How I spend my ~~summer~~ autumn vacation teaching computer science at the University of Washington

Retrospective on teaching UW CSE 403

Autumn 2012

Wayne K. Yamamoto

Summary of the quarter in 5 bullets

- Taught CSE 403 -- software engineering
- Students learned basic and advanced principles of *practical* software engineering
- Forty students successfully executed six software projects
- I learned a lot about teaching in great computer science department and enjoyed the experience
- Didn't get to dig into much research (my personal agenda for the quarter)

How did I get here?

- Casual conversation with UW CSE Professor Dan Grossman and Karen Howard Leikin in June 2012
- I inquired: *How can I help the UW CSE department?*
- Response: *Maybe teach CSE 403 in the Fall?*
- **Sounds great!**
- Sync'd with Paul Beame; Notkin greases the skid
- Here's what I learned

What I wanted students to learn

- An understanding of the fundamentals of software engineering
- Experience building a software project using software engineering fundamentals
- Useful “stuff” to help you as a software professional

Specific software engineering takeaways I wanted students to learn

- "Divide and conquer" is your friend
- Change, ambiguity, and uncertainty are inevitable
- Can't have it all: Cost-benefit analyses are key
- Modularity and abstraction are important tools
- You might be late: What are you going to do?
- You are in a great position for a great career in software engineering!

Curriculum development

- Talked with Notkin, Ernst, and Stepp
- Read all past quarters' websites
- No (hard) directive on what I could/should and couldn't/shouldn't teach
- Considered teaching (nearly) straight up from previous quarter's slide decks...but didn't

Curriculum development

- Decided to use past as a guide, but build significantly different lecture presentations and present (slightly) different viewpoint
- Figured if I just mimicked previous quarters, it would be substandard to the job of "professional" educators and wouldn't be my viewpoint
- Wanted to bring in my unique value -- real world experience on doing software engineering
- Used real examples to highlight software engineering concepts
- Whose objectives did I want to satisfy? Students? Mine? Department? Industry?
  - Hopefully, all. Certainly mine and industry. And right stuff for students.
  - Department? Think so -- feedback would be great

Overall results

- Pleased with project outcomes -- substantive projects were done, with significant software engineering discipline used
- Exams and feedback indicate good understanding of concepts taught
- Prepped students with real world experience to help with future careers

“Can’t have it all” -- a hard but important lesson

- Cost-benefit analysis in software engineering is an important concept
- Difficult lesson for students
- Making a trade off -- or even just “do less” because doing more didn’t matter -- was difficult for students
  - They are used to being able to “do it all”
  - Complaints that they can’t do everything or are “incapable” of doing less
  - Had to carefully manage so they were making trade offs on “intra-class” time and not causing other class work to suffer

Change, uncertainty, and ambiguity are constants in software engineering

- Students hated dealing with these concepts
- Specificity, clarity, completeness, and consistency were expected by many students
- Explicitly and implicitly created situations of change, incompleteness, and ambiguity
- This was difficult for a few students to grok

Group projects

- As usual, a significant part of the coursework and effort
- Was warned by previous instructors that this was a potential cesspool of frustration, hate, and effort --> not the case this quarter
- Took a "professional" approach to managing project, with TAs providing ongoing guidance and evaluation -- management oversight went a long way
- Projects were substantive with good software engineering discipline, using creative problem solving techniques
    - Example: One project had a member that wasn't fully engaged. Project manager creatively decided a solution would be to start pair programming. It worked. (I didn't suggest it -- he figured it out himself)
- Significant exploration of new technologies and techniques

Department IT resources were not very helpful for executing projects

- Spent a lot of time figuring out what department resources were available
- Project groups ended up not using or deploying on department resources for anything
- Students weren't so interested in Microsoft products
- In general, department resources were not well suited for CSE 403
- I have comments for support@cs in general. Who wants to hear them?

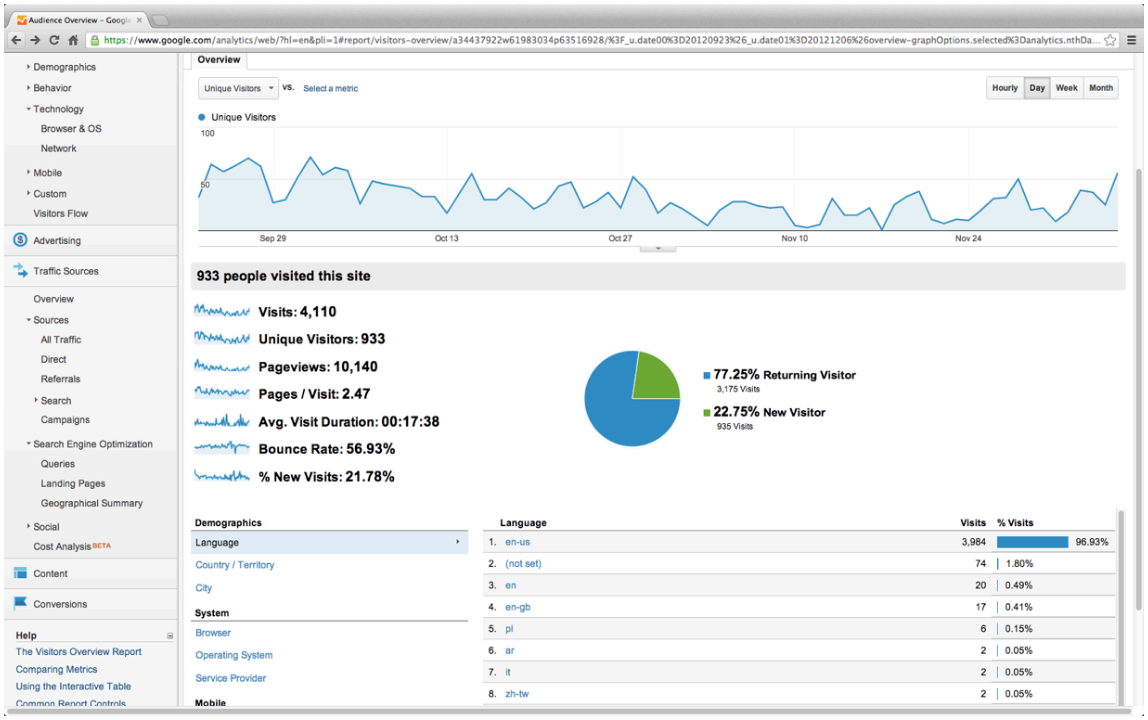
Communication

- Used www.cs.washington.edu/cse403 as a home page but hosted much of the course meta content on Google sites (it was much easier doing this)
- Used Twitter as an experiment
  - Fast way to push short content and links to students
  - Real time feedback in class? (Didn't happen)
  - Ultimately Twitter was good for pushing supplemental content but email was used for essential/urgent content
- Used Google docs for surveys not Catalyst
- Office hours were only sparsely used -- why didn't students to come by more often?

Monitoring

- Required all projects to use github (and therefore git) as a repository. Consequently bug tracking and wikis were also hosted here. Wanted students to use this "state of the art" tool but also aided our ability to track projects in a uniform way
- Allowed us to share experiences across projects and give instructor and TAs insight into the projects
- Created email aliases for all the projects. When used, instructor and TAs could see more details into the projects
- Google analytics provided some insight into how course content was consumed

How are students using course content?



933 people visited this site

Visits: 4,110

Unique Visitors: 933

Pageviews: 10,140

Pages / Visit: 2.47

Avg. Visit Duration: 00:17:38

Bounce Rate: 56.93%

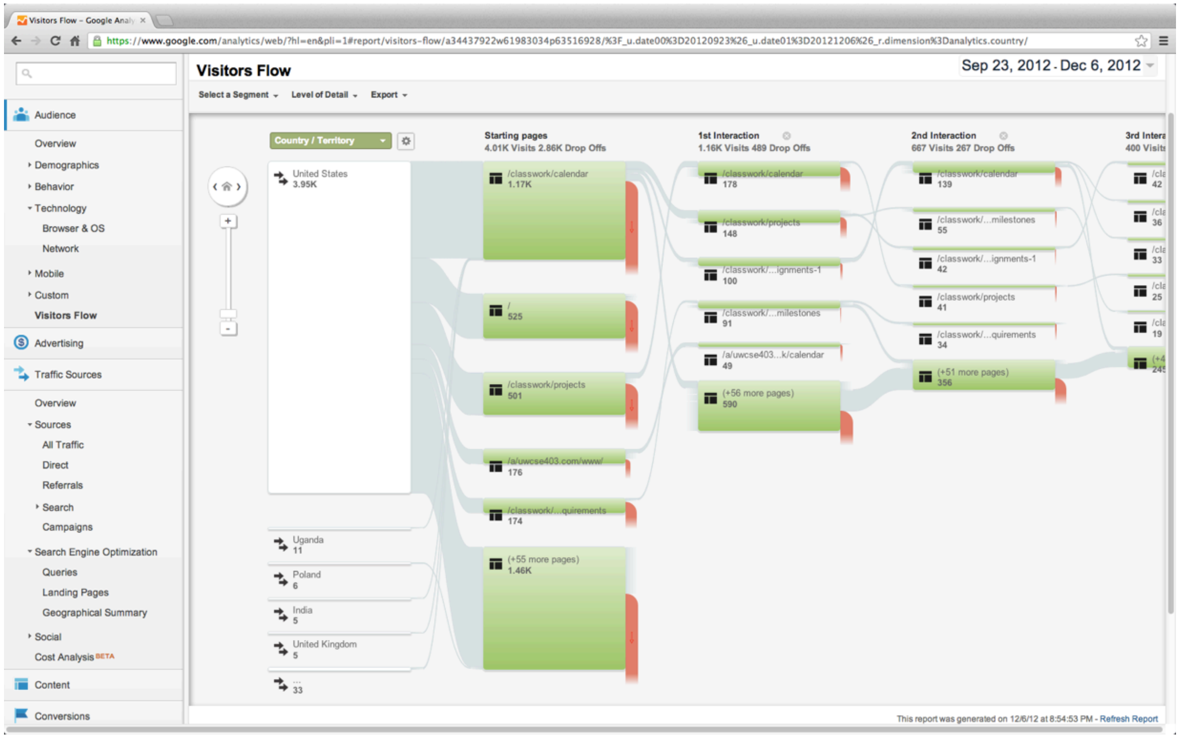
% New Visits: 21.78%

77.25% Returning Visitor
3,175 Visits

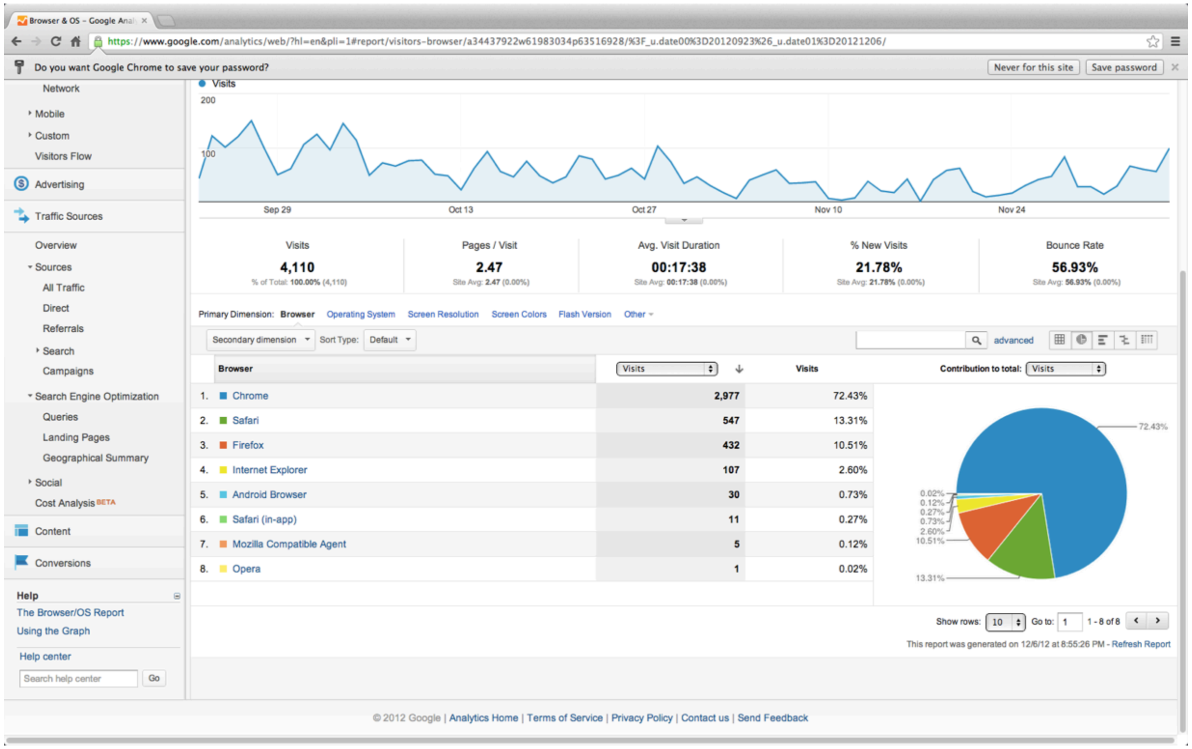
22.75% New Visitor
935 Visits

	Language	Visits	% Visits
1.	en-us	3,984	96.93%
2.	(not set)	74	1.80%
3.	en	20	0.49%
4.	en-gb	17	0.41%
5.	pl	6	0.15%
6.	ar	2	0.05%
7.	it	2	0.05%
8.	zh-tw	2	0.05%

What content are they viewing?



What technologies do they use?



https://www.google.com/analytics/web/?hl=en&pli=1#report/visitors-browser/a34437922w61983034p63516928/%3F_u.date00%3D20120923%26_u.date01%3D20121206/

Do you want Google Chrome to save your password?

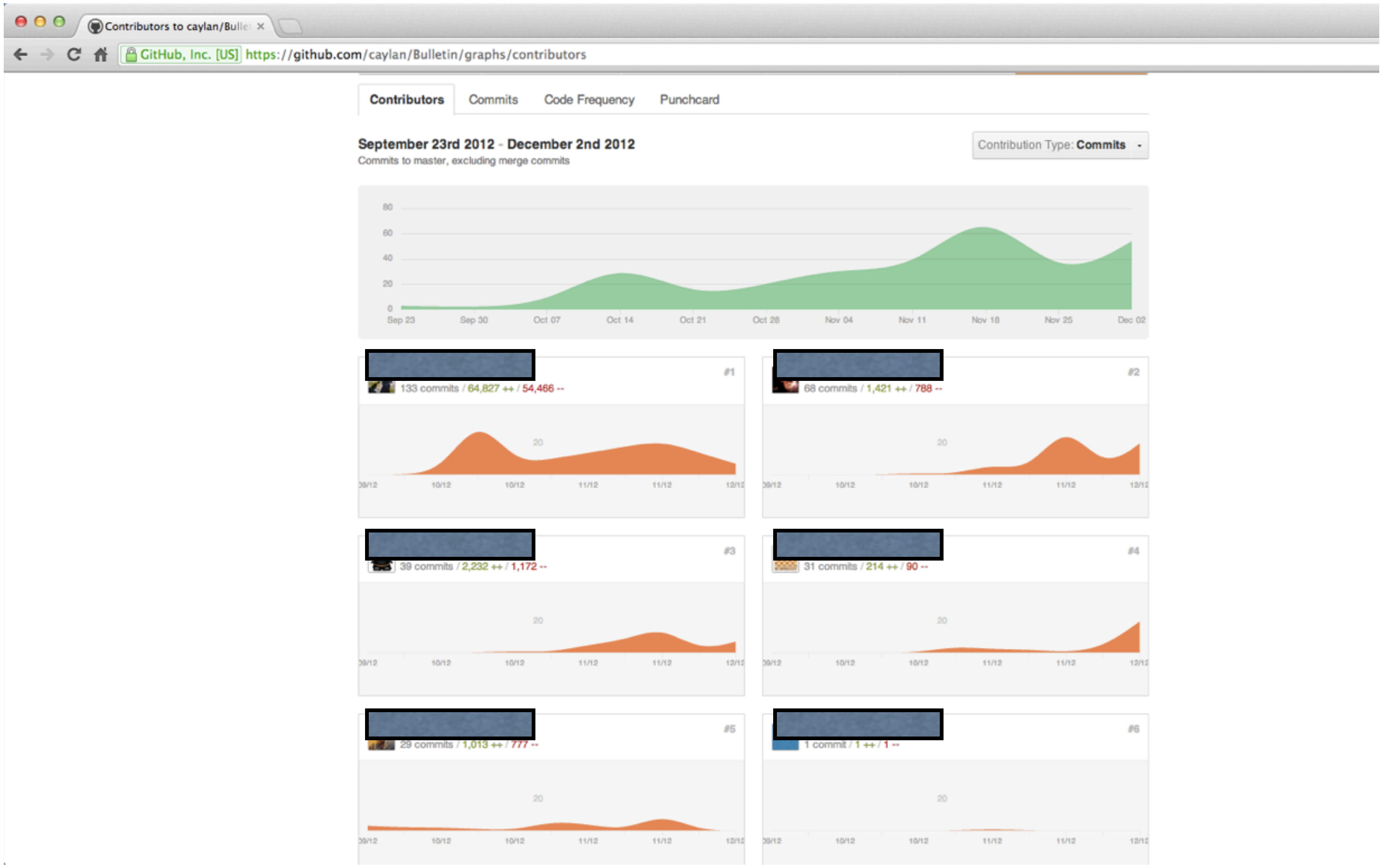
Visits	Pages / Visit	Avg. Visit Duration	% New Visits	Bounce Rate
4,110	2.47	00:17:38	21.78%	56.93%
% of Total: 100.00% (4,110)	Site Avg: 2.47 (0.00%)	Site Avg: 00:17:38 (0.00%)	Site Avg: 21.78% (0.00%)	Site Avg: 56.93% (0.00%)

Primary Dimension: **Browser** Operating System Screen Resolution Screen Colors Flash Version Other

	Browser	Visits	Visits
1.	🟦 Chrome	2,977	72.43%
2.	🟩 Safari	547	13.31%
3.	🟧 Firefox	432	10.51%
4.	🟨 Internet Explorer	107	2.60%
5.	🟦 Android Browser	30	0.73%
6.	🟩 Safari (in-app)	11	0.27%
7.	🟧 Mozilla Compatible Agent	5	0.12%
8.	🟨 Opera	1	0.02%

This report was generated on 12/6/12 at 8:55:26 PM - Refresh Report

Github gives insight on who is doing work



Contributors to caylan/Bulle ×
GitHub, Inc. [US] https://github.com/caylan/Bulletin/graphs/contributors

Contributors | Commits | Code Frequency | Punchcard

September 23rd 2012 - December 2nd 2012
Commits to master, excluding merge commits

Contribution Type: **Commits**

[redacted] #1
133 commits / 64,827 ++ / 54,466 --

[redacted] #2
68 commits / 1,421 ++ / 788 --

[redacted] #3
39 commits / 2,232 ++ / 1,172 --

[redacted] #4
31 commits / 214 ++ / 90 --

[redacted] #5
29 commits / 1,013 ++ / 777 --

[redacted] #6
1 commit / 1 ++ / 1 --

Thoughts

- Pushing the change/ambiguity/uncertainty and cost-benefit analysis memes were successful but caused student "uncomfortableness"/unhappiness
- Magda Balazinska graciously allowed me to sit in on CSE344 -- I learned many subtle teaching lessons from her
- I didn't get too much input/feedback from the department during the quarter -- I was fine with that, but a little concerned that my objectives aligned with the department
- Related: I ran an evening seminar on startups before the Affiliates Meeting. It was good for students and industry. Directionally positive for the department?
- Does the class emphasize the project too much? Maybe there's a better way for "hands on experience" with software engineering. (ask, if you want details)
- Real world, tangible experiences were well received
- As noted by other profs, disconnect between lecture and projects is a potential issue. Real world examples helped.

More thoughts

- Taught less theory and more practice than in previous quarters. Good or bad?
- Spent less time on programming -- design patterns, coding practices, etc. -- than previous courses (Remedial 331?)
- Attempted to couple quiz section assignments with "takeaways" (Software engineering principles in practice, Change/ambiguity/uncertainty. cost benefit analyses). Moderately successful.
- Four group presentations helped focus projects and gave students good experience in oral communication (maybe one presentation too many)
- Had a midterm and final: I was planning on just the final initially. Inserted midterm as an additional learning tool as well as feedback/grading mechanism.
- Student happiness: How important is this? What is the driver for it? How do we deal with this?
- Is there a goal to continuously build and improve a standard (uniform) curriculum and lecture set for the course? We didn't help directly this process if that is a goal. (But maybe indirectly we did.)

Conclusions

- Students learned the desired objectives of what I wanted to teach about software engineering, meeting my goals of helping students and helping industry
- Hopefully, this aligned with department goals
- Glad to be able to help -- hope there are more opportunities
- My experience -- not so different than any first time lecturer?
- How can I reuse/repurpose my work? Teach class again? Here, another UW campus, somewhere else?
- What future engagements that are mutually beneficial for the CSE department and me?