122nd ASEE Annual Conference & Exposition
June 14 - 17, 2015
Seattle, WA
Making Value for Society
Paper ID #12809

Quantitative Correlation between Student Use of Office Hours and Course Performance

Dr. Cinda Heeren, University of Illinois, Urbana-Champaign

Dr. Cinda Heeren is an award-winning Senior Lecturer at the University of Illinois, Urbana-Champaign. She teaches CS225, Data Structures and Programming Principles, to hundreds of enthusiastic and talented undergraduates every year. She is always game to try new pedagogical innovations, and she loves telling young women about her affection for computing.

Dr. Wade Fagen, University of Illinois

Dr. Wade Fagen is a Lecturer in the Department of Computer Science in the College of Engineering at The University of Illinois at Urbana-Champaign (UIUC). He teaches one of UIUC's largest courses, Introduction to Computer Science, known as CS 105. His research aims to improve learning by using technologies that students already bring to the classroom.

Quantitative Correlation between Student Use of Office Hours and Course Performance

0. Abstract:

University courses with a significant computing component typically provide support for student learning in the form of open lab hours attended by instructional staff. Students visit the open lab to work on computer-based assignments, and staff address questions as they arise, thereby providing just-in-time instruction and removing barriers to student progress. We have developed an online queuing system that we use to schedule student assistance in many of our core computing courses. While electronic queuing systems have been used in computing labs for decades, our web tool is instrumented to record a complete historical log of interaction times between students and staff. The analysis presented in this paper is our first attempt to understand who uses the open labs, and when, and what benefit they receive by doing so.

1. Introduction:

Dramatically increasing enrollments in our courses mean we are more dependent than ever on data collection and analysis to make our instruction effective and efficient. The time spent by course staff assisting individual students during office hours is a substantial cost that merits scrutiny, and that analysis is the substance of this report.

In our core programming courses, office hours are held in an open lab environment where students from many courses settle into one of several large rooms in a common building, each filled with lab computers. To provide student assistance on regular assignments during these office hours, we implemented a web-based queueing system. In addition to the queueing features, the queue was instrumented with data collection tools. Every interaction with course staff is logged, creating a near complete picture of how students reach out to course staff in our core courses. This paper presents the data from the queue during the Fall 2013, Spring 2014, and Fall 2014 semesters of three programming-focused courses. In addition to the queue data, students' average grades on programming assignments and average grades on exams are paired with their queue usage and evaluated for improvement.

In total, nearly 14,000 student questions were answered by course staff during office hours across three courses and three semesters (Table 1). The analysis of student use of office hours and their grade data shows: **(1):** student use of the lab resources accelerates near due dates, **(2):** student use of staffed lab hours follows the Pareto 80-20 rule where 80% of the staff time is spent answering questions from 20% of the students, and **(3):** the 20% of students who use office hours most frequently perform significantly better on programming assignments and generally better on exams than their peers who do not use office hours.

This paper first presents an overview of the queue itself (Section 2), then presents an in-depth analysis of the results of the data collection (Section 3), and explores future work related to this data and tool in the final section (Section 4).

Course	Semester Instructor		Student Enroll-ment	Students Receiving In-Person Help		In-Person Questions Answered	Statistics on Questions Answered per Student Receiving In-Person Help			
				#	%	Total	Mean	Median	Range	σ
Data Structures	Fall ‘13	A	502	259	51.6%	2,828	10.9	5	1…77	13.6
	Sp. ‘14	A	586	292	49.8%	2,593	8.9	5	1…100	11.5
	Fall ‘14	A	592	309	52.2%	2,813	9.1	4	1…89	12.3
Computer Architecture	Fall ‘13	B	210	128	61.0%	1,259	11.3	6	1…60	11.7
	Sp. ‘14	C	220	126	57.3%	1,418	9.8	6	1…59	10.2
	Fall ‘14	C	199	135	67.8%	1,503	11.1	8	1…76	11.5
Systems Programming	Sp. ‘14	D	198	98	49.5%	874	8.9	4	1…51	11.7
	Fall ‘14	E	351	128	36.5%	700	6.1	3	1…36	7.2
			2,858	1,475	51.6%	13,998	–	–	–	–

Table 1: Overview of data collected from The Queue. The data displayed shows all questions that were asked by students and marked as answered by a member of the course staff, indicating the total number of in-person interactions students had with course staff in each course.

2. The Queue:

The Queue, our web-based system for ordering student assistance during open lab hours, is modeled after the “take-a-number” system at your local bakery. Students use a web form to place their name, lab location, and a short description of their question at the end of a list of students awaiting the attention of course staff (Figure 1). Student questions are answered face-to-face (and their queue entries are removed) by course staff in first-come-first-served order. Such mechanisms for maintaining fairness have been used in college courses for over 30 years. Our tool is unique in its instrumentation for data collection—we have designed it to record timestamps for every student interaction (among other things): join time, answer begin time, answer end time. It is this data, together with assignment timelines and student course performance, that we assemble and illuminate.

The Queue application is a Ruby-on-Rails web app developed over a semester by a student employee. Its student and staff interfaces are simple, responsive, and clean. The queue is publicly visible, so we allow students to use pseudonyms for identification, though they gain access to the system by authenticating with their school ID. Students typically enter their names into the application from the lab machines, and staff typically process the students from the queue using their mobile device as they walk around the lab center.

The Queue deployment policy varies over course and semester. For example, in some semesters the availability of course staff was engineered so as to be uniformly distributed over the hours in the week, and in others, lab staffing was deliberately increased nearer due dates. Our characterization of student behavior and outcomes is independent of these policies.

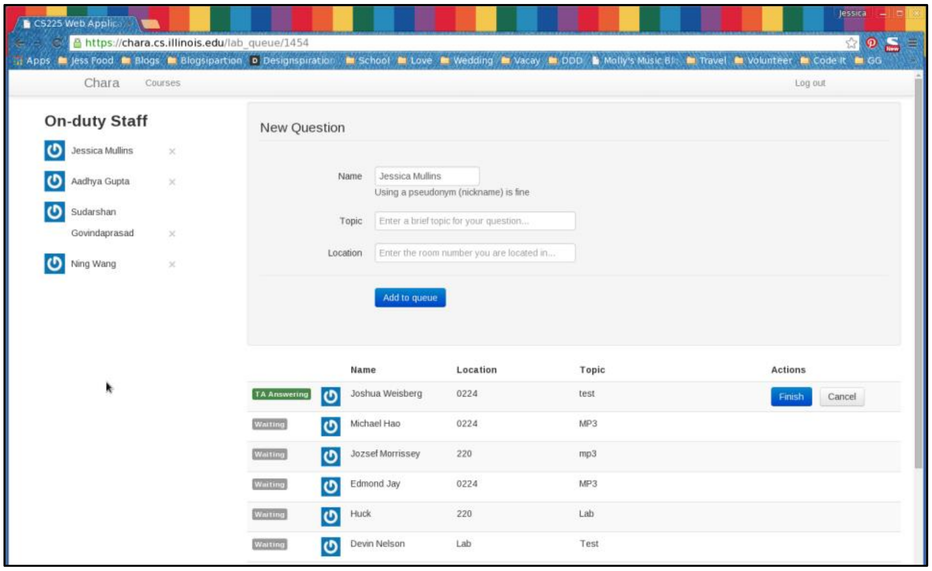


Figure 1: The course assistant interface to the queue while he/she is helping Joshua Weisberg. The student interface to the queue is similar, though different "Action" options appear in the right panel. When the staff person clicks "Finish," he/she is prompted for coarse-grained feedback on the interaction before moving to the next student in the list.

This study was conducted on assignment, queue, and grade data collected from three core, non-introductory computing courses at a large research institution, over three successive semesters, and five different instructors. In total, we observe 14,000 staff-student interactions (Table 1). This is an undercount of the actual interactions, due to inconsistent queue use during slow hours, and due to the elimination of noisy data (i.e. entries that were answered, but not ever deleted).

Though we can find no studies whose focus is an analysis of the utility of student queuing systems, Harvard's CS50 team evaluated the *quantity* of students using a similar electronic queue, and found student wait times to be untenable.[4] There are many enterprise systems available for ordering face-to-face service to customers,[1, 2] though no analysis of their application has been published.

3. Results:

Our analysis answers three fundamental questions related to queue efficacy:

- What are the work patterns of students within a two-week assignment window?
- What fraction of the students in the course use course staff assistance?
- Does personalized instruction on assignments increase student learning?

Work patterns: Unsurprisingly, student use of open lab course assistance accelerates with approaching deadlines. Figure 2 shows queue length over a typical 2-week assignment cycle in a moderately difficult programming course. The average queue length hovers around 5 students during initial lab hours, but then doubles in each of the last two days of the assignment

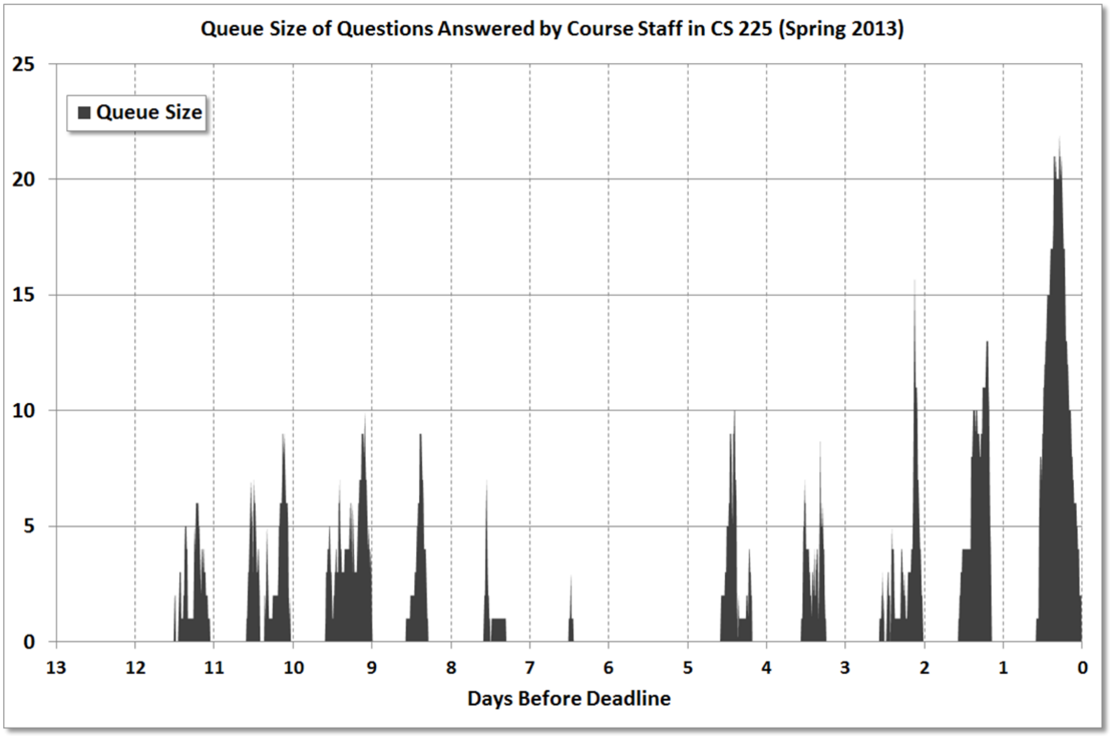


Figure 2: Queue usage within a single assignment period.

period, to ten and then twenty. The particulars of this example reflect lab availability and a course policy that awards extra credit for starting early, but the concentrated use of the open lab at then end of the window was consistent across assignments, courses, and semesters.

In response to this data, one of the courses began to assign more staff to cover the late date lab hours, but they consequently observed a feedback loop wherein students became more likely to start their assignments later, increasing the length of the queue again. The increased staffing was balanced with extra-credit incentives for starting assignments early, and warnings about long waits, but queue management is a persistent course policy concern.

What fraction benefit? Open lab hours are a resource available to all students in a class, and they are staffed copiously throughout the semester in all the course offerings we observed. Casual and anecdotal glances into our labs show productive and engaged students, all the time, but we had no empirical results demonstrating the breadth of participation among the general course populations. Our goal was to understand the cost-benefit tradeoff of the personalized instruction embodied by course staff assistance in the open labs. The data, summarized in Table 2, and visualized in Figure 3, show that roughly 20% of the students are responsible for 80% of course staff resources, following a Pareto distribution. Most notably, that rough figure is consistent within each course, across semesters, and it varies little, even between disparate courses.

Course	Semester	Instructor	Student Enrollment	Cumulative Use of In-Person Assistance					
				50%		80%		90%	
				#	%	#	%	#	%
Data Structures	Fall ‘13	A	502	37	7.4%	96	19.1%	134	26.7%
	Sp. ‘14	A	586	43	7.3%	109	18.6%	151	25.8%
	Fall ‘14	A	592	41	6.9%	104	17.6%	151	25.5%
Computer Architecture	Fall ‘13	B	210	23	11.0%	53	25.2%	72	34.3%
	Sp. ‘14	C	220	24	10.9%	49	22.3%	68	30.9%
	Fall ‘14	C	199	25	12.6%	58	29.1%	76	38.2%
Systems Programming	Sp. ‘14	D	198	13	6.6%	32	16.2%	48	24.2%
	Fall ‘14	E	351	19	5.4%	53	15.1%	77	21.9%
Average Across All Courses:					8.5%		20.4%		28.4%

Table 2: Use of in-person assistance, based on in-person questions answered

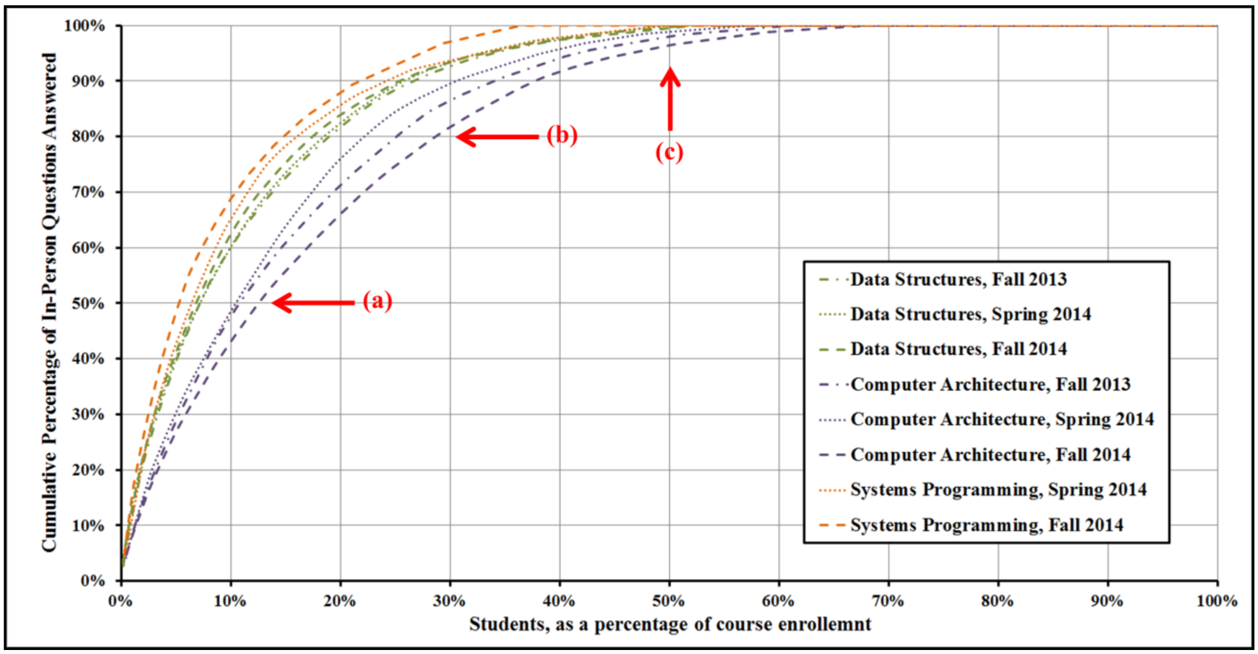


Figure 3: Cumulative use of in-person help, based on students as a percentage of course enrollment. Three areas of interest on the graph are marked:
(a): 50% of all questions per course, roughly asked by 10% of students
(b): 80% of all questions per course, roughly asked by 20% of students
(c): Nearly all in-person questions come from only 50% of enrolled students

This result provokes additional questions centered on the approximately 50% of students who do *not* use open lab assistance. Are they creating small study groups of peers? Are they using the on-line moderated forum in isolation? Is the work challenging enough for them? How do they perceive the available resources? Answers to these questions could inform pedagogical adaptation for those who do use the open lab, perhaps helping them to become more independent.

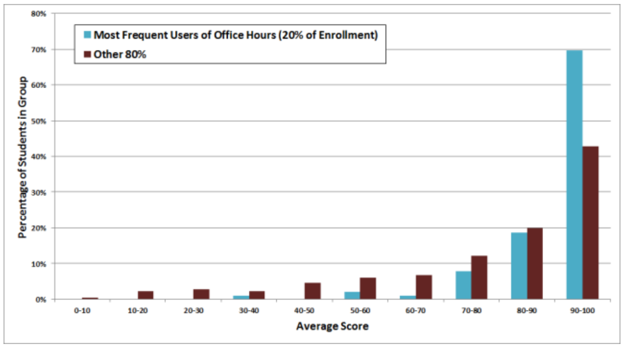
Student Learning: The punch line for this research is the comparison of student learning outcomes between those who use open lab assistance, and those who do not. As proxy measures for learning, we use 1) average scores on the particular assignments for which students seek assistance, and 2) average course exam scores. For each course offering students were partitioned into two groups: the 20% most frequent users of open lab course assistance were compared against the other 80%. This partition was chosen because among all such k% vs. (100-k)% partitions, it maximized the difference in course performance while at the same time accounting for approximately 80% of questions posed during office hours.

In all courses, across semesters, the frequent lab assistance users did significantly better on programming assignments than their infrequent lab-visiting peers, which we expect, since the lab assistance focuses on those assessments. On the other hand, there was essentially no difference between the two groups on average exam performance, except in the data structures course. We tested the hypotheses of equal average scores via a two tailed t-test with unequal variances on the samples of those 20% of frequent lab users compared with the 80% of infrequent users. Though we could not conclude that exam scores were significantly better for lab assistance users in the systems and architecture courses, the trend does show slightly better scores in these cases. Table 3 summarizes the data and t-test significance, and Figure 4 illustrates the score histograms for the frequent users vs. non-users across programming assignments and exams for all three courses in Fall, 2014.

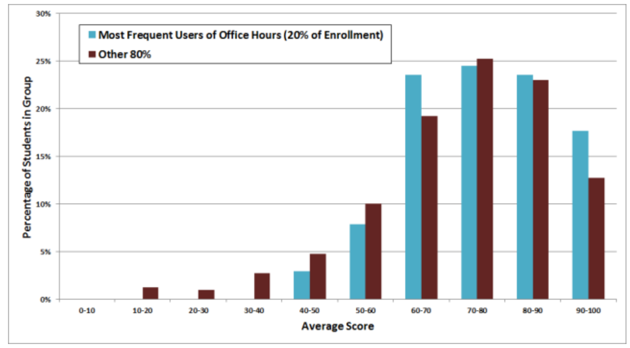
Course	Assessment	Group	#	Range	Average	Std dev	t-test significance
Data Structures (Fall 2014)	Programming Assignments	All	502	0…89	81.40	21.86	
		20%	102	8…89	91.73	10.78	Significant (p < 0.001)
		80%	400	0…7	78.76	23.16	
		Change:			+12.97		
	Exams	All	502	0…89	73.26	16.05	
		20%	102	8…89	76.48	12.82	Significant (p < 0.01)
		80%	400	7…0	72.44	16.68	
		Change:			+4.04		
Computer Architecture (Fall 2014)	Programming Assignments	All	197	0…76	82.43	17.26	
		20%	40	15…76	89.98	7.77	Significant (p < 0.002)
		80%	157	14…0	80.51	18.46	
		Change:			+9.47		
	Exams	All	197	0…76	87.06	13.33	
		20%	40	15…76	88.51	8.47	None (p > 0.05)
		80%	157	14…0	86.68	14.07	
		Change:			+1.83		
Systems Programming (Fall 2014)	Programming Assignments	All	326	0…36	87.67	18.61	
		20%	71	3…36	94.60	11.29	Significant (p < 0.001)
		80%	255	0…2	87.74	19.77	
		Change:			+8.86		
	Exams	All	326	0…36	74.50	12.52	
		20%	71	3…36	75.63	9.63	None (p > 0.05)
		80%	255	0…2	74.18	13.21	
		Change:			+1.45	--	

Table 3: Exploration of course grades on different types of assessments based on the number of open lab questions answered for each student.

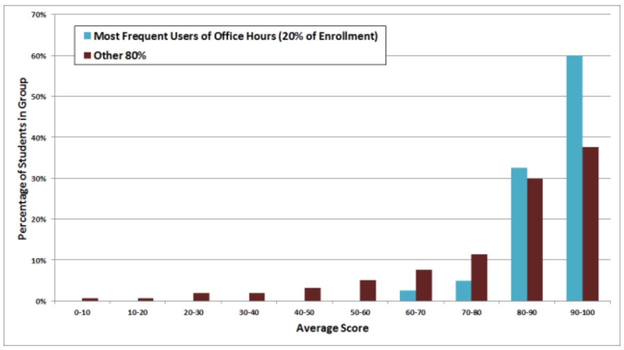
Programming Assignments	Exams



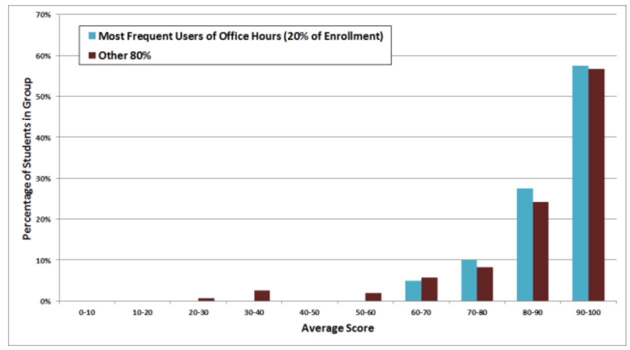
(a): Data Structures (Assignments), Fall 2014



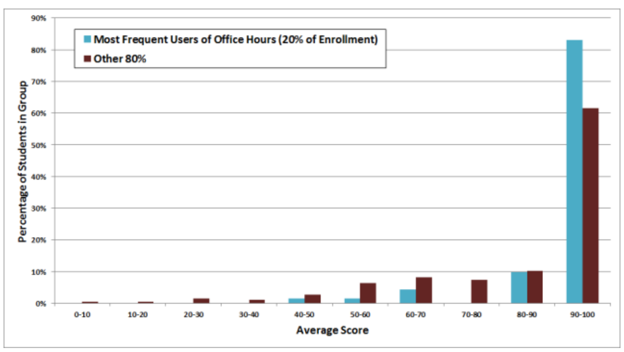
(b): Data Structures (Exams), Fall 2014



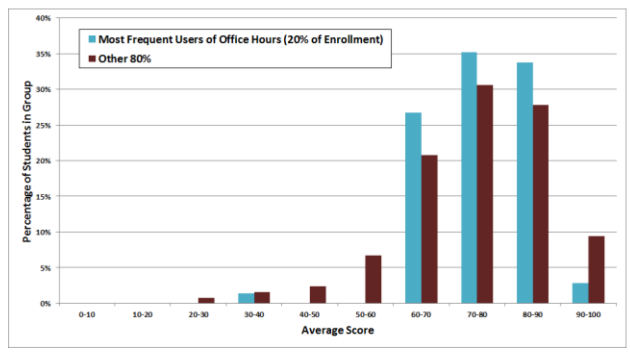
(c): Computer Architecture (Assignments), Fall 2014



(d): Computer Architecture (Exams), Fall 2014



(e): Systems Programming (Assignments), Fall 2014



(f): Systems Programming (Exams), Fall 2014

Figure 4: Relative frequencies of scores for two different types of assessments (programming assignments and exams) across three different courses.

In each chart of Figure 4, the value on the horizontal axis corresponds to an average score on a 100 point scale for either programming assignments or exams, and the table illustrates the relative frequency of each score range. Each chart contrasts the score distribution of frequent lab users (light bars) with the score distribution of lab non-users (dark bars).

4. Future Work:

These results primarily serve to evoke additional questions we might address using this data source. There are four immediate avenues of research we intend to explore:

1) Recent research by Piazza[3], the online moderated forum used by our courses, indicates that women use that resource differently than men, even if they are anonymous to their peers. A similar analysis of the queue data could tell us specifically whether or not women (who are dramatically under-represented in our general population) benefit differently than men do.

2) We intend to use the data on student study patterns to design computer simulations that allow us to test lab staffing policies with the aim of optimizing student throughput, and minimizing queue length.

3) In addition to time stamp data, the Queue is instrumented with a coarse-grained feedback mechanism whereby staff can evaluate student preparedness on any question. We will use this data to prompt early intervention for struggling students.

4) Finally, together with data from the online forum, grade data, attendance, assignment submissions, and lab exercise scores, we will use the queue data to characterize successful students and their study habits, so we can prescribe behaviors that we believe will result in positive course outcomes.

References

[1]: "NEMO-Q | Line Management Systems", http://www.nemo-q.com
[2]: "Appointment Scheduling Software, Scheduling System | Q-nomy", http://www.qnomy.com/Products/Queue-Management.aspx
[3]: "STEM Confidence Gap | Piazza Blog", http://blog.piazza.com/stem-confidence-gap/
[4]: MacWilliam, Malan. "Scaling Office Hours: Managing Live Q&A in Large Courses." Journal of Computing Sciences in Colleges 28.3 (2013): 94-101.