Our experience with self-assessment and peer critique in design education

Chinmay Kulkarni
Stanford University HCI Group
Stanford, CA 94305
chinmay@cs.stanford.edu

Scott R. Klemmer
Stanford University HCI Group
Stanford, CA 94305
srk@cs.stanford.edu

ABSTRACT

Accurately assessing the quality of one's work is an crucial skill for designers and students who want to learn design. This paper describes our experience creating materials that help students self-assess their work in a design course at Stanford University. Using these materials, students have been successful in evaluating themselves—in 69% of all student submissions, self-assessed scores matched independent staff-assessments. Self-assessment accuracy also improved through the quarter ($F(1, 1332) = 31.72, p < 0.001$). We also outline our plans to extend these materials for use with peer-evaluation and for assessments in large online design classes.

Author Keywords

design; education; online education; self evaluation; data analysis.

ACM Classification Keywords

H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

INTRODUCTION

What is good design? Learning the answers to this question is a key part of student's design education. Furthermore, the ability to accurately assess the quality of one's work is a useful skill for the designer [3] and, perhaps, for all creative professions. Unfortunately, students (especially in engineering schools) often struggle with both these issues. In part, this is because design places "emphasis on synthesis rather than predominantly on analysis" [3] but also because design often has no clear right or wrong answers. What educators need, then, are techniques that help students learn the characteristics of good design, and enable them to examine their own work for these characteristics.

We think that self-assessment is one such technique. Prior work has demonstrated that self-assessment improves learning and student performance. In critical writing, students exposed to self-evaluation outperformed those who weren't. Furthermore, these gains were larger for weaker students [9]. In engineering design, self evaluation has been demonstrated to help students accurately gauge their own strengths and weaknesses [3]. Student feedback also suggests that self-evaluation changes the role of the teaching staff, making them coaches rather than evaluators.

This paper describes materials built around self-assessment for an introductory HCI course at Stanford University. The goal of these materials is to help students learn to evaluate their own design work. These materials comprise four main components: 1) a set of detailed weekly assignments that help students learn user-centered design by doing user-centered design; 2) a set of rubrics students use to evaluate their work in these assignments with; 3) an assessment system that combines self-assessment, which encourages reflection, with independent staff assessment that provides critical feedback; and 4) an analytic toolset that enables us iterative improvement of the other components.

The next section describes this self-assessment system. Later sections describe how it has helped improve the class, and some challenges we have faced. We conclude with plans for the future, including how these self-assessment materials could be used for other applications, such as online HCI education.

SELF-ASSESSMENT IN ACTION: CS147

CS147 is a introductory project-based HCI class. In 2011, the class had 156 enrolled students (both undergrad and graduate) from 22 majors (Figure 1). Students participate in a 10-week user-centered design process by designing and implementing a working prototype of a mobile web-application of their choosing. Creating a mobile web-app allows students to pursue ideas that are fairly unique, yet sufficiently constrained that they provide homogeneity for grading. Furthermore, this homogeneity allows students to learn from each other.

Class lectures are supplemented by 12-15 person weekly studios where students present their work and share ideas. Studios last 50 minutes and are held by TAs throughout the day every Friday. Besides providing an opportunity to students to interact and learn from each other, studios provide a platform to receive critical feedback from peers.

While this paper focuses on self-assessment as a teaching tool, we think success with self-assessment is influenced by the other aspects of the class (such as weekly assignments and

CHI'12, May 5–10, 2012, Austin, Texas, USA.

the submission system), which we describe below.

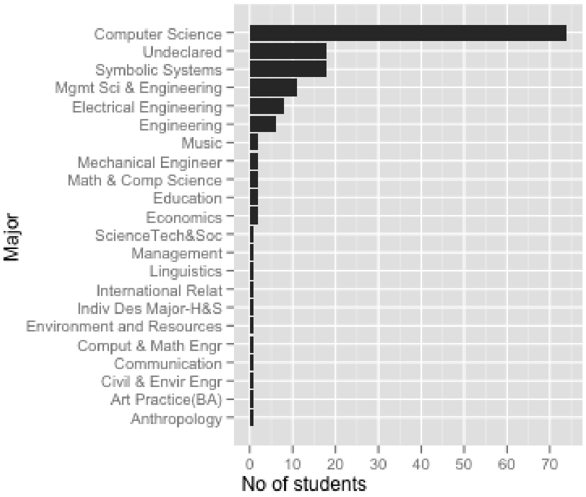


Figure 1. CS147 enrollment: while a large minority of students (47%) major in Computer Science, other majors are well-represented.

Weekly assignments

Weekly assignments guide students through the design and implementation of a mobile web-app of their choosing. Early assignments are individual, while those in the later half of the term are performed in teams.

All assignments are submitted online. When assignments include paper prototypes or other physical artifacts students exhibits these in studio, and upload pictures online. Online submissions (but not the grades) are visible to all students in the class, along with the students' name. Informally, we have noticed that public visibility enables students learn from each other's online work, incentivizes higher quality work, and helps grading to be seen as fair.

Rubrics

Evaluation that is performed with the help of well-defined rubrics leads to students developing a deeper understanding of what constitutes high-quality work and makes the grading seem more fair and transparent [1]. Rubrics also provide students with more detailed and easily understandable feedback [2].

Every assignment in cs147 includes a rubric (see Table 1 for an example). We think that a clear understanding of the rubric is essential to the success of self-assessment. Therefore, at the end of each studio, a teaching assistant walks students through the goals and expectations of the next assignment, describes each rubric item in detail and answers any student questions.

Self-assessment and Staff grading

Self-evaluation occurs at the end of each studio session, after all students have presented and discussed their work. By encouraging self-reflection, performing self-evaluation immediately after discussion helps students gain understanding of the relative standing of their work and evaluate it objectively.

Once the students have submitted their self-evaluation, teaching staff grade the students using the same rubrics that the students used for self-evaluation. This grading is done blind to the grades the students gave themselves. Our online submissions system automatically calculates the student's final grade as the self-assigned grade if the self-assigned and staff-assigned grades are close (the 2011 version of the class allowed a 3% difference between the grades). If the grades are not close, then the student is automatically assigned the teaching staff grade. Grades are then released to students, such that feedback from the staff can inform work on the next assignment.

Incentivizing accurate self-assessment

The ability to self-assess is a useful skill in its own right, so we incentivize students to self-assess accurately. For the 2011 year, students were awarded credit based on how close their self-evaluation was to staff grades, with the maximum bonus worth 2.5% of the points from all assignments. Besides incentivizing a useful skill, this also mitigates the effect of "gaming" self-evaluation, by providing students close to the maximum a student could gain by consistently grading herself above the staff-grade. In practice, no student evaluated themselves more than 2.5% above the staff grade across the quarter.

How accurately do students self-assess?

Having two independent measures of a student's performance (self- and staff-assessment) enables us to measure the effectiveness of self-grading. Overall, for the 2011 year, 69% of all student submissions got their self-assigned grade. Even when students didn't get their own grade, these grades correlated well with the staff assigned grades. The overall Pearson correlation between the two measures for all submissions was 0.91 ($t(2028) = 103.32, p < 0.0001$).

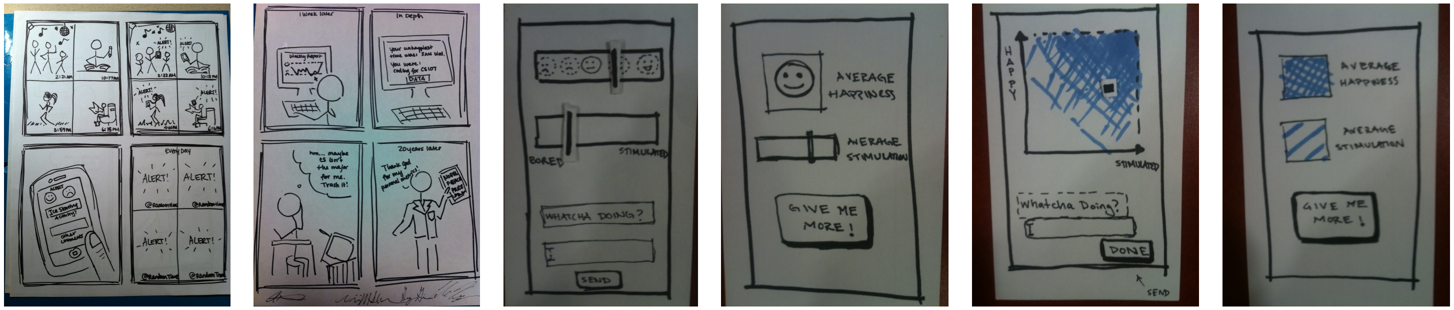
Making rubrics more flexible

Because the class involves a project of a student's own choosing, implementation tasks for projects vary widely. As such, it is not possible to come up with a reasonable single rubric that would fit all possible projects. Could students develop their own rubric for evaluating their projects? We experimented with this idea by splitting the rubric for project progress into two parts. Students first create their own schedule and the first part of this rubric assesses how realistic this schedule is. Students could make changes to their schedule based on this feedback. In subsequent weeks, they evaluate their progress against this schedule using the second part of this rubric. This enables using a fixed two-part rubric to assess tasks that are inherently diverse.

Grade value 100 points

Guiding questions	Bare minimum	Satisfactory effort & Performance	Above & Beyond
Point of view. Does your point of view relate to the design brief, clearly express a problem / opportunity, and clearly convey what a good solution needs to accomplish? (Max 20: 10 for the problem, 10 for the solution requirement)	0-7: The problem is unclear / missing, the solution requirement is unclear / missing, or the point of view is unrelated to the design brief.	8-15: The point of view relates to the brief and the problem and solution requirement are clearly stated, but the solution requirement is either too general (anything that solves the problem meets the requirement) or too specific (only one particular implementation meets the requirement).	16-20: The problem and solution requirement are clearly stated. The requirement provides focus without demading one specific implementation.
Storyboards. Do they both address your point of view? Do they diverge in the solutions? (Max 40: 20 per storyboard)	0-16: The storyboards are hard to follow or do not address the point of view.	17-33: The storyboards reasonably address the point of view, but either a reader may have lingering questions about the situations depicted or the solutions don't diverge much.	34-40: The storyboards are easy to follow and have diverging solutions. Someone else could come up with distinct prototypes just from looking at your storyboards.
Paper prototypes. Did you explore two clearly different interfaces implementing the same idea? How was the quality of paper prototype? Does it feel dynamic, like a working application? Were you creative when implementing the interactions? (Max 40: 20 per prototype)	0-16: The prototypes are incomplete in significant ways. Many screens refer to screens that are not prototyped, and it's often unclear what a certain screen does.	17-33: The prototypes are mostly complete. The purpose of each screen is clear. But maybe the interfaces are not that distinct and share many similarities. Or maybe a user looking at the prototype may sometimes have a question about how to navigate between screens, how to use a form on a screen, or what some element on a screen is doing there.	34-40: The prototypes explore two different interfaces and are detailed enough so that (1) a user can get a good feel for how the application works and flows and (2) a programmer can use the prototypes to implement a skeleton web-application that has working forms and links.

Table 1. Rubric for assignment in Week 3: Prototyping. Rubrics are detailed, and provide objective measures of performance wherever possible.



(i) (ii) (iii) (iv) (v) (vi)

Figure 2. A sample submission by students. Staff and students used the rubric in Table 1 to evaluate this submission. Only one of the two storyboards the students submitted is shown, along with its prototype-screens.

Splitting the rubric into these two separate components has worked well: for the 2011 year, course-staff assessed 98% of students as making "adequate progress" at all their milestones, and 42% as going "above and beyond" at all milestones (milestones were weekly, excepting holidays). This success suggests that such a split approach involving students in their own evaluation could allow rubrics to be used even when the evaluation task is not well-defined. Furthermore, Andrade [2] suggests involving students in creating a rubric as a way to help them understand pedagogical goals more clearly; this split-rubric method may make such collaboration practical in large classes.

DATA ANALYTICS

Online submissions of assignments leads to a large amount of data in an easily analyzable form—the time assignments are turned in, how students grade themselves (and how staff grade them), even the actual text of the submission. At Stanford, we use this data towards three main goals: 1) to ensure that the course is meeting goals that are hard to measure through other means; 2) to identify potential areas for improving teaching; and 3) to help staff identify and focus on students that may need help.

Below, we offer vignettes of each of these applications. All numbers are from the 2011 class.

Do students get better at self-assessment?

Answering this question is very hard based on formal/informal student feedback—formal University feedback doesn't ask about self-assessment, and students themselves may not be able to accurately gauge themselves.

However, using assignment submission data, we see that students indeed improve. We performed a repeated measures ANOVA on the accuracy of the self-evaluations across the term and found that accuracy improves across the course of the term $F(1, 1332) = 31.72, p < 0.001$ (Figure 3).

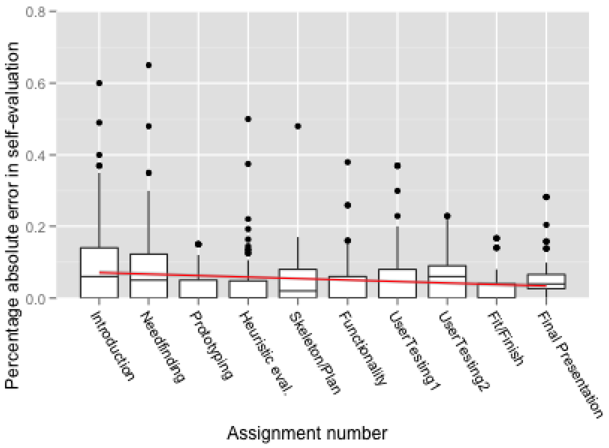


Figure 3. Percentage error of self evaluation over staff assigned grades improves over the term (lower is better).

Improving rubrics

Analyzing submission data also enables us to improve rubrics by identifying items that are confusing. For instance, we see that given the general trend of decreasing differences between self evaluated and staff assigned scores (Figure 3), Assignment 7 (User Testing 1) has an unusually poor correlation between self-assessed and the staff-assessed scores (Pearson = 0.86, vs. the mean 0.91). Looking deeper, we see that the rubric item in assignment 7 which has the lowest self-and-staff correlation asks students to come up with a list of "who did what" that week toward the quarter-long project. This is the first time in the quarter that such a status update was required, and the data suggest that students did not understand requirements clearly.

We later found student complaints at the end of the quarter echoed this concern "Our group knew what we had to accomplish, and wrote the implementation plan to the level that we needed to, but it never seemed to be good enough..."

While formal student feedback is always helpful, it is often obtained too late to improve the course for the current term. Using submission data directly, staff could identify the problem, solicit informal feedback, and improve rubrics later in the quarter that asked the "who did what" question.

Similarly, rubrics also help identify areas for improving teaching. For the 2011 year, for instance, a rubric item for a need-finding assignment had the lowest grades. This item required students to list the ideas they brainstormed to solve a user need they'd identified. The rubric graded students exclusively on the number of ideas brainstormed, and did not give more credit to more "insightful" ideas (following Osborn's advice on brainstorming that "quantity is wanted" [8]). However, students seem to have not understood why credit was given to quantity: one student complained in end-of-quarter feedback, "Generate 20 good ideas instead of 25 silly ones? Expect to lose points."

Identifying students who may potentially perform poorly

Based on student submissions for the last two years, we have identified questions in early assignments that are predictive of poor performance toward the end of the term. We identify such questions by looking at correlations between the student's grade on the assignment, and her final grade in the course. For the 2011 class, the three most predictive questions asked students to (a) brainstorm ideas for their project, (b) create storyboards for the use of their application, and to (c) create paper prototypes of their application. In general, questions with high predictive power test both the student's commitment to work on class assignments (such as number of prototypes created) and the student's grasp of concepts taught in class (e.g., storyboards should show the prototype being used in context).

Based on these and other features (student major and year), we have built a random-forest classifier [4] to identify students who may perform poorly. We plan to use this classifier to identify students who may need help understanding course material or budgeting time for the class.

CHALLENGES AND FUTURE DIRECTIONS

The biggest challenge to building a successful self-assessment system is building effective rubrics. To create rubrics that help learners improve, one needs essentially to be able articulate heuristics for excellence that are concrete, but not limiting. While experience, student feedback and data analytics can help, the tension between concrete and limiting may be inherent to all creative domains.

Second, self-assessment also changes the role of the teaching staff. Instead of simply being graders, TAs are viewed as allies and advisors who help students do well on the rubric. This new role places greater responsibility on TAs, and is often unfamiliar.

Lastly, since assignments and studios in our current system work in lock-step, physical constraints become limiting. For instance, all studios happen on Friday, and it's difficult finding classrooms for them.

FUTURE DIRECTIONS

Scaling HCI education

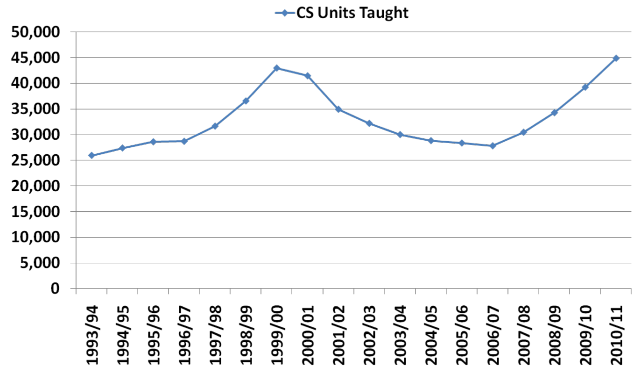


Figure 4. The upward trend in CS enrollment units at Stanford

While the importance of user-centered design has long been recognized [7], there is now increasing awareness of its importance in industry. This has led to a greater need and demand for design education. Enrollment in CS147 at Stanford has increased every year since 2007, even considering the general upward trend in CS enrollment at Stanford (Figure 4). As another example, the online version of this class, `hci-class.org`, has enrollments in the tens of thousands. With classes getting larger, the ability of traditional staff-driven evaluation to accurately assess and help the student learn is being challenged.

Self-assessment and related evaluation mechanisms (like peer assessment) offer one of the few hopes of scaling design education. Unlike other engineering disciplines, evaluating design is an inherently human endeavor. Jonathan Grudin writes that "A central tenet of HCI is that we cannot design interaction of any complexity by the application of formal rules. It is always necessary to try interfaces out with people..." [6]. How then could we automate, say, the evaluation of a prototype? Could the storyboard for a new interactive system be evaluated by algorithm?

Unlike other hard-to-automate problems, however, design evaluation may also be difficult to crowdsource. Tohidi et. al. warn that "...design and creativity are specialized skills. There is no reason to expect them to be spontaneously manifest in those not trained in the field" [11]. Leveraging platforms like Mechanical Turk can thus be both invalid and also potentially harmful. Peer-evaluation may solve this problem by using evaluators who, though non-expert, are interested in acquiring relevant skills and in learning the practice of design.

Peer evaluations

Prior work has established that given clear grading criteria, students "can make rational judgements on the achievements of their peers" [10]. For `hci-class.org`, it is impractical for teaching staff to evaluate every student's submission. Therefore, we plan to use peer-assessment to supplant assessment by the teaching staff. Rubrics we've developed for CS147 will be used both for self- and peer- assessment.

Since peer-review is used to supplant staff evaluation, consistency of grading is a potential concern. Calibrated peer review [5] offers a way to mitigate inconsistency by first training the peer-grader on a number of training assignments, and only allowing them to grade other students once they grade training assignments close enough to staff grades. Such a calibrated peer review system has been used successfully at other universities, e.g., for the Distributed Cognition class at UC San Diego by Edward Hutchins.

ACKNOWLEDGEMENTS

We thank Maria Sandberg, who, as a teaching assistant the 2009 version of the class, first implemented self-assessment, and to students in CS147 who helped improve the system with their feedback.

REFERENCES

1. Andrade, H. The effects of instructional rubrics on learning to write. *Current Issues in Education 4*, 4 (2001).
2. Andrade, H. Teaching with rubrics: The good, the bad, and the ugly. *College Teaching 53*, 1 (2005), 27–31.
3. Boud, D. *Enhancing learning through self assessment*. Routledge, 1995.
4. Breiman, L. Random forests. *Machine learning 45*, 1 (2001), 5–32.
5. Carlson, P., and Berry, F. Calibrated peer review and assessing learning outcomes. In *FRONTIERS IN EDUCATION CONFERENCE*, vol. 2, STIPES (2003).
6. Grudin, J. Ai and hci: Two fields divided by a common focus. *AI Magazine 30*, 4 (2009), 48.
7. Kelley, D., and Hartfield, B. The designer's stance. In *Bringing design to software*, ACM (1996), 151–170.
8. Osborn, A. *Applied imagination; principles and procedures of creative problem-solving*. Scribner, 1963.
9. Ross, J., Rolheiser, C., and Hogaboam-Gray, A. Effects of self-evaluation training on narrative writing. *Assessing Writing 6*, 1 (1999), 107–132.

10. Stefani, L. Peer, self and tutor assessment: relative reliabilities. *Studies in Higher Education 19*, 1 (1994), 69–75.

11. Tohidi, M., Buxton, W., Baecker, R., and Sellen, A. Getting the right design and the design right. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, ACM (2006), 1243–1252.