Assessment of Self-Identified Learning Struggles in CS2 Programming Assignments

Matthew Zahn
Department of Computer Science
North Carolina State University
Raleigh, NC, USA
mszahn@ncsu.edu

Isabella Gransbury
Department of Computer Science
North Carolina State University
Raleigh, NC, USA
igransb@ncsu.edu

Sarah Heckman
Department of Computer Science
North Carolina State University
Raleigh, NC, USA
sarah_heckman@ncsu.edu

Lina Battestilli
Department of Computer Science
North Carolina State University
Raleigh, NC, USA
lbattestilli@ncsu.edu

ABSTRACT

Students can have widely varying experiences while working on CS2 coding projects. Challenging experiences can lead to lower motivation and less success in completing these assignments. In this paper, we identify the common struggles CS2 students face while working on course projects and examine whether or not there is evidence of improvement in these areas of struggle between projects. While previous work has been conducted on understanding the importance of self-regulated learning to student success, it has not been fully investigated in the scope of CS2 coursework. We share our observations on investigating student struggles while working on coding projects through their self-reported response to a project reflection form. We apply emergent coding to identify student struggles at three points during the course and compare them against student actions in the course, such as project start times and office hours participation, to identify if students were overcoming these struggles. Through our coding and analysis we have found that while a majority of students encounter struggles with time management and debugging of failing tests, students tend to emphasize wanting to improve their time management skills in future coding assignments.

CCS CONCEPTS

• **Social and professional topics → Student assessment**; • **Applied computing** → *Interactive learning environments.*

KEYWORDS

CS2; CS Education Research; Self-Regulated Learning

ACM Reference Format:
Matthew Zahn, Isabella Gransbury, Sarah Heckman, and Lina Battestilli. 2023. Assessment of Self-Identified Learning Struggles in CS2 Programming Assignments. In *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1 (ITiCSE 2023), July 8–12, 2023, Turku, Finland.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3587102.3588786

1 INTRODUCTION

The identification of student struggles as they complete programming assignments is important to academic success. Prior studies on student struggle contribute to a deeper understanding of what struggles student face when it comes to self-regulated learning (SLR) [1, 7].

In addition, some studies have explored the effects that student reflections have on the learning process in introductory computer science courses [2, 8, 9]. Through reflecting on their learning, students have identified that the two best ways they can improve themselves are to read the assignment specifications more carefully and start the assignments earlier [9]. This supports the idea that students who participate in reflective activities ultimately have an academic competitive edge, in the long run [8]. Finally, time management is an important skill in computer science that can greatly influence student success when completing programming assignments. It can even be utilized to predict how students will perform in future courses [4].

Despite these findings, it is unclear whether they apply to CS2 - the second course in a set of introductory courses for computer science - coursework and if reflection may lead to changes in how students work on future assignments [2]. These courses are designed to introduce students to the basic programming concepts and data structures that they will use later in the curriculum. To our knowledge, there has not yet been an identification of student struggles in CS2 programming assignments using data gathered from students' reflections. Further, we determine if time management behaviors change between assignments.

We focus on self-identified learning struggles that students report through the submission of a four-question project reflection form at the end of major programming assignment milestones in a CS2 course at a research-intensive university in the southeastern

United States. The objective of this research is *to analyze students' reflections on CS2 programming assignments to identify common struggles students encounter and examine whether or not there is evidence of improvement in these areas of struggle between projects, specifically with time management.*

Our two research questions are:

- **RQ1:** What are the common struggles CS2 students face when completing programming projects?
- **RQ2:** Do students show evidence of improvement in the specific area of struggle, time management, between assignments?

The contributions of this work are: 1) The analysis of the student reflections on completing programming assignments in a CS2 course; what went well, what students struggled with, and what they can improve upon, which have been identified by the students themselves. 2) Comparing students' time management on programming assignments for signs of improvement after a previous reflection on improving time management skills.

2 RELATED WORK

2.1 Student Reflections

When exploring student learning through self-reflection, VanDeGrift et al. examined 236 students in a CS1.5 course who were completing programming assignments and asked students to reflect on how they improved their skills between assignments [9]. Students identified that two major ways to improve were to read the specifications carefully and start assignments earlier.

Stone and Madigan created a pilot project designed to see how reflective writing could be used in the computer science (CS) and information sciences (IS) [8]. The focus was on getting freshman students from different stages of their program, in two separate freshmen level courses, to write reflections for experiential learning activities. Stone and Madigan ultimately found that students who consistently reflect on their learning skills are going to have a competitive advantage in the long run.

Bergin et al. composed a study to evaluate the relationship between SRL and programming performance to determine if SRL can be used as a predictor of introductory programming (e.g., CS1) performance [2]. Students in the study were given the Motivated Strategies for Learning Questionnaire (MSLQ) to measure the student's SRL. Findings from this study support the idea that students with high levels of task value (e.g., they see the importance of the learning task) and intrinsic goal orientation (e.g., they undertake the task for its own sake) used more self-regulating strategies and were more academically successful in introductory programming.

2.2 Student Struggles

When exploring student struggles with CS2 programming assignments, Arakawa et al. composed a study to better understand what students struggle with when trying to learn how to code [1]. The results of their study identify challenges in understanding requirements, assessing task difficulties, and assessing and completing problems. Furthermore, they discovered that students face a struggle with self-control as they get closer to assignment deadlines.

Salguero et al. constructed a study to examine how multiple potential sources of student struggle relate to outcomes across multiple CS1 computer science courses [7]. Utilizing exploratory factor analysis, four factors emerged: personal obligations, lack of sense of belonging, in-class confusion, and lack of confidence. Salguero et al. found that students who are struggling tend to do so on multiple fronts. Students with no prior experience report considerably more struggles than students with prior experience [7].

2.3 Time Management

Time management is an important skill to have when working on assignments in computer science. Castro et al. composed a study to examine how deadlines impact student behaviors in three separate introductory computer science courses [3]. The study concludes that the placement of deadlines in the beginning of the week had the best impact on spurring student activity. Ultimately, this study had varying success with deadlines placed at different days and times throughout the week, concluding that there are other factors at work which impact how students are "spurred" into taking action towards completing and submitting their assignments.

2.4 Synthesis & Contributions

There are several gaps that exist in the current literature on student struggles in introductory computer science coursework. There are few studies that analyze the impacts that self-regulated learning has on the struggles students face while working on programming assignments [1, 7, 9]. We are contributing to the current research on exploring how self-regulated learning techniques, such as reflections, can help students identify the struggles they face with CS2 programming assignments. Furthermore, we investigate these reflections between multiple programming assignments to determine whether or not students improve in these areas of struggle, such as time management, after identifying them in an earlier project reflection.

3 METHOD

The objective of this research is *to analyze students' reflections on CS2 programming assignments to identify common struggles students encounter and examine whether or not there is evidence of improvement in these areas of struggle between projects, specifically with time management.* Our two research questions are:

- **RQ1:** What are the common struggles CS2 students face when completing programming projects?
- **RQ2:** Do students show evidence of improvement in the specific area of struggle, time management, between assignments?

3.1 Course Context

The CS2 course at Anonymous Institution, located in the southeastern United States, is the second of a three-semester introductory sequence for computer science majors and minors. The course consists of two 75-minute lectures and a separate 110-minute lab co-requisite per week for a 15-week semester. The course covers advanced object-oriented programming; introductory software engineering; finite state machines; use and implementation of linear data structures (array-based lists, linked lists, stacks, queues, iterators); and recursion (general recursion overview and recursive lists). The programming language of instruction is Java.

There are multiple pathways to enter CS2: 1) students pass the introductory course with a grade of C or higher; 2) students earn a 4 or 5 on the Advanced Placement Computer Science A exam; 3) students transfer credit from an "equivalent" introductory course; and 4) students complete a credit-by-exam process. Students are expected to enter the course with knowledge of creating objects, composition relationships, file input/output, and arrays.

Since there are multiple pathways into the course, students first complete three Guided Projects (GP), which review prerequisite materials and progressively introduce software engineering tooling and new concepts through a combination of guided practice and independent tasks. Students additionally complete two, multi-part projects. The projects are broken into Part 1, where students propose a design and system tests given a set of requirements, and Part 2, where students implement and test a teaching staff design. In Part 2, students are expected to unit test their code, achieve at least 80% statement coverage, remove all static analysis notifications from source and test code, pass the teaching staff reference tests, and pass their system tests from Part 1. Students have three weeks to complete Part 2s and they submit their code to GitHub and an automated build system evaluates each submission and provides feedback.

Project Part 2's also contain two Process Points milestones, which are designed to help students maintain a steady pace as they complete their assignment during the three-week period. In Process Points 1, students are expected to have constructed the skeleton of their solution, including source and test files, and removed all Checkstyle warnings from their project. This is due by the end of the first week of the assignment. In Process Points 2, students are expected to have 60% statement coverage on the majority of their code base to reveal feedback from the teaching staff reference tests. This is due by the end of the second week of the assignment. Students then have one more full week to complete the assignment, including passing teaching staff reference tests, before the deadline. If needed, there is a standard 48-hour late window with increasing deductions. At the end of the late window, write permission is removed from GitHub repositories closing the project.

During the 2022 calendar year considered in this study, students completed two midterm examinations and a final exam. In the associated lab, students work in small teams to complete 12 lab activities that build upon each other over the course of the semester. Each lab focuses on a key topic in the course. Our focus for this study is three reflections that were administered as in-class exercises during lectures. The reflections were administered after 1) Guided Project 3, 2) Project 1 Part 2, and 3) Project 2 Part 2. The focus was on the student's experience completing the most recent project.

3.2 Participants

This study considered students enrolled in CS2 during the Spring 2022 (S22) and Fall 2022 (F22) semesters at a large public university in the southeastern United States. The participants were at least eighteen and consented [1] to the de-identified use of their course data. The breakdown of participants' demographics is available in Table 1.

[1]The research study is approved by Anonymous University IRB #anonymized.

Table 1: Consenting participant demographics for the Spring 2022 and Fall 2022 semesters.

	Consent Rate	Men	Women	Other Identities	Total
S22	41.56%	70.54%	19.38%	10.08%	129
F22	24.24%	67.03%	26.37%	6.60%	91

3.3 Data Collection via Reflections

Student reflections on their CS2 project experiences were collected as responses to a Google Form. The project reflections contained four questions that asked students to identify:

(1) *What went well* with their project
(2) What they *struggled* with
(3) What they *want to improve* upon for the next project
(4) Any suggestions for office hours as a course resource that would improve their participation in them.

The first three questions were assigned as required questions, meaning students had to provide an answer to these questions before being able to submit the reflection. The suggestions for improving office hours participation was an optional question. The breakdown of the number of submissions to each project reflection form is available in Table 2.

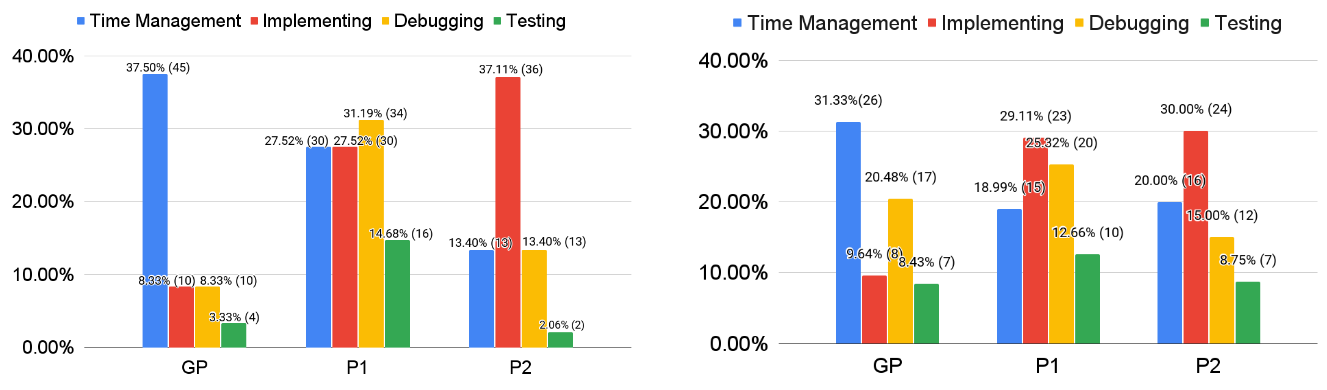
In the S22 semester, we had 77 students (59.7% of our S22 consented participants) who submitted a response to each reflection form; for the Guided Projects (GP), Project 1 (P1), and Project 2 (P2). When we discuss these reflections in later sections, we will use the term *complete reflections* to define this. In the F22 semester, we had 63 (69.23% of our consented F22 population) students who submitted a response to each reflection form. Table 3 provides the gender distribution for students with complete reflections.

Table 2: Total number of students to submit reflections each semester. All three reflections refers to the total number of students that submitted reflections for all three projects.

	GP	P1	P2	All Three Reflections
S22	120	109	97	77
F22	83	79	80	63

3.4 Thematic Analysis

We applied thematic analysis to the responses students submitted to the project reflection form [5]. Thematic analysis involves a team of researchers coding qualitative data, then condensing that data into common themes. This methodology is often used to analyze dialog in study participant interviews [6]. We coded the responses of S22 and F22 student reflections to create themes of student struggles and areas students wanted to improve for the CS2 course. For instance, if a student identified struggling to keep up with the imposed deadline, this would result in the response being tagged as a



(a) Spring 2022. **(b) Fall 2022.**

Figure 1: Struggles students identified across all three reflections.

Table 3: Distribution of students who completed all three project reflections

	Male	Female	Other Gender Identity	Total Count
S22	72.72%	20.78%	6.50%	77
F22	74.60%	25.40%	0.00%	63

struggle with time management. It is also possible that a response may be associated with more than one code. During our thematic analysis we completed the following 4 phases: (1) Review data and rectify errors. (2) Two researchers coded the first 10 responses together and discussed codes. This was done to establish a normal for the coding process. (3) Two researchers coded the remaining student responses. (4) These researchers met to group codes into themes and discuss anomalies.

3.5 Time Management Analysis

We utilized bash script automation to retrieve the date of the first commit students made to their assigned repositories. We then took this date and compared it against the final deadline of the associated program assignment. We then evaluated correctness by making sure all program assignments were being measured on a 0-7 scale for number of days early. This meant mapping 0-21 days on a 0-7 days scale by dividing by 3 and rounding to the nearest whole number. The goal of this was to ensure equal comparisons between assignments with a one week work period and a three week work period. From here we were able to compare whether or not students started earlier on their future assignment after reflecting that they wanted to improve their time management skills

4 RESULTS

4.1 RQ1: Student Struggles

First, we discuss the results of the emergent coding examining the distribution of our themes separated by each of the three questions in the project reflections that students were required to answer to identify the common struggles CS2 students face when completing programming projects. For this particular question we are considering all responses to the reflection forms, not just students who submitted complete responses for all project reflections

4.1.1 What Went Well. When asking S22 students what they thought went well with their Project 1, we recorded 36 (33.03%) unique occurrences that the implementation of the assignment went well, 35 (32.11%) occurrences that everything went well with the assignment, 17 (15.60%) responses that time management went well, and 11 (10.09%) responses that testing the implemented code went well. When asking S22 students what they thought went well with their Project 2, we recorded 31 (31.96%) unique occurrences that everything went well with the assignment, 31 occurrences that time management went well, 13 (13.40%) responses that understanding the instructions went well, and 11 (11.34%) responses that implementing the assignment went well.

When asking F22 students what they thought went well with the Guided Projects, we recorded 36 (43.37%) unique occurrences that everything went well with the projects, followed by 25 (30.12%) for the implementation of the project, 20 (24.10%) for the understanding of the assignment instructions, and 12 (14.46%) for time management. When asking F22 students what they thought went well with their Project 1, we recorded 22 (27.85%) unique occurrences that the implementation of the project went well, followed by 18 (22.78%) for time management, 17 (21.52%) for everything going well with the project, and 13 (16.46%) for testing of the source code.

4.1.2 What Students Struggled With. The next question of the reflection form was concerned with student struggles. There were four main areas of identified struggles: time management, implementing the solution, debugging the solution, and writing tests. Together these four areas accounted for 74.54% of the results in S22 and 76.45% of the results in F22. Figure 1a provides the distribution of the struggles experienced by students in the S22 semester and Figure 1b provides the distribution of the struggles for the F22 semester.

For the S22 semester, time management was the primary struggle when working on the Guided Projects at the start of the semester. As the semester progressed and the programming assignments increased in difficulty, we see a shift from struggles with time management to struggles with implementation and debugging.

We see similar results in F22. The height of time management struggles were during the Guided Projects. Students then struggled more with implementation and debugging in the later projects.

4.1.3 Areas Students Wanted to Improve. The final required section of the reflection asked students to think about how they wanted to improve their performance on future programming assignments in CS2 (or for Project 2, in future coursework). When analyzing the responses from students, we constructed Table 4 which displays the count and percentage distribution of the top areas that students want to improve upon. The most frequent response across both semesters and all projects is *time management* followed by *implementation*, *debugging*, and *documentation*.

4.2 RQ2: Time Management Improvements

Our results from RQ1 showed that time management was an area where students wanted to improve in completing CS2 programming assignments. We wanted to identify if students who listed time management as an area of improvement demonstrated actual improvement. We first grouped all of the students who acknowledged that they felt they needed to improve their time management skills by examining their responses to the third question in the reflection form, as shown in Table 4.

For each student, we collected the date and time of their first commit to their assignment GitHub repository, which we will call *start time*. We then calculated the difference, in days, between the assignment deadline and the start date; a larger difference translates to an earlier start. Due to differences in the time scales with the three Guided Projects and P1, we had to apply a mapping function to the three-week period of work students had for P1 so that it was equivalent to the one week of work for each of the GPs. Therefore all distances were on a 0-7 scale, representing how early students started, with 0 being late and 7 being early. Since the three Guided Projects build on each other, the assignments and GitHub repos were created during the second week of the semester. A proactive student may have started the assignment before it was the "active", or current, assignment with the upcoming deadline in the class. If a student happened to start an assignment before it was the "active" assignment, the time before the deadline number was mapped to 7. Due to the way assignments are presented in this course, this phenomenon was only possible for the Guided Projects and not the Project 1 or Project 2.

For each student interested in improving time management, we compared the three student start times to see if the student had started their next assignment, after a reflection, earlier than the previous assignment. This created comparisons between the GPs and P1, and between P1 and P2. Table 5 shows how many students improved their time management skills between assignments by starting their work earlier than they did previously. While we are able to see improvement in over half the students from the GPs to P1, this is not always the case between P1 and P2 as seen in the F22 semester. This suggests that there are other factors that may influence skills with time management when considered as starting earlier.

5 DISCUSSION

5.1 Threats to Validity

The threats to *internal validity* are associated with the study design. We coded the reflection responses using an emergent coding technique. Two coders worked together to identify the codes. Others coding the reflection responses may have coded things differently. However, the common patterns between semesters and the frequency of certain codes, like time management, suggest that the main findings of this paper hold.

The threats to *external validity* are associated with the generalizability of the results. We provide additional evidence that reflections may support student behavior change as shown in the number of students who started the next project earlier. Additionally, the projects that students work on are different in the S22 and F22 semesters. While they ultimately teach the exact same concepts the context of the assignment's problem differs from assignment to assignment. Finally, the P2 assignment is an optional team assignment, allowing students to submit an opt-in form that will place them on a two person team for the assignment. The inclusion of a partner when working on this assignment can change the student experience compared to individual assignments.

The threats to *construct validity* concern the automation that we used to calculate the Days Early metric. We utilized bash script automation to retrieve the date of the first commit students made to their assigned repositories. We then took this date and compared it against the final deadline of the associated program assignment. We then evaluated correctness by making sure all program assignments were being measured on a 0-7 scale for the days early. This meant mapping 0-21 days on a 0-7 days scale by dividing by 3 and rounding to the nearest whole number. The goal of this was to ensure equal comparisons between assignments with a one week work period and a three week work period.

5.2 RQ1: Student Struggles

In our first question, we sought to understand the common struggles that students encounter when working on programming assignments in a CS2 course. Different from previous studies, we examined how the self-regulated learning technique of writing reflections after completing these assignments can help students self-identify the struggles they are encountering and areas for improvement. Furthermore, we sought to do this in the realm of CS2 where the programming assignments are much more complex, involving multiple interacting classes and challenging concepts like finite state machines and linear data structures. .

Figure 1a and Figure 1b show how student struggles shift between the programming assignments in this course. In the GPs we see time management as the main struggle students are encountering. However, as the assignments shift from more guided tasks to individual work, the assignment difficulty increases which presents

Table 4: Count and percentage of areas where students wanted to improve.

	Spring 2022						Fall 2022					
	GP		P1		P2		GP		P1		P2	
Themes	Count	%	Count	%	Count	%	Count	%	Count	%	Count	%
Time Management	46	38.3%	64	58.7%	35	36.1%	34	41.0%	39	49.4%	32	40.0%
Documentation	13	10.8%	0	0.0%	3	3.1%	28	33.7%	2	2.5%	5	6.3%
Testing	4	3.3%	10	9.2%	7	7.2%	6	7.2%	17	21.5%	8	10.0%
Implementation	14	11.7%	10	9.2%	10	10.3%	5	6.0%	7	8.9%	11	13.8%
Debugging	13	10.8%	7	6.4%	4	4.1%	6	7.2%	7	8.9%	8	10.0%
Understanding	4	3.3%	6	5.5%	18	18.6%	9	10.8%	11	13.9%	1	1.3%
UML	4	3.3%	14	12.8%	6	6.2%	0	0.0%	8	10.1%	6	7.5%
Help Seekng Behaviors	0	0.0%	0	0.0%	1	1.0%	2	2.4%	7	8.9%	3	3.8%
Organization	0	0.0%	0	0.0%	3	3.1%	5	6.0%	1	1.3%	3	3.8%
Partner Communication	0	0.0%	0	0.0%	3	3.1%	0	0.0%	1	1.3%	3	3.8%
Office Hours	0	0.0%	9	8.3%	3	3.1%	1	1.2%	1	1.3%	1	1.3%
Partner Accountability	0	0.0%	0	0.0%	1	1.0%	0	0.0%	0	0.0%	2	2.5%
Communication With Professor	0	0.0%	1	0.9%	0	0.0%	0	0.0%	0	0.0%	0	0.0%
Contributions	0	0.0%	0	0.0%	2	2.1%	0	0.0%	0	0.0%	0	0.0%
Practice	0	0.0%	0	0.0%	1	1.0%	0	0.0%	0	0.0%	0	0.0%
Skeleton	5	4.2%	1	0.9%	0	0.0%	0	0.0%	0	0.0%	0	0.0%
Use Of Resources	0	0.0%	0	0.0%	1	1.0%	0	0.0%	0	0.0%	0	0.0%

Table 5: Count and percentage of how many students improved their time management

	Spring '22		Fall '22	
Project Transition	Count	%	Count	%
GP3 -> P1	28	60.87%	27	81.82%
P1 -> P2	32	50.79%	15	38.46%

new struggles that students now face. In P1 and P2 we can see that struggles with implementing the project and debugging failing tests are the most commonly identified struggles. This trend is consistent between both semesters.

These findings also support the work of previous studies such as Arakawa et al.'s work examining student struggles in CS2; supporting the finding that students do encounter difficulties with CS2 assignment implementation [1]. However, our study differs in that we solely examined the responses of the students to identify the struggles faced in CS2. Arakawa et al.'s finding on self-control challenges students face as they approach assignment deadlines could be a reflection of difficulties with time management at the beginning of the assignment work period; however, this remains unclear.

5.3 RQ2: Time Management Improvement

In our second question, we sought to understand whether the self-identification of common struggles that students encounter when working on their programming assignments could lead them to improve in these areas as they work on future CS2 programming assignments. Of the 46 (38.33%) reflections that indicated improving time management from GP3 to P1, 28 (60.87%) of them did improve as shown in Table 4 and Table 5. These two tables can be used in conjunction to track the rest of the improvements for S22 P1 to P2, F22 GP3 to P1, and F22 P1 to P2.

Comparing the struggles and improvements, we can see from Figures 1a and 1b that time management was the most encountered struggle during the GP assignments. Cross-examining the identified improvements in Table 4 we see that time management was the most reflected improvement for the GP assignments. Finally, we can examine Table 5 and see that for both semesters we examined the most improvement in time management when shifting from GP3 to P1. While we cannot make the claim that these reflections helped students make these improvements, through the students' own reflections we can accurately report that this did indeed happen.

6 CONCLUSION

In this work we analyzed the post-assignment reflections of students in a CS2 course to assess what common struggles students face while working on programming assignments at this level of instruction. Furthermore we examine whether or not students show signs of improvement in these areas of struggle, specifically time management, between programming assignments. We were able to utilize students' reflections to find self-identified struggles that they experience while working on these assignments. These findings were then applied towards finding improvements in time management between programming assignments. We were able to find that some students were able to improve between assignment and some students did not. It is unclear what factors are at work that promote or harm their ability to start an assignment earlier. Future work is needed to uncover what factors truly impact a student's ability to start earlier and whether or not this improves student success on these assignments.

7 ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 1934975. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] Kai Arakawa, Qiang Hao, Tyler Greer, Lu Ding, Christopher D. Hundhausen, and Abigayle Peterson. 2021. In Situ Identification of Student Self-Regulated Learning Struggles in Programming Assignments. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education* (Virtual Event, USA) *(SIGCSE '21)*. Association for Computing Machinery, New York, NY, USA, 467–473. https://doi.org/10.1145/3408877.3432357

[2] Susan Bergin, Ronan Reilly, and Desmond Traynor. 2005. Examining the Role of Self-Regulated Learning on Introductory Programming Performance. In *Proceedings of the First International Workshop on Computing Education Research* (Seattle, WA, USA) *(ICER '05)*. Association for Computing Machinery, New York, NY, USA, 81–86. https://doi.org/10.1145/1089786.1089794

[3] Francisco Enrique Vicente Castro, Juho Leinonen, and Arto Hellas. 2022. Experiences With and Lessons Learned on Deadlines and Submission Behavior. In *Proceedings of the 22nd Koli Calling International Conference on Computing Education Research* (Koli, Finland) *(Koli Calling '22)*. Association for Computing Machinery, New York, NY, USA, Article 25, 13 pages. https://doi.org/10.1145/3564721.3564728

[4] Joonas Häkkinen, Petri Ihantola, Matti Luukkainen, Antti Leinonen, and Juho Leinonen. 2021. Persistence of Time Management Behavior of Students and Its Relationship with Performance in Software Projects. In *Proceedings of the 17th ACM Conference on International Computing Education Research* (Virtual Event, USA) *(ICER 2021)*. Association for Computing Machinery, New York, NY, USA, 92–100. https://doi.org/10.1145/3446871.3469767

[5] Judith A Holton. 2007. The coding process and its challenges. *The Sage handbook of grounded theory* 3 (2007), 265–289.

[6] Moira Maguire and Brid Delahunt. 2017. Doing a thematic analysis: A practical, step-by-step guide for learning and teaching scholars. *All Ireland Journal of Higher Education* 9, 3 (2017).

[7] Adrian Salguero, William G. Griswold, Christine Alvarado, and Leo Porter. 2021. Understanding Sources of Student Struggle in Early Computer Science Courses. In *Proceedings of the 17th ACM Conference on International Computing Education Research* (Virtual Event, USA) *(ICER 2021)*. Association for Computing Machinery, New York, NY, USA, 319–333. https://doi.org/10.1145/3446871.3469755

[8] Jeffrey A. Stone and Elinor M. Madigan. 2007. Integrating Reflective Writing in CS/IS. *SIGCSE Bull.* 39, 2 (jun 2007), 42–45. https://doi.org/10.1145/1272848.1272881

[9] Tammy VanDeGrift, Tamara Caruso, Natalie Hill, and Beth Simon. 2011. Experience Report: Getting Novice Programmers to THINK about Improving Their Software Development Process. In *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education* (Dallas, TX, USA) *(SIGCSE '11)*. Association for Computing Machinery, New York, NY, USA, 493–498. https://doi.org/10.1145/1953163.1953307