# Using the Readiness Assurance Process and Metacognition in an Operating Systems Course

Michael S. Kirkpatrick
Department of Computer Science
James Madison University
kirkpams@jmu.edu

Samantha Prins
Department of Mathematics and Statistics
James Madison University
prinssc@jmu.edu

## Categories and Subject Descriptors

K.3.2 [**Computing Milieux**]: Computer and Information Science Education—*Computer science education*

## General Terms

Human Factors

## Keywords

active learning, education, pedagogy, metacognition

## ABSTRACT

There is significant evidence that active learning techniques facilitate superior learning outcomes when compared to traditional lecture-based techniques. However, adopting an entirely new pedagogy is a time-consuming endeavor that requires considerable effort. In this work, we describe simple transitional steps that we used to increase the amount of active learning in our Operating Systems (OS) course. After introducing these techniques in the Fall 2013 offering of this course, we observed dramatic improvements in a variety of measures of student outcomes, including withdraw-D-failure (WDF) rates and final exam performance. We also observed marked improvement in project completion rates. Our results suggest that adopting components of active learning pedagogies can contribute to positive outcomes with modest investments in time and effort.

## 1. INTRODUCTION

The body of evidence supporting active learning methodologies is significant and continues to grow [4, 6, 15, 5]. Several papers (*e.g.,* [14, 8, 7, 9, 10, 2, 3]), have documented the success of these approaches in computer science specifically. It is becoming increasingly clear that courses that rely on lecture alone are less effective in terms of concept retention and transfer when compared to courses that incorporate in-class activities that provide opportunities for formative assessment. The challenge now is *how* to make the change to active learning. Adopting a new pedagogy is a time-consuming process that can require a significant effort by the instructor, and wary instructors frequently express concerns about the impact of the techniques on coverage of material. In contrast to previous work on new pedagogies, the focus of this paper is to describe our experience with a transitional step of integrating two techniques into our existing interactive lecture[1] course structure, rather than adopting a completely new pedagogy.

Our first technique is based on the Readiness Assurance Process (RAP) as used in Team-Based Learning (TBL) [12].[2] The RAP consists of four components: an individual Readiness Assessment Test (iRAT), a team Readiness Assessment Test (tRAT), Appeals, and Corrective Instruction. In the iRAT and tRAT phases, students take a quiz first as individuals then again as a team; the questions are identical in both cases. The iRAT holds individuals accountable for doing preparation work on their own; failure to prepare will impact their course grade. The tRAT provides a valuable opportunity to discuss concepts with peers, which leads to improved understanding by the group, even if none of the students originally knew the correct answer [16]. During the Appeals phase, teams can submit a written appeal for additional points if they can provide evidence for why their answer should also be considered correct. During the Corrective Instruction phase, the instructor clarifies material based on feedback from the iRAT and tRAT phases.

Our second technique is a variant on two metacognitive Classroom Assessment Techniques (CATs) known as the RSQC$^2$ and muddiest points [1]. (Note that the $^2$ is an exponent and does not indicate a footnote; we use the name as it was used in [1] for consistency.) The RSQC$^2$ stands for *recall, summarize, question, connect, comment.* Students are asked to *recall* and *summarize* the most important point, then craft a precise *question* that they would like answered; students are then asked to *connect* the concepts to previous material in the course or the discipline and then *comment* on their learning process. This activity provides an opportunity to bring metacognition into the course, which is beneficial


*ITiCSE'15,* July 6–8, 2015, Vilnius, Lithuania.

ACM 978-1-4503-3440-2/15/07 ...$15.00.
http://dx.doi.org/10.1145/2729094.2742594.

[1]Note that *interactive lecturing* is distinct from the *traditional lecturing* that has been criticized by studies such as [5]. The former technique involves the deliberate integration of in-class activities, such as think-pair-share, that are designed to avoid many of the problems of the latter.

[2]While we adopted elements of TBL beyond just the RAP, it would be an overstatement to classify our Fall 2013 offering as a TBL course. We are currently redesigning this course to use TBL, and this paper focuses on a transitional step in that direction.

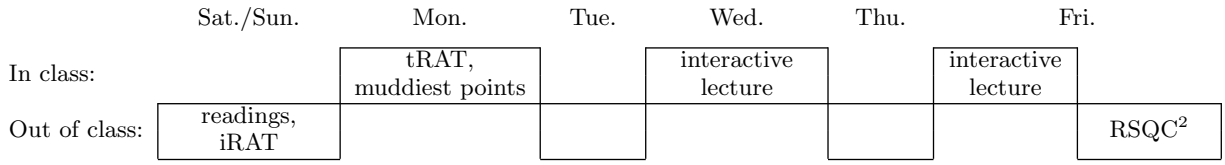


**Figure 1: Castle-top diagram illustrating timing of interventions**

for both students and instructors [11]. In our approach, which we describe in the next section, we combined RSQC$^2$ with the muddiest points activity, where students identify the topic that they found the most confusing.

We introduced these activities into our Fall 2013 offering of our Operating Systems (OS) course (CS 450). We compared the outcomes of this offering (two sections with 52 students total) with our Fall 2012 offering (two sections with 49 students total). Despite demographic similarities in the two offerings, we observed dramatic improvements in student performance, specifically in terms of the withdraw-D-failure (WDF) rate, final exam performance, and course project completion. We performed a number of statistical tests on these factors and found statistically signifcant improvements in the intervention group.

While programming project completion rates also increased in Fall 2013, building a predictive model for completion rates proved difficult. From 2012 to 2013, the students had a statistically significant increase in self-reported C programming experience. As the projects involve modifications to the Pintos kernel [13] and are written exclusively in C, it seems intuitive that the increased experience would contribute to success on these projects. Interestingly, though, we found that C experience had no significant impact on project success once the significant intervention factor was considered. However, statistical tests indicated that the model was weak.

Based on our results, we find compelling evidence that these techniques, which require a very modest effort on the part of the instructor, can play a part in improving student learning outcomes. Informally, we observed that classes after the intervention were more interactive and successful. Statistical analysis supported our observations, as we found significant improvements in the intervention group, while controlling for a variety of demographic factors. The use of an online learning management service, such as Canvas (which we use) or Blackboard, and Immediate Feedback - Assessment Techniques (IF-ATs) can further alleviate the overhead of accommodating these techniques. The result of this combination is a low-cost and effective transitional step toward active learning.

# 2. IMPLEMENTATION

This section describes our specific versions of the interventions in the Fall 2013 offering. Figure 1 shows a *castle-top diagram* that illustrates the timing of these activities during the week. This illustration does not show homework and project work, which students completed on their own outside of class according to their own schedules.

## 2.1 RSQC$^2$ technique

In the traditional approach to the RSQC$^2$, students answer the questions during class time and are given approximately 10 minutes to respond. We found this time commitment to be burdensome, given that our sections were only 50 minutes long. As such, we used a Canvas quiz to conduct the RSQC$^2$. Students had until Friday evening each week to complete the quiz. Furthermore, we modified the phrasing of the questions as follows (the term in brackets was not part of the question given to students and is only intended to map to RSQC$^2$ and metacognition):

1. What one idea from this week did you find most interesting and why? [recall]
2. Explain one concept that you understand more now than you did a week ago. [summarize]
3. What helped you to learn the most this week? [comment, metacognition]
4. Identify one thing from this week that you are confused about in some way. This could be a concept from the lectures and/or book, a requirement of the current project, the relevance of this material, course policies, etc. In addition, try to *explain what you don't understand or why it is confusing.* [question, muddiest point, connect, metacognition]
5. What can *you* do to overcome the confusion you just identified? [metacognition]
6. Is there anything else you would like to add? [comment]

## 2.2 RAP technique

In TBL, the RAP is designed to ensure students have completed required readings prior to class and are prepared to engage in more advanced activities. This activity includes the iRAT and tRAT, both of which are to be conducted during class. In a true TBL course, both of these activities are in-class and graded to ensure accountability for both the individual and the team. In our variation, we moved the iRAT to before class as a weekly online Canvas quiz due each Sunday night. Students could take the iRAT up to 3 times. Immediately after each attempt, students were given their score but Canvas provided no indication of which questions were correct and which were not; consequently, retaking the quiz required students to re-evaluate all of their answers independently. Only the highest score of the 3 attempts was used for grading. Figure 2 shows sample questions that were used on the iRAT/tRAT.

On each Monday, students would retake the quiz as an ungraded tRAT. In the early part of the semester, students reported their responses with a show of hands. The advantage of this approach was that, if different groups came to different conclusions, students had the opportunity to debate and discuss the correct answer. Later in the course, we introduced IF-AT forms, which are pre-printed scratch-off cards. Scratching off the correct response reveals a star,

| | | |
|---|---|---|
| One technique for speeding up thread creation is to automatically create several threads when the process starts. What is the term for this technique?<br>(a) thread pools<br>(b) implicit threading<br>(c) many-to-many model<br>(d) lightweight processes | On the command-line, if I type `ls \| head`, the output produced by running the `"ls"` program is redirected as the input to the `"head"` program. What type of IPC is this?<br>(a) shared memory<br>(b) message passing<br>(c) pipe<br>(d) socket | Which of the following describes the notion of a safe state?<br>(a) Enough resources are available to satisfy the next request<br>(b) All resources can be granted in any order and deadlock is impossible<br>(c) Resources can be granted in some particular order that makes deadlock impossible<br>(d) The system is currently not deadlocked |

**Figure 2: Sample iRAT/tRAT (individual/team Readiness Assessment Test) questions**

while the others are blank. The advantage of the IF-AT approach is that students can discuss the questions in their small groups; this allows students who are reticent to participate in large-group discussions to voice their opinions in a more comfortable setting. However, even with the IF-AT approach, most classes included a large-group discussion to clarify questions at the end of the tRAT.

After completing the tRAT, we then began a discussion of the most common responses to question 4 from the RSQC$^2$. This is similar to the standard TBL approach, which often includes a muddiest point discussion as part of the Clarifying Instruction after the tRAT. The key distinction is that, in standard TBL muddiest point discussions, the instructor can respond to points of confusion that arise during the tRAT, whereas our approach involved collecting this information before class time. The disadvantage of the TBL approach is that fewer voices may be heard (*e.g.,* if some students are shy), and it is difficult to discern larger trends at the time. In our approach, the instructor was able to analyze the questions (which are asked on Friday) to identify fundamental misunderstandings that may be influencing the muddiest points. On the other hand, the disadvantage of our approach is that the students may come up with new questions during the iRAT and tRAT that get missed.

## 2.3 Grading procedures

Grading the iRATs involved no work, as this task was handled automatically by Canvas. For the RSQC$^2$, we graded each submission on a 10-point scale with two points per question (question 6 was ungraded). If the student provided a good faith effort to respond, they would receive full credit. If the student responded to the question with an inadequate response (*e.g.,* providing a nonsense response such as "blargh"), they would receive one point. This low-stakes approach allowed students to respond freely without concern about providing the "correct" answer. The iRATs and RSQC$^2$ activities accounted for 5% each of the total grade for the course.

## 2.4 Additional activities

The primary teaching style for this course consists of interactive lecturing. That is, material is presented in a lecture format that is punctuated with informal application activities, such as in-class exercises, worksheets, and discussions. This style was used in both offerings considered in this paper, so there was no overt attempt to introduce a new pedagogy other than the previously described activities. However, it is our observation that introducing these new techniques made more class time available for in-class activities, as we had to spend less time covering foundational concepts. That is, incorporating the RAP and RSQC$^2$ allowed us to devote more class time to active learning without making a radical change to our pedagogy.

# 3. ANALYSIS

To analyze the possible impact of our intervention on this course, we compared the demographics and outcomes for the Fall 2012 ($N = 49$) and 2013 ($N = 52$) offerings with a number of statistical tests. Note that, due to incomplete questionnaire responses, the histograms and linear regression that we report are based on a subset of this data ($N = 44$ and $N = 43$ for Fall 2012 and 2013, respectively).

## 3.1 Demographics

In our analysis, we identified three demographic factors that required careful consideration. The first difference was an increase in the amount of self-reported C programming experience, which was a consequence of a change made in the prerequisite course. This was measured on a 4-point Likert scale (1 = "No experience," 2 = "Very limited experience," 3 = "Written several small programs (less than 100 lines)," and 4 = "Written large programs"). Figure 3 shows the distribution of responses. For reasons we will describe in Section 3.3, we combined the 1 and 2 responses, as well as the 3 and 4 responses. After combining the groups, a $\chi^2$ test showed that there was a significant difference in C programming experience between the two offerings ($p = 0.001214$).

Next, there was a significant difference in the number of students who completed the prerequisite (CS 350) as a community college equivalent. The number of students who took the equivalent course decreased from 17 in Fall 2012 to 8 in 2013. A $\chi^2$ test yielded $p = 0.04374$.

The third demographic factor that we considered was the students' self-reported[3] grades (based on a 4.0 GPA scale) in the prerequisite. While teaching the course, we felt that the students were fairly comparable and neither group was clearly better than the other. To validate this intuition, we compared the proportions of students who earned either As or Bs in the prerequisite. In Fall 2012, this ratio was 23/44 whereas the ratio for Fall 2013 was 29/43. A $\chi^2$ test yielded $p = 0.221$, indicating no significant difference between the overall quality of students in the two offerings.

## 3.2 Impact on course grades

Despite having performed similarly in the CS 350 prerequisite, the students in the intervention group (Fall 2013)

[3]The decision to use self-reported grades, rather than accessing student records, was made after consultation with our Institutional Review Board (IRB).

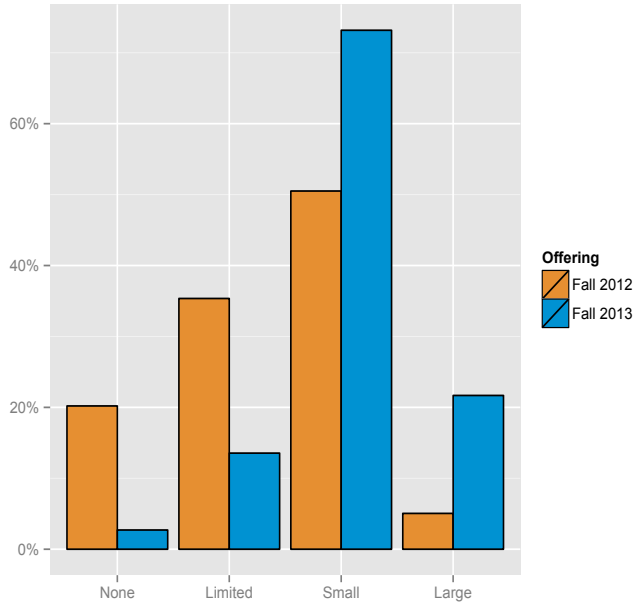


**Figure 3: Self-reported C programming experience**

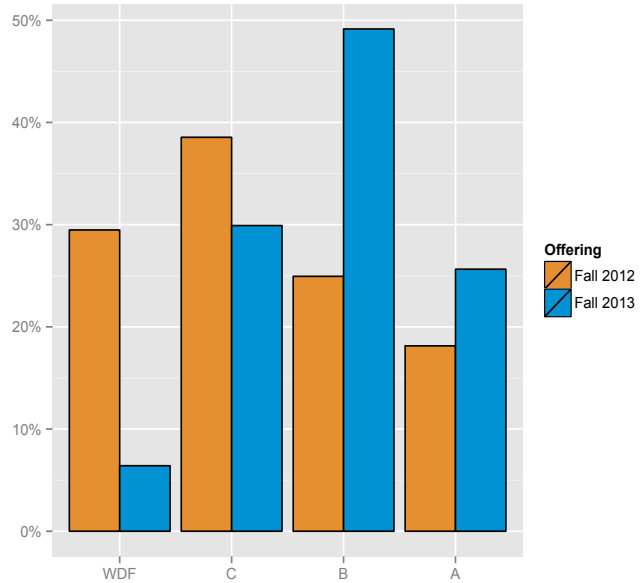


**Figure 4: Fall 2012 vs. Fall 2013 CS 450 grade distribution**

performed significantly better in CS 450 than those in the control group. Figure 4 shows the grade distributions for the two offerings. The WDF rate improved from 13/49 in Fall 2012 to 3/52 in Fall 2013; a one-sided $\chi^2$ test yielded $p = 0.004893$, indicating significance. Similarly, if we look at the students who did well in CS 450 (those earning As and Bs, and only considering the subset of students who responded), we see that the ratio of As and Bs improved from 17/44 in Fall 2012 to 35/43 in Fall 2013, which yields $p = 0.0000596$ in a one-sided $\chi^2$ test. Thus, the intervention group showed a significant improvement in course grades.

In addition to overall course grades, we also observed a significant improvement on programming projects. For instance, both offerings were required to complete (working in groups) portions of the Pintos OS projects [13]. In these courses, the `userprog` project was completed at the end of the semester. The average number of test cases (only a subset were required) that the teams passed increased from 28/50 to 40/50. A $t$ test yields $p = 0.02391$, thus indicating significance. As these were team projects, we attempted to perform a linear regression that modeled the number of test cases passed as a function of the team's average C programming experience and the intervention. The only factor found to be significant was the intervention ($p = 0.0678$), but the adjusted $R^2$ value was 0.1535 (overall $p = 0.09181$ for the model), indicating the model produced by the regression is rather weak. However, what is interesting about this regression is that teams' average C programming experience is **not** a significant predictor of test case success, regardless of whether or not the intervention was present. This finding is counterintuitive, as the Pintos projects are all written in C, but it is welcome: *It suggests that teams with less C experience in both groups were able to catch up with those with more experience by the end of the semester.*

## 3.3 Impact on final exam performance

As noted above, our intervention group (Fall 2013) demonstrated significant improvements in the WDF rate, overall course grade, and programming projects. We also observed that students performed better on the exams, as well. To analyze exam performance in more detail, we performed a multiple linear regression analysis aimed at describing final exam scores as a function of demographics (C programming experience, community college equivalency of the prerequisite, and prerequisite grade) and our intervention. Linear relationships between final exam scores and each of the demographic and intervention variables were reasonable, and residual diagnostics for the final model (described below) revealed no issues with model form or assumptions. We found the community college equivalency factor to be insignificant with the other factors present, and removed it based on ANOVA results ($p = 0.4958$).

Recall that the C programming factor was self-reported Likert-scale data. While we initially considered all four Likert responses as factors in the regression, we found the increase to the third level, "Written several small programs," relative to having "No experience," to be the only significant factor ($p = 0.0551$). Consequently, we found it appropriate to group the responses into two categories: those with limited or no experience, versus those with at least some experience with C programming.

Table 1 shows our model, which is significant with an overall $p = 1.153 * 10^{-12}$ and $R^2 = 0.5114$, indicating that this model captures the behavior well. According to this model, the y-intercept for students in the intervention group was 24 points higher on the final exam when all other factors are held constant. Furthermore, there was a significant interaction between the prerequisite grade and the intervention, which changes the slope of the line for that class. The result is that the weakest students (those who scored a D in the prerequisite) in the intervention group benefited the most. At the same time, though, the strongest students in the intervention group *also* benefited but to a lesser extent.

Figure 5 illustrates the model in graphical form. In this figure, the dotted lines indicate the model for the students who had limited or no C programming experience, whereas the solid lines indicate those who came into CS 450 with prior C experience. In all cases, holding prerequisite grade

| Factor | Estimate | Pr(>\|t\|) |
|---|---|---|
| (Intercept) | 37.530 | $1.6 * 10^{-12}$ |
| C programming | 8.562 | 0.000961 |
| Prerequisite grade | 7.658 | $2.9 * 10^{-7}$ |
| Intervention | 24.228 | 0.001015 |
| Prerequisite grade:Intervention | -4.301 | 0.076666 |

**Table 1: Final linear regression model of final exam scores in two offerings of CS 450**

and C experience constant, students in the intervention group did significantly better than those in the control.

We emphasize that this analysis should be taken as illustration, rather than generalizable quantitative evidence. There were differences between the exams that our statistical analysis does not consider. For instance, there were some questions asked on the Fall 2012 final that appeared on the Fall 2013 midterm instead. Similarly, some questions from the Fall 2012 midterm were added to the Fall 2013 final. Consequently, our exam cannot be considered a valid instrument and we acknowledge the possibility that the exam differences were responsible for some of the improvement. To counter this objection, we observed improved performance on the parts of the exam that were similar, and students also did better on the midterm, as well. Thus, we find enough corroborating evidence that suggests our regression model is consistent with student performance, and students in the intervention group did significantly better than the control.

## 3.4 Limitation of interpretations

To be clear, our analysis presented here shows that students in our intervention group performed better in a variety of measurements even while considering the impact of certain demographic factors. There are many other factors that could have played a role that were not considered. For instance, our analysis did not consider the possible effect of having different instructors for CS 350. Also, the Fall 2013 students may have selected either into or out of our sections of CS 450 based on what they heard from the Fall 2012 students. This selection bias could have included differences in motivations between the groups. As another example, the increased number of students who completed the prerequisite in our department may have led to a shared cultural identity that helped the Fall 2013 students to work together more effectively. Our statistical model does not capture either of these affective characteristics. Thus, we do not assert a causal link between these techniques and the improved outcomes. Rather, we reiterate the point that these techniques allowed us to increase active learning opportunities in the classroom, and the statistical findings corroborate our personal observations that the collective whole of the Fall 2013 experience led to improve outcomes.

# 4. DISCUSSION

Overall, we feel that there is a significant amount of anecdotal evidence and empirical results to suggest that these techniques were beneficial and contributed to improved student outcomes in the Fall 2013 offering. In this section, we will discuss additional points to consider when implementing these techniques in future courses.

On the RSQC$^2$ variation, questions 4 and 5 proved problematic and we abandoned this structure in later courses, returning to a more traditional RSQC$^2$ format. The problem with question 4 was its attempt to serve multiple purposes. Some students responded with specific questions (*i.e.,* muddiest points) while others focused on how the material related to other topics (*i.e.,* connection). Question 5 was designed to complement in-class discussions of metacogniition that introduced students to the idea of reflecting on how they learn. In class, they were provided guidance to think about what other resources (*e.g.,* Wikipedia) they would consider using. Despite this guidance, we found student responses to this question quite varied and unhelpful, and we have omitted it in subsequent courses.

In more recent semesters, we have also changed the grading system used for the RSQC$^2$. Rather than awarding two points per question, we now award a single point for each, regardless of the quality of the students' responses. We had initially feared that this approach, which rewards insightful comments and gibberish equally, would lead to a reduction in the quality of responses. In practice, we have no evidence that this fear is warranted and the overall quality of students' responses has stayed at the same level as previously.

Responses to the RSQC$^2$ questions provide valuable feedback to the instructor regarding student progress. However, in a subsequent course[4], we moved the muddiest point discussion from class time to the class discussion forum on Canvas, and we shifted the responsibility of answering the questions to the students. In this approach, we enumerated the questions, paraphrasing as necessary, in a weekly discussion forum post that was published on Saturdays. Students then collectively provided answers in the style of a Wiki, with the stipulation that each student could only provide the initial answer for a single question; once a question was answered, anyone could edit. The advantage of this approach is that it provides students with a valuable opportunity for peer teaching and formative assessment by allowing them to answer a question with no grade repercussions. In almost all cases, students were able to come up with a correct answer with minimal feedback from the instructor. This variation also reduced the instructional overhead, as we no longer needed to prepare specific responses.

While we observed significant improvements in measurable student outcomes, these techniques also produced a number of intangible improvements, as well. The introduction of the RAP and muddiest points, in particular, had an influence on the flow of classroom activities. By reducing the need to cover basic definitions and fundamental concepts in class, more time was available to devote to the exercises and in-depth discussions. Furthermore, by starting the class with an activitiy that required discussion, students were primed to be engaged with the class that day. We hypothesize that this engagement may have had more influence than the actual pre-class preparation, but we made no attempt to distinguish between the preparation and the engagement components, and we do not see an easy way to do so. However, we do not see a need to at this point, as the primary goal was to find a way to bring active learning into our systems courses.

[4]Our analysis and results sections focus on the outcomes in the Fall 2013 OS course. We introduced this additional variation in Spring 2014 in a different course, so our analysis does not consider its impact on student outcomes. Anecdotally, we found this variation beneficial and include it here for other instructors to consider as they see fit.

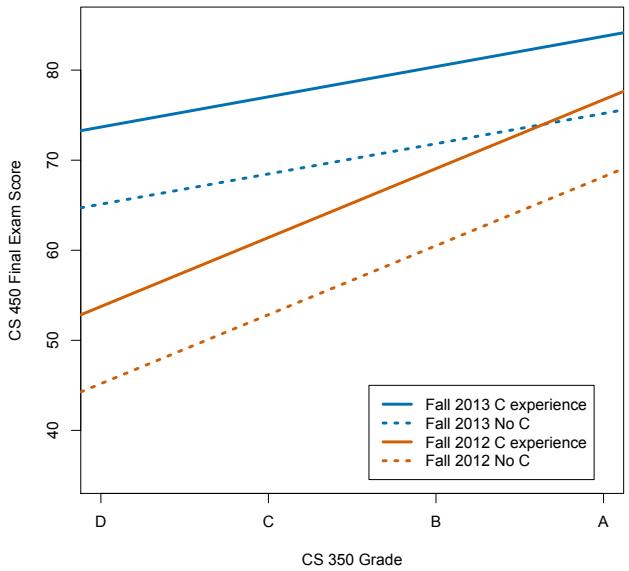


**Figure 5: Visual representation of the linear regression model of final exam score**

Finally, we also need to address a commonly cited concern about switching to active learning techniques: its impact on coverage. That is, instructors often worry that adopting a new pedagogy will interfere with their ability to cover all required material. In this instance, we do not feel that such a concern is warranted. After doing an informal retrospective mapping to the ACM 2013 curriculum recommendations, we found no differences between coverage of Core Tier-1 and Core Tier-2 topics. Any differences between coverage of Elective topics was negligible. Consequently, we find that adopting these techniques will not have any measurable impact on coverage.

## 5. CONCLUSION AND FUTURE WORK

In Fall 2013, we introduced two new techniques into our OS course in an attempt to increase the amount of active learning that happened in the classroom. These two techniques, the RAP and the RSQC$^2$, require a very modest amount of effort on the part of the instructor, but initial evidence indicates that they can be powerful first steps toward improving student learning. Our statistical analysis, which compares the Fall 2012 and Fall 2013 offerings of OS, shows that there were statistically significant improvements in a variety of measurements, even when demographic differences were considered. Thus, for any instructor contemplating a switch to an active learning pedagogy, we recommend these two techniques as a low-cost transitional step.

## 6. REFERENCES


[1] T. A. Angelo and K. P. Cross. *Classroom Assessment Techniques: A Handbook for Faculty*. Jossey-Bass, San Francisco, CA, 1993.

[2] J. Campbell, D. Horton, M. Craig, and P. Gries. Evaluating an inverted CS1. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, SIGCSE '14, pages 307–312, New York, NY, USA, 2014. ACM.

[3] J. Davis. Experiences with just-in-time teaching in systems and design courses. In *Proceedings of the 40th ACM Technical Symposium on Computer Science Education*, SIGCSE '09, pages 71–75, New York, NY, USA, 2009. ACM.

[4] R. M. Felder and R. Brent. Active learning: An introduction. *ASQ Higher Education Brief*, 2(4), August 2009.

[5] S. Freeman, S. L. Eddy, M. McDonough, M. K. Smith, N. Okoroafor, H. Jordt, and M. P. Wenderoth. Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences*, 2014.

[6] R. R. Hake. Interactive-engagement vs. traditional methods: A six-thousand-student survey of mechanics test data for introductory physics courses. *American Journal of Physics*, 66(1):64–74, 1998.

[7] S. Horwitz, S. H. Rodger, M. Biggers, D. Binkley, C. K. Frantz, D. Gundermann, S. Hambrusch, S. Huss-Lederman, E. Munson, B. Ryder, and M. Sweat. Using peer-led team learning to increase participation and success of under-represented groups in introductory computer science. In *Proceedings of the 40th ACM Technical Symposium on Computer Science Education*, SIGCSE '09, pages 163–167, New York, NY, USA, 2009. ACM.

[8] H. H. Hu and T. D. Shepherd. Using POGIL to help students learn to program. *Trans. Comput. Educ.*, 13(3):13:1–13:23, Aug. 2013.

[9] P. Lasserre and C. Szostak. Effects of team-based learning on a CS1 course. In *Proceedings of the 16th Annual Joint Conference on Innovation and Technology in Computer Science Education*, ITiCSE '11, pages 133–137, New York, NY, USA, 2011. ACM.

[10] C. B. Lee, S. Garcia, and L. Porter. Can peer instruction be effective in upper-division computer science courses? *Trans. Comput. Educ.*, 13(3):12:1–12:22, Aug. 2013.

[11] M. Mani and Q. Mazumder. Incorporating metacognition into learning. In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*, SIGCSE '13, pages 53–58, New York, NY, USA, 2013. ACM.

[12] L. K. Michaelsen, A. B. Knight, and L. D. Fink. *Team-Based Learning: A Transformative Use of Small Groups in College Teaching*. Stylus Publishing, Sterling, VA, 2002.

[13] B. Pfaff, A. Romano, and G. Back. The Pintos instructional operating system kernel. In *Proceedings of the 40th ACM Technical Symposium on Computer Science Education*, SIGCSE '09, pages 453–457, New York, NY, USA, 2009. ACM.

[14] L. Porter, C. Bailey Lee, and B. Simon. Halving fail rates using peer instruction: A study of four computer science courses. In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*, SIGCSE '13, pages 177–182, New York, 2013. ACM.

[15] M. Prince. Does active learning work? a review of the research. *Journal of Engineering Education*, 93(3), July 2004.

[16] M. K. Smith, W. B. Wood, W. K. Adams, C. Wieman, J. K. Knight, N. Guild, and T. T. Su. Why peer discussion improves student performance on in-class concept questions. *Science*, 323, January 2009.