A Study of Students' Progress Through Introductory Computer Science Programming Courses

Jaime Raigoza
Computer Science Department
California State University, Chico
Chico, USA

Abstract— **Given the growing demand for skilled workers from the Computer Science field, the high attrition rate of entering Computer Science students is a serious problem at most universities. Much research exists on evaluating the reasons of failure with the introductory Computer Science curriculum referred to as the CS1, CS2 and CS3 courses. Current research has proposed methods to predict patterns and characteristics to offer early detection of students likely to fail. The problem is difficult to understand due to the existence of many possible reasons students drop-out from the computer science curriculum. Factors such as student transfers, course diversity, and students repeating failed courses are seldom considered.**

The goal of this paper is to extend the understanding of the attrition rates for entering Computer Science students by analyzing the progress of student success through 10 years of course data. The impact of transfer students is considered as well as the frequency students repeat the CS1, CS2, and CS3 courses, and their success. Analysis is done by following the students up until graduation. An analysis is also made to determine how courses tend to predict the graduation success rate.

Keywords- Computer Science; student attrition; course curriculum; student success; transfer students

I. INTRODUCTION

Of the 400+ students who took the first Computer Science introductory programming course at California State University (CSU), Chico within the 3-year period between the Fall 2006 and Spring 2009 semesters, only 41 students graduated from the Computer Science Department by the Spring 2016 semester. As alarming as this statistic is, a similar graduation rate was found in [1]. This high attrition rate is a serious and unfortunately, a common problem at many universities. Much interest exists in research to tackle this problem due to the growing demand of STEM-related jobs. The introductory Computer Science curriculum at most universities typically consists of courses which are commonly referred to as the CS1, CS2 and CS3 course series. These names will be adopted in this paper.

The high rate of failure has been attributed to the student's lack of aptitude, misconceptions on what computer science is, advising and a multitude of other reasons [1]. Research work in this area has also examined methods to predict patterns or characteristics that can offer early detection of student success or failure. The many well-founded reasons for failure makes it difficult to solve the attrition rate problem. A common approach to addressing this problem has been to determine methods that predict a student's success/failure, perform intervention methods, and make curriculum changes [5, 10]. The authors in [11] performed a systematic research work review of 13 different teaching approaches and determined that educators have made positive changes but, the differences in effectiveness tend not to be statistically significant.

The goal of this paper is to first study the progress of student success through 10 years of course data. The work is unique such that the results are based on tracking and studying several characteristics of a large cohort of students as they progress through the computer science curriculum up until graduation. The attrition rate and the impact of transfer students' enrollment is analyzed. Furthermore, since students can repeat failed courses, the repetition distribution and success rate is measured for the CS1, CS2, and CS3 courses. An analysis is also made between several courses to determine which one better predicts the success rate to graduation.

II. RELATED WORK

Being able to program a computer is a fundamental skill which explains why teaching institutions often begin their Computer Science curriculum with a programming course. Based on an extensive study of 63 teaching institutions, the authors in [2] determined a failure rate of 33% for students who take the CS1 programming course. One of the major reasons of student failure is commonly attributed to a lack of good study habits which typically leads to poor project management. Students very often do not properly estimate the amount of time it will take to successfully complete their software assignment. The student may also prematurely start coding without a true understanding of the problem they are attempting to solve, and they may have not completed a proper analysis and design. Students eventually become frustrated by trying to force their entire set of code for a program to run without having built and tested it gradually in a modular fashion. The outcome is unfortunately, failure to get their code to compile and/or execute.

Students also have a common misconception of what Computer Science entails by not realizing it requires logic and math skills to develop algorithms that solve problems. It is not enough to be able to query the Internet, use applications and play games [1]. The results of a study done in [3] indicate that students with a high aptitude in mathematics tend to achieve a good programming ability since both require similar cognitive skills. Many entering computer science students do not have the skill to develop a procedural algorithm irrespective of the

978-1-5090-5920-1/17/$31.00 ©2017 IEEE

syntax. The authors in [1] suggest that extra help with problem solving, such as by providing a course in introductory problem solving would lead to higher retention rates in the computer science curriculum, *problem solving is a skill that can be learned*. The work in [8] suggests that by pairing students which is commonly used as an agile method in industry, the students' retention rates improves without negatively impacting the students' final exam scores.

Not only is understanding the possible reasons for student failure a start in addressing the high rate of attrition but, also being able to predict success and failure is of benefit. An extensive study in [12] across 51 institutions revealed that the class size is a significant predicter such that a higher CS1 pass rate is achieved given smaller classes. The authors in [4] built classifiers (decision trees) to predict the dropout rate of freshmen with accuracies between 75% to 80%. With an accurate model of prediction, students were better advised to pursue a field of study. Additionally, the department was better able to allocate resources to aid students. The work in [4] found that the grades for linear algebra and calculus classes were strong predictors of success in the Computer Science curriculum. Their results also helped make improvements in classifying the students' performance metrics.

III. METHODOLOGY

The approach taken in this study was to follow the success of students rather than just simply counting the percentage of pass and fails. Students were assigned unique IDs. The actual student IDs were not used for privacy protection. Hence, the school database administrator ran all the actual student IDs through a hashing function to produce a pseudo student ID number. The pseudo student ID number was essential in order to follow each student through the Computer Science curriculum.

To follow the progress of each student through the curriculum, a script written with the Python language was created. A second reason the analysis was easier done via code was that the analysis required to maintain intermediate statistics on the distribution of courses being repeated with respect to each student, and whether the student ultimately graduated.

The initial step of the analysis was to determine a count on how the students progressed through the CS1, CS2, and CS3 courses. Figure 1 shows a visual representation of the student progress flow based on 10 years of enrollment data between the Fall 2010 and Spring 2016 semesters. A brief description of the nomenclature used is:

T: Transfer students

P: Pass

F: Fail with a grade of D+ or lower

D: Students who Pass but Drop from the CS1, CS2, and CS3 series curriculum

CSCI: Computer Science

CINS: Computer Information Systems

The circular roundabout design represents the possibility that students may repeat a course. And the arrows indicate students entering and exiting the CS1, CS2, and CS3 curriculum series.

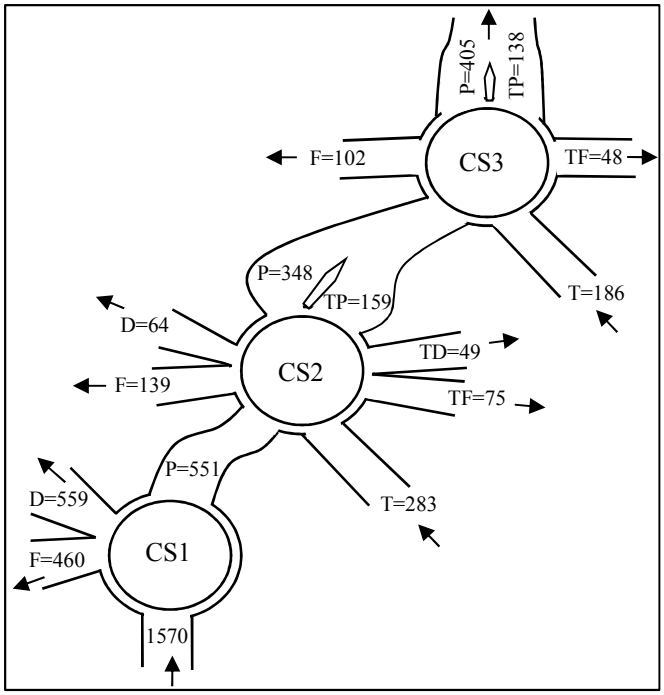

Fig. 1. Student progress flow diagram

The contents in Table 1 shows the courses from which student enrollment data was analyzed. Three early math or remedial courses were included in the study, as well as the required math courses. The requirements for a B.S. in Computer Science (CSCI) are Math 20 and Math 21 whereas, a B.S. in Computer Information Systems (CINS) requires Math 05 and Math 09.

TABLE 1. Courses from which enrollment data was analyzed

Course ID	Course Description
CS1	Programming and Algorithms I
CS2	Programming and Algorithms II
CS3	Algorithms and Data Structures
Math 51	Intermediate Algebra
Math 18	Trigonometry
Math 19	Pre-Calculus Mathematics
Math 20	Analytic Geometry and Calculus I
Math 21	Analytic Geometry and Calculus II
Math 05	Statistics
Math 09	Survey of Calculus

IV. RESULTS

The server used to execute the data analysis consisted of a Windows 10 Home 64-bit with an Intel Core i7 5500U@2.4 GHz and 16GB Dual-Channel DDR3@798MHz. The work was executed using VMWare Version 12 Pro with Ubuntu Version

16.04LTS. The version of Python was 3.4. A total of 2GB RAM was configured on the virtual server.

The dataset consisted of 10 years of course enrollment data from courses within the Computer Science curriculum between the Fall 2006 and Spring 2016 semesters. The data consisted from all majors and not just of students pursuing Bachelor's degrees in CSCI and CINS.

The results from Figure 1 are shown in Table 2. An unexpected result is that a far greater number of students transfer in taking CS2 compared to CS3; the standard approach for students transferring in from a 2-year college is to have completed CS1 and CS2. A likely reason of a higher percentage of CS2 students than expected is that transfer students re-take CS2 especially if he/she lacks a good understanding of the C++ programming language with pointers.

Another point of interest is that approximately one third of the students drop from the CS series curriculum after successfully passing CS1. The reason for this is that 90% of these students who drop belong to the Computer Engineering, Electrical / Electronic Engineering, or Mechatronic major. These three majors require CS1 as part of their curriculum towards a B.S. degree. The remaining 10% transfer to a different major. It is also of interest to note the high numbers of transfer students into the department. A total of 31% (159 / (159 + 348) of the students completing CS2 are first-time transfer students and a total of 25% (138 / (138 + 405) of the students completing CS3 are first-time transfer students. Given this flow of transfer students, 48% of the students advancing CS3 are transfer students. The percentage of transfer students is actually higher because transfer students may start with CS1.

Further results indicate a minimal difference between the success performance between the non-transfer and transfer students as shown in the right-most column of Table 2. The results also show a trend of improved success performance as students progress through the CS series.

TABLE 2. Student progress through CS series

	Total	Pass	Pass (drop CS)	Fail	Success %
CS1	1570	551	559	460	71%
CS2	551	348	64	139	75%
CS2-T	283	159	49	75	73%
CS3	507	405	-	102	80%
CS3-T	186	138	-	48	74%

A. Student's progress based on a 3-year period

The next task is an attempt to answer the question, "Suppose a given student were to enroll in a CS1, CS2, or CS3 course, can I then predict their graduation status?" This was accomplished by first extracting the first 3 of the 10 years of CS1 course data and then follow the progress of these students for the next 7 years. The goal was to determine if these students either graduated with a CSCI/CINS degree, graduated with a non-CSCI/CINS degree or did not graduate. The results indicate that given the 411 students enrolled in CS1 over a period of 3 years, only 41 students graduated with a B.S. degree in CSCI/CINS. A similar evaluation was done with 3 years given only CS2 course data and thereafter, with only CS3 course data. Table 3 represents the results with actual student counts.

TABLE 3. Student progress from 3-year period

	Total	CSCI Grad	CINS Grad	Non-CS Grad	Non-Grad
CS1	411	26 (6%)	15 (4%)	226 (55%)	144 (35%)
CS2	206	36 (17%)	19 (9%)	84 (41%)	67 (33%)
CS3	163	47 (29%)	26 (16%)	48 (29%)	42 (26%)

Note that the same students within the CS1 dataset may also be included in the CS2 and CS3 datasets. For example, a student who enrolled CS1 during the Fall 2016 semester would be included in the CS1 data set. And if the same student successfully proceeded to CS2 in the Spring 2017 semester, the student would also be included in the CS2 dataset. The goal of this analysis phase was to determine how likely a student will succeed as they continue through the CS1, CS2, and CS3 curriculum. The results show that students who reach CS3 are i) over 4 times (29% : 6%) more likely to graduate with a CSCI degree compared to students enrolled in CS1 ii) and even after successfully passing CS3, only 29% of these students graduate with a CS degree.

Since the CS1, CS2, and CS3 courses are taken by students of different majors, the earlier analysis does not thoroughly depict the success of CSCI/CINS students. Let's now analyze the success performance of only students whose major is CSCI or CINS. The results shown in Table 4 indicate a much higher attrition for CSCI students. Table 5 shows that a lower percentage of CINS students know they will graduate with a CINS major compared to CSCI students.

TABLE 4. CSCI Students

	Total	Declared as a CSCI major and did not graduate as a CSCI major	Declared as a CSCI major and graduated as a CSCI major	Not declared as a CSCI major and graduated as a CSCI major
CS1	130	104 (80%)	22 (17%)	4 (3%)
CS2	90	54 (60%)	30 (33%)	6 (7%)
CS3	81	34 (42%)	46 (57%)	1 (1%)

Figure 2. shows a further comparison graduation analysis between the CSCI and CINS students. All though the linear regression lines show a similar slope, an increase of students leads to a lower relative number of CSCI students compared to CINS students. Specifically, in 2007, the relative ratio is 15:5 and in 2016 it is 25:15 (lower). Between the Fall 2006 and

Spring 2016 semesters, 209 students graduated with a B.S. in Computer Science and 101 students graduated with a B.S. in Computer Information Systems.

TABLE 5. CINS Students

	Total	Declared as a CINS major and did not graduate as a CINS major	Declared as a CINS major and graduated as a CINS major	Not declared as a CINS major and graduated as a CINS major
CS1	36	21 (58%)	7 (20%)	8 (22%)
CS2	38	19 (50%)	14 (37%)	5 (13%)
CS3	36	10 (28%)	23 (64%)	3 (8%)

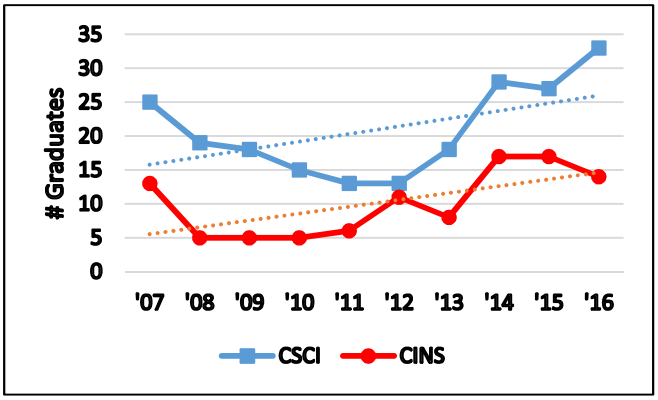

Fig. 2. Grade distribution of CS students who graduate

B. Analysis on course repetition

To further understand the success of students as they progress through the CS series, an analysis was done on how often students repeat courses. The frequency of course repetition was measured for all three courses, CS1, CS2, and CS3. Based on the CS department needs, it was also necessary to determine the graduation rates of the students who tended to repeat the courses. In this analysis phase, the percentage of students graduating is irrespective on being a CSCI/CINS major.

The values for the "# of students" are non-cumulative such that the sum of students for each course equals the number of students enrolled from Table 2. Thus, the table can be read as "87 students or 61% of the students who took the CS1 course, were able to pass on their second attempt. Of these 87 students, 45% eventually graduated but not necessarily with a Computer Science degree". This implies that the 87 students failed CS1 on their first attempt and are not part of the first attempt failure metric.

The results in Table 6 indicate that very few students attempt any of the CS courses a 4th time. This was important to know because one of the decision CS departments may make is to limit the number of times a course can be re-taken, such as was done recently at CSU, Chico.

It is evident from Table 6 that the total number of students progressing through the CS series decreases and that the success percentage increases. Given this trend, an anomaly is that the frequency for attempts greater than or equal to 3, CS3 is twice of either CS1 and CS2. The frequencies are 21, 20, and 42 for CS1, CS2, and CS3, respectively. Possible explanations for this are: i) students in CS3 have more invested in the CS curriculum and are more persistent to continue, ii) CS1 and CS2 students are more likely to change their major since they are just beginning, and iii) incoming transfer students who begin at CS3 are not fully prepared and will more likely re-take CS3 rather than reverting to CS2. The authors in [9] also found that the frequency of CS3 repeats is greater than for CS1 and CS2.

TABLE 6. Repetition and success of courses

	# Attempts	Outcome	# Students	% Grad
CS1	1	Pass	1014 (72%)	52%
		Fail	393 (28%)	25%
	2	Pass	87 (61%)	45%
		Fail	55 (39%)	22%
	3	Pass	8 (42%)	62%
		Fail	11 (58%)	18%
	4	Pass	0 (0%)	NA
		Fail	1 (100%)	0%
	5	Pass	1 (100%)	0%
		Fail	0 (0%)	NA
CS2	1	Pass	539 (76%)	54%
		Fail	174 (24%)	36%
	2	Pass	69 (68%)	52%
		Fail	32 (32%)	25%
	3	Pass	9 (56%)	56%
		Fail	7 (44%)	14%
	4	Pass	3 (100%)	67%
		Fail	0 (0%)	NA
	5	Pass	0 (0%)	NA
		Fail	1 (100%)	0%
CS3	1	Pass	464 (83%)	61%
		Fail	98 (17%)	26%
	2	Pass	58 (65%)	55%
		Fail	31 (35%)	10%
	3	Pass	**19 (58%)**	47%
		Fail	**14 (42%)**	21%
	4	Pass	2 (29%)	50%
		Fail	5 (71%)	40%
	5	Pass	0 (0%)	NA
		Fail	2 (100%)	0%

C. Which course(s) better predicts a CS Grad?

Based on the same 3-year data from the earlier section, the next analysis was conducted to determine which course(s) are better predictors for successful graduation with a CS degree.

Figure 3 shows the students who graduate with a CS degree are more likely to have received an 'A' grade in CS1/CS3, and at least a 'B' grade in CS2. The large number of C grades in CS3 of students contradicts the notion that a grade of 'A' or 'B' is necessary in CS3 to be an indicator of graduating with a CS degree. Hence, the CS1 grade distribution consistently shows that given a higher grade, the likelihood of graduating with a CS degree is higher. A similar analysis was performed with students from Under Represented Minority (URM) groups and the results were similar. An URM group refers to students classified as female, Hispanic/Latino, African-American, and/or Native-American. More work is needed to determine if these results are statistically significant.

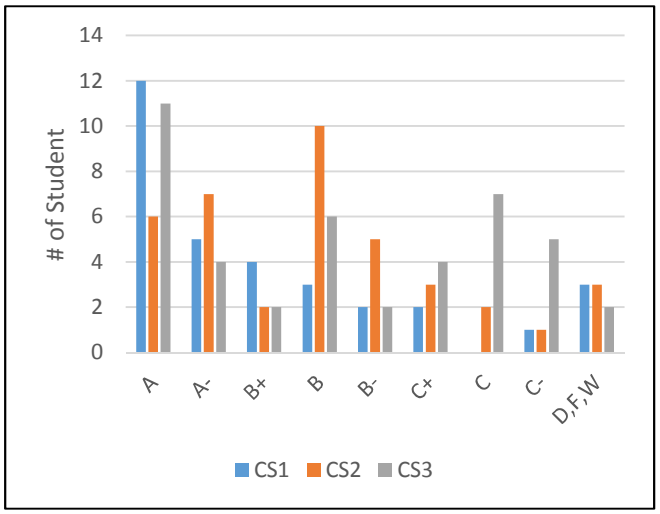

Fig. 3. Grade distribution of CS students who graduate

The results in Table 7 further suggest that success in CS1 may be a good predictor by analyzing the grade distribution of some courses the CSCI students took towards their graduation. Notice that the totals in Table 7 do not add up to 209 since many students transferred in from a 2-year college. The data is also summarized using a weighted GPA formula.

TABLE 7. Grade distribution for CSCI students

	Course						
	CS1	CS2	CS3	M18	M19	M20	M21
A	47	44	56	4	7	11	12
B	15	47	60	6	6	18	32
C	5	15	36	3	3	25	26
D	1	2	0	0	1	3	4
F	1	1	0	0	0	2	2
W	1	1	1	2	0	1	0
TOTAL	70	110	153	15	17	60	76
GPA	3.52	3.17	3.08	3.15	3.08	2.51	2.57

A similar course distribution analysis was done for "ALL" students independent of major as shown in Table 8.

TABLE 8. Grade distribution for ALL students

	Course						
	CS1	CS2	CS3	M18	M19	M20	M21
A	525	675	553	343	379	508	348
B	478	910	878	466	738	867	700
C	342	685	886	454	823	1156	957
D	142	149	137	150	223	279	152
F	259	158	106	195	250	325	258
W	110	101	79	139	172	249	148
TOTAL	1856	2678	2639	1747	2585	3384	2563
GPA	2.48	2.68	2.62	2.36	2.30	2.28	2.27

Table 9 shows that CS students tend to have a higher success in CS1 compared the average scores.

TABLE 9. Grade distribution comparison

	Course						
Major	CS1	CS2	CS3	M18	M19	M20	M21
CS	3.52	3.17	3.08	3.15	3.08	2.51	2.57
ALL	2.48	2.68	2.62	2.36	2.30	2.28	2.27
Difference	1.04	0.48	0.46	0.79	0.78	0.23	0.30

D. Are there any patterns from students changing majors into and out of Computer Science?

At CSU, Chico, it is easy for students to change their declared major. It was of interest to determine any patterns of these changes and when were these changes being made. Since the course enrollment data contained the declared major of the student at the time the course was taken, it was possible to study patterns in major changes. For example, by sorting the data in chronological order, it is possible to determine changes to a student's declared major based on the courses taken.

The initial analysis consisted of first determining the Computer Science graduates within the Fall 2011 through Spring 2016 semesters. There was a total of 121. Based on these graduates, the next step was to determine what their declared major was upon taking a given course. The course data was taken from the full 10-year period between the Fall 2006 and Spring 2016 semesters. This way, if a student graduated in the Fall 2011 semester, 5 years of prior data for this student was available. The results are shown in the top sections of Table 10 and Table 11.

TABLE 10. Computer Science Major

Enrollment	Student Course Enrollment				
	M51	M18	M19	M20	M21
Total Grads	7	12	15	51	65
# Grads declared as a CSCI major	1 (14%)	5 (42%)	12 (80%)	43 (84%)	51 (78%)
# Grads declared as a non-CSCI major	6 (86%)	7 (58%)	3 (20%	8 (16%)	14 (22%)
# Students declared as a CSCI major	26	27	53	117	83
# Students who graduated as a CSCI major	2 (8%)	4 (15%)	9 (17%)	34 (29%)	38 (46%)
# Students who never graduated	8 (31%)	8 (30%	20 (38%)	39 (33%)	20 (24%)
# Students who graduated in a non-CSCI major	16 (61%)	15 (55%)	24 (45%)	44 (38%)	25 (30%)

The next phase was to study possible patterns of students transferring out of Computer Science. This was accomplished by first determining all the students who took a given course as a declared Computer Science major. Then from these students, it was determined if the student graduated as a Computer Science major, or possibly if the student did not graduate. The bottom sections of Table 10 and table 11 contain these results.

TABLE 11. Computer Information Systems Major

Enrollment	Student Course Enrollment				
	M51	M18	M19	M05	M09
Total Grads	4	8	21	31	36
# Grads declared as a CINS major	2 (50%)	6 (75%)	10 (48%)	23 (74%)	33 (92%)
# Grads declared as a non-CINS major	2 (50%)	2 (25%)	11 (52%)	8 (26%)	3 (8%)
# Students declared as a CINS major	8	6	5	41	30
# Students who graduated as a CINS major	1 (12%)	2 (33%)	1 (20%)	16 (39%)	12 (40%)
# Students who never graduated	2 (25%)	2 (33%)	1 (20%)	6 (15%)	8 (27%)
# Students who graduated in a non-CINS major	5 (63%)	2 (33%)	3 (60%)	19 (46%)	10 (33%)

Table 12 and Table 13 contain more detailed results pertaining to the majors that Computer Science students transferred into and out from.

TABLE 12. Major transfers for CSCI majors

	Major	# of Students
Top majors prior to declaring and graduating as a CSCI major	Computer Animation: Game Development	10
	Math	9
	Mechanical Engineering	7
Top majors CSCI students transfer to	Management Info. Systems	30
	Computer Info. Systems	24
	Pre-Bus Info Systems	20
	Computer Animation: Animation	18
	Computer Animation: Game Development	17

Most of the students who transfer out of the Computer Science major change to the Business Information Systems and Computer Animation: Game Development majors. This is a trend the Computer Science Department suspected.

TABLE 13. Major transfers for CINS majors

	Major	# of Students
Top majors prior to declaring and graduating as a CINS Major	Computer Science	30
	Pre-Bus Administration	11
	Mechanical Engineering	5
Top majors CINS students transfer to	Management Info. Systems	22
	Pre-Bus Info Systems	6
	Pre-Bus Administration	4

To gain a better understand on how Math impacts the success rate of students whose major is Computer Science, the graduation percentages were plotted as shown in Figure 4. As expected, the percentages of graduating in Computer Science increases steadily as the student continues through more advanced math courses. Conversely, the probability that a Computer Science student transfers to a different major and graduates, also becomes less as progress is made through the Math curriculum. The results indicate a statistical significance using the chi-square test ($\rho < 0.01$) that Math 119 (Pre-Calculus) is a critical point. Upon passing Math 119, the probability that a student does not graduate is less. Related research suggests that Calculus I is the critical class [6, 7]. A possible explanation is that the Math competency level may be lower in this study compared to other schools.

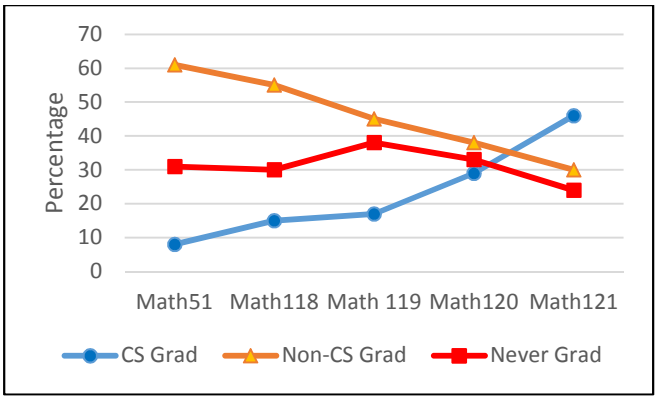

Fig. 4. Math series success

E. Graduation duration rates

A goal of the university from which this study was conducted is to reduce the number of years that students take to graduate. The 3-year CS1 data was used to compare the time duration between students graduating with a CSCI/CINS degree and students graduating with a non-CS degree. To visualize the difference between the CS students and non-CS students, the results were scaled to 100. For example, from the time students enrolled in CS1, the largest group of students (consisting of 35 students) took 3.5 years to graduate. Hence, the value of 35 was scaled to 100, and similarly for the remaining values.

Figure 5 shows the results which indicate a slightly longer duration period for students to graduate with a CS degree from the time they first enroll in the CS1 course. Note that a value of zero for a non-CS student indicates that the student took CS1 in his/her last semester before graduating.

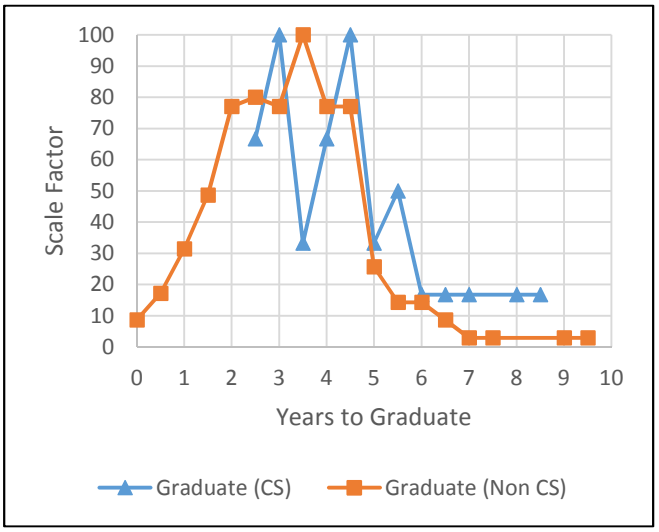

Fig. 5. Graduation duration comparison

V. CONCLUSION AND FUTURE WORK

A deeper understanding on the students' progress through the CS1, CS2, and CS3 was gained, such as determining that the success performance metrics between non-transfer and transfer students were similar. Roughly half of the graduates from the Computer Science Department originated as transfer students. It is also evident that as the student makes progress through the computer science curriculum, the success rate increases. More data analysis is necessary to validate that the grade for CS1 is indeed the best predictor of success.

Given that the rate of attrition with Computer Science students is commonly high among teaching institutions, and prevalent at CSU, Chico, it suggests more work is needed to better address the problem. Future work consists of analyzing the grade outcomes of other courses such as Physics and Chemistry to determine if they can better predict success in the computer science curriculum. More data will also be requested and analyzed to determine what other majors, and their distribution, are enrolling in the CS1, CS2, and CS3 courses; and also, to better understand the time needed for students to graduate.

REFERENCES

[1] T. Beaubouef and J. Mason, "Why the High Attrition Rate for Computer Science Students: Some Thoughts and Observations," ACM SIGCSE Bulletin, New York, USA, Volume 37, Issue 2, pp. 103-106, June 2005.

[2] J. Bennedsen and M.E. Caspersen, "Failure Rates in Introductory Programming," ACM SIGCSE Bulletin, New York, USA, Volume 39 Issue 2, pp. 32-36, June 2007.

[3] P. Byrne and G. Lyons, "The effect of student attributes on success in programming," ACM SIGCSE Bulletin, New York, USA, Volume 33, Issue 3, pp. 49-52, September 2001.

[4] G. Dekker, M. Pechenizkiy, and J. Vleeshouwers, "Predicting Students Drop Out: A Case Study," Proceedings of the 2nd International Conference on Educational Data Mining - EDM 2009, Cordoba, Spain, pp. 41-50, January 2009.

[5] J. Elarde, "Toward improving introductory programming student course success rates: experiences with a modified cohort model to student success sessions," Journal of Computing Sciences in Colleges, Volume 32 Issue 2, pp. 113-119, December 2016.

[6] J.E. Froyd, K.J. Shryock, M. Tripathy, A.R. Srinivasa, and R.C. Simon, "Patterns of Students' Success: How Engineering Students Progress Through a Course Sequence. Presented at Persistence and Retention," 122nd American Society for Engineering Education (ASEE) Annual Conference & Exposition. Seattle, WA, June 14-17, 2015.

[7] J.J. Pembridge and M.A. Verleger, "First-Year Math and Physics Courses and their Role in Predicting Academic Success in Subsequent Courses," 120nd American Society for Engineering Education (ASEE) Annual Conference & Exposition, Atlanta, GA, June 23, 2013.

[8] L. Porter, M. Guzdial, C. McDowell, and B. Simon, "Success in introductory programming: what works?," Communications of the ACM, volume 56, number 8, pp. 34-36, August 2013.

[9] C.F. Reilly and E. Tomai, "An examination of mathematics preparation for and progress through three introductory computer science courses," Frontiers in Education Conference (FIE) 2014.

[10] C. Reilly, E. Tomai, and L.M. Grabowski, "An evaluation of how changes to the introductory computer science course sequence impact student success," Frontiers in Education Conference (FIE) 2015. 32614 2015. IEEE, pp. 1-6, 2015.

[11] A. Vihavainen, J. Airaksinen, and C. Watson, "A systematic review of approaches for teaching introductory programming and their influence on success," Proceedings of the 10th Conference on International Computing Education Research, Glasgow, United Kingdom, pp. 19-26, August 2014.

[12] C. Watson and F.W.B. Li, "Failure rates in introductory programming revisited," Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education, Uppsala, Sweden, pp 39-44, June 2014.