Procrastination vs. Active Delay: How Students Prepare to Code in Introductory Programming

Elizabeth B. Cloude
elizabeth.cloude@tuni.fi
Tampere University
Tampere, Finland

Jiayi Zhang
jzhang7718@gmail.com
University of Pennsylvania
Philadelphia, PA, USA

Ryan S. Baker
ryanshaunbaker3@gmail.com
University of Pennsylvania
Philadelphia, PA, USA

Eric Fouh
efouh@cis.upenn.edu
University of Pennsylvania
Philadelphia, PA, USA

ABSTRACT

When students procrastinate on programming assignments, it can hinder the quality of their code and negatively impact their grades. In contrast, when students actively delay working on assignments to prepare to code (e.g., reading or seeking help), it can be an effective self-regulated learning (SRL) strategy beneficial to programming performance. However, distinguishing active delay from procrastination is methodologically challenging. To address this, we tracked what students did when they behaviorally delayed starting an assignment. Most students prepared to code by using multiple course resources across programming assignments. We found that many students delayed starting to code by seeking help in the Q&A platform, and this was beneficial to the quality of their code. Also, some pre-coding activities were related to behavioral delay in starting to code, but benefitted students' grades, and thus may indicate active delay, but not all pre-coding activities were beneficial. By considering pre-coding activities, we gain a comprehensive view of students' approach to coding in CS education.

CCS CONCEPTS

• **Social and professional topics → Computing education**; • **Applied computing → E-learning**.

KEYWORDS

procrastination, active delay, CS1, learning behaviors

ACM Reference Format:
Elizabeth B. Cloude, Jiayi Zhang, Ryan S. Baker, and Eric Fouh. 2024. Procrastination vs. Active Delay: How Students Prepare to Code in Introductory Programming. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2024), March 20–23, 2024, Portland, OR, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3626252.3630907

1 INTRODUCTION

Self-regulated learning (SRL) is an important skill for students to develop in computer science (CS) education [2, 17, 21]. Effective SRL involves students playing an active role in their learning, by continuously monitoring and evaluating learning processes to achieve learning goals [23]. When students are less effective self-regulated learners, e.g., they procrastinate on their assignments, they perform worse compared to better self-regulated learners [8]. However, studies find inconsistent results, where sometimes procrastination is beneficial to the quality of code and performance [13].

The mixed findings on procrastination and performance may stem from two limitations in literature. First, the methods used to define when students procrastinate are problematic [6]. Some studies leverage self-report data to measure procrastination, while others use behavioral measures of delay, such as how late students start to code an assignment after it is released [8, 25], or how close the assignment is started to the deadline [15]. Second, solely relying on delay measures may not be the best way to measure procrastination, especially when many students actively delay their work as an adaptive strategy [2]. Active delay allows students to prepare in advance and often benefits their grades [5, 6, 25].

The inconsistent findings on the relationship between procrastination and performance could be due to challenges in properly distinguishing between procrastination and active delay. Leveraging single measures of delay may not holistically represent if, when, and to what extent students procrastinated or actively delayed working on their assignments. To address these limitations, the objective of this work was to examine students' pre-coding behaviors *prior* to completing multiple programming assignments, such as their degree of preparation for assignments based on how often they accessed course resources, e.g., digital textbook, online office hours, before they started coding. Next, we examined whether pre-coding activities predicted students' behavioral delay in starting to code and grades in an introductory CS course. Our research questions are below:

- Do students prepare to code an assignment, and do they prepare more for difficult assignments? We hypothesize that some students will engage with course resources prior to coding. In addition, we expect that students will engage with course resources more often for difficult assignments.

SIGCSE 2024, March 20–23, 2024, Portland, OR, USA

ACM ISBN 979-8-4007-0423-9/24/03.
https://doi.org/10.1145/3626252.3630907

- Does the extent of preparing to code predict when students will start to code a programming assignment? We hypothesize that if students engage with course resources more often prior to coding, this will predict when they start working on the programming assignment.
- Does preparing to code predict students' grades in an introductory programming course? We hypothesize that students who engage with course resources prior to starting an assignment will achieve better grades in the introductory course.

2 RELATED WORK

A meta-analysis [13] revealed that the relationship between procrastination and performance was mediated by how researchers defined and measured the construct [10, 18]. A critical way to tease apart the relationship between procrastination and performance lies in utilizing proper and consistent methods. Historically, most studies of procrastination in computer science utilize single measures of behavioral delay to determine if a student is procrastinating, such as starting a task early [9] or late [8, 12, 18]. But not all delay is a failure to engage in SRL and thus may not be detrimental to performance [5]. Some students strategically delay their work to prepare for the task. In this way, their intention to delay is based on their metacognitive awareness of their abilities, and preparing in advance when needed is evidence of an adaptive SRL strategy.

To discern between procrastination and active delay, [6] examined whether students who reported more active delay differed in grades, and self-reported SRL strategy use and goal orientation compared to students who reported more procrastination. Students who reported more procrastination had lower grades, but those who reported more active delay had higher grades. Procrastinators utilized less metacognitive strategies; in contrast, students who reported more active delay had higher self-efficacy. These findings suggest that students who actively delay differed from procrastinators in metacognitive strategy use and feelings of confidence. However, there are limitations with using self-report data; partly because many students' self-report data do not align with their actual learning behaviors (e.g., [3]).

Lindt et al. [16] extended this work by examining whether students who actively delayed or procrastinated differed in grades using a mixed-methods approach. First, they administered the Procrastination Assessment Scale [11] and Active Procrastination Scale [4]. Next, semi-structured interviews were conducted to more deeply understand the differences between procrastinators and active delayers. The results revealed that many students reported delaying their work on assignments until days or hours before the deadline was not intended as a strategy. Procrastinators explained they unintentionally delayed their work as a form of avoidance, possibly because of anxiety or fear of failure [16]. Students who actively delayed their work did so in order to prepare.

Most studies utilize single measures of behavioral delay to determine if a student is procrastinating, such as starting a task early [9] or late [12, 18]. In a study of over 1100 CS1 students collected across five years [8], the results showed that students who started working on their assignments earlier demonstrated better quality of work on their assignments. The authors argued that this result may suggest that active delayers start the assignment earlier than procrastinators based on its relationship to performance; still, questions remain unanswered about whether students were actively delaying intentionally instead of procrastinating. Wessel et al. [22] extended this work by comparing behavioral delay measures collected using experienced sampling (ESM) with self-reported data in a longitudinal study. Self-reported measures were administered at randomly varying moments to gauge their procrastination and active delay while students worked on multiple programming assignments throughout the course. Results showed that self-reported procrastination was positively associated with behavioral delay (completing an assignment); whereas, self-reported active delay was not associated with behavioral delay.

In sum, prior studies suggest that discerning between active delay and procrastination is possible, and perhaps, students who actively delay their work are effective self-regulated learners. Leveraging other data channels to supplement behavioral delay measures may better represent what students are doing when they are delaying an assignment. Few studies in CS education measure what students do outside of a programming activity to define their level of procrastination or active delay. [25] found that students who voluntarily practiced programming problems submitted their assignments earlier and had better grades, possibly indicating a form of active delay. Results also showed that many students submitted their assignments later more often based on how difficult the assignment was, possibly due to anxiety or fear of failure [16].

To better distinguish between active delay and procrastination, researchers need to supplement behavioral delay measures regarding work on the assignment with measures of what students are doing to prepare for the assignment to capture the nature of the students' delay. If the student is delaying work on an assignment, are they also engaging with course resources during this delay and then working on the assignment? What preparation activities are students engaging in when they have access to multiple course resources designed to support their progress and success in the course (e.g., office hours, electronic textbook, Q&A discussion forum, lecture videos, etc.). and is this a predictor of their behavioral delay? Collecting what students do to prepare for an assignment may reveal if, and when, students may intentionally delay their work compared to students who procrastinate to better understand its impact on performance.

3 PARTICIPANTS AND COURSE DESIGN

Three hundred and one undergraduate students (n=301) completed this study by enrolling in a CS1 course at a large, private university in the northeastern USA during the Fall 2020 semester (14 weeks). The course was taught online due to the COVID pandemic and was designed for students who had little to no prior computing experience, most of whom were in their first semester of college and had not declared a major. An ethical review committee approved this study prior to data collection, and demographic information on the students was not released by the university to the researchers.

The course used Java as the programming language and required the students to complete nine programming homework assignments and two timed exams (Table 1). The programming assignments were designed to assess programming knowledge, while also providing

Table 1: Homework assignments, topics, and exams.

HW	Exam	Topics
0	1	Java syntax, print statements
1	1	Conditionals/loops, variables, primitive types
2	1	Functions, array manipulation
3	1	Functions, string manipulation, ASCII encoding
4	1	Recursion
5	2	Object-Oriented, unit testing
6	2	OOP, bitwise operations, number systems
7	2	Linked nodes, unit testing
8	2	OOP, bitwise operations, number systems

students with practice opportunities for solving programming problems associated with the computing topics introduced in the course (e.g., recursion, abstract data types). The course provided several resources to students which collected their interaction data:

- Digital interactive textbook (Codio)[1], that delivered lecture notes as an interactive electronic textbook.
- Learner management system (LMS; Canvas)[2], delivered pre- and live-recorded lecture videos. The videos covered course content and live coding sessions.
- Online help-seeking: students could access a Q&A platform (24/7; Piazza)[3] and online office hours (OH) management software that students could use to seek help from both peers and/or instructor(s) at specific times.
- Automated feedback tool (Gradescope)[4] that automatically graded and provided immediate feedback to students.

Depending on the assignment, students had a week or two (HW0, HW8) to complete each programming assignment. Students were given an unlimited number of submissions prior to the assignment deadline, and they received immediate feedback with each submission using an automated feedback tool. The last homework (HW9) was removed from our analysis because it was a self-designed project. While 301 students were enrolled in the course, some students did not submit all assignments. Hence, the number of participants varied slightly across programming assignments.

4 DO STUDENTS PREPARE TO CODE AN ASSIGNMENT, AND DO THEY PREPARE MORE FOR DIFFICULT ASSIGNMENTS?

4.1 Methods

To address research question one, students' engagement in pre-coding activities was calculated based on their utilization of course resources, specifically after the assignment was released, but before their initial engagement with the assignment in the IDE each week. To measure when and how often students prepared for each assignment, we collected their interactions with multiple course resources throughout the introductory programming course.

The number of students who used course resource platforms across all programming assignments was collected. For students who prepared, we calculated the number of days they used each platform while preparing for each assignment. In particular, for each student and homework assignment, we examined how often a student used the Q&A and LMS platforms by summarizing the number of 1) days the Q&A platform was used, 2) days the LMS platform was used, 3) posts viewed on the Q&A platform, and 4) total minutes spent watching videos on the LMS platform. As Table 2 shows, of the 301 students enrolled in the class, the majority of students used the Q&A and LMS platforms across all assignments (except for HW0 where no students used the LMS platform). Due to low OH and digital textbook usage, we narrowed our analysis to only focus on Q&A and LMS data.

A 2-parameter item response theory (IRT; [7]) model was used to determine assignment difficulty. IRT was used to calculate a difficulty parameter for each assignment (HW) based on the student's final grade (see results in [25]). Similar to platform usage, the assignments were ranked from the easiest to the most difficult based on the difficulty estimates. Next, we examined the relationships between students' pre-coding activities and assignment difficulty. The higher the difficulty estimate, the higher the ranking was for that assignment (Table **??**).

Last, a Spearman correlation examined associations between difficulty rank and students' pre-coding activities usage rank for each assignment. A Benjamini-Hochberg [1] correction was applied to each correlation to limit the false discovery rate using the 'alpha.correction.bh' package in R [20].

4.2 Results

Students who prepared to code allocated their time differently across the platforms for each assignment. The majority of students spent their time preparing to code using the Q&A forum and LMS platform across assignments (except for HW0). These results suggest that students typically prepare to code by engaging with the Q&A and LMS platforms, but their selection of preparation activities varied across the assignments.

In Table **??**, we present the homework and the rankings of each assignment, determined by the difficulty level and the average student usage of the Q&A and LMS platforms prior to coding. As illustrated, HW0 was the least difficult assignment and had the lowest ranking on all four pre-coding activity measures. This indicated that students either did not use or used the Q&A and LMS platforms the least often to prepare for HW0.

The Spearman correlations found no significant associations between assignment difficulty and students' average use of the Q&A and LMS platforms while preparing to code ($ps > .05$). Although this relationship was marginally significant ($r = .6, \alpha = .013, p = .097$), it is worth noting that a positive (moderate) correlation was found between assignment difficulty and the number of posts viewed in the Q&A. This finding suggested that students who engaged in pre-coding activities, particularly viewing more posts on the Q&A platform, tended to do so increasingly as the difficulty of the programming assignment escalated. This partially

[1] https://www.codio.com/
[2] https://www.instructure.com/canvas
[3] https://piazza.com/
[4] https://www.gradescope.com

Table 2: Pre-coding Activities across homework assignments.

	Q&A		LMS		OH		Textbook	
HW	*n*	#	*n*	#	*n*	#	*n*	#
0	212	358	0	0	6	6	96	34
1	263	1220	275	625	17	16	97	36
2	208	508	275	814	16	15	84	35
3	217	550	229	251	32	30	97	39
4	226	622	200	211	31	30	123	39
5	207	503	223	269	80	30	237	39
6	211	457	154	154	30	52	89	27
7	231	668	218	282	35	35	105	37
8	232	853	217	256	28	27	135	38
9	246	1021	232	606	121	62	199	38

n: Number of students; #: Number of interactions.

supported prior findings [25], where students prepared more before they started coding the more difficult the assignment was.

5 DOES THE EXTENT OF PREPARING TO CODE PREDICT WHEN STUDENTS WILL START TO CODE A PROGRAMMING ASSIGNMENT?

5.1 Methods

We calculated rank-based regression for each homework assignment, to estimate whether engaging in pre-coding activities before coding relates to the number of days students delay coding the programming assignment. The number of days that students did not code after the assignment was released was very right-skewed, meaning that almost all of the students started coding their assignment within 10 days after its release. As such, rank-based regression was the appropriate method since it estimates the vector of coefficients in a general linear model. In rank-based regression, Jaeckel's dispersion function is used to minimize the error distance, instead of Euclidean distance as with Least Squares in linear regression [19].

We used the "Rfit" package in R to calculate our models [14] (data and code are available in Appendix A). For each assignment, the four pre-coding measures (see Section 4.1) were used to predict when students' started to code a programming assignment. To capture the full range of pre-coding activities across the course resources, for each assignment, we included all students who have worked on the assignment, including those who did not use any of the platforms before starting to code. For students who did not use Piazza or Canvas, a value of zero was imputed if any of the four pre-coding activity measures were missing.

5.2 Results

As shown in Table 3, we found that other than HW3 and HW7, the number of days students used the Q&A Piazza negatively predicted when students started coding their assignment. This suggests that the more days students spent using the Q&A platform, the earlier they started to code the assignment. We also found that the more often students viewed posts in the Q&A (except HW3) and videos on the LMS was positively associated with starting to code the assignment later after the release date, possibly indicating active delay by students gathering information from posts on the Q&A. For example, a student may read a lot of posts on the Q&A platform for an assignment or topic if they were confused or struggling to start the assignment before their questions were answered [24].

Similarly, we found the number of days students used the LMS was not related to when they started coding the assignment. A possible explanation could be that the LMS can be used in multiple ways beyond preparing to code, such as using the platform for general purposes like reviewing the syllabus. These findings support our hypothesis that engaging in pre-coding activities, specifically when students prepare for assignments with the Q&A platform, was related to students' coding start date. However, the extent of delay varied based on the degree of students' preparation and how their preparation was measured (i.e., negative relationships for the number of days using the Q&A versus positive relationships for the number of posts viewed on the Q&A platform; 3. The inverse relationship between the two metrics and when coding started could indicate a difference in how students are oriented to different goals (e.g., approach vs. avoidance) and the learning strategies they use. For instance, frequent but short visits on the Q&A platform were correlated with starting to code earlier, whereas frequent and long visits might indicate less-than-optimal SRL strategies.

Students may also spend more time reading posts on the Q&A platform if they do not have an accurate sense of their knowledge gaps and needs, and are thus unable to target specific topics. It is also worth noting that the R^2 values in the models were moderate for the analyses in Table 3. The highest R^2 value was 24% (HW5), indicating 24% of the variability in when students started to code was explained by pre-coding activities, suggesting other factors are likely at play in explaining when students start to code.

6 DOES PREPARING TO CODE PREDICT STUDENTS' GRADES IN AN INTRODUCTORY PROGRAMMING COURSE?

6.1 Methods

Similar to RQ2, we utilized rank-based regression to examine whether pre-coding activities were related to students' grades on the assignment. Since the grades from each assignment were left-skewed (due to an unlimited amount of resubmissions using the automated feedback tool), with the majority of the students scoring a letter grade A, grades were transformed into ranks to represent their performance. The same set of students was included in this analysis as in RQ2. For each assignment, students' pre-coding activities were used to predict their grade ranking on the assignment.

6.2 Results

For each model, the unstandardized coefficient and significance level are reported. As shown in Table 4, the number of days students used the Q&A platform positively predicted their HW grade ranks for all assignments, except for HW2 and HW8. This positive relationship indicated that the more days students used the Q&A platform before they started to code, the higher their grade was

Table 3: Predicting the degree of procrastination using pre-coding activity measures.

HW	Q&A	LMS	Q&A Posts	Videos	Model Fit Indices
0	$-.26^{***}$	NA	$.03^{***}$	NA	$F = 19.02, p < .0001, R^2 = .21$
1	$-.17^{*}$	$-.09$	$.06^{***}$	$.02^{***}$	$F = 10.03, p < .0001, R^2 = .13$
2	$-.24^{**}$	$.19$	$.05^{**}$	$.02^{***}$	$F = 8.880, p < .0001, R^2 = .12$
3	$.05$	$.30$	$.03$	$.01^{***}$	$F = 7.90, p < .0001, R^2 = .11$
4	$-.23^{*}$	$-.05$	$.05^{***}$	$.01^{***}$	$F = 6.39, p < .0001, R^2 = .09$
5	$-.18^{*}$	$.85$	$.03^{***}$	$.02^{***}$	$F = 18.49, p < .0001, R^2 = .24$
6	$-.22^{**}$	$.08$	$.05^{***}$	0.01^{***}	$F = 17.03, p < .0001, R^2 = .21$
7	$-.11$	$.99^{**}$	$.05^{***}$	0.02^{***}	$F = 17.12, p < .0001, R^2 = .21$
8	$-.26^{**}$	$-.02$	$.12^{***}$	0.01^{***}	$F = 15.15, p < .001, R^2 = .19$

$^{*}p < .05$; $^{**}p < .01$; $^{***}p < .0001$; F statistics & p-values correspond to the Drop in Dispersion Test.

on the assignment. The lack of relationships between grades and pre-coding activities for HW2 could be due to the nature of the programming problem, which required students to implement a physics equation correctly. In this way, many students were confused by the physics equation instead of the programming topic. HW8 on the other end, was the last "assigned" homework. One of the course grading policies dropped the lowest assigned homework grade, as long as the student received a grade of at least 30% on all HWs. Thus, many students did just enough work to earn 30% on this last assignment ("who can blame them").

Similar positive (strong) relationships were found between the number of times students used the LMS platform and grade ranks, but only for HW2, HW5, and HW7. A possible explanation for this could be due to the fact that HW2, HW5, and HW7 exposed students to concepts (parameter passing, primitive values, objects, references, etc.) that take practice and time to master, often requiring thorough code tracing before achieving proficiency. Since the instructors go over code examples during lectures, students visited the LMS frequently to rewatch lecture videos while preparing for those HWs.

Interestingly, the number of times students viewed posts on the Q&A platform was not related to HW grade ranks ($ps > .05$). A possible explanation for this could be that simply viewing a post on the Q&A platform did not ensure that the student had received an answer to their question or addressed their need to prepare prior to coding. Not all posts on the Q&A platform were created equal, meaning some posts may have been more or less relevant to programming topics and this may not have been helpful to students' before starting a specific programming assignment. More research is needed to determine the type and quality of the posts that students viewed. For example, simply viewing posts may not be indicative of actively delaying work to prepare for the assignment. Rather it may be indicative of an ineffective strategy to improve the quality of their code on the assignment. This could be due to the student a) viewing low-quality posts on the Q&A, or b) adopting an ineffective SRL strategy to prepare for learning by seeking out the content and preparation needed to improve the quality of code.

Similar results were obtained for the time students prepared by watching videos on the LMS. The minutes students watched videos had no relationship with their code quality, with the exception of HW7. There was a negative (weak) relationship between the amount of time viewing videos on the LMS and HW7 grade rank. This was surprising and did not support our hypothesis, where we expected more time engaging in pre-coding activities would benefit grades. Yet, there were no relationships or a negative relationship between the time students viewed the videos on the lecture or live coding session with the quality of their code. Moreover, if students spent too much time watching lecture videos, then it was negatively related to their grades, on some of the homework assignments.

This could be explained by the fact that students who needed to rewatch the entire video were probably very confused about the assignment compared to students who did not need to watch longer. This could also indicate that the student did not necessarily know what to look for in the video, based on their need to prepare. In contrast to students who accurately identified their knowledge gap, they could have fast-forwarded to the specific part of the video that addressed their need to prepare. These results partially supported our hypothesis: pre-coding activities across all homework assignments would be positively related to performance outcomes. Instead, we found that some preparation activities were more beneficial than others.

7 DISCUSSION

The objective of this study was to examine the extent that students prepared for coding assignments, such as their level of preparation by utilizing course resources, e.g., digital textbook, online office hours, Q&A forum, etc., and its relation to behavioral delay measure and grades in an introductory computer science (CS1) course. Our first hypothesis, where we expected students to engage with course resources prior to coding, and that students would engage with course resources more often for difficult assignments, was supported in **RQ1**. The results suggested that most students engaged with course resources prior to coding, but they allocated their time differently for varying programming assignments and across multiple course resources. However, most of the time spent preparing to code was by utilizing the Q&A forum and the digital textbook. We also found a marginally significant, and moderate positive correlation between assignment difficulty and the number of posts viewed on the Q&A forum.

Table 4: Predicting homework performance (rank) using pre-coding activities.

HW	Q&A	LMS	Q&A Posts	Videos	Model Fit Indices[5]
0	7.51^{**}	8.88	−.24	−.18	$F = 4.37, p = .0019, R^2 = .06$
1	13.93^{**}	7.95	−.96	−.17	$F = 3.42, p = .0095, R^2 = .05$
2	4.43	29.89^{**}	.55	−.23	$F = 3.33, p = .011, R^2 = .05$
3	11.67^{*}	20.98	−.14	.1	$F = 4.95, p = .0007, R^2 = .07$
4	15.49^{***}	−1.49	−.53	−.07	$F = 2.8, p = .0265, R^2 = .04$
5	14.07^{***}	46.14^{***}	−.06	−.08	$F = 8.63, p < .0001, R^2 = .11$
6	11.03^{**}	11.40	−0.78	−.13	$F = 2.74, p = .0293, R^2 = .04$
7	13.5^{***}	23.13^{*}	−.69	$-.21^{*}$	$F = 7.29, p < .0001, R^2 = .04$
8	5.39	15.04	−.44	.12	$F = 2.57, p = .0383, R^2 = .04$

$^{*}p < 0.05$; $^{**}p < 0.01$; $^{***}p < 0.0001$; F statistics & p-values correspond to the Drop in Dispersion Test.

For **RQ2**, our second hypothesis was partially supported, where we expected that the more often students engaged with course resources prior to coding, would predict their extent of behavioral delay in starting to code the assignment. The results showed that the number of pre-coding activities students engaged in across multiple course resources predicted their extent of behavioral delay in starting to code sometimes, but not always. We found that the extent of behavioral delay varied based on the student's degree of preparation and also how preparation was measured. Collecting data on days spent using a course resource and the number of posts viewed in the Q&A forum, for instance, revealed differences among students, possibly due to differences in how they were oriented to goals and learning strategy use.

Last, **RQ3** partially supported our third hypothesis, where we expected that students who engage with course resources prior to starting the assignment would achieve better grades in the introductory course. The results suggest that Q&A posts and videos were not as useful for preparing to code an assignment. Instead, we found a different pattern. When students actively delayed starting to code the assignment by actively seeking help online in the Q&A platform–going beyond mere viewing of posts, which is more passive–this approach was associated with higher quality code for most assignments. However, other pre-coding activities were not beneficial to grades on the homework, such as the amount of time watching videos on the LMS or viewing posts on the Q&A.

In sum, these findings advance our understanding of if, when, and how students prepare to build programs and its relation to their behavioral delay and grades in an introductory CS course. Some pre-coding activities improve grades on programming assignments, despite students' delaying their work on starting assignments. However, not all instances of preparation may be strategic or effective. It is important to note that for students to engage in effective preparation they must identify the task demands and accurately evaluate their current knowledge base [23, 24]. From here, students can strategically target materials or course resources that would allow them to prepare to code.

Since the study was conducted in a course setting, the data collected possess high ecological validity, as they were observed in the natural context where learning to program occurs. This authenticity increases the likelihood that our findings may generalize to other CS classroom contexts, as they align closely with the reality of teaching and learning experiences. We recommend educators and researchers adopt multiple measures of behavior that go beyond a single delay measure to distinguish between students who procrastinate vs. actively delay their work on assignments.

Future researchers may want to consider the role of goal orientation and metacognitive accuracy on how students engage in pre-coding activities. Collecting additional pre-coding activity data (e.g., Stack Overflow) and conducting qualitative interviews may provide a more comprehensive understanding of how students prepare to code in introductory programming. Overall, this study extends prior findings by providing a better picture of the assignment completion life-cycle to CS educators and researchers. The implications of this work may allow future work to better identify when students procrastinate vs actively delay working on programming assignments.

7.1 Threats to Validity

This work was limited in two ways: 1) students' first interaction with the IDE defined when they started coding; yet, it is possible that they may have opened the IDE and not started to code until later; and 2) IRT was used to determine assignment difficulty. Given that students had unlimited submissions, utilizing their final grades on assignments does not accurately reflect their initial grades. Instead, their final grades were products of potentially multiple attempts and adjustments with the automated feedback tool. In addition, it is possible that some students delayed starting to code, not due to procrastination, but other responsibilities outside of the course (e.g., part-time work, hobbies, etc.).

8 ACKNOWLEDGMENTS

This study was supported by the National Science Foundation (NSF; DUE-1946150). Any conclusions expressed in this material do not necessarily reflect the views of NSF.

A ONLINE RESOURCES

Data, software (R script) for data processing, and the results are publicly available at: https://github.com/SERI-CS/pre-coding-analysis/.

REFERENCES

[1] Yoav Benjamini and Yosef Hochberg. 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)* 57, 1 (1995), 289–300.

[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*. 81–86.

[3] Heeryung Choi, Philip H Winne, Christopher Brooks, Warren Li, and Kerby Shedden. 2023. Logs or Self-Reports? Misalignment Between Behavioral Trace Data and Surveys When Modeling Learner Achievement Goal Orientation. In *LAK23: 13th International Learning Analytics and Knowledge Conference*. 11–21.

[4] Jin Nam Choi and Sarah V Moran. 2009. Why not procrastinate? Development and validation of a new active procrastination scale. *The Journal of social psychology* 149, 2 (2009), 195–212.

[5] Angela Hsin Chun Chu and Jin Nam Choi. 2005. Rethinking procrastination: Positive effects of" active" procrastination behavior on attitudes and performance. *The Journal of social psychology* 145, 3 (2005), 245–264.

[6] Danya M Corkin, L Yu Shirley, and Suzanne F Lindt. 2011. Comparing active delay and procrastination from a self-regulated learning perspective. *Learning and Individual Differences* 21, 5 (2011), 602–606.

[7] Rafael Jaime De Ayala. 2013. *The theory and practice of item response theory.* Guilford Publications.

[8] Stephen H Edwards, Jason Snyder, Manuel A Pérez-Quiñones, Anthony Allevato, Dongkwan Kim, and Betsy Tretola. 2009. Comparing effective and ineffective behaviors of student programmers. In *Proceedings of the fifth international workshop on Computing education research workshop*. 3–14.

[9] Anthony Estey and Yvonne Coady. 2016. Can interaction patterns with supplemental study tools predict outcomes in CS1?. In *Proceedings of the 2016 acm conference on innovation and technology in computer science education*. 236–241.

[10] Eric Fouh, Wellington Lee, and Ryan S Baker. 2021. Nudging students to reduce procrastination in office hours and forums. In *2021 25th International Conference Information Visualisation (IV)*. IEEE, 248–254.

[11] Andrew J Howell and David C Watson. 2007. Procrastination: Associations with achievement goal orientation and learning strategies. *Personality and Individual Differences* 43, 1 (2007), 167–178.

[12] Viggo Kann and Anna-Karin Högfeldt. 2016. Effects of a program integrating course for students of computer science and engineering. In *Proceedings of the 47th ACM technical symposium on computing science education*. 510–515.

[13] Kyung Ryung Kim and Eun Hee Seo. 2015. The relationship between procrastination and academic performance: A meta-analysis. *Personality and Individual Differences* 82 (2015), 26–33.

[14] John D. Kloke and Joseph W. McKean. 2012. Rfit: Rank-based estimation for linear models,. *The R Journal* 4, 2 (2012), 57–64.

[15] Juho Leinonen, Francisco Enrique Vicente Castro, and Arto Hellas. 2021. Does the early bird catch the worm? Earliness of students' work and its relationship with course outcomes. In *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1*. 373–379.

[16] Suzanne F Lindt, Danya M Corkin, and SI Yu. 2014. Using multiple methods to distinguish active delay and procrastination in college students. *American International Journal of Contemporary Research* 4, 2 (2014), 28–32.

[17] Dastyni Loksa, Lauren Margulieux, Brett A Becker, Michelle Craig, Paul Denny, Raymond Pettit, and James Prather. 2022. Metacognition and self-regulation in programming education: Theories and exemplars of use. *ACM Transactions on Computing Education (TOCE)* 22, 4 (2022), 1–31.

[18] Joshua Martin, Stephen H Edwards, and Clfford A Shaffer. 2015. The effects of procrastination interventions on programming project success. In *Proceedings of the eleventh annual International Conference on International Computing Education Research*. 3–11.

[19] Joseph W McKean and Thomas P Hettmansperger. 2016. Rank-based analysis of linear models and beyond: A review. *Robust Rank-Based and Nonparametric Methods: Michigan, USA, April 2015: Selected, Revised, and Extended Contributions* (2016), 1–24.

[20] Michael Mogessie. 2023. *alpha.correction.bh: Benjamini-Hochberg Alpha Correction.* R package version 0.0.1.

[21] James Prather, Brett A Becker, Michelle Craig, Paul Denny, Dastyni Loksa, and Lauren Margulieux. 2020. What do we think we are doing? Metacognition and self-regulation in programming. In *Proceedings of the 2020 ACM conference on international computing education research*. 2–13.

[22] Jason Wessel, Graham L Bradley, and Michelle Hood. 2019. Comparing effects of active and passive procrastination: A field study of behavioral delay. *Personality and Individual Differences* 139 (2019), 152–157.

[23] PH Winne and AF Hadwin. 2008. *The weave of motivation and self-regulated learning.* Erlbaum. 297–314 pages.

[24] Philip H Winne and Nancy E Perry. 1998. Measuring self-regulated learning. In *Handbook of self-regulation*. Elsevier, 531–566.

[25] Jiayi Zhang, Taylor Cunningham, Rashmi Iyer, Ryan Baker, and Eric Fouh. 2022. Exploring the Impact of Voluntary Practice and Procrastination in an Introductory Programming Course. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education-Volume 1*. 356–361.