Engineering Educators Bringing the World Together
2025 ASEE Annual Conference & Exposition
Palais des congrès de Montréal, Montréal, QC • June 22–25, 2025 ASEE
Paper ID #47852

Assessing Communication Skills Across Upper-Level CS Courses

Dr. Ryan Edward Dougherty, United States Military Academy

Dr. Ryan Dougherty is an assistant professor at the United States Military Academy at West Point. He earned his Ph.D. and B.S. degrees in Computer Science from Arizona State University. His research interests include theoretical Computer Science, evolutionary computing, combinatorial algorithms and Computer Science education. Dr. Dougherty spent a year as a visiting assistant professor at Colgate University. He has also been on several program committees in Computer Science education.

Dr. Maria R. Ebling, United States Military Academy

Maria R. Ebling is an assistant professor at the United States Military Academy at West Point in the Department of Electrical Engineering and Computer Science. She earned both a Ph.D. and M.S. degree in Computer Science from Carnegie Mellon University and a B.S. in Mathematics from Harvey Mudd College. Her research interest includes pervasive computing, the Internet of Things, and Computer Science education. Prior to joining the faculty of West Point, Dr. Ebling spent three years as the Chief Technology Officer at Medaptive Health and nearly 20 years at the IBM T. J. Watson Research Center. Dr. Ebling is an ACM Distinguished Scientist.

Assessing Communication Skills Across Upper-Level CS Courses

Abstract

Computer Science courses demand commitment to not only a deep understanding of the course material but also an ability to communicate that understanding. We implemented a writing assignment and several formal group presentations for undergraduates in a Theory of Computing (ToC) course. The writing assignment included creating a research-style paper solving ToC problems and anonymously refereeing other students' paper submissions. In this course students also orally presented several times in groups about related ToC problems. We address the impact that such writing assignments and presentations have on follow-on courses in our CS curriculum: a similar writing assignment in an Operating Systems (OS) course, and both written reports and formal presentations in a two-semester capstone course. We found that participation in ToC had a significant effect on the OS course's outcomes, and similarly was a significant predictor for those of the capstone courses. OS course participation was an accurate predictor of capstone course performance, and similarly the first semester of capstone accurately predicted the second. Additionally, we found that peer reviewing in ToC predicted OS writing performance and that the final ToC presentation was predictive of capstone's presentation scores. These results suggest that specific elements of prior instruction for technical communication can positively impact performance in subsequent upper-level CS courses. The most surprising finding is the importance of peer review in technical CS courses for follow-on course performance, which merits further study.

1 Introduction

The third ABET criterion for computer science [1] says that "Graduates of the program will have an ability to communicate effectively in a variety of professional contexts." Professional writing and communication skills are very important for all computer science students. These skills develop from a large variety of experiences: writing code, writing research papers and formal reports, giving research talks, among many others [2, 3]. Often instructors structure their courses based on one of these: CS1 courses usually only focus on writing code, algorithms and theory courses predominantly focus on formal proofs, ethics or professional responsibilities courses based on individual reports, and research seminars on research talks and papers. To date, there has been little investigation as to how communication skills developed in one CS course impact similar skills in a subsequent course.

This paper aims to begin to understand the above issues with a focus on writing and presentation skills. We pose the following research questions:

- **RQ1**: Does a student's overall performance in a technical CS course that has a communication skills assignment impact their performance in a later technical CS course with a similar such assignment?
- **RQ2**: Does a student's performance on a writing assignment in one course impact their performance on a similar writing assignment in a later CS course?
- **RQ3**: Does a student's performance on a oral presentation in one course impact their performance on a similar presentation in a later CS course?

We created a study[1] that answers these research questions by examining communications activities across three upper-level courses that all CS majors at our institution must take: Theory of Computing (ToC), Operating Systems (OS), and a year-long Capstone course. The remainder of this paper is organized as follows: Section 2 contains relevant work regarding impacts of CS courses on follow-on ones. Section 3 sets the context for both our institution and the three relevant courses. Section 4 details our method. Section 5 contains the results regarding the above research questions. Section 6 discusses our results. Section 7 details future work and concludes the paper.

2 Background and Related Work

Many researchers have worked to predict factors that influence success in CS1. An ITiCSE working group performed a systematic literature review in 2012 [4] examining the literature on predicting academic performance in computer science courses. Alvarado et al. [5] looked at the impact of pre-college CS classes on future success across the Computer Science curriculum. They found statistically significant differences across nearly all of the mid- and upper-level CS courses for students who took the AP CS A course as compared to their peers who did not. Quille and Bergin [6] re-validated the Predict Student Success (PreSS) model [7] more than ten years after its introduction. They found that PreSS was able to identify weak CS1 students with more than 70% accuracy based on data collected approximately 10% into the introductory programming module.

Researchers have also studied predictors of success in CS2 courses. Hooshangi et al. [8] confirmed that CS1 grades predict performance in CS2. Ellis and Hooshangi [9] further validated this work using seven years of data from 5300 CS2 students. Beck et al. [10] studied the question of whether responses to individual CS1 exam questions could predict student success overall in a CS2 class and determined this is the case. Bjorn and Kann [11] took an inductive approach to determine the factors of success in CS2. Danielsiek and Vahrenhold [12], and Layman et al. [13], found similar results.

Other researchers have examined predictors of success more broadly. Falkner and Falkner [14] discovered that a simple metric–turning in the first assignment late–was an indicator of risk with nearly 1 in 4 of students who do so at risk of failing the course.

[1]Approved by our local institutional review board.

The work above is entirely focused on predicting success in a course, whether that course is CS1, CS2, or CS courses more generally. In contrast, our work looks at predicting future communication assessments based on current communication performance. Researchers have also studied communication skills among CS graduates. French [15] created a communication-intensive computer science course; within their classification, our work falls into the research paper and writing skills categories. Givens [16] used a writing assignment in her algorithms course to give students practice with written and oral communication and experience with algorithms beyond the scope of the class. Kumar [17] describes the concept of a mock conference and the scaffolding to support students. Keesee and Bauer-Reich [18] discuss using a mock conference to prepare students for research-style papers, whether in graduate school or in industry. Dougherty [19] describes his experience in using a research-style paper as part of an undergraduate Theory of Computing (ToC) course. Our paper expands on that experience by tracking the communication outcomes, both written and oral, of the same cohort of students as they move forward into future classes that have written and presentation assessments.

3 Context

Our institution is a small liberal-arts college in the northeastern United States. Students in our department mainly consist of CS majors, Cyber majors, and Electrical Engineering majors. Sixty-five students in these majors graduated at the conclusion of this study. There are three courses at our institution relevant to this paper: a Theory of Computing (ToC) course, an Operating Systems (OS) course, and a year-long capstone course. We describe each course below.

3.1 ToC Course

Our ToC course is a requirement for all CS majors, and students most often take it in the Spring semester of their junior year. This offering had 39 students and no students from other majors. The only two pre-requisites for ToC are discrete mathematics and data structures, which is a CS2 course; there are no formal post-requisites for ToC.

The ToC course held a “mock conference” graded event in which students were placed in groups of two or three, given a unique ToC problem, and tasked with writing a research-style paper about it. Each group submitted their (anonymized) paper, and each student individually and anonymously provided two conference-style reviews. Papers were then graded, and some were accepted to the mock conference, based on accuracy of proofs and arguments and on the clarity of exposition. There were no formal presentations about their research papers. Grades for the anonymous reviews were based nearly entirely on the accuracy and completeness of the criticisms with some credit for accurate choice of accept/reject and paper summary. For more details, see [19].

Throughout the semester, students were placed in groups to present a single problem in a short in-class presentation. These problems involved standard ToC concepts [20, 21]: regular languages, context-free languages, Turing machines and decidability, and undecidability. In a separate assignment at the semester’s end, different groups of students presented a formal proof of a claim. For this assignment, each group had 4 or 5 students and were assigned an NP-completeness problem to present to the instructor. All presentation group grades were assigned based on the correctness and clarity of presentation along with group cohesion and ability to answer the instructor's questions.

3.2 OS Course

Our OS course is taken by all Computer Science majors and about half of the Cyber Science majors. The course is taken in the first semester of their senior year with very few exceptions. ToC is not a pre-requisite for OS. In this offering, 44 students took OS, 28 of whom had taken ToC during the previous semester.

Traditionally, the OS course had students do a performance analysis of different OS schedulers using Pintos [22]. Students would present their "results" orally in class. The results of this assignment were repeatedly unsatisfying because the students did not seem to have done a thorough evaluation. Following the introduction of the mock conference in ToC, the OS course changed the presentation format of this assignment to be a research-style paper.

Students were placed into groups for this assignment. Each group selected their evaluation criteria, performed their experiment(s), and wrote their papers. They also implemented a system call that allowed them to change the priority of processes. Many, though not all, groups also implemented additional instrumentation to collect necessary data. Students received peer reviews prior to submitting their final research paper to the instructor for grading. The papers were not presented in class.

3.3 Capstone Course

Our capstone course is a year-long course taken by all seniors in the department, not just Computer Science majors. In this course, students are placed in groups of four to six to complete a major project across two semesters. We split the course into the first and second semester offerings: Capstone 1 and Capstone 2 respectively. The capstone courses had 65 students complete it with 29 who had previously taken ToC. Occasionally non-EECS majors are part of some groups due to the interdisciplinary nature of the projects, but most group members are EECS majors. Deliverables include a project analysis, lab notebook, formal group presentations (such as sprint reviews), as well as written documents (such as a design report and a continuity document).

4 Method

In the Spring 2023 offering of our ToC course, we recorded grades of all students regarding the mock conference assignment–both the paper and refereeing parts–and the in-class presentations. In the following semester (Fall 2023), we recorded the grades of all students taking OS regarding the paper writing assignment. Also in that semester and the following one, we recorded grades of all students taking Capstone 1 and Capstone 2. Any students in any of these classes who dropped the course, changed majors, or voluntarily withdrew from

Table 1: Least Squares Regression Results for RQ1: ToC Overall Performance vs. OS Overall Performance.

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
Intercept	-3.6619	103.446	-0.035	0.972	-216.297	208.973
ToC Overall	1.0364	0.127	8.137	< 0.001	0.775	1.298

Table 2: Least Squares Regression Results for RQ1: ToC Overall Performance vs. Capstone 1 Overall Performance.

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
Intercept	504.8428	92.5568	5.4544	< 0.001	314.5896	695.0961
ToC Overall	0.5117	0.1140	4.4900	< 0.001	0.2774	0.7459

our study were removed before analysis, and any remaining students were anonymized by an independent party.

5 Results

5.1 RQ1: Impact or Not?

For RQ1, to determine whether student performance in a course with communication-focused assignments influences their overall performance in follow-on courses that also have such assignments, we used ordinary least squares (OLS) to test the effect of having taken Theory of Computing (ToC). Tables 1 to 5 contain our statistical results.

We found that performance in ToC was significant for predicting OS performance ($p < 0.001$) and was significant for predicting both Capstone 1 and Capstone 2 performance ($p < 0.001$ for both). We also found statistical significance for participation in the OS course's predicting Capstone 2 performance ($p < 0.001$). Finally, we found significance for performance for Capstone 1's predicting Capstone 2 performance ($p < 0.001$).

5.2 RQ2: Writing

For RQ2, we used multiple linear regression with individual assignment scores from the ToC course as predictors. Tables 6 to 10 contain our results. For the OS writing assignment, the model including both ToC paper and refereeing scores was statistically significant ($p = 0.001$), with the refereeing score being the stronger predictor ($p = 0.022$). For Capstone

Table 3: Least Squares Regression Results for RQ1: ToC Overall Performance vs. Capstone 2 Overall Performance.

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
Intercept	504.8428	92.557	5.454	< 0.001	314.590	695.096
ToC Overall	0.5117	0.114	4.490	< 0.001	0.277	0.746

Table 4: Least Squares Regression Results for RQ1: OS Overall Performance vs. Capstone 2 Overall Performance.

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
Intercept	644.5890	67.056	9.613	< 0.001	508.955	780.223
OS Overall	0.3258	0.080	4.086	< 0.001	0.165	0.487

Table 5: Least Squares Regression Results for RQ1: Capstone 1 Overall Performance vs. Capstone 2 Overall Performance.

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
Intercept	206.0339	100.493	2.050	0.047	2.768	409.300
Cap1 Overall	0.7993	0.113	7.086	< 0.001	0.571	1.027

Table 6: Multiple Linear Regression Results for RQ2: ToC Paper Writing and Refereeing vs. OS Writing.

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
Intercept	46.3018	12.024	3.851	0.001	21.538	71.065
ToC Paper	0.2444	0.171	1.429	0.165	-0.108	0.597
ToC Ref	0.4940	0.202	2.451	0.022	0.079	0.909

Table 7: Multiple Linear Regression Results for RQ2: ToC Paper Writing and Refereeing vs. Capstone 2 Continuity Documentation.

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
Intercept	99.5765	6.675	14.918	< 0.001	85.829	113.324
ToC Paper	-0.0993	0.095	-1.046	0.306	-0.295	0.096
ToC Ref	0.0538	0.112	0.481	0.634	-0.177	0.284

Table 8: Multiple Linear Regression Results for RQ2: ToC Paper Writing and Refereeing vs. Capstone 2 Final Report.

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
Intercept	127.4958	10.742	11.869	< 0.001	105.372	149.619
ToC Paper	-0.1278	0.153	-0.836	0.411	-0.442	0.187
ToC Ref	0.2694	0.180	1.497	0.147	-0.101	0.640

Table 9: Linear Regression Results for RQ2: OS Writing vs. Capstone 2 Continuity Documentation.

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
Intercept	95.4531	5.800	16.458	< 0.001	83.722	107.184
OS Writing	0.0209	0.062	0.336	0.739	-0.105	0.147

Table 10: Linear Regression Results for RQ2: OS Writing vs. Capstone 2 Final Documentation.

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
Intercept	141.7193	10.507	13.488	< 0.001	120.467	162.972
OS Writing	-0.0742	0.113	-0.656	0.515	-0.303	0.154

Table 11: Multiple Linear Regression Results for RQ3: ToC In-Class/Final Presentations vs. Capstone 1 Presentation.

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
Intercept	-13.0348	30.575	-0.426	0.674	-76.005	49.935
ToC InClassPres	0.2316	0.817	0.283	0.779	-1.452	1.915
ToC FinalPres	0.4592	0.185	2.480	0.020	0.078	0.840

2's continuity documentation and final report, no statistically significant predictive value was found for either ToC paper or refereeing scores ($p = 0.306, 0.634$ and $p = 0.411, 0.147$, respectively). For both written assignments in Capstone 2 (continuity documentation and final report), the OS writing assignment showed no statistical significance ($p = 0.739$ and $p = 0.515$, respectively).

5.3 RQ3: Presentation

For RQ3, we also used multiple linear regression with individual presentation scores from the ToC course as predictors. Tables 11 and 12 contain our results. To analyze the effect of ToC presentation experience on later presentations, we computed average in-class presentation scores (ToC InClassPres) and included the final presentation (ToC FinalPres) as an additional predictor.

For Capstone 1's presentation, the model was not statistically significant ($p = 0.674$); however, the final ToC presentation score showed a significant individual effect ($p = 0.020$). For Capstone 2's final presentation, the overall model was not significant ($p = 0.347$), and individual predictors did not reach significance alone.

Table 12: Multiple Linear Regression Results for RQ3: ToC In-Class/Final Presentations vs. Capstone 2 Final Presentation.

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
Intercept	-32.2324	33.594	-0.959	0.347	-101.421	36.956
ToC InClassPres	1.7065	0.898	1.900	0.069	-0.143	3.556
ToC FinalPres	0.2946	0.203	1.448	0.160	-0.124	0.713

6 Discussion

For RQ1, we did find a statistically significant effect of ToC participation on overall performance in all three surveyed follow-on courses: OS, Capstone 1, and Capstone 2. This suggests that simply taking a course with writing and presentation assignments appears to translate to immediate performance differences in follow-on courses. For all three results, the p-value was very small, less than 0.001. However, the r^2 value was only moderately high, such as predicting OS performance with $r^2 \approx 0.72$ and Capstone 2 with $r^2 \approx 0.44$. The reason for such a relatively low value resides with the calculation of the t-statistic; the slope for the model involving these courses is small, and thus the t-statistic is large, yielding a small p-value.

For RQ2, we observed a statistically significant relationship between ToC writing scores and OS writing performance, especially with the anonymous refereeing component. This may indicate that students who engaged with evaluating their classmates' work were better prepared to communicate technical information in OS. We also found a significant predictive effect for both Capstone 2's writing assignment and final design documentation assignment, suggesting that the benefits of ToC writing do stay with students throughout the rest of the technical CS course curriculum. This assignment occurs approximately one year after ToC and is an intensive writing assignment in the Capstone 1/2 sequence. The improvement suggests a longer-term benefit of early exposure to structured writing in technical CS courses, but more investigation is warranted. We note in both cases that this significance only comes from writing *and* peer-reviewing as neither effect reached significance individually.

For RQ3, we found that the final presentation in ToC was not a statistically significant predictor of performance in Capstone 1's presentations, although the more formal presentation at the end of the ToC course individually reached significance. For Capstone 2, the regression models for presentation outcomes were clear: the models for presentations were not significant and no individual predictor stood out apart from the final presentation in ToC. These results appear to indicate that presentations in a technical CS course are generally not as significant as writing assignments.

Taken together, these results suggest that specific components of the ToC course, not just participation, are correlated with later technical CS course performance. Writing and presentation practice in ToC may have stronger effects when the skills are directly aligned with subsequent assignments.

6.1 Limitations

The main limitation is the applicability of some of our results. For example, we achieved statistical significance in some of our linear regression models. This might be explained as high-achieving students in one class are typically also high-achieving in another class.

In all three courses, these assignments were group assignments consisting of small teams of students. The team sizes in ToC and OS were teams of 2, possibly 3 in the case of an odd number of students. For capstone, teams were typically between 4 and 6 students. We did not account for the writing abilities of the individuals on any teams.

A limitation in the Capstone results for RQ1 is that the groups in these courses had both CS and non-CS students because some projects were multi-disciplinary. In all cases, the group presentation/writing assignment grade for Capstone was the same for all group members; thus there were some dependencies in the data. The only individual assignments in our Capstone courses were peer evaluations, which were not relevant to our study.

The grades in all three courses were assigned by the individual instructors, who may have had different grading standards. This concern is particularly pronounced in the Capstone course where each group had one instructor and each instructor was assigned to between one and three groups. Because there was a fairly large number of instructors, differences in grading standards could have had an impact on our results.

Finally, we did not consider the writing assignments students had outside these computer science courses during their junior and senior years. There is at least one significant writing event that take place in humanities classes during either junior or senior year. Because these events were not considered, it is possible that the writing improvements can be attributable to skills learned in those humanities classes and not in their major.

7 Future Work & Conclusion

A longitudinal study tracking and analyzing a given student cohort across all writing and presentation assignments within a curriculum is warranted, not just for CS courses. This would be useful for CS educators to determine where and why students have issues with writing and presenting. Implementing such a system would be challenging in terms of determining all the writing events across all courses taken. In addition, if CS courses do not already contain writing events, it may be challenging to convince faculty that incorporating one or more such events will not take away from their technical content [23].

In this paper we sought to determine if there is any impact of formal writing and presentation assignments in a ToC class on similar types of assignments in follow-on courses. We first determined that participation in such a ToC course does yield statistically significant changes in performance for the OS course and the Capstone course sequence. Next, we determined that writing in one course generally does not have a significant effect on writing assignments in a subsequent course, except for when both writing and peer reviewing. We also determined that oral presentations in one course do not have a significant effect on such assignments in a subsequent course. Finally, perhaps our most surprising finding is the importance of peer review in subsequent written assignments. Although more investigation is warranted, peer review appears important to improving one's own writing.

Acknowledgments

We thank the CS class of 2024 for helping support CS education research. The views expressed in this article are those of the authors and do not reflect the official policy or position of the Department of the Army, Department of Defense, or the U.S. Government.

References

[1] A. B. of Delegates Computing Area. *Criteria for Accrediting Computing Programs*. Baltimore, MD, Oct. 2022. URL: https://www.abet.org/wp-content/uploads/2023/05/C001_CAC-Criteria_2023-2024.pdf.

[2] V. W. Pine and M. L. Barrett. "What kinds of communication are required on the job?" In: *J. Comput. Sci. Coll.* 21.2 (Dec. 2005), pp. 313–321. ISSN: 1937-4771.

[3] K. Anewalt and J. Polack. "Industry Trends in Software Engineering: Alumni Perspectives". In: *J. Comput. Sci. Coll.* 39.3 (Oct. 2023), pp. 159–170. ISSN: 1937-4771.

[4] A. Hellas et al. "Predicting academic performance: a systematic literature review". In: *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*. ITiCSE 2018 Companion. Larnaca, Cyprus: Association for Computing Machinery, 2018, pp. 175–199. ISBN: 9781450362238. DOI: 10.1145/3293881.3295783. URL: https://doi.org/10.1145/3293881.3295783.

[5] C. Alvarado, G. Umbelino, and M. Minnes. "The persistent effect of pre-college computing experience on college CS course grades". In: *ACM Inroads* 9.2 (Apr. 2018), pp. 58–64. ISSN: 2153-2184. DOI: 10.1145/3210551. URL: https://doi.org/10.1145/3210551.

[6] K. Quille and S. Bergin. "Programming: predicting student success early in CS1. a re-validation and replication study". In: *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*. ITiCSE 2018. Larnaca, Cyprus: Association for Computing Machinery, 2018, pp. 15–20. ISBN: 9781450357074. DOI: 10.1145/3197091.3197101. URL: https://doi.org/10.1145/3197091.3197101.

[7] S. Bergin and R. Reilly. "Programming: factors that influence success". In: *Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education*. SIGCSE '05. St. Louis, Missouri, USA: Association for Computing Machinery, 2005, pp. 411–415. ISBN: 1581139977. DOI: 10.1145/1047344.1047480. URL: https://doi.org/10.1145/1047344.1047480.

[8] S. Hooshangi, M. Ellis, and S. H. Edwards. "Factors Influencing Student Performance and Persistence in CS2". In: *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education - Volume 1*. SIGCSE 2022. Providence, RI, USA: Association for Computing Machinery, 2022, pp. 286–292. ISBN: 9781450390705. DOI: 10.1145/3478431.3499272. URL: https://doi.org/10.1145/3478431.3499272.

[9] M. Ellis and S. Hooshangi. "Replication and Expansion Study on Factors Influencing Student Performance in CS2". In: *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*. SIGCSE 2023. Toronto ON, Canada: Association for Computing Machinery, 2023, pp. 896–902. ISBN: 9781450394314. DOI: 10.1145/3545945.3569867. URL: https://doi.org/10.1145/3545945.3569867.

[10] L. Beck, P. Kraft, and A. W. Chizhik. "Predicting Student Success in CS2: A Study of CS1 Exam Questions". In: *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education - Volume 1*. SIGCSE 2022. Providence, RI, USA: Association for Computing Machinery, 2022, pp. 140–146. ISBN: 9781450390705. DOI: 10.1145/3478431.3499276. URL: https://doi.org/10.1145/3478431.3499276.

[11] C. Björn and V. Kann. "Variables Affecting Students' Success in CS2". In: *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1*. ITiCSE 2023. Turku, Finland: Association for Computing Machinery, 2023, pp. 257–263. DOI: `10.1145/3587102.3588856`. URL: `https://doi.org/10.1145/3587102.3588856`.

[12] H. Danielsiek and J. Vahrenhold. "Stay on These Roads: Potential Factors Indicating Students' Performance in a CS2 Course". In: *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*. SIGCSE '16. Memphis, Tennessee, USA: Association for Computing Machinery, 2016, pp. 12–17. ISBN: 9781450336857. DOI: `10.1145/2839509.2844591`. URL: `https://doi.org/10.1145/2839509.2844591`.

[13] L. Layman, Y. Song, and C. Guinn. "Toward Predicting Success and Failure in CS2: A Mixed-Method Analysis". In: *Proceedings of the 2020 ACM Southeast Conference*. ACM SE '20. Tampa, FL, USA: Association for Computing Machinery, 2020, pp. 218–225. ISBN: 9781450371056. DOI: `10.1145/3374135.3385277`. URL: `https://doi.org/10.1145/3374135.3385277`.

[14] N. J. Falkner and K. E. Falkner. "A fast measure for identifying at-risk students in computer science". In: *Proceedings of the Ninth Annual International Conference on International Computing Education Research*. ICER '12. Auckland, New Zealand: Association for Computing Machinery, 2012, pp. 55–62. ISBN: 9781450316040. DOI: `10.1145/2361276.2361288`. URL: `https://doi.org/10.1145/2361276.2361288`.

[15] J. H. French. "Evaluating a communication-intensive core course in the CS curriculum". In: *J. Comput. Sci. Coll.* 28.2 (Dec. 2012), pp. 197–209. ISSN: 1937-4771.

[16] R. M. Givens. "Expanding communication opportunities in algorithms courses". In: *J. Comput. Sci. Coll.* 34.3 (Jan. 2019), pp. 47–54. ISSN: 1937-4771.

[17] K. Kumar. "A Learner-Centred Mock Conference Model for Undergraduate Teaching". In: *Collected Essays on Learning and Teaching* 4 (2011), pp. 20–24.

[18] C. L. Keesee and C. Bauer-Reich. "Using a Mock Conference to Develop Academic Writing and Presentation Skills in Undergraduate Students". In: *2022 IEEE Frontiers in Education Conference (FIE)*. Uppsala, Sweden: IEEE, 2022, pp. 1–8. DOI: `10.1109/FIE56618.2022.9962719`.

[19] R. E. Dougherty. "Experiences Using Research Processes in an Undergraduate Theory of Computing Course". In: *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*. SIGCSE 2024. Portland, OR, USA: Association for Computing Machinery, 2024, pp. 310–316. DOI: `10.1145/3626252.3630849`. URL: `https://doi.org/10.1145/3626252.3630849`.

[20] R. E. Dougherty et al. "A Survey of Undergraduate Theory of Computing Curricula". In: *Proceedings of the 2024 on ACM Virtual Global Computing Education Conference V. 2*. SIGCSE Virtual 2024. New York, NY, USA: Association for Computing Machinery, Dec. 2024, pp. 281–282. DOI: `10.1145/3649409.3691070`. URL: `https://doi.org/10.1145/3649409.3691070` (visited on 02/06/2025).

[21] A. N. Kumar et al. *Computer Science Curricula 2023*. New York, NY, USA: Association for Computing Machinery, 2024.

[22] 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. Chattanooga, TN, USA: Association for Computing Machin-

ery, 2009, pp. 453–457. ISBN: 9781605581835. DOI: `10.1145/1508865.1509023`. URL: `https://doi.org/10.1145/1508865.1509023`.

[23] E. Giangrande. “Communication skills in the CS curriculum”. In: *J. Comput. Sci. Coll.* 24.4 (Apr. 2009), pp. 74–79. ISSN: 1937-4771.