Involving Testing Students in Software Projects

Patrick J. Schroeder
University of Wisconsin - Milwaukee

“Experimental Evaluation of Six Test Techniques” - L. Lauterbach, W. Randal

A conclusion of the study:

… choice of tester was seen to be a larger factor in the resulting effectiveness of the testing than was choice of test technique.

What Skills Does a Tester Need to Succeed?

- Communications Skills – persuade, influence, educate managers, developers, customers
- Interpersonal Skills – remain sensitive and constructive in complex, political, stressful situations
- Creative, Intuitive, Critical Thinking Skills – understanding the possibilities of the product and how it can fail

Approach: Learn by Doing

- CS536 Intro to Software Engineering (SE) – Grad/Undergrad
- CS790 Software Testing and Quality Assurance – Grad, w/CS536 Pre-req.

CS536 SE teams paired with CS790 test teams in execution of a software project

Application

Shari Pfleeger's Loan Arranger System

- Mortgage-backed security business
- Data Repository Architecture
- Multi-user system
- Complex calculations – accrued interest

SE's Assignment

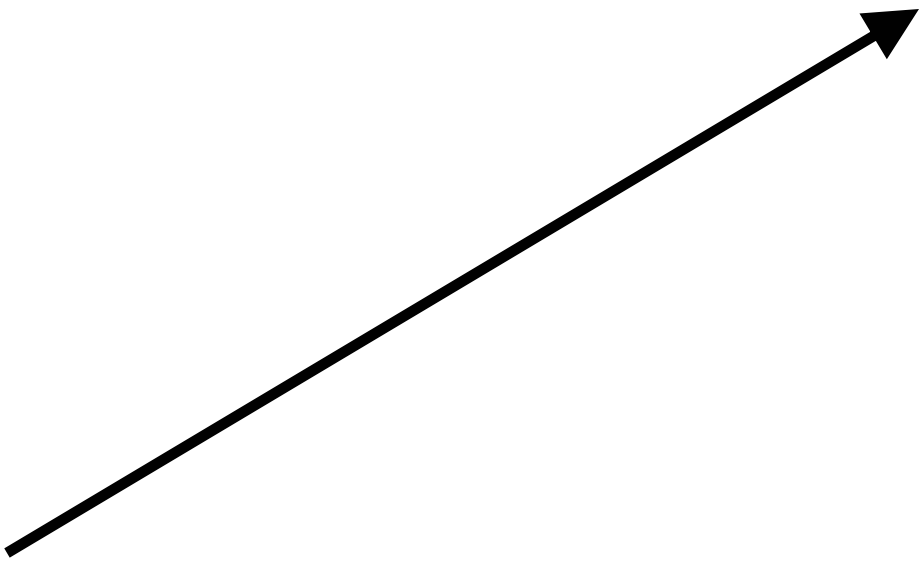
- ⇨ Complete the requirements specification (use cases, high-level class diagram)
- ⇨ Design, implement and deliver software on an incremental delivery schedule
- ⇨ Negotiate the scope of the project
- ⇨ Provide software releases, release notes, installation instructions to the testers
- ⇨ Respond in a timely fashion to bug reports

Tester's Assignment

Find a way to help the your SE team deliver a better product, by:

- NOT demoralizing them
- Finding important bugs quickly
- Writing good bug reports
- Helping them understand the product
- Understanding their software capabilities
- Generating a light-weight test plan and test logs

Tester's Journey



First Person on the Airplane

"Damaged Goods"

Fallback Plan

- Incremental deliveries made of the *TA-Prof* version of the software

- Allowed testers to execute tests even if their SE team was not delivering

Project Execution

- ☺ Much excitement generated!
- ☺ Fewer than expected flame wars in the bug tracker/email
- ☹ Many software deliveries late or missed
- 😐 Good, and really bad, software delivered
- ☹ Some requests by testers to abandon their SE teams
- ☺ Improvement of deliveries as the semester progressed

Bug Reports

- Over 250 bug reports generated
- Initial bug reports poor
- Reviews of bug reports in class caught tester's attention
- SE's and testers learned to recognize good bug reports

Final Project Evaluation

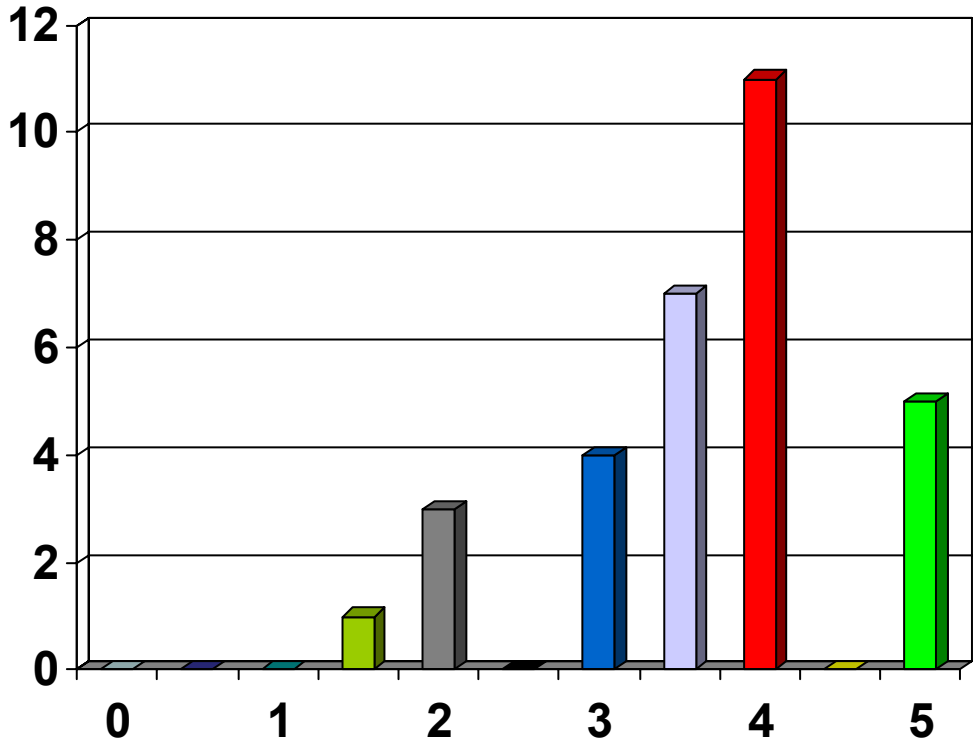
- SE teams filled out evaluations of their test teams
- Test teams filled out evaluations of their SE teams
- Each software project was tested with the help of the TA

Do you think that working on the LAS project gave you insight into the challenges of software testing?

0 1 2 3 4 5

|----------|----------|----------|----------|----------|

No Yes

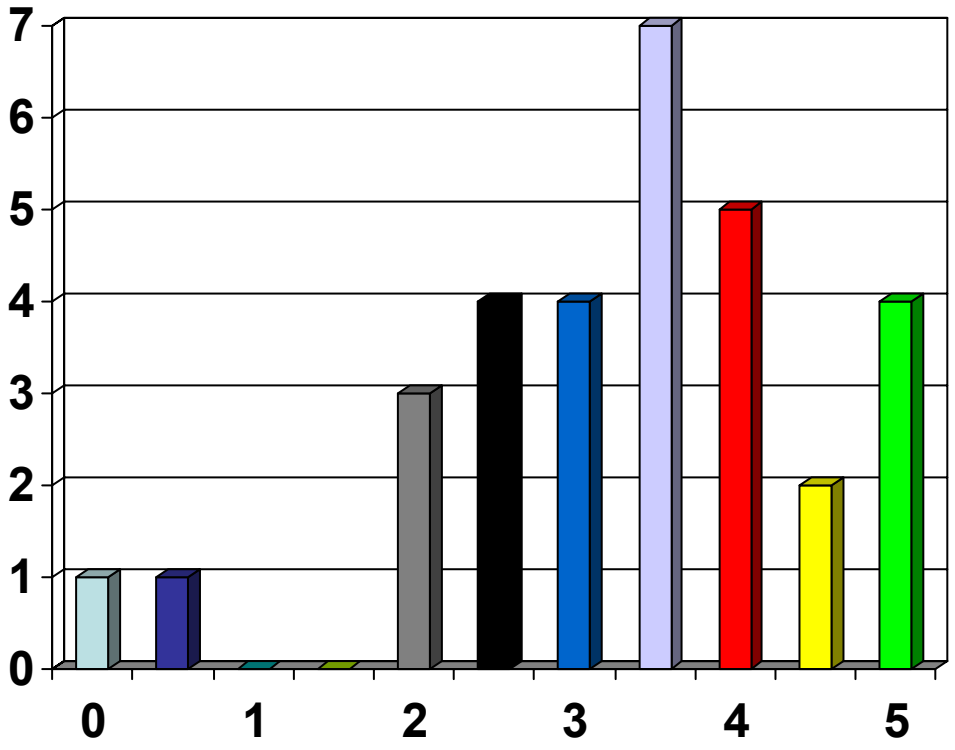


Do you think your testing team's effort contributed to the quality of the final LAS product?

0 1 2 3 4 5

|----------|----------|----------|----------|----------|

No Contribution Full Contribution

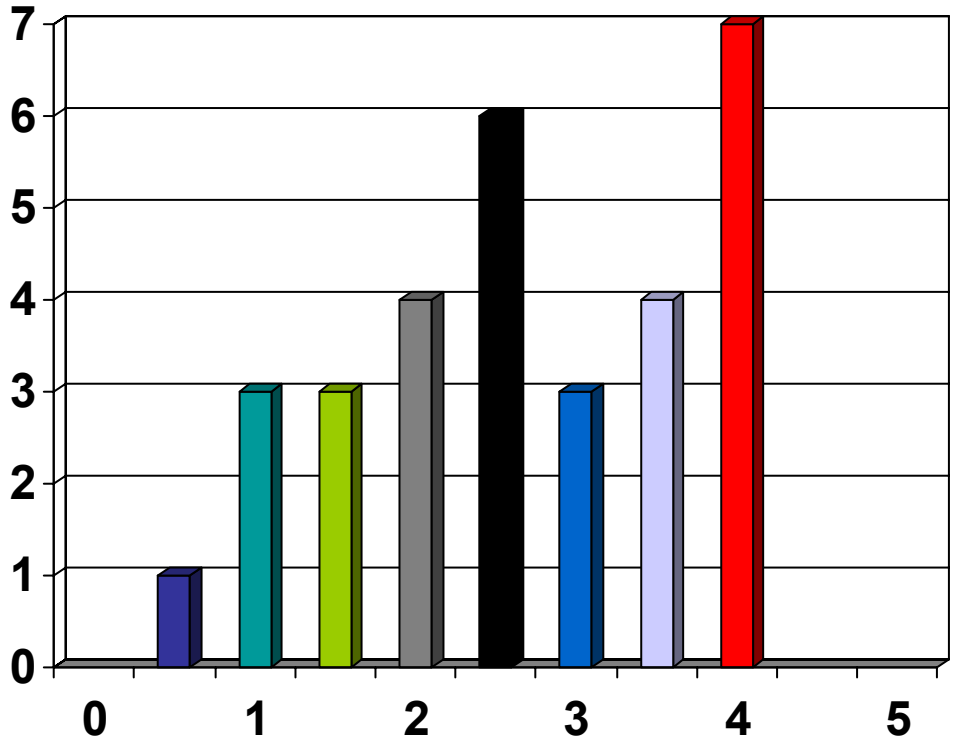


Do you think ***your*** testing team over tested or under tested the LAS software?

0 1 2 3 4 5

|----------|----------|----------|----------|----------|

Under Tested Over Tested

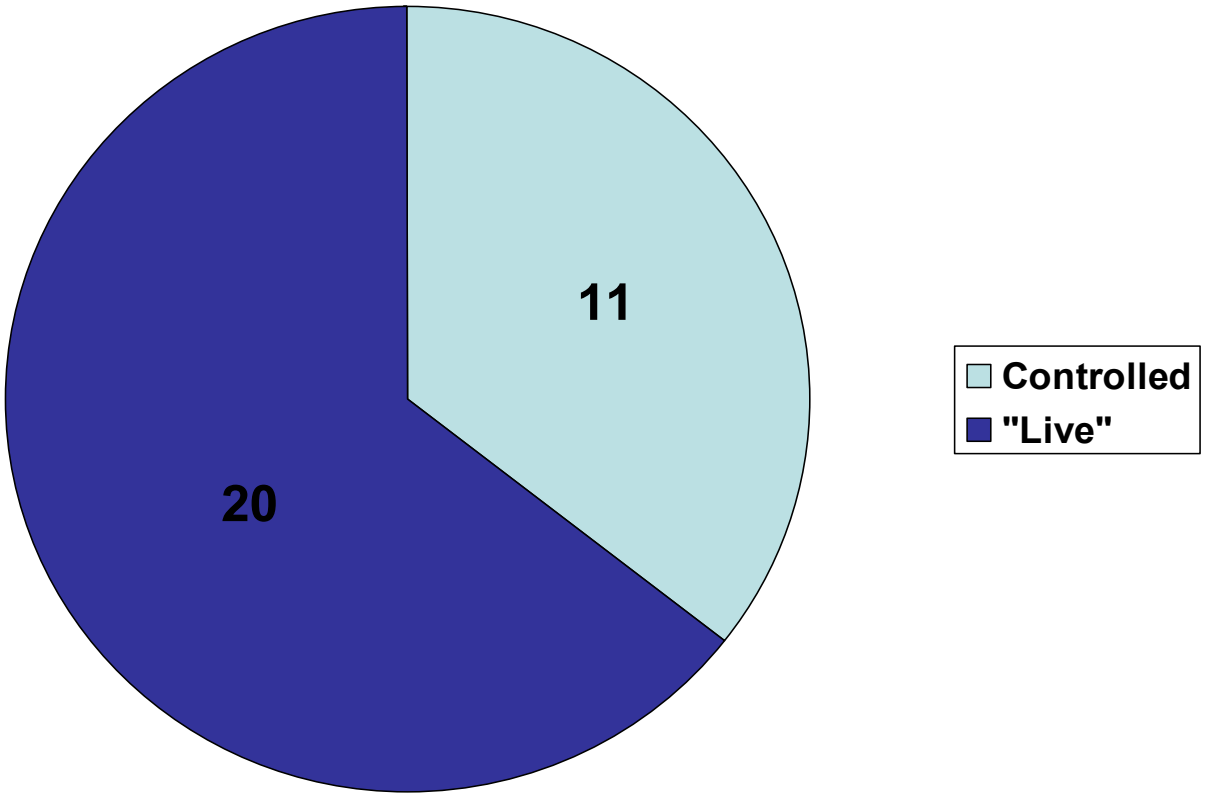


The number of tests I personally executed was (roughly): _____

Avg	36.8
Min	5
Max	200
Mode	20

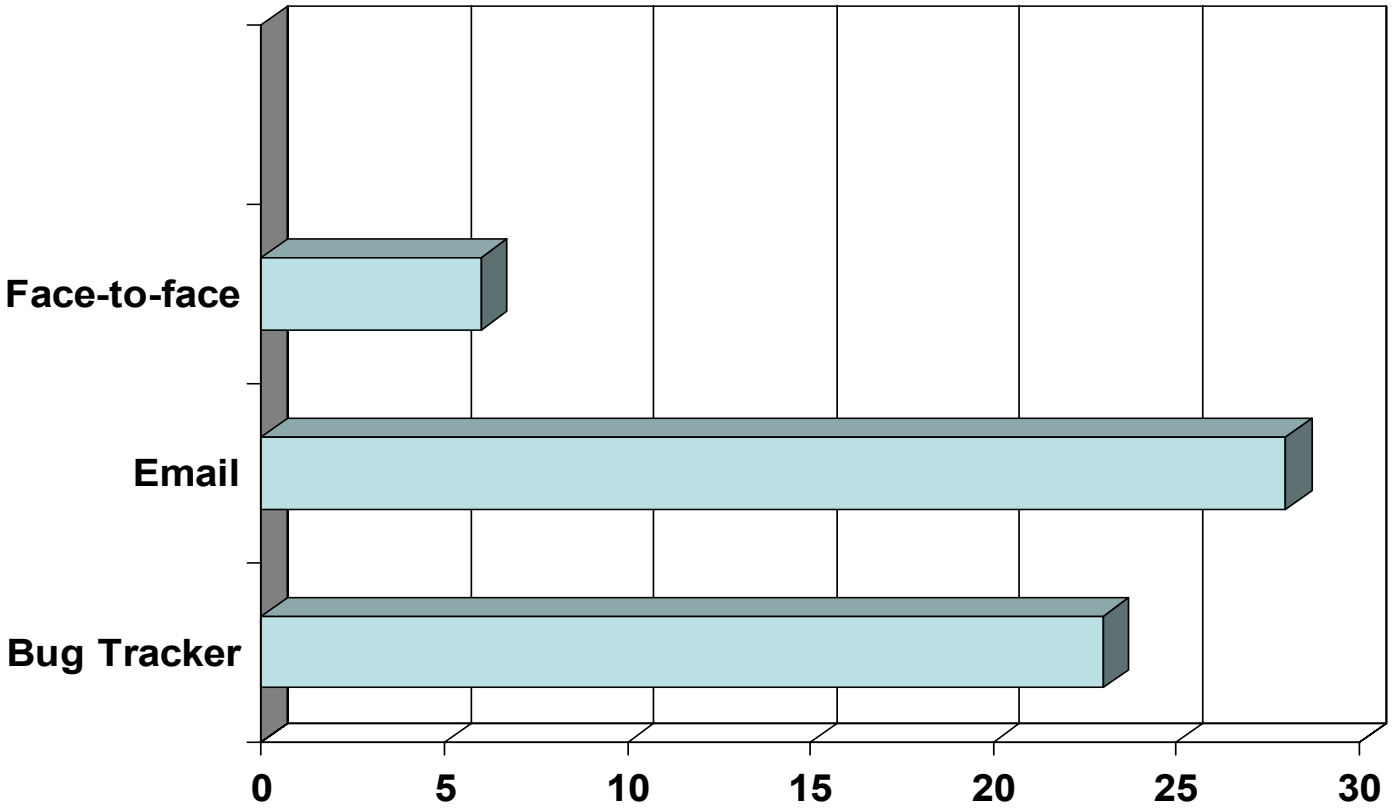
A different approach to teaching software testing is to use a smaller, controlled, previously completed program. This approach omits interaction with software developers.

- I would prefer testing an existing program in a controlled environment
- I prefer working with on a “live” project with software developers, despite the chaos



How did you communicate with the SE team? (Check all that apply)

- Bug tracker
- Email
- Face-to-face meetings (other than requirements review)



	A	B	C	D	E	F	G
1	**Semester:**	Sem. I, 02-03					
2	**Instructor:**	P. Schroeder					
3	**Course Number:**	790-002					
4	**Course Title:**	Advanced Topics in Computer Science: Software Testing and QA					
5							
6		**Average**	**1**	**2**	**3**	**4**	**5**
7	**1. Knowlege (1-5)**	4.64	5	4	5	5	5
8	**2. Understandable (1-5)**	4.64	5	5	5	5	4
9	**3. Questions (1-5)**	4.50	4	4	5	5	4
10	**4. Practice (1-5)**	4.45	5	5	5	5	4
11	**5. Assignments (1-5)**	4.12	4		5		4
12	**6. Exams (1-5)**	4.26	4	3	5		4
13	**7. Availibility (1-5)**	4.52	5	4	5		3
14	**8. Motivate (1-5)**	4.18	4	4	5	4	3
15	**9. Recommend Instructor (1-5)**	4.36	4	4	5	4	4
16	***Instructor Average***	**4.41**	4.444444	4.125	5	4.666667	3.888889
17	***Instructor Median***	4.50					
18	**10. Syllabus (1-5)**	3.88	4	3	4	3	
19	**11. Follow Syllabus (1-5)**	4.12	4	5	4	3	
20	**12. TA Performance (1-5)**	3.00			1		2
21	**13. Relevant (1-5)**	4.15	5	4	4	4	3
22	**14. Recommend Course (1-5)**	3.96	5	3	4	4	4
23	**15. Textbook (1-5)**	3.65	4	3	4	4	3
24	**16. Learning Technology (1-5)**	2.59		1	5	5	1
25	**17. Cheating (1-2)**	2.00	2	2	2	2	2
26	**18. Difficulty (1-5)**	2.82	1	4	1	3	3
27	**19. Workload (1-5)**	3.07	3	4	2	3	3
28	***Course Average(10, 11, 13-15)***	3.94	4.5	3.5	4	3.75	3.333333
29	***Course Median***	4.00					
30	***Toughness Average***	2.95					
31	***Toughness Median***	3					

Conclusion

- Great project experience for the SE teams
- SE students short-changed on testing
- Good testing experience for the test teams, in the context of a chaotic project
- Good exposure to a wide variety of challenges that testers face
- Problem: not enough time
- Problem: evaluating projects costly
- Problem: application too complex

Thank you!

Pat Schroeder

University of Wisconsin Milwaukee

pats@uwm.edu