Measuring increased engagement using Tablet PCs in a code review class

S. Kamin  
Computer Science Dept.  
University of Illinois  
kamin@illinois.edu

W. Fagen  
Computer Science Dept.  
University of Illinois  
wfagen2@illinois.edu

C. Cortez  
Computer Science Dept.  
University of Illinois  
cortez3@illinois.edu

ABSTRACT
The Programming Studio in the University of Illinois Computer Science department is a required course in which small groups of students participate in weekly code reviews of each other’s programs. To increase the liveliness of the discussions, Tablet PCs were introduced. We measure the effectiveness of this intervention both by surveys and by performing statistical tests on audio recordings of the Studio with and without the tablets. The latter suggests that speech is more evenly distributed among different students over the course of a code review when Tablet PCs are used; we view this as an indication that they are producing livelier discussions.

Categories and Subject Descriptors
K3.2 [Computer and Information Science Education]: Computer Science Education

General Terms
Education, software engineering

Keywords
Code reviews, Tablet PC

1. INTRODUCTION
Students in the Computer Science department at the University of Illinois (Urbana-Champaign) are required to take CS 242, Programming Studio. Normally taken in the first semester of the junior year — after the core programming courses — the Studio is intended to build individual programming skills. The overall approach is drawn from studio courses in architecture and fine arts: students present their work “in public,” and get feedback. Specifically, students meet every week in discussion sections of 3 to 6 students, plus a moderator, where each student has twenty minutes to present their program. With some exceptions, the discussions focus on the code itself, rather than the functionality of the program. Assignments are not as structured as in traditional classes, but the students generally have an idea of what the other students’ programs do. (The main exception is in the last four weeks of the semester, where student-selected term projects are presented; during that time, program demos are mixed with code discussions.)

We reported on the Studio course shortly after it was introduced in 2007 [WK07]. We said at that time that although the Studio seemed to be useful to most students, there were some difficulties we were confronting; we noted, “We need to continually work on maintaining the quality of the discussion in the discussion sections.” Programs are complicated, and it is hard to have a stimulating discussion on something that only one of the interlocutors understands. We feared that the discussions would be closer to monologues, the critiques offered would be superficial, and the discussion would be dull and unengaging. In [WK07], we discussed various mechanisms we had used to help stimulate discussion. Using Tablet PCs is another such a mechanism. This paper discusses how they were used and how we attempted to verify that they were having the desired effect.

Tablet PCs are ordinary PCs with the added feature of a screen that can be written upon. The machines we used are “Tablet PCs,” employing Microsoft’s digital ink API (rather than “tablets” in the sense of the iPad). These machines have an “active digitizer” screen which can only be written upon with a special stylus. Writing on them is very natural, and students adjust to them easily. In the Studio, each student has a Tablet, and all comments written by any student are instantly shared with all other students. The system we used provides in effect a shared whiteboard, with a student’s code forming the background image.

Without the Tablet PCs, the Studio operates by having each student explain his or her program as it is displayed on a large screen. The Tablets provide two benefits: First, writing, which makes it easier to ask and answer questions about code. Second, private copies of
The presenter’s code, allowing each student to explore it independently, rather than following the presenter. In introducing the Tablets, we hoped these benefits would help make the discussions more engaging and efficient.

The study reported here was performed in the Spring, 2011, semester. Its goal was to test whether the tablets had a positive effect on the discussions. We set up a simple experiment: In five sections, we had students alternate between Tablet use and the traditional Studio. We gathered data in two ways: surveys of participating students, and statistical analysis of audio recordings of the Studio sections. In this paper, we report on both, but mainly on the analysis of the recordings.

This paper makes three contributions to the study of Computer Science education:

- We describe a Tablet PC system supporting code reviews in the Programming Studio, and explain why it is helpful.
- We give a method for mechanically measuring “engagement,” or “liveliness,” of the Studio discussions, based on analyzing audio recordings of the participants.
- We describe an experiment to test the hypothesis that the introduction of Tablet PCs lead to livelier discussions. We present data from both student surveys and voice recording analyses that confirm the hypothesis.

The next two sections of the paper describe the Programming Studio and the use of Tablet PCs. Section 4 describes the experimental set-up, and then sections 5 and 6 give the results obtained from the surveys and the recordings, respectively. These are followed by a discussion of related work, and our conclusions.

2. PROGRAMMING STUDIO

CS 242, Programming Studio [WK07], is a required class for CS majors at the University of Illinois, providing individualized instruction on programming by having students critique each other’s programs in a code review setting. The class grew out of a concern that, despite many hours spent programming in CS courses, not all of our students were becoming proficient at it. A student could get through our program — we are undoubtedly not unique in this regard — without ever having a detailed, one-on-one, discussion about a program he or she had written.

The Studio is taken in the junior year, after all the core courses have been completed. Students get fairly open-ended assignments each week, and are expected to show up at the weekly discussion with a working program. In a group of three to six students, plus a moderator — a graduate student or upper-level undergraduate — each student presents his or her program for 20–25 minutes. The presentation may include a demo, but usually it is simply a discussion of the code. (This changes in the last few weeks of the semester, when students do self-chosen projects; then, more of the discussion is devoted to demos.) In the presentation, the presenter stands in front of the group and explains his or her program, while showing it on a large display. The arrangement is shown in Figure 1.

Most students find the studio beneficial, and many report that it played a key role in their development as a Computer Scientist. However, others complain that the discussions tend to focus on superficial aspects of their programs. This points to a fundamental issue that we identified in [WK07]: how can a discussion of programs be kept lively and engaging? Some mechanisms for addressing this issue are described in [WK07]. The Tablet PCs are our latest effort in this direction.

**Terminology:** In this paper, we refer to the presenting student as the **presenter**, non-presenting students as **reviewers**, and everyone in the studio, including the moderator, as **participants**. A **section** is a regular meeting of a particular group of students; the experiment involved five sections (Thursdays at 5PM and 7PM, Fridays at 9AM, 11AM, and 1PM). A **meeting** is a single meeting of a section, lasting from one to two hours depending upon the number of participants.

3. TABLET PCS IN THE STUDIO

The system used in the Studio is a straightforward “shared whiteboard” application. It was developed using the SLICE framework [Kam08,Slice], and is customized for our course. As each student takes a turn as presenter, he or she loads the program files into the system, where each is represented by a tab. Within each tab, the text of the file forms the background of the whiteboard; the text cannot be directly edited, but simply annotated upon. Figure 2 shows the SLICE screen during a studio discussion.

Each student in the discussion, as well as the moderator, has a Tablet, and all share the same data. That is, any program loaded into the presenter’s Tablet, and any stroke made by anyone, is immediately sent to all others. The arrangement is otherwise identical to that shown in Figure 1.

The Tablets provide two capabilities: (1) The ability to write, draw, and point. Drawing is useful to explain the deep structure of a program, data structure, or algorithm; writing makes it easier for reviewers to...
ask “what if you did it this way?” questions; pointing makes it easier to draw attention to a particular spot in the program. In general, writing helps make the discussion more efficient. (2) The ability to explore the presented program independent of the presenter. Each student can switch tabs and scroll through the files in the presented programs. This allows a reviewer to spend time understanding a section of code after the presenter has moved on, or simply to explore it in a different order than the presenter uses. (A button on the interface allows a student to synch up with the presenter.) This can allow a reviewer to gain a better understanding of the presented program — or at least of some part of it — and thus make more substantive comments about it.\footnote{We made little attempt to distinguish the impact of these two features. It would be worthwhile to do so, in particular, because this second capability could be provided with ordinary computers, which may be more readily available than Tablets.}

4. EXPERIMENT

The Studio discussion can switch easily from the traditional structure to the Tablet PC structure. The idea of our experiment was to have the discussions switch from traditional to Tablet back to traditional, while we took two types of measurements: surveys and audio recordings. We then analyzed these measurements to detect differences between the two modalities. The surveys would reveal differences in student perceptions, and the recordings would reveal actual differences in the discussions.

We ran the experiment in the Spring, 2011, semester, with five sections participating, with a total of 18 students and 3 moderators. We began recording students and moderators, using small personal recorders with lapel microphones on each participant, in week 5 and continued through the end of the semester (week 14). In weeks 5–6 and 11–14, the Studio ran in its traditional structure; in weeks 7–10, the Tablets were used (with the exception of a single meeting in week 7). We ran the surveys once, in week 14.

As noted earlier, the Studio was somewhat different in the final four weeks, when discussions were not entirely around code, but also involved demos. It seems reasonable to surmise that the nature of the discussions would change during this period: demos would inspire more lively discussions; on the other hand, reading the code for each project would be more difficult. In analyzing the voice data, we therefore distinguish three periods, or “epochs”: weeks 5–6, plus one meeting in week 7 (traditional structure; discussion of code); weeks 7–10 (Tablet structure; discussion of code); weeks 11–14 (traditional structure; demos and discussion of code).

5. SURVEY RESULTS

The survey was administered on paper, anonymously, in week 14 — that is, after six weeks of traditional structure,\footnote{To be clear: Before we started recording audio in week 5, — the Studio had already been running, in the traditional structure, for four weeks.} four weeks of tablet use, and three more weeks of traditional structure. It was taken by 18 of the 21 participants.

The survey was a single page, with nine questions. The first seven were on a sliding scale, the eighth allowed multiple selections from a list, and the last was an open-ended written response question.

Space prevents us from presenting the entire survey, but we present the first five questions (numbered 1(a)–1(e)). They are introduced by the statement, “Since starting to use the Tablet PCs in studio, please note how your experience in the studio has changed.”

1(a) Do you find the discussions more engaging?
1(b) Do you find the discussions more efficient?
1(c) Do you feel better able to contribute to the discussion?
1(d) Have you been better able to understand presented code?
1(d) Have you felt that you can present your code more effectively?

The responses for questions a, b, and e were, from 1 to 5, “much more,” “a little more,” “same amount,” “a little less,” and “a lot less,” and for c and, “yes, significantly,” “yes, a little,” “no change,” “no, a little less,” and “no, much less.” The responses are shown in Figure 3.

In retrospect, we should have offered intermediate positive responses to these questions. Possibly because of the limited choices, participants generally chose “a little” rather than “much.” But what is noteworthy is the unanimity of the responses. We consider these responses very positive.

6. ANALYSIS OF AUDIO RECORDINGS

We did not originally expect that the use of the Tablets would have very much effect on speaking in the Studio. If anything, we thought the communication channel through the networked tablets might partially supplant the oral channel. However, in a preliminary trial with Tablets in Fall, 2010, we began to consider that the tablets might have a positive effect on the oral channel.
In particular, we postulated that the tablets would help make the discussions more lively and engaging. There are two properties of the studio recordings that we think might represent “liveliness:”

**Total amount of speech.** This is an obvious measure, but in this case, we wouldn’t expect it to increase, for this reason: In both the traditional and Tablet PC sections, to a first order of approximation, there is always exactly one person speaking at any time.

**Distribution of speech.** What we mean by liveliness is “more participants speaking more often.” With no interaction, the presenters give a monologue for about twenty minutes. Increased interaction means more students talking during other students’ presentations. This is the effect we hoped to see.

We first needed to process the audio recordings to determine when “speech” happened. Our analysis assumes that speech, as opposed to noise, is generally characterized by higher volumes, and that speech of the wearer — the person being recorded by a particular microphone — is louder than other sounds. Our processing of the data is described in detail below (section 6.2).

The output of the first part of the analysis is a list of time intervals in each recording, representing “speech events” — times when speech is occurring. We then analyze the speech events for differences in speech quantity and distribution between the different epochs.

### 6.1 The data

The voice recordings were obtained from each participant in each of the five discussions sections in the experiment, during weeks 5–14, with a personal recorder and lapel microphone. These produce files in proprietary “VOC” format, sampled at 8000 Hz, which we translated to 16–bit WAV format.

Table 4 gives the basic statistics about the recordings. For various reasons, we did not always obtain a complete recording for every participant in every section. Because incomplete recordings are not usable for some of the comparative statistics that interest us, we have based most of the numbers presented in the following on the complete recordings.

### 6.2 Speech events

Our analysis is based on the assumption that the loudest sounds in a recording probably represent speech of the person being recorded. This is not always true, but it is the best approximation we can make. The alternatives are to manually code each recording — but doing so accurately would take hundreds of hours — or to mechanically distinguish among the types of sounds, and the specific speakers, in each recording — a technology that is still only experimental [SDV00].

We first applied a small smoothing effect to the data, effectively decreasing the sample rate to 800 Hz. We then identified the background static that was constant across all the recordings and found it to be at -27.4 dB. To distinguish the periods of “silence” (only background noise) from “volume,” we examined each sample and looked for periods of silence followed by five consecutive samples of -27.4 dB or greater. These five and all subsequent samples, until the next five consecutive samples of silence, were together marked as “volume events.” To eliminate “pops,” volume events less than 0.1 seconds in length were removed.

A “speech event” is a single, uninterrupted length of speech by one person. We started the process of combining volume events into speech events by looking at average volumes of consecutive volume events. We found that they usually differed from one another slightly, but were nearly always within 0.8 standard deviations of one another, if the speech was from the same person. Volume events were combined if the averages fell within 0.8 standard deviation and the length of time between the two was less than two seconds.

At this point, thousands of volume events in a single recording had been condensed down to several hundred speech events. We were still left with a significant amount of cross-talk (non-wearer speech). We found that the amplitude inside a volume event of cross-talk generally falls below a certain amplitude level and the wearer’s speech is often vastly above that level. From comparing hand-coded files, we found that the average time-weighted amplitude of -23.7 dB across all the volume events that make up a speech event distinguishes...
the wearer’s speech from cross-talk.

The output of this process is a series of time intervals that represent the start and end of speech events. To validate the algorithm, we had four different humans hand-code 13 audio files. As part of the hand coding, two files were done by two different humans independently. The mismatch of audio marked as speech events between two different humans coding the same file (10%-15%) was in line with the mismatch between a human coding of the file and the result of the algorithm.

### 6.3 Comparing speech quantity

As noted above, we did not expect to see a large effect on total speech quantity from this intervention, but it is still an obvious question to ask. This table gives average speech time, as a percentage of meeting time, for various subgroups of meetings (rows represent subgroups of participants, columns epochs):

<table>
<thead>
<tr>
<th></th>
<th>All</th>
<th>Non-tab</th>
<th>Tablet</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>23</td>
<td>22</td>
<td>24</td>
<td>23</td>
</tr>
<tr>
<td>Students</td>
<td>23</td>
<td>21</td>
<td>25</td>
<td>23</td>
</tr>
<tr>
<td>Moderators</td>
<td>22</td>
<td>24</td>
<td>23</td>
<td>22</td>
</tr>
</tbody>
</table>

Student speech was about 20% higher when using tablets. We note also that of 17 students for which we have readings in both the tablet and non-tablet epochs, 11 spoke more, on average, during the tablet epoch, while 6 spoke more during the non-tablet epoch.

### 6.4 Comparing speech distribution

Consider two scenarios at opposite ends of the “liveliness” spectrum: At one extreme, every student speaks only when presenting, and in the other, all students speak constantly. The first represents the minimum amount of engagement, and the second the maximum. In reality, of course, the meetings fall between these extremes.

The methods we use to measure distribution of speech would clearly distinguish these extreme cases, if they ever occurred. We give three comparisons (each described in detail in the sections that follow):

1. **Window-based speech quantity analysis:** To mitigate the impact of a student’s presentation on the speech quantity measure, we divide each meeting into discrete time windows and give a point for each window that is “occupied.”

2. **Average speech starting time:** Consider the average time at which a student speaks in a particular meeting. If the students speaks only during his or her presentation, this time will be in the middle of the presentation, while if the student speaks equally often during every presentation, the time will be in the middle of the meeting. Thus, a more even distribution of speech would correspond to a lower standard deviation in the average starting times among the students in a meeting.

3. **Non-presentation speech quantity:** The main question for us is, how much do students speak during other students’ presentations? If we knew exactly when each student gave his or presentations, we could remove that speech and just count the quantity of speech that remains.

We find that by each measure, the Tablet sections display broader distribution of speech. The next three sections discuss these comparisons in detail. In giving the results of these analyses, moderators should act as a “sanity check,” in that we would expect their speech to be more evenly distributed than that of students.

#### 6.4.1 Window-based comparison

The theory behind this analysis is that in any summation of speech quantity, a student’s presentation is liable to dominate, but by dividing the meeting into discrete windows and counting a window only once — no matter how much speech occurred in that window — we can mitigate this dominance.

Consider the two extreme scenarios. In both, the amount of speech may be the same, but if we divide the meeting into windows of, say, five minutes, the presentation can only contribute about four points. In the “anarchic” scenario, many more windows may be occupied, albeit sparsely.

For this analysis we use a window size of 5 minutes; a window is considered occupied if there is one minute of speech in those five minutes. Each recording gives a percent occupancy (occupied window/total windows):

<table>
<thead>
<tr>
<th></th>
<th>All</th>
<th>Non-tab</th>
<th>Tablet</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>45</td>
<td>43</td>
<td>48</td>
<td>44</td>
</tr>
<tr>
<td>Students</td>
<td>43</td>
<td>39</td>
<td>47</td>
<td>43</td>
</tr>
<tr>
<td>Moderators</td>
<td>52</td>
<td>56</td>
<td>52</td>
<td>50</td>
</tr>
</tbody>
</table>

As expected, the moderators’ speech is fairly evenly distributed; the largest value (non-tablet) is about 11% greater than the smallest (project). For students, we see a stronger increase for the tablet sections, with an occupancy about 21% higher during the tablet epoch than the non-tablet epoch.

#### 6.4.2 Average speech starting time

This measure of evenness of distribution of speech is a bit more subtle. We can calculate the average start time of a speech event in a particular recording as

\[
\text{Average start time} = \frac{\sum_{e} \text{start-time}(e) \times \text{duration}(e)}{\sum_{e} \text{duration}(e)}
\]

Again consider the two scenarios. In the first — monologues only — we would find that the first presenter’s average speech event start time would be at about 10 minutes (i.e. halfway through the presentation), the second presenter’s would be at about 30 minutes (halfway through the second presentation, which starts at 20 minutes), and so on. All together, we would find average starting times for these students at 10, 30, 50, 70, 90, and 110 minutes. In the second scenario — complete anarchy — we would find every student’s average speech event starting time would be in about the middle of the meeting. The two scenarios are therefore distinguished by the variance in average speech event starting times of the participants.
Here are statistics for various groupings of participants and epochs, giving standard deviations of average start time (taken relative to the duration of the meeting):

<table>
<thead>
<tr>
<th></th>
<th>All</th>
<th>Non-tablet</th>
<th>Tablet</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>.123</td>
<td>.121</td>
<td>.093</td>
<td>.132</td>
</tr>
<tr>
<td>Students</td>
<td>.129</td>
<td>.133</td>
<td>.102</td>
<td>.143</td>
</tr>
<tr>
<td>Moderators</td>
<td>.099</td>
<td>.038</td>
<td>.047</td>
<td>.139</td>
</tr>
</tbody>
</table>

The means are always near 0.5, ranging from a low of .47 to a high of .52. The standard deviations vary more widely. Project presentations seem anomalous, giving the highest values for of all epochs, across subgroups. Considering just the tablet and non-tablet epochs, we find the moderators have considerably smaller values than students, as we would predict. For students, the standard deviation during the tablet epoch is about 25% smaller than during the non-tablet epoch.

6.4.3 Non-presentation speech quantity

Unfortunately, we do not have data on exactly when each student presented in each meeting. So we approximate: We assume that each student speaks more during their presentation than at any other time, and look for when each student spoke the most. We assume that presentations are 15 minutes (there is actually some variation, but to avoid removing non-presentation speech, we used a shorter duration than what we believe to be the actual average). We find the 15 loudest minutes in each recording, remove them, and then calculate the total speech time in what remains. The following numbers give speech time as a percentage of meeting time, for students only (since removing the maximum 15 minutes of a moderators’ speech makes no sense).

<table>
<thead>
<tr>
<th></th>
<th>All</th>
<th>Non-tablet</th>
<th>Tablet</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>Students</td>
<td>11</td>
<td>10</td>
<td>13</td>
<td>10</td>
</tr>
</tbody>
</table>

During the tablet epoch, the average amount of speech for students, outside of their own presentation, was 30% higher than during the other epochs. Also, of 17 students for which we have readings in both epochs, all but one spoke more during the tablet epoch than during the non-tablet epoch.

6.5 Discussion

In interpreting these results, the biggest question for us is whether the notion of “speech event” that we calculate is “correct.” We cannot know that we are using volume levels that always distinguish the speaker’s voice from other sounds. Furthermore, it is difficult to be sure comparing to accurate manual codings, which, as we have noted already, are extremely tedious to produce.

For now, all we can say is that the automated codings seem fairly good, and tend to match our manual codings about as well as the manual codings match each other. Moreover, we cannot see how the processing we do is biased toward use or non-use of Tablet PCs, so that the consistent findings in favor of the Tablet epoch are, taken together, suggestive of a meaningful effect.

7. RELATED WORK

We drew the idea of using individual lapel microphones from “Conversation Clock” [BK07], a system that visually displays each individual’s contribution to a conversation at a table. It has proved difficult to segment an individual’s voice from the background audio without domain knowledge [SDV00]; in [BK07], domain knowledge was obtained by training the system with the speaker’s voices.

8. CONCLUSIONS AND FUTURE WORK

We have described how, in a course in which Computer science students gather for mutual code reviews, Tablet PCs were introduced to help facilitate and enliven the discussions. From the results of student surveys, and various analyses of recordings of the discussions, we have argued that the Tablet PCs did indeed have the desired effect.

We would not claim that the effect of Tablet PCs in the Programming Studio has been fully studied. Our experiment involved only five sections, with 18 students. Furthermore, we do not have any idea of how sensitive the results may be to the software itself; this was the first time the Slice software was used extensively for this purpose. Aside from simple improvements like fixing bugs, we can imagine added features like having different interfaces for different “roles” [Wi01], especially for the discussion moderators.

The main technical issue the arises from this work is how to distinguish among the voices in a recording. If we were able to reliably distinguish among different voices, not only could we get more accurate results for these data, we could run future experiments with much greater ease. It would mean we could record a class with one or two free-standing microphones, instead of individual lapel mikes, making these kinds of experiments much more practical. The quest for increased student engagement is a long one; a Google search for “strategies for increasing student engagement” gets over two million hits. In those circumstances where talkativeness is a good proxy for engagement, we believe our work can contribute to studying engagement using more objective measurements.

References


[SDV00] K. Shearer, C. Dorai, S. Venkatesh, Incorporating Domain Knowledge with Video and Voice Data Analysis in News Broadcasts, Proc. Sixth ACM Intl. Conf. on Knowledge Discovery and Data Mining, Boston, MA, 2000

[Slice] SLICE website: slice.cs.illinois.edu
