Case Quality vs. Partnership



All 3 differences in mean are statistically significant.

Limitations

- Projects in our experiment may have varied in difficulty.
- Control and experiment groups came from different semesters of same course.
  - Note: Both semesters were very consistent in organization and material.
- Students chose whether to work with a partner, who their partner would be.

Conclusion

- Students who received additional feedback on their test cases wrote higher-quality test cases, even after augmented feedback was taken away.
- Students who worked with a partner consistently wrote higher-quality test cases.
- Our work can help inform CS educators in their decisions on how to evaluate student tests and what automated feedback to provide.