ISSN: 2473-3857
v2 n4066

How Much Time Do Students Spend On Programming Assignments? A Case for Self Reporting Completion Times

Mark Segall
segall@msudenver.edu
Department of Computer Information Systems
Metropolitan State University of Denver
Denver, CO, 80217

Abstract

Professional program development with Agile methods require the team to estimate how much work can be completed in a given period of time. This paper reports on asking students in a Computer Information Systems programming course to self-report the time it took to complete weekly programming exercises and to rate the difficulty of the assignments. This can develop a self-awareness about the time it takes to complete tasks. In addition, it provides feedback to the instructor about the difficulty of the various assignments, plus the data from previous semesters can be provided to students to allow them to plan for future assignments that require more effort. There was a wide range of time spent on assignments. The average (± SD) time spent on each exercise ranged from a low of 41 (± 44) to a high of 239 (± 229) minutes. The average time each student spent on all assignments was 119 (± 64) minutes, ranging from a low average of 18 minutes to a max average of 307 minutes per assignment. Statistical analysis did not show that the time spent completing exercises could predict performance in the course. A component analysis with the time spent, difficulty rating and course average as three variables indicated three clusters of students that took about 80, 160, and 240 minutes on average.

Keywords: Pedagogy, Programming, Homework Time, Agile

1. Introduction

In Computer Information Systems (CIS) courses we have students complete exercises to practice the skills being taught. These are often formative assessments (Taras, 2005); students will get grades for the assignment, but also feedback about how they have done to improve their understanding of the material.

In a CIS programming course weekly homework exercises can take a lot of time to complete. Novice students are learning programming concepts, a specific programming language, and often an integrated development environment (IDE), while students with experience in other languages might only be learning the syntax of a new language and the IDE. In addition to creating the program students also learn to deal with the errors, testing and debugging that inevitably occur when creating a computer program.

Given the demands on a student to manage their academic studies, work and family activities it is important to understand the time students spend completing assignments for classes they are taking. This paper reports on the collection of student self-reports on how much time was spent completing weekly exercises in a junior level CIS programming class. This practice of requesting student self-reports will be discussed in how it can be used to judge the difficulty of the programming assignments, give feedback to the instructor about the student's experience, and promote project management skills.

ISSN: 2473-3857
v2 n4066

2. Background

The course Business Application Development with Java (CIS 3145) is a required for all CIS majors in the College of Business at XYZ University. All sections of the course turn in assignments through the Blackboard Learn course management system. The Blackboard Learn system has a discussion forum for posting questions about assignments labeled "Problem Solving Board". This is the description for this board on the website.

"This is the place to discuss problems and errors in your programs. Make sure to clearly, and with detail, describe the error messages you see. When responding to questions, make sure you do not provide others with the code you are using. A good strategy is to refer to the appropriate parts of the textbook or help sections that answer the question."

There are 14 weekly exercises (each worth about 3% of the total grade) from the textbook (Murach, 2011), along with 4 projects, and two exams. The exercises give students a project that must be modified to complete the project. Students turn in a zip file of the complete NetBeans project folder on Blackboard Learn. Each assignment has the following instructions:

"Report the time (in minutes) it took to complete the project. Rate the difficulty of the project on an ordinal scale as either: "Easy", "Moderate", "Hard", or "Challenging". Choose just one of these four string values."

Most students simply reported the minutes and rating, but some did not report either the minutes or rating. A few students would write more details. This is an example of a student reporting one of the longest times spent on an exercise.

"This assignment took me over a week and half to complete, as I simply was unable to make heads or tails of it before the exam. After reviewing the information for the exam, I was better able to tackle this project but ran in to a few issues towards the end I could not figure out for the multiple output piece. I would rate this as "Challenging." Total time spent approx. = 20 hours = 1200 minutes."

3. Results

By Exercise

The data was collected for two semesters (Fall 2015 and Spring 2016). Two sections of the course were taught each semester with a total of 57 students. All sections were taught by the same instructor. Appendix A shows a summary of the average, median, minimum and maximum time spent for each of the 14 exercises, over the two semesters. All calculations were done with Minitab 17.

For the Fall semester the lowest average of time spent was 44 minutes for first exercise turned in for chapter 2. Even this first assignment had a wide range of reported times with a minimum of 5 and a maximum of 300 minutes. The Median of 30 minutes, which is lower than the average, indicates that the average was skewed to the high end. The spring semester also showed the lowest average time for the first exercise with a mean of 39 minutes and a median of 30. The range of times was not as large (5 to 120 minutes).

Exercise #2.2

	Avg	Median	Min	Max
Fall 2015	44	30	5	300
Spring 2016	39	30	5	120

The chapter 9 exercise on object interfaces was the longest in both semesters with an average and median of 268 and 185 minutes respectively for the fall semester, and 230 and 180 for the spring semester. The minimum and maximum for this assignment was 60 and 840 minutes for the fall, and 20 and 1200 minutes for the spring.

Exercise #9.1

	Avg	Median	Min	Max
Fall 2015	268	185	60	840
Spring 2016	230	180	520	1200

The two semesters show consistent average times per exercise in Appendix A. The Pearson correlation between the average time per exercise for spring and fall is 0.898. This is an indication that the semesters can be combined for further analysis.

The percent of each of the four difficulty ratings for each exercise is also displayed in Appendix A.

In the Fall of 2015 four exercises had the highest number of students rating the exercise as Easy (44% to 67%). Five of the exercises had the

ISSN: 2473-3857
v2 n4066

highest number of Moderate ratings (38% to 45%), while four exercises had the most Hard ratings (40% to 64%). Only exercise 9.1 had 50% Challenging ratings, which is also the exercise with the largest average time to completion.

In the Spring 2016 semester the same exercises had the highest percentage for the Easy and Challenging ratings. However, the four Hard rated exercises shifted to the Moderate category in the spring.

To summarize the results, the data was combined for the two semesters while keeping the plurality rating from the fall semester. Appendix 2 shows the data sorted by the average time to complete the exercises. The Easy and Challenging exercise clearly take the least and most time respectively. The Moderately rated exercises tended to take less time on average but overlapped with the Hard rated exercises.

Using the Rating as a categorical variable the table below shows the average and median times to complete an exercise for all student ratings.

Rating Category	Median	Avg.	StDev
Easy	45	61	55
Moderate	90	117	96
Hard	120	156	156
Challenging	180	239	229

The Easy exercises took about an hour on average for students to complete, Moderate exercises took about 2 hours, Hard about 2.5 hours and the challenging exercise was 4 hours. The median times where approximately 15, 30, 30, and 60 minutes less than the average times for the Easy, Moderate, Hard, and Challenging exercises respectively.

By Student

If some students need more time to complete an assignment, it may mean they will not do as well as other students in the course, in which case time to complete an assignment could be used as a risk factor to predict which students need extra assistance. A regression equation was used to see if the average time that a student spent on exercises could predict the final course grade. The regression equation, displayed as a fitted line plot in Appendix C, was not significant (F = 0.14, P =0.674).

Course Grade = 81.17 - 0.00841 Avg. Minutes

In order to perform an exploratory cluster analysis, the difficulty ratings were converted to a numeric scale of 1 for Easy, 2 for Moderate, 3 for Hard, and 4 for Challenging. Using the three variables of the average minutes to complete all exercises, the average difficulty rating, and the course average for the student a cluster of observations (students) was the cluster analysis was performed at a 50% similarity level. Three clusters were created. See Appendix D for a bubble plot of the clusters.

Variable	Cluster1	Cluster2	Cluster3
Avg. Minutes	156.1	83.3	239.9
Rating Avg.	2.5	2.0	2.6
Course Grade	72.8	81.7	83.0
Cluster Size	11	39	8

The three clusters of students showed that the quickest group was cluster 2 which took about one hour and 20 minutes to complete assignments. This group was 67% of the students and their overall difficulty rating as Moderate (2) with a course average of about 82%. Cluster 3 spent an average about three times longer (about 4 hours) than cluster 2 to complete the exercises, and while they also rated the exercises harder (2.6) their course average was a little more than 1% larger. Cluster 1 was between the other two spending an average of 2.5 hours to complete the exercises and rating the exercises between moderate and hard, similar to cluster 3. Interestingly cluster 1's overall average was about 10 points less than the other two groups.

4. Discussion

By Exercise

The time it takes a student to complete assignments is related to their subjective assessment of its difficulty. Easy assignments take an hour or less to complete while moderately rated assignments take 1.5 to 2 hours to complete. Challenging exercises take three hours or more. This technique can be useful when creating a new course or adopting a new textbook to ensure the exercises are an optimal level of difficulty.

Collecting the time that students report it takes to complete a homework exercise can give instructors insight into the difficulty that students have with different assignments. After collecting the data for the Fall semester changes were made to the online notes provided to all students in an attempt to clarify possible misunderstanding on the material. It was encouraging to see that the

ISSN: 2473-3857
v2 n4066

exercises that fell under the Hard ratings shifted to moderate and that the percentage of those rating exercise 9.1 as Hard or Challenging dropped from 92% to 76%. If interventions are effective they could result in an improvement in the reported times and ratings.

By Student

When initially collecting this data it was hoped that the variability in the time to complete assignments could provide insight into how students perform in a programming class. This was not the case. As noted in a paper by Nonis and Hudson (2010) the total time spent studying is not a consistent predictor of academic performance. However, they did find that study habits, such as the ability to concentrate, can interact with the time spent studying to impact academic performance. Given that one cluster of students that took three times as long to complete exercises performed overall as well as another cluster could mean that some students are more efficient at studying, but both groups are equally effective.

It is obvious that there are different types of students. Robins, Roundtree, & Rountree (2003) review the notion that there are different kinds of novice learners. For example, Perkins, Hancock, Hobbs, Martin, & Simmons (1989) suggest that novice programmers can be categorized as *stoppers* who easily give up when they encounter a problem and *movers* who work through problems when they encounter them. This categorization of stoppers and movers could be related to the fixed versus growth mindset proposed by Dweck (2006) where students who believe that they are not capable of learning are more likely to give up on a task versus those who believe that they can learn if they work harder. These distinctions could be the basis for the cluster of students that score a grade lower than the other two clusters. They might not have the motivation or time to work harder on the programming exercises.

Project Management Considerations

Asking students to report the time it takes to complete a project supports project management skills. In Agile methods teams start each sprint estimating how many story points can be accomplished (Moreira, 2013). Being self-aware of the time it takes to write code will put students on the path to this important skill.

A future exercise that can be incorporated into a programming course is to also ask students to estimate the time it will take to complete the exercise before they start the exercise. Then when they record the actual time the difference can be used to adjust the estimates for the next exercise. This activity would introduce to students the Agile principle of "At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly" (Beck et. al. 2001). This could be a short paper or ongoing online journal.

Student Benefit

An additional use of this data is to provide students at the beginning of the semester a summary of the time completion data from previous semesters. Student would have the chance to prepare for exercises that typically take longer to complete. Taking some uncertainty out of completing the exercises could reduce anxiety about the projects.

5. CONCLUSIONS

The practice of asking students to report the time it takes to complete an assignment, as well as subjectively rate the difficulty of an assignment can benefit both instructors and students. Instructors can learn about topics that might be harder than expected and students can gain experience in skills useful for project management. As is, this data cannot help predict student performance. Further research can be done to see if the time to complete assignments, in addition to other factors, will provide insights to improving the teaching of computer programming.

6. REFERENCES

Beck, K., Beedle, M., Bennekum, A. V., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J. And Thomas, D. (2001). Manifesto for Agile Software Development. Retrieved July 31, 2016 from http://www.agilemanifesto.org/principles.html

Dweck, C.S. (2006). Mindset: The new psychology of success. New York: Random House.

Moreira, M.E. (2013). Being Agile Your Roadmap to Successful Adoption of Agile. Apress, Berkely.

Murach, J. (2011). Murach's Java Programming. Mike Murach & Associates, Fresno.

ISSN: 2473-3857
v2 n4066

Nonis, S.A, & Hudson, G.I. (2010) Performance of College Students: Impact of Study Time and Study Habits. *Journal of Education for Business*, 85(4), 229-238.

Perkins, D.N, Hancock, C., Hobbs, Martin, F., & Simmons, R. (1989). Conditions of learning in novice programmers. In E.Soloway & J.C. Sphorer (Eds.), *Studying the novice programmer* (pp. 261-279). Hillsdale, NJ: Lawrence Erlbaum.

Robins, A., Roundtree, J., & Roundtree, N. (2003). Learning and Teaching Programming: A Review and Discussion*,* 13(2), 137-172.

Taras, M. (2005). Assessment – summative and formative – some theoretical reflections. *British Journal of Educational Studies,* 53(4), 466-478.

ISSN: 2473-3857
v2 n4066

Appendix A

Summary of Exercise Time Estimates from Previous Semesters

Fall 2015

Exercise	Exer. #	Avg	Median	Min	Max	Percent reporting each rating / exercise: Easy	Moderate	Hard	Challenging
While loop	**2.2**	44	30	5	300	67%	29%	0%	5%
Data manipulation	**3.2**	69	57.5	10	180	31%	38%	23%	8%
Data Validation	**5.2**	145	110	25	600	14%	21%	57%	7%
Classes	**7.2**	72	35	7	240	56%	25%	13%	6%
Inheritance	**8.2**	60	50	10	180	44%	39%	17%	0%
interfaces	**9.1**	268	185	60	840	8%	0%	42%	50%
Dates	**13.2**	123	120	10	390	17%	39%	28%	17%
Arrays	**11.2**	97	105	10	200	19%	44%	25%	13%
GUI	**15.1**	147	120	15	480	25%	45%	20%	10%
GUI List Model	**16.1**	185	140	30	720	20%	20%	40%	20%
Applets	**17.1**	119	105	15	240	36%	21%	43%	0%
Text I/O	**18.1**	124	120	15	240	17%	42%	33%	8%
Derby DBMS	**20.1**	71	50	15	300	57%	36%	0%	7%
JDBC	**21.1**	200	200	60	390	21%	0%	64%	14%

Spring 2016

Exercise	Exer. #	Avg	Median	Min	Max	Easy	Moderate	Hard	Challenging
While loop	**2.2**	39	30	5	120	53%	41%	6%	0%
Data manipulation	**3.2**	83	90	10	300	24%	41%	24%	12%
Data Validation	**5.2**	106	90	15	360	15%	39%	24%	21%
Classes	**7.2**	64	45	10	240	44%	34%	16%	6%
Inheritance	**8.2**	71	60	8	300	43%	43%	11%	3%
interfaces	**9.1**	230	180	20	1200	3%	21%	35%	41%
Dates	**13.2**	129	120	20	360	13%	50%	16%	22%
Arrays	**11.2**	159	110	15	540	16%	50%	22%	13%
GUI	**15.1**	117	110	20	360	25%	59%	9%	6%
GUI List Model	**16.1**	204	120	20	1600	13%	50%	21%	17%
Applets	**17.1**	142	120	30	300	10%	52%	14%	24%
Text I/O	**18.1**	108	90	20	480	24%	55%	10%	10%
Derby DBMS	**20.1**	71	60	15	240	65%	23%	8%	4%
JDBC	**21.1**	176	165	30	600	0%	50%	33%	17%

ISSN: 2473-3857
v2 n4066

Appendix B

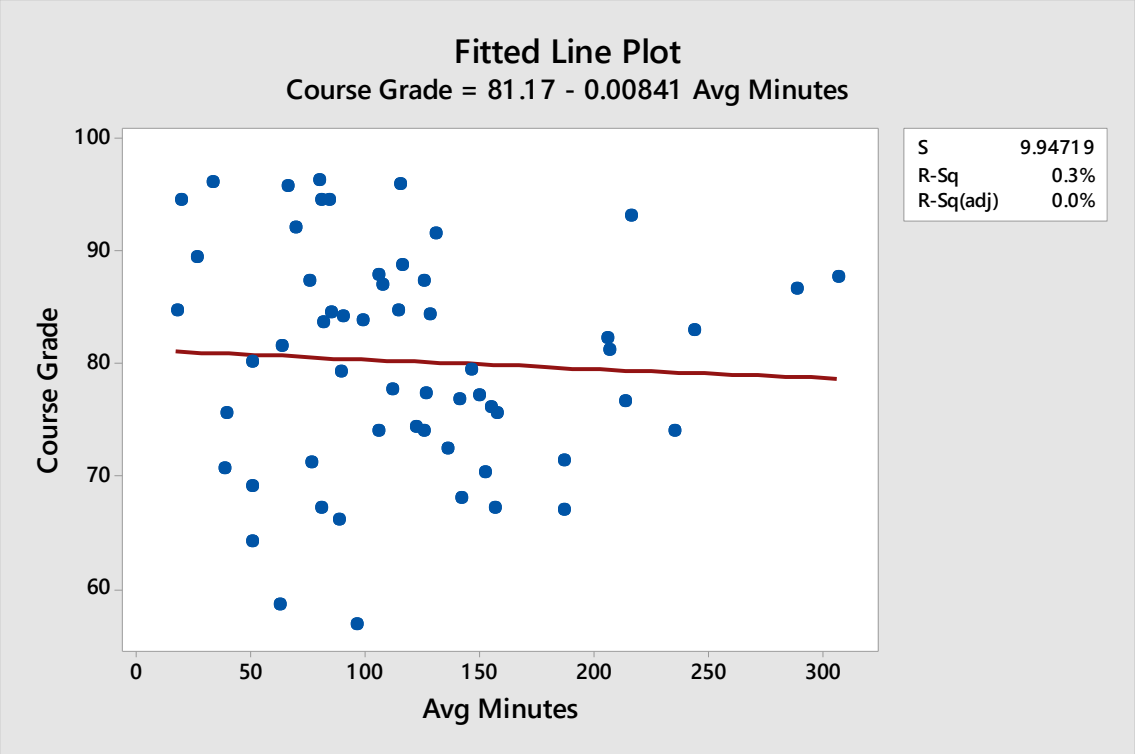
Summary of Exercise Time Estimates Combined & Sorted by Average

Exercise	Exer. #	Avg	StDev	Min	Median	Max	n
While loop	2.2	40.8	43.9	5	30	300	54
Classes	7.2	66.2	59.8	7	45	240	47
Inheritance	8.2	67.8	51.3	8	60	300	48
Derby DBMS	20.1	71.1	62.3	15	60	300	46
Data manipulation	3.2	78.8	56.9	10	60	300	54
Text I/O	18.1	112.4	83.8	15	90	480	43
Data Validation	5.2	118.0	102.4	15	90	600	49
GUI	15.1	127.2	95.8	15	120	480	47
Dates	13.2	127.3	87.3	10	120	390	49
Applets	17.1	134.5	71.6	15	120	300	37
Arrays	11.2	139.5	133.3	10	105	540	44
JDBC	21.1	183.0	115.4	30	180	600	42
GUI List Model	16.1	196.9	273.8	20	120	1600	41
interfaces	9.1	238.8	229.2	20	180	1200	44

Appendix C

Regression for Exercise Time Versus Overall Course Grade

ISSN: 2473-3857
v2 n4066



Appendix D

Cluster Analysis of Observations (Students)